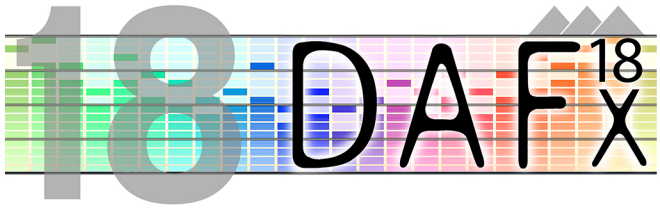
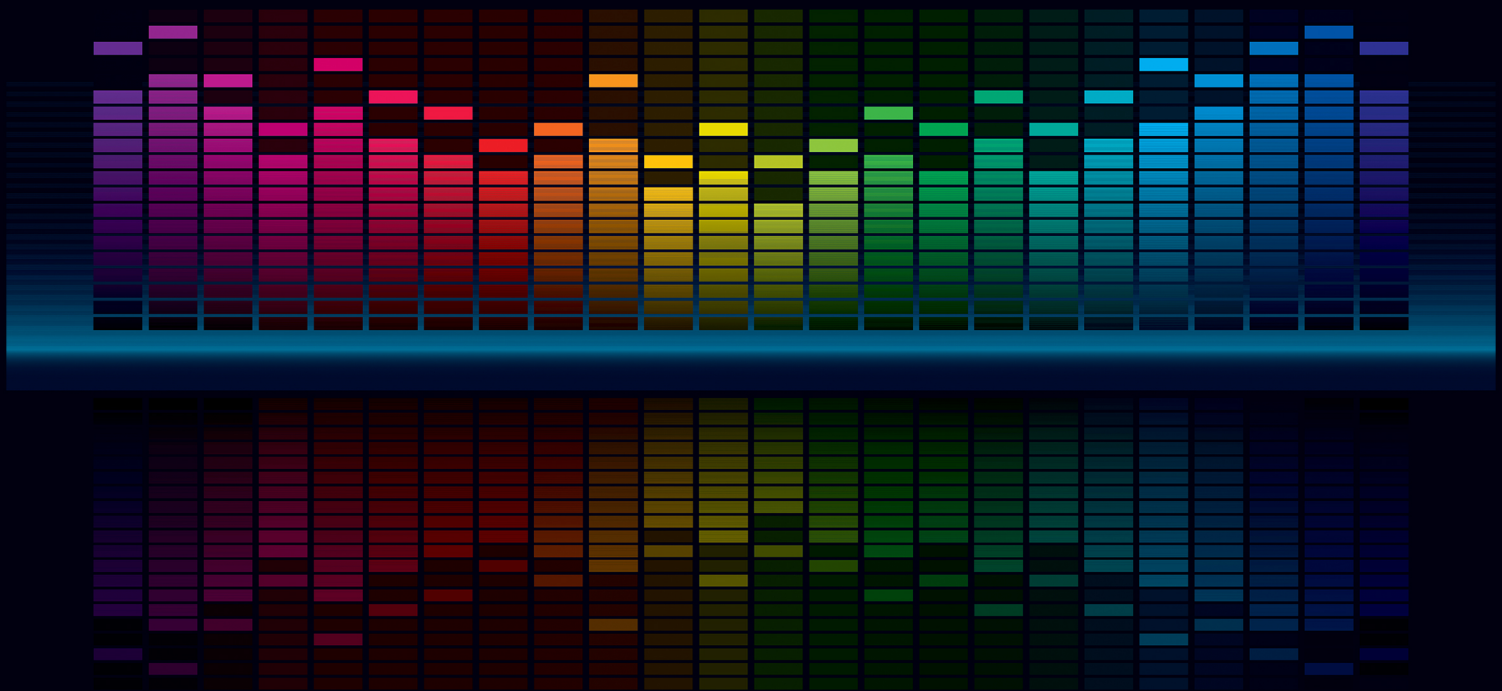
universidade de aveiro
theoria poiesis praxis



Proceedings

21st International Conference on Digital Audio Effects
September 4th - 8th, 2018
Aveiro, Portugal

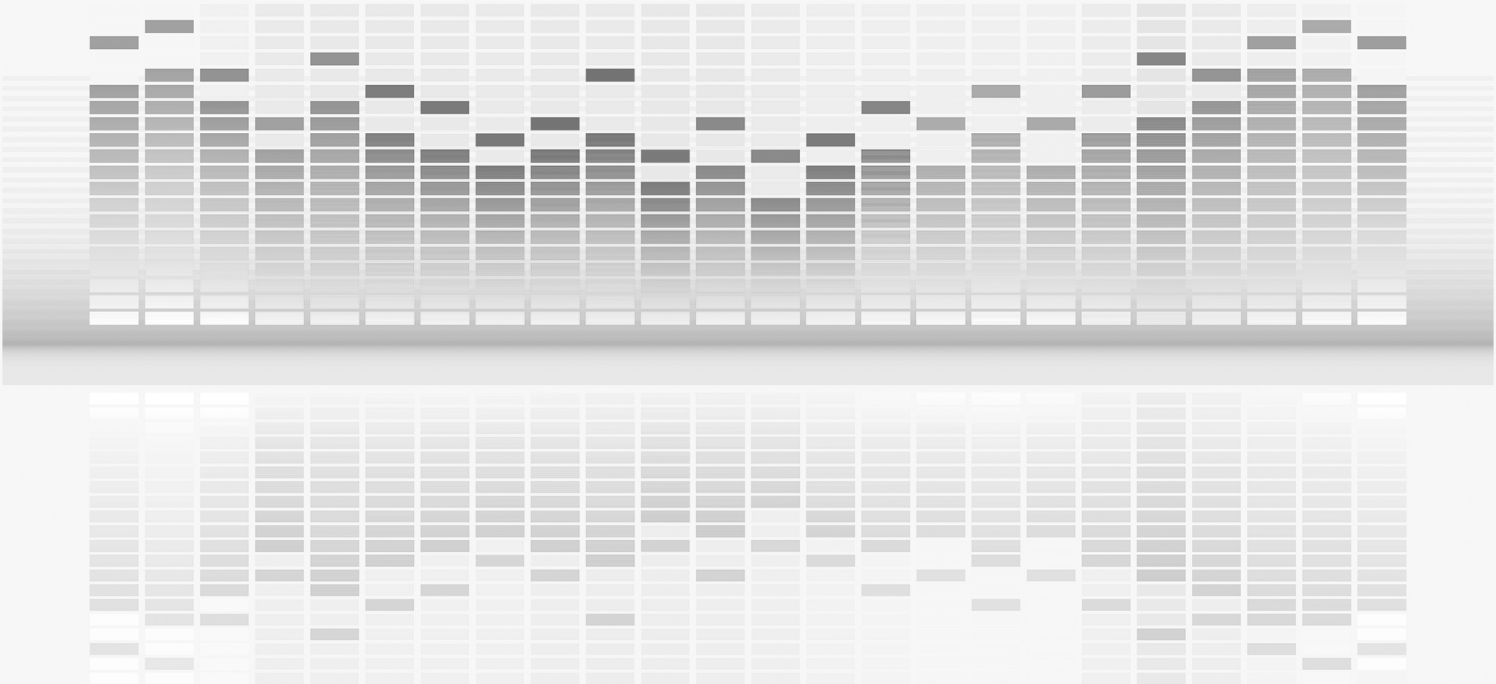


universidade de aveiro
theoria poiesis praxis

DAFx18

Proceedings

International Conference on Digital Audio Effects
September 4th - 8th, 2018
Aveiro, Portugal



Credits:
Proceedings edited and produced by
Matthew Davies, Aníbal Ferreira, Guilherme Campos, Nuno Fonseca
LaTeX style by Paolo Annibale

ISSN 2413-6700 (Print)
ISSN 2413-6689 (Online)

http://dafx2018.web.ua.pt
www.dafx.de

DAFx18 is proudly sponsored by

Foreword

The University of Aveiro and the Portuguese Association of Audio Engineering are delighted to host the 21st International Conference on Digital Audio Effects. This privilege is amplified by two factors: the celebration of two decades since its inaugural edition, held in Barcelona in 1998, and the opportunity of welcoming the DAFx community to Portugal for the first time.

Organising such a prestigious event is not a simple task. We are particularly indebted to former organisers, especially DAFx17 committee members Stefan Bilbao, Brian Hamilton and Michael Newton, for kindly sharing their experience and support from the first call for papers right up to compiling these proceedings. The generous contribution of our industry sponsors – Eventide, Steinberg, Arturia, Yamaha, Audiokinetic, Izotope, Imaginando, Sonnox, Ableton, Newfangledaudio and NeuralDSP – and the support of the city councils of Aveiro, through the councillor for culture Miguel Capão-Filipe, and Arouca, through mayor Margarida Belém, were crucial. We must also acknowledge the gracious collaboration of the *Voz Nua* choir, directed by Aoife Hiney, the *Xperimus Ensemble*, led by Helena Marinho, and Carlos Brito, head of the Brotherhood of Queen St. Mafalda.

Concerns over gender balance were brought to the fore last year. Stimulating female participation is part of the broader challenge of fostering inclusion and diversity. The DAFx community is keen to embrace this challenge and we kept it in mind at all times.

We believe that highlighting the interdisciplinarity of audio and the broad range of applications of DAFx technology is likely to attract interest from a more diverse audience. Reflecting this idea, our call for papers explicitly encouraged interdisciplinary submissions and the exploration of digital audio processing as a tool for inclusion. Out of 68 submissions, 52 were accepted, resulting in a 76.5% acceptance ratio, which cover the full range of proposed topics. Analog systems and machine learning techniques attracted the most interest. Reviewers praised the high quality of many papers, the best of which will be invited for publication in the Journal of the Audio Engineering Society.

In building our panel of keynote speakers, we strived for balance between different perspectives on digital audio effects, namely from academic research, industry development and artistic application. We are very fortunate that Joshua Reiss, Yvan Grabit and David Farmer were able to accept our invitation, as they perfectly match this criterion. A similar balance was sought in the remaining invited sessions. The first day tutorials, covering a diverse range of stimulating topics, are split between academy and industry; on the final day, we will host the inaugural *DAFx jam session* for musical exploration of effects submitted by participants. The programme opens and closes with two outstanding female academics, from Portugal, whose backgrounds are Psychology and Music respectively.

Attendance figures confirm significant gender imbalance, with only 8.9% female registrations, and the ratio for sponsor delegates was 7.1%. Our call for student volunteers resulted in just one girl among 23 candidates (4.3%). Our invited panel features 16.7% female participation. Geographically, attendance is also very unevenly distributed. While there are attendants from every continent, only 8.7% come from outside Europe or North America. Lowering registration fees and providing inexpensive student accommodation seems to have had little or no impact on this front.

Inclusion is a strong motivation for sharing the DAFx concert with the local community. This involves bridging the gap between electroacoustic music and more mainstream genres. There could hardly be a better choice to meet this challenge than the virtuoso bassoonist Paul Hanson, an expert in modern performance techniques with roots both in jazz and classical music, to whom we are deeply grateful.

Reflecting our commitment to embracing diversity and promoting inclusion, the remaining events of the social programme, and especially the Saturday tour to Arouca, seek to celebrate the remarkable diversity of our region's natural and cultural heritage.

We sincerely hope that you enjoy your DAFx experience in Aveiro!

The DAFx18 Local Organising Committee

Conference Committees

DAFx18 Local Organizing Committee

Guilherme Campos (General Chair)
Nuno Fonseca (General Chair)

Aníbal Ferreira (Paper Chair)
Matthew Davies (Paper Chair)

José Vieira (Finance Chair)
Diamantino Freitas (Finance Coordinator)

Rui Penha (Concert Chair)
Daniel Albuquerque (Conference Material)
Pedro Pestana (Communication)
Salviano Soares (Sponsorship)
Anabela Viegas (Secretariat)

DAFx18 Programme Committee

Regis Rossi Alves Faria (University of São Paulo)
Federico Avanzini (University of Milan)
Peter Balazs (Acoustics Research Institute)
Stefan Bilbao (The University of Edinburgh)
Luiz Biscainho (Universidade Federal do Rio de Janeiro (UFRJ))
Marcelo Caetano (INESC TEC)
Guilherme Campos (University of Aveiro)
Mark Cartwright (MARL: Music and Audio Research Laboratory (NYU))
Vasileios Chatziioannou (Institute of Music Acoustics, Vienna)
Matthew Davies (INESC TEC)
Brecht De Man (Birmingham City University)
Giovanni De Poli (University of Padova)
Philippe Depalle (McGill University)
Sascha Disch (Fraunhofer)
Michele Ducceschi (The University of Edinburgh)
Gianpaolo Evangelista (University of Music and Performing Arts Vienna)
Bruno Fazenda (University of Salford)
Anibal Ferreira (Faculdade da Engenharia - Universidade do Porto)
Nuno Fonseca (Instituto Politécnico de Leiria)
Federico Fontana (University of Udine)
Brian Hamilton (University of Edinburgh)
Jason Hockman (Birmingham City University)
Robert Hoeldrich (University of Music and Performing Arts Graz)
Martin Holters (Helmut Schmidt University)
Jean-Marc Jot (Magic Leap, Inc.)
Richard Kronland-Martinet (CNRS-PRISM)
Esteban Maestre (McGill University)
Sylvain Marchand (L3i, University of La Rochelle)
Dave Moffat (Queen Mary University of London)
Damian Murphy (University of York)
Juan Miguel Navarro Ruiz (Universidad Católica de Murcia, UCAM)
José Vieira (University of Aveiro)
Michael Newton (The University of Edinburgh)
Julian Parker (Native Instruments)
Rui Penha (Faculdade da Engenharia - Universidade do Porto)

DAFx Board

Contents

Oral Session 2: Percussive Sound Separation / Transcription

Poster Session 2

Oral Session 3: Intelligibility & Perception

Poster Session 3

Oral Session 4: Analysis / Synthesis 2

Oral Session 7: Frequency / Impulse Estimation

Oral Session 8: Digital Audio Effects & Processing

Keynotes

Joshua D. Reiss
Disruptive Innovation in Sound Design and Audio Production

Abstract In films, games, music and virtual reality, we recreate the sounds around us, or create unreal sounds to evoke emotions and capture the imagination. But there is a world of fascinating phenomena related to sound and perception that is not yet understood. If we can gain a deep understanding of how we perceive and respond to complex audio, we could not only interpret the produced content, but we could create new content of unprecedented quality and range. This talk is targeted at a general audience, and considers the possibilities opened up by such research. What are the limits of human hearing? Can we create a realistic virtual world without relying on recorded samples? If every sound in a major film or game soundtrack were computer-generated, could we reach a level of realism comparable to modern computer graphics? Could a robot replace the sound engineer? Investigating such questions reveals surprising aspects of auditory perception, and has the potential to revolutionise sound design and music production.

David Farmer
Confessions from a plugin junkie

Abstract Here, the intention is simply to give a window into an actual users experience. Some examples will be shown of how the use of plugins is applied in a typical day. This will include what draws somebody to use certain plugins over others that may do similar things. Some GUI features will be explored that are found useful and also what is a hinderance. It will be also discussed what it's like to be an end user in a saturated market of products and just how it is to discover, try, and buy developers products.

Yvan Grabit
The top ten things you have to know as Developer from the idea to a product, based on the History of Audio Plugin formats

Abstract From an idea of an algorithm to a final commercial plugin, there is a lot of steps you have to know and understand as a developer in order to make the best of your idea. I will talk about such top ten things from DSP design to UX/UI design including such concerns like latency, bypassing, parameters, precision, automation, surround, persistency,... using reference to the development and history of Audio plugin formats, mainly based on VST 3. The goal of this keynote is to help future or already established plugin developers to be prepared and aware of what should be not forgotten during development.

Tutorials

Catarina Mendonça
Perceptual and cognitive factors for VR audio

Abstract There are many challenges faced by those aiming to render and reproduce convincing virtual audio. This tutorial defines key concepts and goals to allow for the feeling of presence in a simulated audio world. The specific role of factors such as individualization of HRTFs and headphones, sensory adaptation, room cues, motion cues, real-time rendering, and multimodal interfaces is addressed. There is a complex interplay between the ideal sound accuracy and several of these factors. When is accuracy perceptually relevant? When can we fool the listener? These questions are answered having in mind indicators such as localization accuracy, externalization, multimodal interactions and attentional effects. There are three main conclusions: 1) what the listener perceives depends on what we ask, 2) sensory adaptation ultimately allows to overcome most technical limitations, and 3) more accurate rendering will always have benefits.

Vesa Välimäki
Digital Audio Filters

Abstract This tutorial will review the basic digital filters used in audio and music processing, such FIR, allpass, and equalizing filters. FIR filtering is carried out by convolving the samples of the input signal with the filter coefficients. An allpass IIR filter has a flat magnitude response and a nonlinear phase response. It is useful in numerous audio applications, such as in artificial reverberation and in delay equalization. Equalizing filters enable enhancement of sound reproduction systems. The tutorial will include sound examples and interactive demonstrations to explain how the digital filters work and what they can achieve.

Julian Storer
Building plugins and DSP with JUCE

Abstract This talk is an introduction to how the JUCE library provides classes and tools that can help developers who are building plugins (or plugin hosts) and writing DSP algorithms. Topics I'll cover are:

- A quick high-level overview of JUCE and the functional areas it covers;
- A dive into how the audio plugin abstraction layer works and how you'd use it to build a simple plugin;
- An overview of how our plugin hosting classes work and how they might be used to write a simple plugin host;
- A dive into what our DSP module provides;
- If time permits, a quick introduction to some JUCE GUI library concepts.

No familiarity with JUCE is expected, but the talk will require some experience with C++ to get the most out of it.

Shahan Nercessian
Machine Learning with Applications to Audio

Abstract Machine learning is an exploding field which over the past few years has seen great advances, received arguably excessive hype, and has become ubiquitous in our every-day lives. In its correct application, machine learning enables and has already demonstrated borderline science-fiction-like processing and decision making of data, particularly in the domain of image processing and analysis. In this tutorial, we will de-mystify machine learning and its associated buzzwords, explaining what it is, what it isn't, and how it works. Upon formulating some common machine learning problems and giving a short overview of more "classical" machine learning approaches, we will take a deeper dive into neural networks and touch on some modern deep learning architectures. Throughout, we will explore applications of machine learning to audio problems and show how it is used in iZotope products for carrying out various audio classification and restoration tasks.

EFFICIENT EMULATION OF TAPE-LIKE DELAY MODULATION BEHAVIOR

Vadim Zavalishin, Julian D. Parker

Native Instruments GmbH
Berlin, Germany
firstname.lastname@native-instruments.de

ABSTRACT

A significant part of the appeal of tape-based delay effects is the manner in which the pitch of their output responds to changes in delay-time. Straightforward approaches to implementation of delays with tape-like modulation behavior result in algorithms with time complexity proportional to the tape speed, leading to noticeable increases of CPU load at smaller delay times. We propose a method which has constant time complexity, except during tape speedup transitions, where the complexity grows logarithmically, or, if proper antialiasing is desired, linearly with respect to the speedup factor.

1. INTRODUCTION

Delay and echo effects have been fundamental tools for manipulating space and rhythm in music production since the 1950s, with the first commercial units being based on loops of magnetic tape. Over the following decades, other methods for producing such effects were developed, including the use of magnetic drums, and bucket-brigade chips [1, 2, 3]. Starting in the 1970s, digital implementations of delay-lines became available.

A delay line can broadly be thought of as a black-box into which a signal is passed, and which outputs it at some later ('delayed') time. Independent of the technology involved, all delay lines work in fundamentally the same way. The signal is injected into a medium at a particular point, it travels through that medium for some time, and is received at another point. This medium can be magnetic tape, a chain of capacitors, a digital ring-buffer, or even potentially an acoustic or mechanical system. Despite the change in sound-character imposed by the medium, the broad difference between different types of delay-line is the way in which they allow the delay time to be varied. Some allow the distance between the entry point and exit point to be varied (we call these *length-type* delays), whilst others instead manipulate the speed at which the sound traverses the medium (we call these *speed-type* delays). This difference is exemplified by two famous tape-based delay devices - the EchoPlex [4], and the Roland Space Echo series. The former allows the read head of the tape machine to be moved, whereas the latter allows the speed of the motor driving the tape to be changed. This distinction is important, because it greatly influences the pitch-change perceived when manipulating the delay-time, especially when the system is subjected to feedback as is the case in echo effects. In the case of length-type delays, the pitch-change perceived in the output (and recirculated when feedback is present) is dictated purely by the rate of change of the length. In the case of speed-type delays, the change in pitch is defined by the ratio of speeds between the instant the signal entered the medium and the instant it exits. It turns out that the latter behaviour is desirable musically, as it leads to much more consistent control over pitch. For example, a repeating echoing sound can be cleanly pitched up and down by varying the delay-time in the latter case, whereas in the former the same manipulation will result in erratic overlapping pitch changes.

Typical implementations of digital delays are based on a variable length ringbuffer, where the delay time parameter controls the distance of an interpolated read-point from the write-point [5]. This implementation clearly falls into the length-type category, and exhibits the expected problems. Straightforward speed-style digital delays can be implemented using a fixed array and a varying internal sample-rate. Thus, there must be sample rate conversion at the write head (from the outside sampling rate to the sampling rate implied by the speed) and at the read head (from the implied to the outside sample rate). Moreover, the number of processed "internal samples" per one "outside sample" is proportional to the speed. Thus, the time complexity of such emulation is $O(v)$, where v is the speed. This particularly means that at low delay times, where the tape speed is high, the CPU load significantly increases. Another interpretation of variable-sample-rate digital delay is given by Rocchesso [6] who frames the process using interpolated read and write points. Holters extends this variable-sample-rate paradigm to use the BBD circuits input and output filters to perform the necessary interpolation [3].

Huovilainen [1] proposed a way to recompute speed variations into length variations, which can then be used to simply control the read position of a ringbuffer-based digital delay. However his method of recomputation, even though reducing the overall computation costs, still has an $O(v)$ complexity.

In this paper, we propose a new method of implementing a digital speed-style delay which in steady-speed situations and during slowdowns has an $O(1)$ complexity, regardless of the actual speed. During speedup transitions the method has an $O(\log K)$ complexity, where K is the speedup factor. If proper antialiasing of speedups is desired, then during speedup transitions the computation complexity grows to $O(K)$.

For the sake of a more intuitive language we will be talking of emulation of a tape delay. However the discussion will equally apply to other types of speed-style delays.

In Sec. 2, we introduce a continuous-time model for the relationship between tape speed and delay-time, which we call the *tape equation*. We then discuss some results that can be derived directly from the model. In Sec. 3, we describe a numerical scheme for solving the tape equation in the case of arbitrary changes in speed, including consideration of aliasing distortion in cases where tape speed is increasing. In Sec. 4 we present some measurements of the computation efficiency of the described method, in comparison to existing methods. In Sec. 5, we conclude.

2. THE TAPE EQUATION

In real-world tape-delays the tape is often looped, with an erasing head placed before the recording head. Since the erasing head removes the previous signal recorded to the tape, without loss of generality we can consider this configuration to be equivalent to a tape of infinite length in both directions. We will associate a one-dimensional coordinate system with the tape, so that each point on the tape has a coordinate.

Let $x_w(t)$ be the coordinate of the tape point positioned against the write head at the time moment t, and let $x_r(t)$ be the coordinate of the tape point positioned against the read head. Let the tape move in the direction from the write head to the read head and let's orient the tape coordinate axis x so that $x_w(t)$ and $x_r(t)$ will increase with time as the tape moves. Then, if $v(t)$ denotes the speed of the tape at time moment t,

$$\dot{x}_w(t) = \dot{x}_r(t) = v(t) > 0$$

Let the distance between the heads be fixed at

$$x_w(t) - x_r(t) \equiv L > 0 \tag{1}$$

and let $T(t)$ denote the *effective delay time* at time moment t, meaning that the signal value which is being picked up by the read head at the time moment t was written by the write head at the time moment $t - T(t)$:

$$x_r(t) = x_w(t - T(t)) \tag{2}$$

Clearly, during the time range $[t-T(t), t]$ the tape has travelled the distance equal to $\int_{t-T(t)}^{t} v(\tau)\,\mathrm{d}\tau$ and this distance must be equal to L:

$$\int_{t-T(t)}^{t} v(\tau)\,\mathrm{d}\tau = L \tag{3}$$

The above formula is the *tape equation* in the integral form. It relates the tape speed function $v(t)$ to the effective delay time $T(t)$. In case of constant speed on the range $[t - T(t), t]$ the formula turns into $v \cdot T(t) = L$ giving

$$T(t) \equiv L/v \tag{4}$$

We can refer to L/v as *steady-state delay time*.

By introducing the "total distance travelled by the tape"

$$V(t) = \int v(\tau)\,\mathrm{d}\tau \tag{5}$$

(which is understood in the sense that $V(t)$ is *some* arbitrary antiderivative of $v(t)$) the tape equation can be rewritten in the difference form

$$V(t) - V(t - T(t)) = L \tag{6}$$

It is convenient to choose the constant of integration in $V(t)$ in such a way that

$$x_w(t) = V(t) \tag{7a}$$
$$x_r(t) = V(t) - L \tag{7b}$$

Alternatively, if we wish to choose the constant term of $V(t)$ from some other considerations, (7) can be enforced by the choice of the origin of the coordinate axis x. From this point on we will assume that (7) always holds.

By taking the time derivative of (6):

$$v(t) - v(t - T(t)) \cdot \left(1 - \frac{\mathrm{d}}{\mathrm{d}t}T(t)\right) = 0$$

and peforming algebraic transformations, we obtain the tape equation in the differential form:

$$\frac{\mathrm{d}}{\mathrm{d}t}T(t) = 1 - \frac{v(t)}{v(t - T(t))} \tag{8}$$

By describing the relationship between tape-speed and delay-time, the tape equation can allow us to produce tape-like behaviour using an ordinary variable-length ringbuffer-based digital delay. In order to achieve this, the equation must be solved either analytically or numerically.

2.1. Suitability of equation forms for numerical solution

The differential form (8) of the tape equation doesn't contain information about the distance between the heads. Thus, there is no "built-in error correction mechanism" in (8) and a straitforward numerical solution could exhibit errors which accumulate into an indefinitely large drift of $T(t)$. Intuitively, consider the following example. Suppose $v(t)$ varies for a while and then the variations stop: $v(t) = \mathrm{const}\ \forall t \geq t_0 - T(t_0)$. In this case the right-hand side of (8) will be zero $\forall t \geq t_0$ and thus any error accumulated in $T(t)$ will stay there forever.

Consequently, the differential form is not well suited for use in practical delay implementations. However in theoretical work it can be useful in combination with differential equation solvers, such as the ones found in CAS (computer algebra system) software, which often expect the equation to be supplied in the differential form.

The integral form (3) contains a different potential numerical drift source. (3) suggests an incrementally computed moving sum as a numerical implementation, where the new terms of the form $v(t)\Delta t$ will be incrementally added and the old terms $v(t - T(t))\Delta t$ will be subtracted. However, even if what we add exactly equals what we subtract later, addition and subtraction of the same value might not totally cancel each other due to limited precision of floating point calculations. This error also can accumulate.

Note that, if we instead use fixed point calculations in the moving sum, the addition and subtraction of equal values will exactly cancel. So, if we can make sure that we add and subtract exactly the same values, there will be no drift. However, as we shall see in the further discussion, it will be even easier to simply use the difference form (6), where we spare the subtraction of $v(t-T(t))\Delta t$ and therefore don't need to consider the resulting error.

2.2. Analytical solution for an instantaneous jump in speed

It is educative to consider the case of a single instantaneous jump in the tape speed, the speed being constant at all other times. In this case there is a simple analytic solution to the tape equation. Indeed, let

$$v(t) = \begin{cases} v_0 & \text{if } t < 0 \\ v_1 & \text{if } t \geq 0 \end{cases}$$

Let $V(0) = 0$, giving

$$V(t) = \begin{cases} v_0 t & \text{if } t < 0 \\ v_1 t & \text{if } t \geq 0 \end{cases} \tag{9}$$

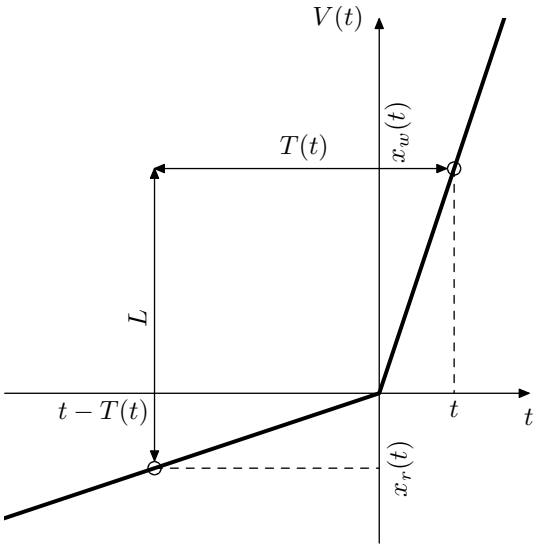


Figure 1: Graphical interpretation of (6).

and we choose the origin of coordinate x so that (7) holds.

Now we want to substitute (9) into (6), thus obtaining $T(t)$. It is highly instructive to use the graphical interpretation of (6) given in Fig. 1. The figure represents the graph of $V(t)$ defined by (9) with highlighted points $x_w(t)$ and $x_r(t)$ corresponding to the write and read head positions at time t. By (1) the vertical distance between these points is L, and by (2) the horizontal distance between these points is $T(t)$. Thus, both write and read heads move along the curve $V(t)$ in such a way that the vertical distance between these points is always L, thereby defining the horizontal distance between them, which is $T(t)$.[1]

We could use Fig. 1 to construct the explicit formula for $T(t)$. From (7a) we have $x_w(t) = V(t)$. Then using Fig. 1 we obtain $x_r(t) = x_w(t) - L$ and $t - T(t) = V^{-1}(x_r(t))$. Combining all these formulas together yields

$$T(t) = t - V^{-1}(V(t) - L) \tag{10}$$

where $V^{-1}(V(t) - L)$ is simply the time at which the signal, which is currently being picked up, was recorded.[2]

Now returning to the specific form of $V(t)$ given by (9) and looking at Fig. 1 we are having two obvious results:

$$\begin{aligned} T(t) &= L/v_0 \quad \text{if } t \le 0 \\ T(t) &= L/v_1 \quad \text{if } t \ge L/v_1 \end{aligned} \tag{11}$$

For $0 \le t \le L/v_1$ (and this is specifically the case shown in Fig. 1) to find the total time $T(t)$ we have to add the time duration corresponding to the right semiplane part of $T(t)$ and the one corresponding to the left semiplane part:

$$T(t) = t + \frac{L - v_1 t}{v_0} = \frac{L}{v_0} + \left(1 - \frac{v_1}{v_0}\right) t \tag{12}$$

[1]Note that this interpretation and Fig. 1 itself are not limited to the specific shape of $V(t)$ defined by (9), but apply for arbitrary $V(t)$.

[2]Of course, (10) could have been directly obtained from (6). By obtaining it using Fig. 1 instead, we have given an intuitive interpretation to (10).

(note that (12) gives $T(0) = L/v_0$ and $T(L/v_1) = L/v_1$, the same values as given by (11), corresponding to the fact that $T(t)$ must be continuous).

Thus $T(t)$ varies linearly on the transition range $t \in [0, L/v_1]$. More specifically, $T(t)$ changes from the old steady-state delay time to the new one and the transition duration is equal to the new steady-state time. This change produces the commonly known pitch jump effect from a sudden change of the tape speed. This jump has an obvious explanation in terms of tape speed, but now we can also explain in in terms of delay time. The duration of the pitch-shifting transition is exactly equal to the new steady-state delay time, thus, if feedback is present, the pitch-shifted signal will be recorded back into exactly one echo period of the "new steady-state", staying in the feedback loop until it decays or until a new speed change occurs.

3. A NUMERICAL SOLUTION FOR ARBITRARY VARIATIONS IN SPEED

If $v(t)$ is not known in advance, we can't compute (10) analytically and need to develop a numerical method. We want this method to be usable for practical delay implementations, therefore we want it to be computationally efficient and not suffer from the drift problem explained in Sec. 2.1.

3.1. Properties of a digital ringbuffer

We intend to implement a "tape delay" by combining a numerical solution of the tape-equation with a variable-length ringbuffer-based delay. Before we continue to discuss the solution of the tape equation, it is helpful to address some details regarding ringbuffer-based delays. The following discussion assumes that the sampling period and, respectively, the sampling frequency are unity: $f_s = 1$.

Consider the range of delay times supported by an ordinary ringbuffer-based delay. Clearly there is maximum delay time, limited by the ringbuffer's capacity. The minimum delay time is in principle zero. So, we have

$$0 \le T(t) \le T_{\max} \tag{13}$$

However there are two factors further limiting that range.

The first factor is interpolation, which is necessary to support delay times which are not an integer number of samples. Most interpolators need to consider some samples both before and after the interpolation point. Thus (13) (for a symmetric interpolator) turns into

$$\max\{\Delta - 1, 0\} \le T(t) \le T_{\max} - (\Delta - 1) \tag{14}$$

where Δ is half-width of the interpolator's kernel. This limitation can be however worked around by reducing the interpolator's order and/or window size when the interpolation is done at the edges of the range. This might be particularly desirable for $T \approx 0$, e.g. if we're looking for a comb filtering effect.

Another factor affecting (13) and (14) appears if the delay is used inside a feedback loop. If the ringbuffer is structured so that output and input happen synchronously in the algorithm, a unit delay will implicitly be introduced into the loop, as the current output of the delay is never known when calculating the input. (Fig. 2). For an ordinary digital ringbuffer-based delay this solely means that the delay time is off by one sample, which can be either tolerated or compensated by adjusting the offset of the read head

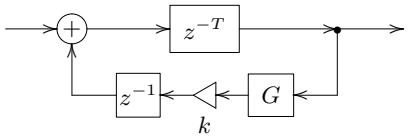


Figure 2: Unit delay in delay's feedback loop. k is feedback amount. G denotes some additional processing (not necessarily linear) which might occur in the loop.

by one sample. However if the ringbuffer delay is used as a basis for a tape delay emulation, either of the two mentioned options will distort the tape equation's solution and thus might potentially break the exact-repetition nature of the tape delay feedback in case of modulated tape speed. Therefore we would rather avoid the introduction of the extra unit delay altogether.

This can be achieved by splitting the processing of the delay's sample tick into two separate parts: the reading and the writing part. The read is processed first, then the feedback path, then the writing part. This eliminates the extra unit delay. In this case, however, the current input sample of the delay is not being written into the delay buffer until the end of processing and thus is not available for the read interpolation. This increases the lower boundary of supported $T(t)$ by one sample compared to (14):

$$\max\{\Delta, 1\} \le T(t) \le T_{\max} - (\Delta - 1) \tag{15}$$

unless we would be willing to solve the implicit equation arising out of the instantaneous dependency of delay's output signal on delay's input.

In this paper we will assume that the tape delay is to be used in a feedback loop and will develop the algorithm details under the assumption of split read/write processing.

3.2. Tape equation variables in discrete time

Let n be the discrete time-index. Since we assumed that the sampling period is unity, we have: $t = n$. Let's also choose the length scale so that the distance L between the heads is also unity: $L = 1$. The tape speed v expressed in these units means the fraction of the distance between the heads travelled over one sample period.

In order to be able to numerically apply the tape equation we need to keep the information about the tape speed values in the past, where the time range of interest is $[t, t - T(t)]$, where t is the current time. In Sec. 2.1 we gave reasons to choose the difference form (6) of the tape equation as the basis of the numerical solution, therefore, rather than storing the past values of $v(t)$, we will store the past values of its integral $V(t)$.

We want to use $T(t)$ obtained via the tape equation to control the delay time of a ringbuffer-based digital delay. It is a natural choice therefore to extend the ringbuffer elements to also contain the values $V[n]$ along with the stored audio signal samples. The value $V[n]$ will be contained in the same element as the audio signal recorded by the delay at time n. Intuitively, the write head simultaneously records the audio signal and the signal $V(t)$ onto the tape. Assuming that V is defined by

$$V(t) = \int_0^t v(\tau)\,\mathrm{d}\tau$$

or, in discrete time

$$V[n] = \sum_{i=0}^{n} v[i] \tag{16}$$

we can compute $V[n]$ incrementally

$$V[n] = V[n-1] + v[n] \tag{17}$$

Note that (16) and (17) imply that $v(t) = v[n]$ is assumed to be constant over a duration of one sample period, which is a common simplification when dealing with changing control values in discrete time. We will work further under this assumption, unless otherwise noted.

3.3. Representation of tape coordinates

$V[n]$ is an infinitely growing sequence, and thus there are dangers of increasing precision losses and/or overflow in (17), if floating point representation is used. Instead of trying to estimate whether this could be an issue with practical sampling rates and running times, we are going to use fixed point representation which will provide an elegant way to avoid such concerns altogether. We will be using this fixed point representation for $V[n]$ and any other values expressing the tape coordinate or derived values such as tape speed.

(15) effectively provides the upper and the lower bound to $v[n]$:

$$\max\{\Delta, 1\} \le 1/v[n] \le T_{\max} - (\Delta - 1)$$

that is

$$0 < \frac{1}{T_{\max} - (\Delta - 1)} \le v[n] \le \frac{1}{\max\{\Delta, 1\}} \le 1 \tag{18}$$

Under the restriction (18) our fixed point numbers will need to have a sign bit and two integer bits,[3] the remaining bits are to be used for the fractional part. Thus from a 64 bit integer we'll make a 2.61 signed fixed point number. So we'll have better precision than if we used 64-bit IEEE 754 floats, which have only 52 fractional bits of mantissa. The fixed point representation also gives constant precision across the entire value range, which is more appropriate for our purposes.

At any particular time the integral (3) and respectively the difference (6) are dependent only on the history within time range $[t - T(t), t]$. Thus, the tape coordinate values, which we are interested in are contained within the range $[V(t) - L, V(t)]$ (or marginally outside). Since $L = 1$ and $t = n$, this range can be written as $[V[n] - 1, V[n]]$. This suggests that we should be fine using fixed point arithmetic modulo 8 for the computations involving tape coordinates.

Under the assumption of two's complement binary representation of 64-bit integers, arithmetic computations modulo 8 will occur automatically for 64-bit integer-based 2.61 fixed point numbers. More precisely, the addition, subtraction and multiplication will be automatically done modulo 8, while comparisons will require special care.

Instead of simply comparing two numbers for $>$, $\ge$, $<$ or $\le$, we will need to compare their signed difference to zero, e.g.

$$(a > b)_{(\mathrm{mod}\ 8)} \overset{\mathrm{def}}{\iff} a - b > 0 \tag{19}$$

[3] We will use the fixed point representation not only for speed, but for anything which includes length into its dimension. That is the reason to have the extra bits in the integer part, which should become more clear through the discussion that follows.

This way the comparison is made "on the shortest path" between a and b (or, more precisely, between elements of the respective congruence classes).

3.4. Ringbuffer index arithmetic

Ringbuffer indexing also uses modulo arithmetic. A standard ringbuffer implementation generally needs only index increments, in which case modulo arithmetic technically means doing a trivial wraparound. Usually such wraparound is implemented either by a conditional check, or, if ringbuffer size N is a power of 2, by bitmasking with $N-1$. For the purposes of this algorithm the index arithmetic will need more of the modulo techniques.

Assuming ringbuffer indices n always take values within the range $[0, N-1]$, let's introduce the concept of a modular range of ringbuffer indices:

$$[n_1, n_2)_{(\mathrm{mod}\ N)} = \begin{cases} [n_1, n_2) & \text{if } n_2 \geq n_1 \\ [n_1, N) \cup [0, n_2) & \text{if } n_2 < n_1 \end{cases}$$

The length of the range is thus

$$\left|[n_1, n_2)_{(\mathrm{mod}\ N)}\right| = \begin{cases} n_2 - n_1 & \text{if } n_2 \geq n_1 \\ n_2 - n_1 + N & \text{if } n_2 < n_1 \end{cases}$$

If N is a power of 2, then, under the assumption of two's complement binary integer representation, modular range length can be computed without evaluating a conditional:

$$\left|[n_1, n_2]_{(\mathrm{mod}\ N)}\right| = (n_2 - n_1)\ \&\ (N-1)$$

where $\&$ denotes bitwise "and".

Notably, we can't use a comparison definition similar to (19) for ringbuffer indices, because we cannot assume that the comparison needs to be done "on the shortest path".[4] Therefore instead of index comparisons, we will be comparing the lengths of index ranges, which is a well-defined operation.

In binary search within the ringbuffer contents we will need to be able to find the middle of a modular range. Clearly, we can't simply take the average of the range's bounds, as the result could be off by $N/2$. Instead, we'll need to divide the range's length by 2 and use this new length to obtain the middle position and the new bounds.[5]

3.5. Tracking of the read position

In a ringbuffer-based delay implementation, the "write head" progressively cycles through the underlying array of the ringbuffer, advancing by one array index per sample. The position of the "read head" is computed each time anew, by subtracting the delay time (in samples) from the write head's position. In the proposed implementation of the "tape delay" we update the ringbuffer's read position incrementally instead. The read position must be fractional in order to support $T(t)$ which are not integer sample counts. We will therefore need to incrementally track the following values:

- the write position in the ringbuffer's array n_w
- the write head's tape coordinate x_w
- the read position in the ringbuffer's array $n_r + \nu_r$, where n_r is the integer part and $\nu_r \in [0, 1)$ is the fractional part[6]
- if ringbuffer size N is not a power of two, we might want to explicitly store the size of the ringbuffer contents (which is equal to $|[n_r, n_w)_{(\mathrm{mod}\ N)}|$) and update it with changes to n_w and n_r, thereby saving one evaluation of a conditional, when ringbuffer content size is needed.

We will further assume that the formal indexing of $V[n]$ is identical to the ringbuffer element indexing modulo N. We will also notate and understand the fractional indexing of $V[n]$ as

$$V(n_r + \nu_r) = V[n_r] + (V[n_r + 1] - V[n_r]) \cdot \nu_r \qquad (20)$$

Note that the linear interpolation in (20) is chosen because it is exactly correct given the previously stated assumption that $v(t)$ is constant over the duration of one sample.

As discussed in Subsec. 3.1, we wish to implement a delay usable in a feedback context, meaning that the reading of the audio output signal from the ringbuffer should occur prior to and separately from the writing of the audio input signal. Using (10) and Fig. 1 we can construct the following algorithm for processing a single sample step of such delay.

0. In the beginning of the step the variables are set like follows:
   - n_w is pointing to the ringbuffer element which is about to be written to
   - x_w contains the previous coordinate of the write head
   - $n_r + \nu_r$ is pointing to the ring buffer element which was read from in the previous step.
1. Compute the new value of x_w using (17) and (in agreement with (7a)) write it into $V[n_w]$. This doesn't depend on the delay's input signal value and therefore can be done in the beginning.[7]
2. Compute $x_r = x_w - L = x_w - 1$ (according to (7b)) and then search for the new $n_r + \nu_r$ such that
   $$V(n_r + \nu_r) = x_r \qquad (21)$$
   The details of the search will be explained in Sec. 3.6.
3. Read the delay's output from the ringbuffer at position $n_r + \nu_r$.
4. Send the delay's output sample through the feedback loop all the way to the delay's input.
5. Write the delay's input into the ringbuffer at position n_w and advance n_w by one array index.

[4] With tape coordinate representation we have introduced additional headroom into the modulus to make sure that the distance between the values which we would want to compare or to subtract doesn't exceed half of the modulus. We could have introduced similar headroom into ringbuffer indices, but that would raise efficiency concerns.

[5] It could be useful to incrementally store the range's length in a separate variable during binary search.

[6] Actually, only n_r needs to be incrementally tracked, while ν_r will be computed each time.

[7] The fact that we compute the new value of x_w in the beginning and advance n_w in the end is matched in the later proposed approach to the algorithm initialization. If cache line aliasing between the read and write positions becomes a concern, we could perform the writing of x_w into $V[n_w]$ in step 5 instead (note that such change doesn't affect the mentioned initialization). However this excludes $V[n_w]$ from the allowed range of the search in step 2, effectively decreasing the upper bound of the tape speed in (18) and respectively increasing the minimum attainable delay time, unless the need to access $V[n_w]$ is handled "manually" during the search.

3.6. Updating of the read position

In step 2 of the "tape delay" processing algorithm introduced in Sec. 3.5 we need to perform a search for the solution of (21). We will split the search in two parts. First (the search itself) we search for n_r such that $x_r \in [V[n_r], V[n_r+1]]$. Having found n_r, we can solve (20) in respect to ν_r, obtaining:[8]

$$\nu_r = \frac{x_r - V[n_r]}{V[n_r+1] - V[n_r]} \tag{22}$$

According to (18) the sequence $V[n]$ is monotonic and we need to search only in the forward direction from the previous value of n_r. The simplest possible implementation of the search therefore is: repeatedly advance n_r by one array index, comparing $V[n_r]$ to x_r. This is actually not as bad as it may seem. Because in the most commonly occuring case, when $v(t)$ doesn't change much during $[t - T(t), t]$, or at least doesn't speedup noticeably, we will need to advance n_r only one or two times, until we found the new value of n_r. In case of a speedup by a factor of K we will however need to perform ca. K steps before we find the new position n_r. Thus, during speedup transitions, the operation count will increase by a factor of K.

This can be improved to an increase only by a factor of $\log_2 K$. Since $V(t)$ is monotone, it can be inverted by bisection [7], which in discrete time case effectively takes the form of binary search followed by the subsample position refinement at the end. Thus, we intend to do binary search within $V[n]$ between the old value of n_r and the new value of n_w.[9] In isolation, this is not such a good idea. Because now the binary search range contains the entire region of the tape between the read and write heads, and we are going to binary-search this entire range (performing ca. 10-20 search steps) each time, even in case of small speed variations. This results in not $O(\log K)$ computation complexity but rather $O(\log T(t))$.

We can, however, improve the selection of the search range. Starting from the old read position n_r we check the value $V[n_r + 2]$. If $V[n_r + 2] \geq x_r$, then we take $n_r + 2$ as the upper bound of the range and n_r as the lower bound. Otherwise we know that the new read position doesn't lie between n_r and $n_r + 2$ anyway, so we update n_r to take the value $n_r + 2$ and now take a step of 4, probing the value $V[n_r + 4]$ (formerly $V[n_r + 6]$) and taking the range from n_r to $n_r + 4$, if successful. Otherwise we update n_r to $n_r + 4$ and take a step of 8 etc, until we finally find the range containing x_r.

Notably, during this process of searching for the initial range we have to be careful not to cross the write position n_w. As mentioned before, we can't use a comparison approach like the one of (19) for ringbuffer indices. Therefore, instead of comparing n_r to n_w we need to compare the step size to the length of the modular range $[n_r, n_w)_{(\mathrm{mod}\ N)}$.

It's not difficult to see that the described way of searching for the initial range has computation complexity of $O(\log K)$ and so does the binary searching on the range found in this way, thus our entire implementation has $O(\log K)$ complexity.[10]

3.7. Initialization

The previous discussion of the tape delay algorithm was assuming that we are somewhere in the middle of the running time and all incremental variables are properly set by the previous sample's processing. However, how do we set these variables initially?

Let's assume that we are using linear interpolation for reading the audio signal at non-integer positions $n_r + \nu_r$. In this case we propose to initially let

$$\begin{aligned} V[0] &= -L = -1 \\ V[1] &= 0 \\ n_r &= 0 \\ n_w &= 2 \\ x_w &= 0 \end{aligned}$$

where $V[0]$ and $V[1]$ are stored in the ringbuffer's array elements 0 and 1, and the audio signal part of these elements is initialized to zero. Then, as long as $0 \leq x_w < 1$, the read head will interpolate between elements 0 and 1 of the array, thereby producing zero output, as if we were reading from a clean tape.[11]

If instead of linear interpolation we use e.g. cubic interpolation, then the same initialization idea will look like:

$$\begin{aligned} V[0] = V[1] &= -L = -1 \\ V[2] = V[3] &= 0 \\ n_r &= 1 \\ n_w &= 4 \\ x_w &= 0 \end{aligned}$$

Other interpolation schemes can be treated similarly.

3.8. Sparse storage of $V[n]$

The need to store $V[n]$ in addition to the audio in the ringbuffer significantly increases the memory usage by the delay. E.g. if the audio consists of two 32-bit float stereo channels, by adding a 64-bit fixed point $V[n]$ to each sample we double the amount of used memory.

The memory requirements can be reduced if the tape speed doesn't change on every sample, but at a lower "control rate". This could however introduce audible artifacts due to "steppy" pitch changes corresponding to the steppy nature of the tape speed. In such case, instead of considering the tape speed constant on the

[8] We should remember that we are using arithmetic modulo 8 when dealing with tape coordinates, and, strictly speaking, the division of two values both having the tape coordinate units in (22) hasn't been defined. This division doesn't require any special treatment though, since the distances from x_r to $V[n_r]$ and from $V[n_r]$ to $V[n_r + 1]$ on the tape coordinate axis cannot exceed $\max\{v[n]\}$, which according to (18) is 1.

[9] Remembering to use the previously discussed modular arithmetic rules, for both tape coordinate and index.

[10] Obviously, if $\log K < 1$ and especially if $\log K < 0$, the complexity bound $O(\log K)$ looks highly questionable. It's not difficult to realize that dropping K below 1 (tape slowdown) doesn't further reduce the number of computations. This suggests that the computation complexity estimation for arbitrary speed changes is, strictly speaking, $O(\max\{1, \log K\})$. For the simplicity of notation, however, we can agree to understand $O(\log K)$ as a somewhat informal way of writing $O(\max\{1, \log K\})$.

[11] Thus, we have a very quick initialization procedure, not requiring to fill the entire ringbuffer, neither with the values of $V[n]$ nor with the zero audio samples. This is quite useful if the implementation is used in a plugin in a DAW, where plugin reset times may become an issue.

range from $V[n]$ to $V[n+1]$ we could consider it linearly increasing.[12] The linear function (20) is thereby replaced by a quadratic one and (22) turns into a quadratic equation solution formula.

Of course, other than linear segments of $v[n]$ could be used, as long as they can be inverted (in reasonable computation time) to find the fractional position.

3.9. Antialiasing of speedups

The interpolated readout of the ringbuffer works well if the delay speed is almost constant or is slowing down. However a speedup effectively shifts the pitch of delay's signal up, thereby creating frequencies above Nyquist threshold, which ordinary interpolation doesn't try to suppress. It is, however, possible to reformulate the polynomial interpolation in a way allowing arbitrary cutoffs below Nyquist.

Any polynomial interpolation of a sequence y_n can be alternatively expressed as a convolution with a kernel:

$$\mathcal{L}[y_n](t) = \sum_n y_n \lambda(t-n)$$

where $\lambda(t)$ is the kernel and $\mathcal{L}[y_n]$ denotes a continuous time function which is the result of the interpolation of y_n. The interpolation's kernel λ is obtained by interpolating the Kronecker delta sequence with the interpolator in question:

$$\lambda(t) = \mathcal{L}[\delta_n](t)$$

and thus consists of polynomial segments. The kernel normally corresponds to a continuous-time lowpass filter with a cutoff close to Nyquist. We can lower the kernel's cutoff by a factor of $K \in \mathbb{R}$ by stretching the kernel K times along the time axis and simultaneously reducing its amplitude K times:

$$\mathcal{L}[y_n](t) = \sum_n y_n \frac{\lambda((t-n)/K)}{K} \tag{23}$$

Thus, by expressing the interpolation as convolution we can arbitrarily change the interpolation filter's cutoff, which allows us to use the interpolator to suppress unwanted frequencies below Nyquist.[13]

The speedup situation can be detected by comparing the reading speed (which is simply the current tape speed) to the speed at which the signal was recorded. According to (17) the recording speed is simply equal to $V[n_r] - V[n_r - 1]$.[14] Note, however, that the computation complexity of such antialiasing is $O(K)$, where K is the speedup factor. We could put an upper bound on the complexity by artificially clamping the factor K used in (23) at the obvious cost of some unfiltered aliasing in case K exceeds the clamping value.

[12] This might require storing $v[n]$ alongside $V[n]$, unless we want to guess $v[n]$ from neighboring values of $V[n]$.

[13] One has to take care to make sure that the interpolator, which is stretched along the time axis by the factor K, does not attempt to read ahead of n_w or too far behind n_r, where the samples either haven't been written into the buffer yet or have been overwritten with newer values. The issue can be addressed e.g. by reducing the cutoff factor K, if we get too close to the boundaries of the valid index range.

[14] The forward difference $V[n_r + 1] - V[n_r]$ also can be in principle taken, since (23) is only an approximation of a proper resampling process anyway.

The equation (23) is only a somewhat rough approximation, done under the assumption that the tape speed doesn't change very quickly, because it assumes that the samples y_n are equally spaced in time. In reality the samples are not traversed by the read head at equal time intervals, this happens at different times and at different speeds. A more correct version of (23) therefore might be

$$\mathcal{L}[y_n](t) = \sum_n y_n \frac{\lambda((t-\tau_n)/\max\{K_n, 1\})}{\max\{K_n, 1\}} \tag{24}$$

where τ_n is the time at which the sample y_n is traversed and K_n is the respective upsampling factor. Note that some of the times τ_n in (24) occur in the future. They still may be known in advance, if we can know the variations of the tape speed in advance, in which case (24) is still perfectly implementable.[15]

3.10. Extending the bounds of tape speed

The tape speed bounds imposed by (15) can be significantly extended.

If we are having no feedback or if we are willing to solve the implicit feedback equation, then (14) applies instead of (15) and we might want to support arbitrarily large speeds. Notably, we still have quite a lot of headroom which we could add into the proposed fixed point format. E.g. we could use 11.52 signed fixed point numbers, thereby increasing the upper boundary of the tape speed headroom (18) by a factor of 512 giving $v \leq 512$.

Conversely, as $1/v$ reaches the maximum capacity of the ring buffer (more precisely defined by (14) and (15)) we attain the maximum delay time possible with our proposed approach based on (10). At this point, however, we could add the ideas of the straightforward implementation of tape delay, which would correspond to slowing down the medium below the natural speed of the 1:1 ratio (the medium moves by 1 sample during one sample tick). In our setup this would correspond to advancing n_w by a fractional amount less than 1. The writing to the ringbuffer will be occurring "in between" the sample ticks with proper interpolation, or, if antialiasing is desired, with proper "downsampling filtering".

In this way we can achieve arbitrarily small and even zero tape speeds, corresponding to arbitrarily large to infinite delay times. In principle one could even go in negative speed direction.[16] A limitation of this approach is that in this case the bandwidth of the signal transmitted through the medium will be reduced, since the "tape's sampling rate" is lower than the outside sampling rate.

4. RESULTS

In order to get an idea of the performance of our method, we have made a comparison between an ordinary ringbuffer-based digital variable-length delay (VL), variable sample-rate delay (VS), Huovilainen's method (H) and the proposed method (P), all methods using SIMD-ified Catmull–Rom interpolation of two stereo

[15] We only need to know the future tape speed but not the future values of the audio signal. This means that (24) is implementable even for a feedback delay without introducing any additional latency into the feedback. The latency will be introduced only into how the delay responds to the speed control signal.

[16] As the speed goes to zero and negative values, $V(t)$ stops being strictly monotonic, and one needs to take additional decisions during the search for the solution of (21).

Method	x1	x2	x10	x100
(VL)	21	21	21	21
(VS)	46	58	163	1320
(H)	37	47	97	725
(P)	39	39	39	39
(P')	39	38	54	110

Table 1: Performance cost (in TSC clocks per processed sample) of different algorithms at different tape speeds.

Method	x1	x2	x10	x100
(VL)	23	75	277	2531
(VS)	48	112	419	3810
(H)	39	101	353	3235
(P')	41	92	310	2620

Table 2: Performance cost of different algorithms during speedups, with enabled antialiasing.

channels of audio[17]. The comparison is presented in Table 1 where (P') is the performance of the proposed method in the case of a speedup from 1x to the specified speed. The measurements were taken by letting the algorithm run for a large number of samples at different equivalent tape speeds, each time taking the average number of TSC[18] clocks per processed sample. The relative measurement error is ca. 5%. The 1x tape speed corresponds to 1 tape 'sample' processed per 1 outside sample. The identical performance costs of the proposed method at x1 and x2 speeds are due to the fact that the bisection method is searching over the same initial range of two entries in both cases.

We also measured the performance of the proposed method with enabled antialiasing of speedups. Whilst not measured directly, we have assumed that the antialiasing overhead would be identical between different methods and added the same overhead to other measured results.[19] The respective performance comparison is presented in Table 2, where the tape speed of the delay line is being switched from x1 to the specified speed.

5. CONCLUSION

In this paper, we introduced the tape equation, which allows the translation of variations of the medium speed into variations of delay time, thus allowing to implement tape-like modulation behavior using ordinary digital delays. In certain cases, when the speed variation pattern is known in advance, this translation can be done analytically. We have also introduced a numerical method to be used in cases where analytical solution is not possible. Compared to previous methods, which have $O(v)$ time complexity, the presented method has mostly $O(1)$ time complexity, except during speedup transitions, where the complexity is $O(\log K)$. If antialiasing of speedups is desired, the time complexity of speedup transitions grows to $O(K)$, however, the complexity can be bounded by artificially clamping the antialiasing filter's cutoff.

The proposed numerical method is also exact in the sense that the error from time discretization manifests solely as the tape speed being assumed constant over the duration of each sample, whereas precision losses occur only in the quantization of the speed values and in the final computation of the subsample read position for the ringbuffer (where it's totally negligible). Thus, all error is effectively contained in the time- and level-quantization of the tape speed, the solution of the tape equation itself being exact. Furthermore, the time-quantization error can be further improved by assuming linearly changing speed during each sample.

The proposed method can be used in implementations of delays where tape-like modulation behavior is desired. Compared to previously used approaches, our method has comparable or better CPU load at all delay times.

[17] Audio samples of each of these methods are available at the accompanying website: `https://github.com/julian-parker/DAFX-Tape`

[18] Time Stamp Counter, a processor's internal high-precision timer.

[19] In (VS) the antialiasing will need to be done constantly unless we take additional effort to store the information about the recording speed. In other methods this information is already available, thus the antialiasing may be done just during the speedups.

6. REFERENCES

[1] A. Huovilainen, "Enhanced digital models for analog modulation effects," in *Proc. Int. Conf. Digital Audio Effects (DAFx-05)*, Madrid, Spain, 2005, pp. 155–160.

[2] C. Raffel and J. O. Smith, "Practical modeling of bucket-brigade device circuits," in *Proc. Int. Conf. Digital Audio Effects (DAFx-10)*, Graz, Austria, Sep. 2010, pp. 50–56.

[3] M. Holters and J. D. Parker, "A combined model for a bucket brigade device and its input and output filters," in *Proc. Int. Conf. Digital Audio Effects (DAFx-18)*, Aveiro, Portugal, 2018.

[4] S. Arnardottir, J. S. Abel, and J. O. Smith III, "A digital model of the Echoplex tape delay," in *Proc. of the 25th Audio Engineering Society Convention*. Audio Engineering Society, 2008.

[5] J. Dattorro, "Effect design, part 2: Delay line modulation and chorus," *Journal of the Audio Engineering Society*, vol. 45, no. 10, pp. 764–788, 1997.

[6] D. Rocchesso, "Fractionally addressed delay lines," *IEEE Transactions on Speech and Audio Processing*, vol. 8, no. 6, pp. 717–727, 2000.

[7] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery, *Numerical recipes in C*, vol. 2, Cambridge University Press, 1996.

A COMBINED MODEL FOR A BUCKET BRIGADE DEVICE AND ITS INPUT AND OUTPUT FILTERS

Martin Holters

Department of Signal Processing and Communication
Helmut Schmidt University
Hamburg, Germany
martin.holters@hsu-hh.de

Julian D. Parker

Native Instruments GmbH
Berlin, Germany
julian.parker@native-instruments.com

ABSTRACT

Bucket brigade devices (BBDs) were invented in the late 1960s as a method of introducing a time-delay into an analog electrical circuit. They work by sampling the input signal at a certain clock rate and shifting it through a chain of capacitors to obtain the delay. BBD chips have been used to build a large variety of analog effects processing devices, ranging from chorus to flanging to echo effects. They have therefore attracted interest in virtual analog modeling and a number of approaches to modeling them digitally have appeared. In this paper, we propose a new model for the bucket-brigade device. This model is based on a variable sample-rate, and utilizes the surrounding filtering circuitry found in real devices to avoid the need for the interpolation usually needed in such a variable sample-rate system.

1. INTRODUCTION

Bucket brigade devices (BBDs) were invented in the late 1960s at Philips Research Labs [1], as a method of introducing a time-delay into an analog electrical circuit. These chips were subsequently used to build a large variety of analog effects processing devices, ranging from chorus to flanging to echo effects. Well-known BBD-based devices include the Memory Man delay/echo pedal and the Electric Mistress flanger effect from Electro-Harmonix, as well as a series of chorus designs produced by Roland, starting in the mid 70s with the chorus circuit of the JC-120 amplifier and culminating with the Dimension-D rack unit and the chorus included in the Juno-60 synthesizer.

There have been a number of approaches to modeling BBD devices digitally. Raffel [2] concentrated on the filtering and non-linear behavior of the BBD, without treating the dynamic behavior of the BBD when the clock-rate is varied. Huovilainen [3] and Mačák [4] both model the BBD in the context of a flanger effect. The latter uses a variable sample rate delay to model the BBD delay behavior, whilst the former uses a method based on storing the times at which an input arrived to the BBD. Variable sample rate digital delay-lines have been described in the past, primarily for the use in physical models of acoustic systems [5]. Recently, methods have been proposed for emulating tape and BBD-like behaviour by storing the previous 'speed' of the system (clock-rate in a BBD) [6].

The presented technique is built on the observation that BBD chips, due to their sampling nature, are typically used in conjunction with low-pass filters to prevent aliasing. We propose a novel approach, modeling the BBD together with these filters. The BBD itself will be trivially modeled as what it is: a fixed length but variable sample rate delay-line. The main novelty of the proposed approach is that the resampling between the audio sampling rate and the variable BBD clock rate utilizes the filters already present in the analog circuit and hence avoids the need for additional interpolation. The lack of direct interpolation results in more accurate fitting of the frequency response of the circuit as no additional filtering is introduced from the interpolation. Additionally, the distortion produced by the constant variation of the interpolation filter is avoided. Experimental results confirm that the method leads to a faithful BBD model.

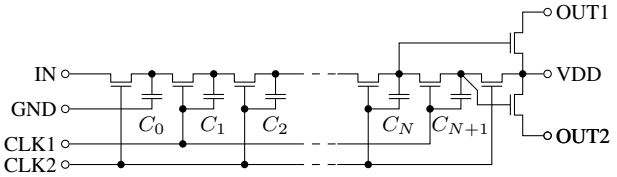


Figure 1: *Simplified BBD schematic*

2. WORKING PRINCIPLE OF BUCKET BRIGADE DEVICES

Figure 1 shows a simplified schematic of a typical BBD. We have omitted additional field effect transistors that, together with the shown ones, form tetrodes to reduce unwanted coupling between the stages. While we leave the detailed explanation of the propagation principle to [1] and the reason for using tetrodes to [7], we shall briefly look at the input and output circuitry.

While the input transistor, controlled by CLK2, is open, capacitor C_0 follows the input voltage $u_{\mathrm{BBD}}(t)$ between the IN and GND terminals. Closing the input transistor hence corresponds to sampling the input signal at the time instant t_0 of the respective clock edge. The two clock signals CLK1 and CLK2 are complementary, so that the transistor connecting C_0 and C_1 opens (nearly) in the same instant and the signal sample $u_{\mathrm{BBD}}(t_0)$ is transferred to C_1 while C_0 returns to the reference voltage [1].

Let the following clock edges occur at times $t_1, t_2, \ldots$. Note that only at every second clock edge t_n, n even, the input transistor transitions from open to closed, sampling the input. Thus at any time, only half the capacitors carry the signal, while the others are at the reference voltage. In the metaphor of the bucket brigade, this corresponds to half the buckets being filled with water and transported in one direction, while the other half is empty and is transported back (to be filled again).

With every edge, the charge representing the signal is propagated to the next capacitor, that is after the clock edge at t_n, capacitor C_{n+1} holds $u_{\mathrm{BBD}}(t_0)$. It follows that the signal arrives at capacitor C_N at t_{N-1} and drives the first output terminal OUT1 while the second output terminal OUT2 is in high impedance state.

After the next clock edge at t_N, capacitor C_{N+1} holds $u_{\text{BBD}}(t_0)$ and drives OUT2 while OUT1 is in high impedance state. This continues until at t_{N+1}, the next signal sample $u_{\text{BBD}}(t_2)$ arrives at capacitor C_N and drives OUT1 while OUT2 is in high impedance state again. Therefore, application circuits combine the two outputs, so that the signal sampled at t_0 is present at the combined output from t_{N-1} to t_{N+1}, that is

$$y_{\text{BBD}}(t) = u_{\text{BBD}}(t_n) \text{ for } t_{n+N-1} \leq t < t_{n+N+1},\ n \text{ even}. \quad (1)$$

In other words, for a constant clock rate, the signal is not only delayed by $N/2$ clock periods (corresponding to N clock edges). It is also convolved with a rectangular pulse of one clock period width giving rise to a high-frequency attenuation depending on the BBD clock rate. For all commercially available BBDs, N is even, so that if the input sampling occurs at every t_n, n even, the output changes its value at every t_n, n odd, which we will assume for simplicity during the development of the proposed model.

In addition to the desired functionality of delaying the signal, due to their analog nature, BBD chips usually also alter the signal in unwanted ways. In particular, the long chain of active semiconductor stages acting upon the signal typically adds noise and may introduce non-linear distortions. Additionally, losses and tolerances in the capacitances may lead to non-unity overall gain. However, this paper focuses on the sampling and delay behavior and does not consider these parasitic effects.

Finally, as a direct consequence of the working principle, there are several inherent sources of aliasing distortion in the BBD system – firstly there are frequency components present at the input of the BBD that exceed the effective Nyquist frequency of the BBD. These components will be reflected around the BBD Nyquist frequency. Most BBD circuits include a filter at the input to suppress this behavior. Secondly, there are the image-spectra created by sample-and-hold nature of the output of the BBD chip. Similarly to at the input, most BBD circuits include an output filter to suppress these images. These types of aliasing (at least when present in small quantities) can be considered to be desirable for the expected sound of a BBD and should be reproduced by a digital model.

3. PROPOSED MODEL

We propose to model the BBD as a delay-line of fixed length, operating at another, potentially varying sampling rate, the BBDs clock rate, similar to [4]. However, instead of using simple interpolation for the necessary sampling rate conversions, we will exploit the fact that typical application circuits contain low-pass filters at the BBDs input and output. These are responsible to prevent aliasing from the sampling and reconstruction process of the BBD. We will make use of exactly these anti-aliasing filters for the necessary resampling. The transformation of these filters to the digital domain will be carried out using a modified impulse-invariant transform similar to the approach taken in [8], as this facilitates dealing with different, asynchronous sampling rates on input and output side.

3.1. Input filter

Perfect reconstruction of the analog signal $u(t)$ from its samples $\bar{u}(k) = u(kT_\text{s})$, where $T_\text{s} = 1/f_\text{s}$ is the sampling interval, can be understood as subjecting a train of Dirac impulses weighted with $\bar{u}(k) \cdot T_\text{s}$ to an ideal low-pass filter, band-limiting it to the Nyquist frequency. Here, we replace the ideal low-pass filter with the input low-pass filter $H_{\text{in}}(s)$ found in front of the BBD, which is assumed to have sufficient attenuation at the Nyquist frequency (of the original audio sampling rate f_s) that an acceptable amount of aliasing remains. Our aim is to obtain samples $u_{\text{BBD}}(t_n)$ of the filter's output (being the BBD's input) at times t_n, n even, at which the BBD samples its input.

Let $H_{\text{in}}(s)$ be expanded into partial fractions as

$$H_{\text{in}}(s) = \sum_{m=1}^{M_{\text{in}}} \frac{r_{\text{in},m}}{s - p_{\text{in},m}} \quad (2)$$

where we may assume no non-negative powers of s to occur as $H_{\text{in}}(s)$ is a low-pass filter and further assume all poles $p_{\text{in},m}$ to be simple to simplify the following development. Then the corresponding impulse response can easily be found to be

$$h_{\text{in}}(t) = \begin{cases} \sum_{m=1}^{M_{\text{in}}} r_{\text{in},m} \cdot e^{p_{\text{in},m} t} & \text{for } t \geq 0 \\ 0 & \text{otherwise.} \end{cases} \quad (3)$$

Exciting the filter with a single Dirac impulse weighted with $\bar{u}(k) \cdot T_\text{s}$ at time kT_s, we obtain the corresponding output

$$u_{\text{BBD,k}}(t) = \begin{cases} \bar{u}(k) \cdot T_\text{s} \sum_{m=1}^{M_{\text{in}}} r_{\text{in},m} \cdot e^{p_{\text{in},m}(t - kT_\text{s})} & \text{if } t \geq kT_\text{s} \\ 0 & \text{otherwise.} \end{cases} \quad (4)$$

Now let the time be decomposed as $t = (l_n + d_n)T_\text{s}$ where l_n is an integer and $0 \leq d_n < 1$. Then

$$u_{\text{BBD,k}}((l_n + d_n)T_\text{s}) = \begin{cases} \bar{u}(k) \sum_{m=1}^{M_{\text{in}}} g_{\text{in},m}(d_n) \cdot \bar{p}_{\text{in},m}^{\,l_n - k} & \text{if } l_n \geq k \\ 0 & \text{otherwise} \end{cases} \quad (5)$$

where $\bar{p}_{\text{in},m} = e^{p_{\text{in},m} T_\text{s}}$ and

$$g_{\text{in},m}(d_n) = T_\text{s} \cdot r_{\text{in},m} \cdot \bar{p}_{\text{in},m}^{\,d_n}. \quad (6)$$

Further rewriting as

$$u_{\text{BBD,k}}((l_n + d_n)T_\text{s}) = \sum_{m=1}^{M_{\text{in}}} g_{\text{in},m}(d_n) \cdot x_{\text{in},m,k}(l_n) \quad (7)$$

with

$$x_{\text{in},m,k}(l_n) = \begin{cases} \bar{u}(k) \cdot \bar{p}_{\text{in},m}^{\,l_n - k} & \text{if } l_n \geq k \\ 0 & \text{otherwise,} \end{cases} \quad (8)$$

we see that the latter can be expressed recursively as

$$x_{\text{in},m,k}(l_n) = \begin{cases} \bar{p}_{\text{in},m} \cdot x_{\text{in},m,k}(l_n - 1) & \text{if } l_n > k \\ \bar{u}(k) & \text{if } l_n = k \\ 0 & \text{otherwise.} \end{cases} \quad (9)$$

Now by the superposition principle, the filter response to the complete input signal is given by

$$u_{\text{BBD}}((l_n + d_n)T_\text{s}) = \sum_k u_{\text{BBD,k}}((l_n + d_n)T_\text{s}) = \sum_{m=1}^{M_{\text{in}}} g_{\text{in},m} \cdot x_{\text{in},m}(l_n) \quad (10)$$

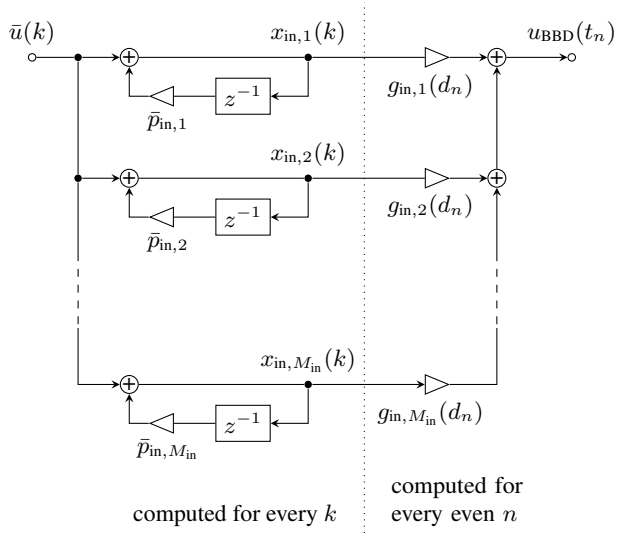


Figure 2: *Digital realization of the input filter where $t_n = (k + d_n) \cdot T_s$, n even, are the sampling instants of the BBD input*

where

$$x_{\text{in},m}(l_n) = \sum_k x_{\text{in},m,k}(l_n) = \bar{p}_{\text{in},m} \cdot x_{\text{in},m}(l_n - 1) + \bar{u}(l_n) \quad (11)$$

constitutes a simple first-order recursive filter. This leads to the digital realization shown in figure 2. For every input sample $\bar{u}(k)$, the parallel recursive parts are updated, and for every sample needed at the BBD input, the weighted sum is evaluated. The weights of the latter depend on the fractional offset d_n of the BBD sampling instant t_n within the audio rate sampling interval.

3.2. Output filter

The development for the output filter is similar but differs in two aspects: Now, the input samples occur at the BBD clock rate while the output samples are needed at the fixed audio sampling rate, and the input samples have to be treated as consecutive rectangular pulses instead of Dirac impulses. That is, $y_{\text{BBD}}(t)$ is piecewise constant in intervals $[t_n, t_{n+2})$, n odd.

For the following development, it is helpful to work with a sequence of differences $\Delta(n) = y_{\text{BBD}}(t_n) - y_{\text{BBD}}(t_{n-1})$, n odd, with associated step functions

$$\epsilon_n(t) = \begin{cases} \Delta(n) & \text{if } t \geq t_n \\ 0 & \text{otherwise} \end{cases} \quad (12)$$

such that

$$y_{\text{BBD}}(t) = \sum_n \epsilon_n(t). \quad (13)$$

Similar to the previous development, we first determine the filter output produced by a single step $\epsilon_n(t)$ and consider the output filter $H_{\text{out}}(s)$ to be decomposed into partial fractions as

$$H_{\text{out}}(s) = \sum_{m=1}^{M_{\text{out}}} \frac{r_{\text{out},m}}{s - p_{\text{out},m}}. \quad (14)$$

The response to a unit step (Heaviside step function) is

$$h_{\text{out}}(t) = \begin{cases} \sum_{m=1}^{M_{\text{out}}} \frac{r_{\text{out},m}}{p_{\text{out},m}} \left(e^{p_{\text{out},m} t} - 1\right) & \text{if } t \geq 0 \\ 0 & \text{otherwise} \end{cases} \quad (15)$$

$$= \begin{cases} H_0 + \sum_{m=1}^{M_{\text{out}}} \frac{r_{\text{out},m}}{p_{\text{out},m}} e^{p_{\text{out},m} t} & \text{if } t \geq 0 \\ 0 & \text{otherwise} \end{cases} \quad (16)$$

where $H_0 = -\sum_{m=1}^{M_{\text{out}}} \frac{r_{\text{out},m}}{p_{\text{out},m}}$. It follows trivially that the response $y_n(t)$ to a single $\epsilon_n(t)$ is

$$y_n(t) = \begin{cases} H_0 \Delta(n) + \Delta(n) \sum_{m=1}^{M_{\text{out}}} \frac{r_{\text{out},m}}{p_{\text{out},m}} e^{p_{\text{out},m}(t - t_n)} & \text{if } t \geq t_n \\ 0 & \text{otherwise.} \end{cases} \quad (17)$$

Now let $\bar{y}_n(k) = y_n(kT_s)$ be samples of the individual responses taken at the original audio sampling rate, and $t_n = (l_n - 1 + d_n)T_s$, where l_n is an integer and $0 < d_n \leq 1$. Then

$$\bar{y}_n(k) = \begin{cases} H_0 \Delta(n) + \Delta(n) \sum_{m=1}^{M_{\text{out}}} \frac{r_{\text{out},m}}{p_{\text{out},m}} \bar{p}_{\text{out},m}^{k - l_n + 1 - d_n} & \text{if } k \geq l_n \\ 0 & \text{otherwise} \end{cases} \quad (18)$$

where $\bar{p}_{\text{out},m} = e^{p_{\text{out},m} T_s}$. We further rewrite as

$$\bar{y}_n(k) = \sum_{m=1}^{M_{\text{out}}} x_{\text{out},m,n}(k) + \begin{cases} H_0 \Delta(n) & \text{if } k \geq l_n \\ 0 & \text{otherwise} \end{cases} \quad (19)$$

with

$$x_{\text{out},m,n}(k) = \begin{cases} \bar{p}_{\text{out},m}^{k - l_n} g_{\text{out,m}}(d_n) \Delta(n) & \text{if } k \geq l_n \\ 0 & \text{otherwise} \end{cases} \quad (20)$$

where

$$g_{\text{out,m}}(d_n) = \frac{r_{\text{out},m}}{p_{\text{out},m}} \bar{p}_{\text{out},m}^{1 - d_n}. \quad (21)$$

Similar to the input filter, we can express this using the recursion

$$x_{\text{out},m,n}(k) = \begin{cases} \bar{p}_{\text{out},m} \cdot x_{\text{out},m,n}(k - 1) & \text{if } k > l_n \\ g_{\text{out,m}}(d_n) \Delta(n) & \text{if } k = l_n \\ 0 & \text{otherwise} \end{cases} \quad (22)$$

and by superimposing all input step functions get the recursive first order subsystem

$$x_{\text{out},m}(k) = \sum_n x_{\text{out},m,n}(k) = \bar{p}_{\text{out},m} \cdot x_{\text{out},m}(k - 1) + \sum_{\substack{n \\ k = l_n}} g_{\text{out},m}(d_n) \Delta(n) \quad (23)$$

where the driving term includes all steps occurring during the past sampling interval, that is all odd n such that $(k-1)T_s < t_n \leq kT_s$. Further, by definition

$$H_0 \sum_{\substack{n \\ l_n \leq k}} \Delta(n) = H_0 y_{\text{BBD}}(kT_s). \quad (24)$$

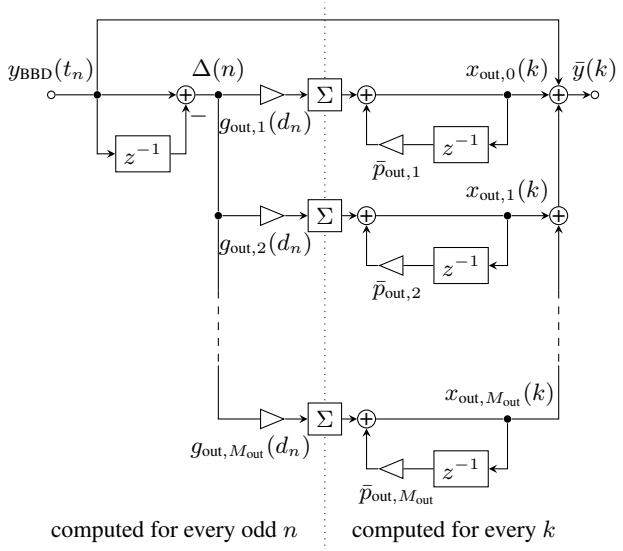


Figure 3: *Digital realization of the output filter where $t_n = (k - 1 + d_n) \cdot T_s$, n odd, are the switching instants of the BBD output*

The final output then is the sum of these first-oder subsystems, i.e.

$$y(k) = H_0 y_{\text{BBD}}(kT_\text{s}) + \sum_{m=1}^{M_\text{out}} x_{\text{out},m}(k), \tag{25}$$

leading to the digital realization shown in figure 3, where the Σ-nodes on the border between the sampling rates shall denote the accumulation of the inputs on the n side over one interval of the k side.

Algorithm 1 shows pseudo code for the complete model of BBD and filters. Note that the inner loop (lines 6–19), which performs the operations running at the BBD clock rate, is executed before the k-th input sample $\bar{u}(k)$ is processed in line 21, and therefore effectively does the processing for the time interval between the $k-1$-th and k-th sample. The BBD samples are assumed to be stored in a queue of fixed length N, accessed with $\text{enqueue}()$ and $\text{dequeue}()$ to insert and retrieve a sample, respectively.

3.3. Real-valued systems

In above derivation, all coefficients are potentially complex-valued. Of course, unless they are already real, they occur in conjugate complex pairs, so that two complex-valued first-order systems can be combined into one real-valued second-order system. The calculation is straight-forward and we only present the resulting equivalent sub-systems in figures 4 and 5, where the m-th and $\hat{m}$-th pole are assumed to form a conjugate pair and the (real-valued) coefficients for the formed input-filter sub-system are given by

$$a_{1,\text{in},m} = 2\cos(\angle \bar{p}_{\text{in},m}) \tag{26}$$

$$a_{2,\text{in},m} = -|\bar{p}_{\text{in},m}|^2 \tag{27}$$

$$b_{0,\text{in},m}(d_n) = \beta_{\text{in},m} \cdot |\bar{p}_{\text{in},m}|^{d_n} \cdot \cos(\angle r_{\text{in},m} + d_n \angle \bar{p}_{\text{in},m}) \tag{28}$$

$$b_{1,\text{in},m}(d_n) = -\beta_{\text{in},m} \cdot |\bar{p}_{\text{in},m}|^{d_n+1} \cdot \cos\left(\angle r_{\text{in},m} + (d_n - 1)\angle \bar{p}_{\text{in},m}\right) \tag{29}$$

where $\beta_{\text{in},m} = 2T_\text{s} \cdot |r_{\text{in},m}|$. Similarly, the coefficients for the

Algorithm 1 Proposed BBD and filters model

```
1:  n ← 0
2:  x_{in,m} ← 0 for m = 1, ..., M_in
3:  x_{out,m} ← 0 for m = 1, ..., M_out
4:  y_{BBD,old} ← 0
5:  for all k do
6:      while t_n < kT_s ∨ (n odd ∧ t_n = kT_s) do
7:          d_n ← t_n − (k − 1)T_s
8:          if n even then
9:              enqueue( Σ_{m=1}^{M_in} g_{in,m}(d_n) · x_{in,m} )
10:         else
11:             y_BBD ← dequeue()
12:             Δ ← y_BBD − y_{BBD,old}
13:             y_{BBD,old} ← y_BBD
14:             for m ∈ 1, ..., M_out do
15:                 x_{out,m} ← x_{out,m} + g_{out,m}(d_n) · Δ
16:             end for
17:         end if
18:         n ← n + 1
19:     end while
20:     for m ∈ 1, ..., M_in do
21:         x_{in,m} ← p̄_{in,m} x_{in,m} + ū(k)
22:     end for
23:     y(k) = H_0 · y_{BBD,old} + Σ_{m=0}^{M_out} x_{out,m}(k)
24:     for m ∈ 1, ..., M_out do
25:         x_{out,m} ← p̄_{out,m} x_{out,m}
26:     end for
27: end for
```

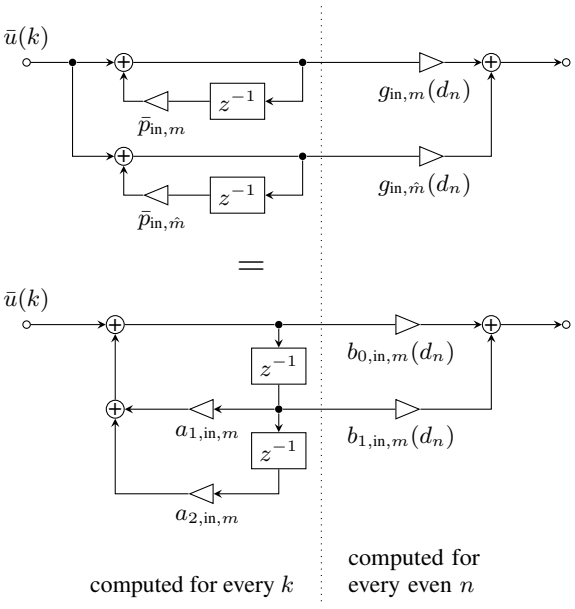


Figure 4: *Two complex-valued first-order systems for a conjugate complex pole pair of the input filter and the equivalent real-valued second-order system*

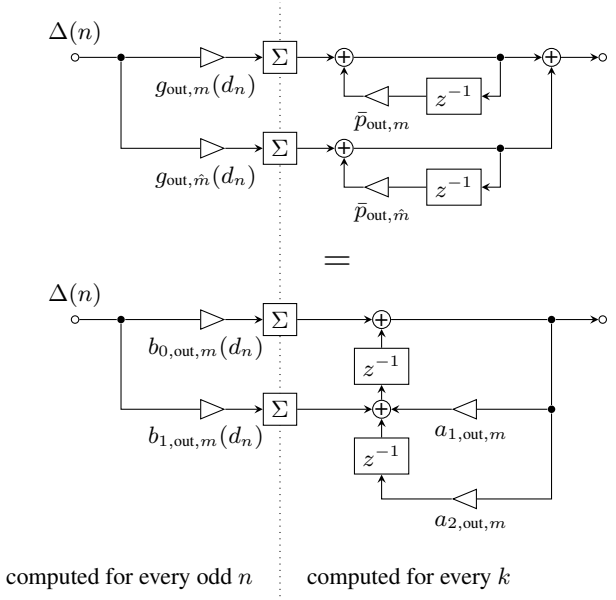


Figure 5: *Two complex-valued first-order systems for a conjugate complex pole pair of the output filter and the equivalent real-valued second-order system*

formed output-filter sub-system are given by

$$a_{1,\text{out},m} = 2\cos(\angle\bar{p}_{\text{out},m}) \tag{30}$$

$$a_{2,\text{out},m} = -|\bar{p}_{\text{out},m}|^2 \tag{31}$$

$$b_{0,\text{out},m}(d_n) = \beta_{\text{out},m} \cdot |\bar{p}_{\text{out},m}|^{1-d_n} \cdot \cos(\angle r_{\text{out},m} + (1-d_n)\angle\bar{p}_{\text{out},m}) \tag{32}$$

$$b_{1,\text{out},m}(d_n) = -\beta_{\text{out},m} \cdot |\bar{p}_{\text{out},m}|^{2-d_n} \cdot \cos\left(\angle r_{\text{out},m} - d_n\angle\bar{p}_{\text{out},m}\right) \tag{33}$$

where $\beta_{\text{out},m} = 2\left|\frac{r_{\text{out},m}}{p_{\text{out},m}}\right|$. Note that by precomputing constants and reusing common terms, computing the b coefficients for one second-order sub-system requires evaluation of one exponential and two cosine functions. Alternatively, given the limited range of d_n, one may use polynomial approximations or look-up tables for the b coefficients. An analysis of the effects of approximation errors is beyond the scope of this paper, however.

4. RESULTS

In the following, we consider the BBD and filter combination as found in the chorus effect of the Juno-60 synthesizer. As a detailed circuit analysis is beyond the scope of this paper, we only state the relevant aspects. Both the input and output filter are sixth-order filters that can be decomposed into a first-order high-pass filter (for adjusting bias voltages) and a fifth-order low-pass filter. We will only include the latter in our combined BBD/filter model. Numerical circuit analysis gives the coefficients of table 1, corresponding to the frequency responses shown in figure 6.

We first validate the model by studying a situation where we can analytically derive the expected output: sinusoidal input and a BBD clock with constant rate f_{BBD} so that the time interval between

Table 1: *Coefficients of the input and output filters*

	H_{in}	H_{out}
r_1	$251\,589$	5092
r_2	$-130\,428 - 4165\text{i}$	$11\,256 - 99\,566\text{i}$
r_3	$-130\,428 + 4165\text{i}$	$11\,256 + 99\,566\text{i}$
r_4	$4634 - 22\,873\text{i}$	$-13\,802 - 24\,606\text{i}$
r_5	$4634 + 22\,873\text{i}$	$-13\,802 + 24\,606\text{i}$
p_1	$-46\,580$	$-176\,261$
p_2	$-55\,482 + 25\,082\text{i}$	$-51\,468 + 21\,437\text{i}$
p_3	$-55\,482 - 25\,082\text{i}$	$-51\,468 - 21\,437\text{i}$
p_4	$-26\,292 - 59\,437\text{i}$	$-26\,276 - 59\,699\text{i}$
p_5	$-26\,292 + 59\,437\text{i}$	$-26\,276 + 59\,699\text{i}$

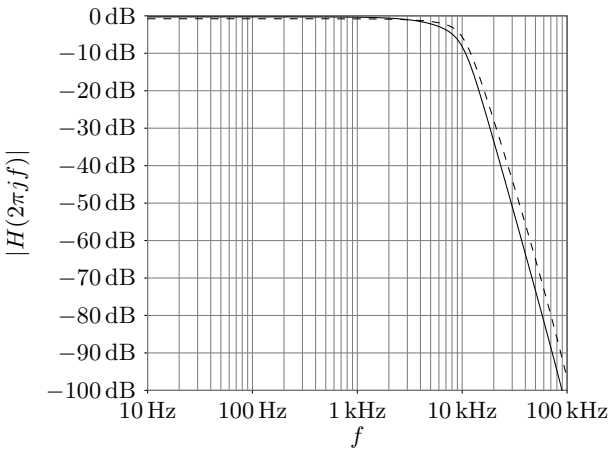


Figure 6: *Frequency response of the input filter (—) and output filter (- -)*

two clock edges is $t_n - t_{n-1} = \frac{1}{2f_{\text{BBD}}}$, where we assume the clock to have $50\,\%$ duty cycle, i.e. $t_n - t_{n-1} = t_{n-1} - t_{n-2}$, as is typical in BBD applications. For signals band-limited to $f_{\text{BBD}}/2$ and ignoring aliasing distortion introduced by the BBD, the BBD may then be treated as the linear filter

$$H_{\text{BBD}}(i\omega) = e^{-i\omega\frac{N}{2f_{\text{BBD}}}} \cdot \operatorname{sinc}\left(\frac{\omega}{2\pi f_{\text{BBD}}}\right) \tag{34}$$

where N is the number of stages of the BBD and $\operatorname{sinc}(x) = \frac{\sin(\pi x)}{\pi x}$. The first factor is the phase shift due to the delay, the second factor the amplitude distortion due to presenting rectangular pulses at the output. For the input signal

$$u(k) = \sin\left(2\pi\tfrac{f_0}{f_\text{s}}k\right) \tag{35}$$

we therefore expect the output

$$y(k) = a \cdot \sin\left(2\pi\tfrac{f_0}{f_\text{s}}k + \varphi\right) \tag{36}$$

where

$$a = \operatorname{sinc}\left(\tfrac{f_0}{f_{\text{BBD}}}\right) \cdot |H_{\text{in}}(2\pi i f_0)| \cdot |H_{\text{out}}(2\pi i f_0)| \tag{37}$$

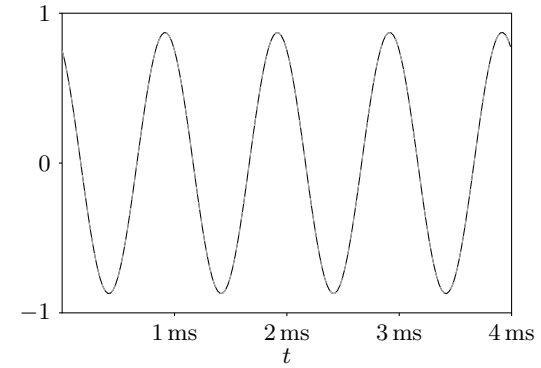
$$\varphi = -\pi f_0 \tfrac{N}{f_{\text{BBD}}} + \angle H_{\text{in}}(2\pi i f_0) + \angle H_{\text{out}}(2\pi i f_0). \tag{38}$$

This expected output is compared in figure 7 with the output computed using the proposed model. Here, we choose $f_0 = 1\,\text{kHz}$, $f_\text{s} = 44.1\,\text{kHz}$, $N = 256$, and $f_{\text{BBD}} = 50\,\text{kHz}$. As can be seen, model output and theoretically expected output are in good agreement, small differences remain however. These are caused by aliasing due to the non-perfect attenuation of the filters at and above the Nyquist frequency. Figure 8 shows the same configuration, but with an instant step in f_{BBD} occurring at $t = 5\,\text{ms}$. A smooth change in the frequency of the output can be seen, as is expected for a BBD. This is in contrast to a simple digital delay-line, which would exhibit a discontinuity at the output when subjected to a discontinuous change in delay time. This behaviour arises because the effective pitch of the output of the BBD compared to its input depends on the ratio of f_{BBD} between the instant when the signal was sampled by the BBD and when it exits the BBD.

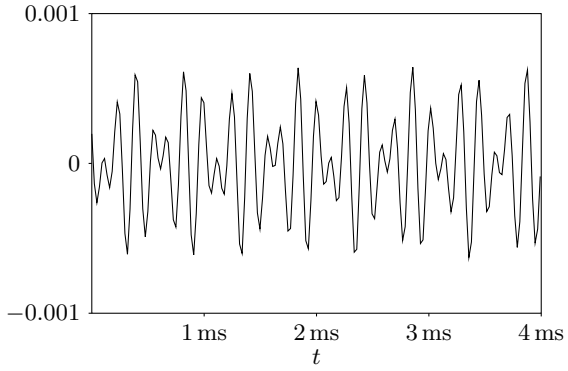
As a more practically relevant scenario, we compare the model output to the output of the BBD output filter recorded from a real Juno-60 synthesizer. The BBD clock period is controlled by a triangular LFO signal, leading to piecewise constant pitch shifts, alternatingly upwards and downwards. It is worth noting that the minimum BBD clock rate is about $26\,\text{kHz}$, so that the analog circuit may already introduce aliasing distortion itself, as discussed previously.

To allow a meaningful comparison, several extra considerations are necessary:

- The first-order high-pass filters previously omitted have to be included. This is done by converting them to digital filters using the bilinear transform and applying them to the input signal before and the output signal after running the BBD model.
- Measurements in the circuit showed that the BBD amplifies the signal by approximately $2.3\,\text{dB}$ which is also included in the simulation.



(a) Output of the proposed model (—) and the theoretically expected output (– –)



(b) Difference between proposed model output and theoretically expected output

Figure 7: *Comparison of the output of the proposed model and theoretically expected output for a sinusoid at* $f_0 = 1\,\text{kHz}$ *sampled at* $f_s = 44.1\,\text{kHz}$ *delayed by a BBD with* $N = 256$ *stages clocked at* $f_{BBD} = 50\,\text{kHz}$

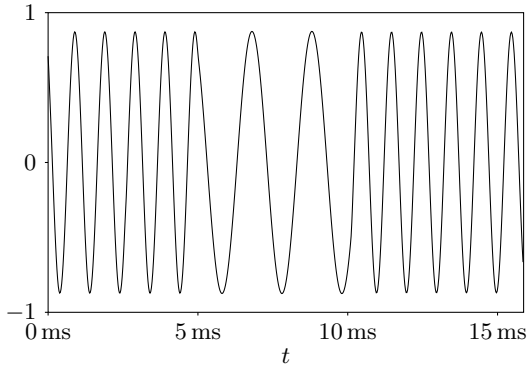


Figure 8: *Model output (—) for a sinusoid of* $f_0 = 1\,\text{kHz}$ *sampled at* $f_s = 44.1\,\text{kHz}$ *delayed by a BBD with* $N = 256$ *stages clocked at* $f_{BBD} = 50\,\text{kHz}$ *before* $t = 5\,\text{ms}$ *and* $f_{BBD} = 25\,\text{kHz}$ *afterwards.*

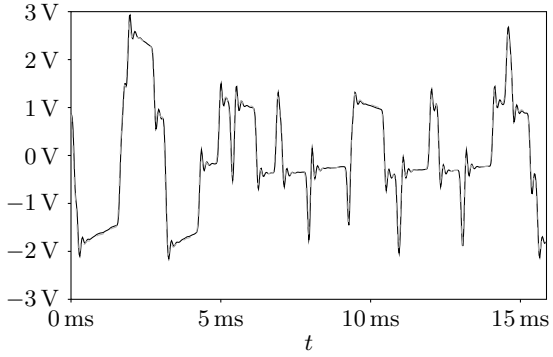


Figure 9: *Model output (—) and recorded output (– –) for a C_{maj} chord input*

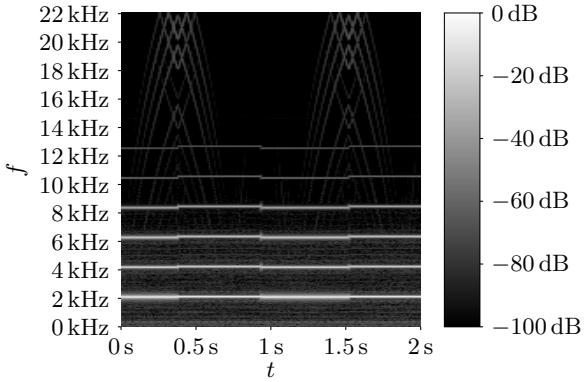
- Instead of recording the BBD clock signal, which would necessitate a very high sampling rate, we reconstruct the clock rate by estimating phase, frequency, and amplitude of a triangular oscillator. Visual inspection of the measurements and simulation results reveals a mismatch in obtained delay time of up to $0.13\,\mathrm{ms}$, varying with time, which is likely due to non-perfect clock rate reconstruction.
- While the filter parameters used are derived from nominal component values, the tolerance of the real components will lead to slightly different filtering behavior.

Figure 9 shows a time domain comparison of the model output and the recorded output when driven with a C_{maj} chord (C_4, E_4, G_4). As can be seen, despite the uncertainties mentioned above, very good agreement is achieved. Closer inspection reveals a small time offset (less than $0.04\,\mathrm{ms}$ in the shown excerpt) between model output and recording. This impedes interpretation of the difference signal, as it is dominated by peaks around the steep edges of the signal due to the misalignment.

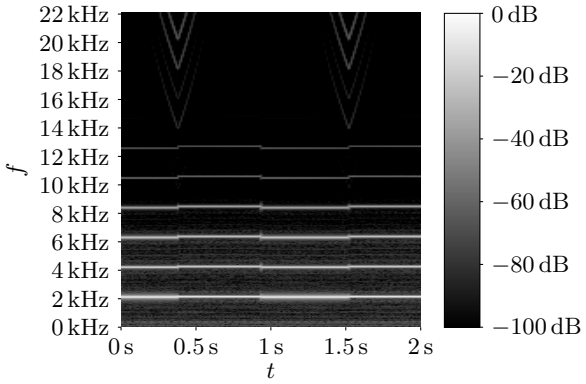
To specifically study the effects of aliasing, the highest note available on the Juno-60, C_7 nominally at 2093 Hz, is used as input. The spectrograms in figure 10 reveal a small amount of aliasing in the recorded output of the analog device (figure 10(c)) and slightly more aliasing in the digital model output at the sampling rate $f_\mathrm{s} = 44.1\,\mathrm{kHz}$ (figure 10(a)), as was to be expected.

This extra aliasing in both examples is produced by the assumption of the input to be an impulse-train, as well as the reflection around the audio Nyquist frequency of the image-spectra generated by the sample-and-hold nature of the BBD output. Helped by the existing presence of aliasing in the analog BBD system, this extra aliasing is not audible. In applications where the extra aliasing is problematic, the easiest remedy is oversampling. This is almost tautological, but note that here, a significant portion of the computation happens at the BBD clock rate and is independent of the audio sampling rate, making oversampling especially attractive. The effectiveness can be seen in figure 10(b), where the sampling rate is doubled to $f_\mathrm{s} = 88.2\,\mathrm{kHz}$ and the extra aliasing due to the model vanishes.

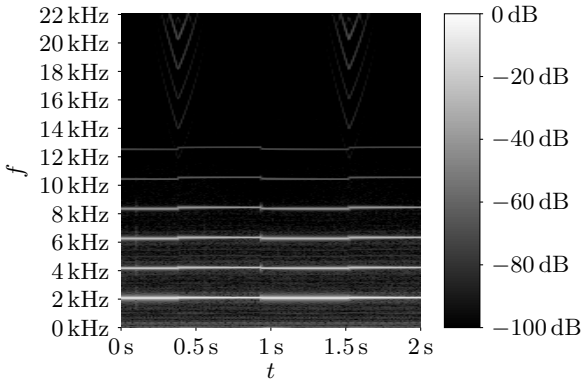
Evaluation based on the Objective Difference Grade (ODG) [9] (advanced mode) as computed with GstPEAQ [10] confirms the high similarity of the simulation to the measurements. Table 2 shows the ODG for the two stimuli discussed above as well as for a



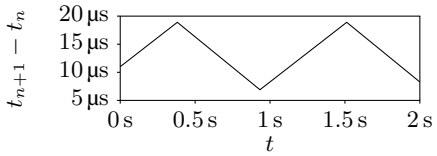
(a) Model output at $f_\mathrm{s} = 44.1\,\mathrm{kHz}$



(b) Model output at $f_\mathrm{s} = 88.2\,\mathrm{kHz}$



(c) Recorded output



(d) Interval between BBD clock edges

Figure 10: *Spectrograms of the model output and the recorded output of the BBD output filter in a Juno-60 synthesizer, excited with a C_7 (nominally 2093 Hz), and the LFO-controlled BBD clock edge interval*

Table 2: *Objective difference grade (ODG) comparing measurement of BBD output and simulation for different stimuli and sampling rates*

stimulus	$f_\mathrm{s} = 44.1\,\mathrm{kHz}$	$f_\mathrm{s} = 88.2\,\mathrm{kHz}$
C_2	-0.696	-0.670
C_7	-0.530	-0.393
C_{maj}	-0.646	-0.611

low pitched note, C_2 nominally at $65.41\,\mathrm{hertz}$[1]. The ODG ranges between 0 ("differences imperceptible") and -4 ("differences very annoying"), where -1 corresponds to "differences perceptible but not annoying". Hence the achieved results could be classified as "differences not annoying if perceptible at all", with the expected slight improvement for the higher sampling rate. Considering that, as outlined above, the BBD model is not the only source of differences between simulation and measurements, this is a clear success.

5. CONCLUSION

BBD chips sample a signal and delay it by a constant number of sampling intervals. To prevent aliasing from high-frequency content present in the input signal and to suppress image-spectra in the output signal, typical application circuits contain low-pass filters at their input and output. In this paper, we have proposed a model for the combination of the BBD and the filters. In fact, the BBD is trivially modeled as delay-line of constant length, working at the same clock rate as in the analog circuit. The key idea is that the resampling between the audio sampling rate and the BBD clock rate utilizes the filters already existing in the circuit. To this end, the filters are transformed into the digital domain by a modified impulse-invariant transform that allows the output to be taken or the input to be given at arbitrary time instants.

As verified with experimental results, the model thus obtained allows faithful reproduction of the analog system's behavior, even including aliasing distortion that may occur. However, the audio sampling rate has to be high enough that the filters have sufficient attenuation at the Nyquist frequency. Otherwise, additional aliasing distortion may be introduced. If necessary, this can be trivially prevented by oversampling.

6. REFERENCES

[1] F.L.J. Sangster and K. Teer, "Bucket-brigade electronics: new possibilities for delay, time-axis conversion, and scanning," *IEEE Journal of Solid-State Circuits*, vol. 4, no. 3, pp. 131–136, jun 1969.

[2] C. Raffel and J. O. Smith, "Practical modeling of bucket-brigade device circuits," in *Proc. 13th Int. Conf. Digital Audio Effects (DAFx-10)*, Graz, Austria, Sep. 2010, pp. 50–56.

[3] A. Huovilainen, "Enhanced digital models for analog modulation effects," in *Proc. 8th Int. Conf. Digital Audio Effects (DAFx-05)*, Madrid, Spain, 2005, pp. 155–160.

[4] J. Mačák, "Simulation of analog flanger effect using BBD circuit," in *Proc. 19th Int. Conf. Digital Audio Effects (DAFx)*, Brno, Czench Republic, 2016.

[1] The evaluated signals are available at `https://www.hsu-hh.de/ant/en/team/martin-holters/dafx2018-bbd`.

[5] D. Rocchesso, "Fractionally addressed delay lines," *IEEE Transactions on Speech and Audio Processing*, vol. 8, no. 6, pp. 717–727, 2000.

[6] V. Zavalishin and J. D. Parker, "Efficient emulation of tape-like delay modulation behavior," in *Proc. 21st Int. Conf. Digital Audio Effects (DAFx-18)*, Aveiro, Portugal, 2018.

[7] F.L.J. Sangster, "Integrated bucket-brigade delay line using MOS tetrodes," *Philips Technical Review*, vol. 31, pp. 266, 1970.

[8] J. Pekonen and M. Holters, "Nonlinear-phase basis functions in quasi-bandlimited oscillator algorithms," in *Proc. 15th Int. Conf. Digital Audio Effects (DAFx)*, York, UK, 2012, pp. 261–268.

[9] ITU Radiocommunication Assembly, *RECOMMENDATION ITU-R BS.1387-1 – Method for objective measurements of perceived audio quality*, ITU, 2001.

[10] M. Holters and U. Zölzer, "GstPEAQ – an open source implementation of the PEAQ algorithm," in *Proc. 18th Int. Conf. Digital Audio Effects (DAFx)*, Trondheim, Norway, 2015.

REMOVING LAVALIER MICROPHONE RUSTLE WITH RECURRENT NEURAL NETWORKS

Gordon Wichern and Alexey Lukin

iZotope, Inc.
Cambridge, MA, USA
alex@izotope.com

ABSTRACT

The noise that lavalier microphones produce when rubbing against clothing (typically referred to as *rustle*) can be extremely difficult to automatically remove because it is highly non-stationary and overlaps with speech in both time and frequency. Recent breakthroughs in deep neural networks have led to novel techniques for separating speech from non-stationary background noise. In this paper, we apply neural network speech separation techniques to remove rustle noise, and quantitatively compare multiple deep network architectures and input spectral resolutions. We find the best performance using bidirectional recurrent networks and spectral resolution of around 20 Hz. Furthermore, we propose an ambience preservation post-processing step to minimize potential gating artifacts during pauses in speech.

1. INTRODUCTION

The lavalier microphone (lav mic) is an invaluable tool for the audio engineer. By inconspicuously attaching near the mouth it allows the person wearing the microphone to move freely, minimizes visual distractions, and also helps to reduce reverberation and noise from the recording environment. Because lav mics are typically attached to a subjects wardrobe, they can sometimes rub against clothing creating an auditory disturbance often described as rustle. Lav mic rustle can overlap with speech in both time and frequency and vary in unpredictable ways based on how the person wearing the microphone moves their body. This makes developing an algorithm to automatically detect and remove rustle extremely challenging.

Traditional techniques for single-channel speech enhancement, e.g., spectral subtraction [1], work well for stationary background noise (e.g., air conditioner hum), but struggle in the presence of non-stationary disturbances, such as lav mic rustle. Recently, source separation approaches have achieved success in separating speech from complex non-stationary background noise, such as music, weather, or even other speech [2]. These techniques typically operate on a time-frequency representation of the signal, e.g., the spectrogram, and often take a supervised learning approach where a collection of clean speech and isolated noise samples are used to learn a model. Once trained, this model can obtain separated speech and noise signals when given noisy speech as input.

Time-frequency masking is one approach to single-channel speech separation which estimates the amount of speech and noise present in each spectrogram bin (i.e., the mask). This mask is then used as a time-varying filter to separate speech from noise. Recent advances in deep neural networks have drastically improved the ability to learn the nonlinear mapping function necessary to estimate time-frequency masks from noisy speech. The approaches of [3, 4, 5, 6] use feedforward network architectures where the mask for a frame of audio is predicted using input features from several surrounding frames. In this architecture, increasing the amount of temporal context requires increasing the dimension of the network input, which extends the size of the entire network. This increases the risk of overfitting and the resources necessary to train and deploy the network.

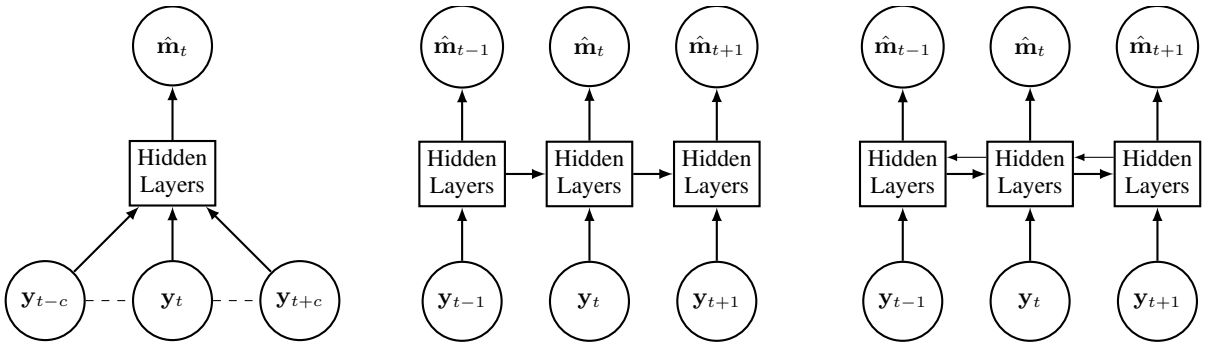
For this reason, recurrent architectures have demonstrated success on several sequential prediction tasks [7] like language translation, video captioning, speech recognition, and speech/noise separation. Recurrent architectures save an internal hidden state between time steps, and the appropriate context for a problem at hand can be learned from data. However, to avoid the vanishing/exploding gradient problem, gated architectures, such as the long short-term memory (LSTM) [8], must be used. Additionally, [9] showed improved performance on a speech noise separation task using a bidirectional LSTM (BLSTM) [10], which performs both a forward and backward pass over the data, thus incorporating future context at the cost of offline operation.

In this paper, we explore deep feedforward, recurrent LSTM, and BLSTM network architectures for removing lav-mic rustle from speech, a specific problem for audio engineers that, to the best of our knowledge, has not previously been explored in the literature. We begin by reviewing mask estimation approaches to single-channel source separation and different deep network architectures in Section 2. We benchmark the performance of our recurrent architectures against feedforward networks in terms of noise reduction and speech intelligibility and explore trade-offs in the spectral features used as network inputs in Section 3. Techniques for removing lav-mic rustle while maintaining a certain amount of background ambience to maintain the natural quality of the recording are explored in Section 4. Finally, conclusions and discussions of future work are provided in Section 5.

2. SPEECH SEPARATION NETWORKS

Our algorithm works on a mono mixture $y(t) = s(t) + n(t)$ of speech $s(t)$ corrupted by lav mic rustle $n(t)$. Given a training set with examples of isolated speech and rustle signals, we create mixtures with known ground truth to learn a mapping that estimates the clean speech signal $\hat{s}(t)$ from noisy mixture $y(t)$. Rather than operating on the time-domain waveform, our neural networks take as input the short-time Fourier transform (STFT) magnitude spectrogram of $y(t)$, denoted as $\mathbf{Y} = [\mathbf{y}_1, \mathbf{y}_2, ..., \mathbf{y}_T] \in \mathbb{R}^{d \times T}$, where d is the number of frequency bins.

We use the magnitude ratio mask as the time-varying filter for



(a) Feedforward (b) Recurrent (c) Bidirectional

Figure 1: Incorporating temporal context via multiple frame inputs (a) or hidden layer state propagation (b) and (c).

separating speech and rustle, which is defined as

$$\mathbf{m}_t = \frac{\mathbf{s}_t}{\mathbf{s}_t + \mathbf{n}_t}, \tag{1}$$

where t is the STFT frame (time) index, $\mathbf{s}_t$ are the spectral magnitude coefficients of clean speech, and $\mathbf{n}_t$ is the rustle magnitude spectrum. The division operation in (1) is performed element-wise. Because magnitude spectra $\mathbf{s}_t$ and $\mathbf{n}_t$ are nonnegative, the mask elements $\mathbf{m}_t$ from (1) are in the interval $[0, 1]$. The output of our neural network is $\hat{\mathbf{m}}_t$, which we use to obtain estimated magnitude spectra for separated speech and rustle, i.e.,

$$\hat{\mathbf{s}}_t = \hat{\mathbf{m}}_t \odot \mathbf{y}_t, \tag{2}$$

$$\hat{\mathbf{n}}_t = (\mathbf{1} - \hat{\mathbf{m}}_t) \odot \mathbf{y}_t, \tag{3}$$

where $\odot$ represents an element-wise product. We use (2) and (3) to obtain the estimated time-domain waveforms $\hat{s}(t)$ and $\hat{y}(t)$ through the inverse STFT, with phase information taken from the noisy mixture $y(t)$.

2.1. Network architectures

We estimate $\hat{\mathbf{m}}_t$ using a feedforward neural network architecture as follows:

$$\hat{\mathbf{m}}_t = \sigma(\mathbf{W}^L \mathbf{h}_t^{L-1} + \mathbf{b}^L), \tag{4}$$

$$\mathbf{h}_t^\ell = ReLU(\mathbf{W}^\ell \mathbf{h}_t^{\ell-1} + \mathbf{b}^\ell),\ \ell = 2, ..., L-1, \tag{5}$$

$$\mathbf{h}_t^1 = ReLU(\mathbf{W}^1 \mathbf{y}_t^c + \mathbf{b}^1), \tag{6}$$

where L represents the number of layers, $\sigma(\cdot)$ the sigmoid nonlinearity, and $ReLU(\cdot)$ the rectified linear unit activation function. The weight and bias parameters of layer ℓ, whose values are learned during training, are denoted by $\mathbf{W}^\ell$ and $\mathbf{b}^\ell$. The input to the feedforward architecture is $\mathbf{y}_t^c = [\mathbf{y}_{t-c}, ..., \mathbf{y}_t, ..., \mathbf{y}_{t+c}]^T \in \mathbb{R}^{d(2c+1)}$, which incorporates temporal context by stacking a small number of frames to use as the network input.

We can alternatively incorporate temporal context using a recurrent network architecture for estimating $\hat{\mathbf{m}}_t$ as

$$\hat{\mathbf{m}}_t = \sigma(\mathbf{W}^L \mathbf{h}_t^{L-1} + \mathbf{b}^L), \tag{7}$$

$$\mathbf{h}_t^\ell = f(\mathbf{h}_t^{\ell-1},\ \mathbf{h}_{t-1}^\ell),\ \ell = 2, ..., L-1, \tag{8}$$

$$\mathbf{h}_t^1 = f(\mathbf{y}_t,\ \mathbf{h}_{t-1}^1), \tag{9}$$

where $f(\cdot)$ represents the nonlinear mapping function of a recurrent layer, and the state of each recurrent hidden layer, i.e., $\mathbf{h}_t^\ell$ for layer ℓ is stored and used as an additional input at the next time step. We use LSTM-style [8] recurrent layers for $f(\cdot)$, which were successfully used for speech denoising in [9, 11]. The input to the network, $\mathbf{y}_t$ in (9), is only a single spectrogram frame.

We can further incorporate temporal context into a source separation architecture by using bidirectional LSTM (BLSTM) archictures [10]. BLSTM networks require offline operation, and we can define a BLSTM layer as

$$\tilde{\mathbf{h}}_t^\ell = f(\mathbf{h}_t^{\ell-1}, \vec{\mathbf{h}}_{t-1}^\ell) + f(\mathbf{h}_t^{\ell-1}, \overleftarrow{\mathbf{h}}_{t+1}^\ell), \tag{10}$$

where $\vec{\mathbf{h}}_{t-1}^\ell$ and $\overleftarrow{\mathbf{h}}_{t-1}^\ell$ are outputs of the forward and backward recurrent layers, respectively. The backward layer $\overleftarrow{\mathbf{h}}_{t-1}^\ell$ consumes the input spectrogram in time-reversed order. Figure 1 illustrates how temporal context is incorporated in feedforward, recurrent (LSTM), and bidirectional recurrent (BLSTM) layers.

2.2. Training objective

Given a training set of isolated speech and isolated rustle noise spectrograms, we can create mixtures with known ground truth for learning the nonlinear mapping between noisy speech spectra $\mathbf{y}_t$ and estimated ratio mask $\hat{\mathbf{m}}_t$. Several studies on neural network based speech separation [2, 3, 11] have shown the utility of using the error in the estimated spectrum $\hat{\mathbf{s}}_t$ (as opposed to the error in the estimated mask $\hat{\mathbf{m}}_t$) as the network training objective. This leads to the so-called signal approximation mean squared error objective function

$$J_{MSE} = \frac{1}{T} \sum_{t=1}^{T} ||\hat{\mathbf{s}}_t - \mathbf{s}_t||_2^2. \tag{11}$$

But using this objective can sometimes cause the network to be too conservative in situations where noise is quieter than speech, yet still perceptible. An alternative objective proposed in [2] is

$$J_{DIS} = \frac{1}{T} \sum_{t=1}^{T} \Big(||\hat{\mathbf{s}}_t - \mathbf{s}_t||_2^2 + ||\hat{\mathbf{n}}_t - \mathbf{n}_t||_2^2 - \gamma||\mathbf{s}_t - \hat{\mathbf{n}}_t||_2^2 - \gamma||\mathbf{n}_t - \hat{\mathbf{s}}_t||_2^2 \Big), \tag{12}$$

where the parameter γ provides a trade-off between interference (i.e., rustle remaining in the separated speech) and artifacts caused by the source separation process. We found setting $\gamma = 10^{-3}$ to work well for removing rustle from speech, and we use that value and the objective function from (12) in all experiments described in Section 3. The objective function in (12) is minimized using backpropagation and stochastic gradient descent.

3. EXPERIMENTS

3.1. Dataset description

To create a training set for rustle noise removal, we needed to collect a large amount of clean speech and isolated rustle data. While several publicly available datasets for speech research offer only low sampling rate data (less than 40 kHz), we have used the pitch tracking corpus from [12], the reverberant speech from the Chime challenge [13], the processed speech from the DAPS experiment [14], and the TSP speech dataset [15]. All of these datasets provide audio at sampling rates of 44.1 or 48 kHz. We have also supplemented our clean speech training data with several hours of audio recorded for iZotope tutorial videos. While no publicly available datasets of isolated rustle exist, we were able to use sound effects from *www.prosoundeffects.com* that shared sonic qualities with lav mic rustle. However, these sound effects alone were insufficient to cover the wide range of lav mic rustle disturbances we wanted our algorithm to remove. We thus collected approximately one hour of isolated lav mic rustle noise, varying microphone type, clothing, movement, and recording environment. All of the audio processed in these experiments had a sampling rate of 48 kHz.

While most rustle disturbances are rather quiet relative to speech (i.e., $\mathrm{SNR} \gg 0$), we also wanted our algorithm to be robust to low-SNR situations, such as lav mics mounted on athletes during competition or while outdoors in extreme weather events. Thus, using these isolated speech and rustle noise datasets, we created mixtures with SNR ranging from -6 to $+9$ dB (SNR has been measured over periods with active rustle). To limit the computational resources necessary for training, all mixtures were limited to 10 seconds in length, and these mixtures sometimes consisted of multiple speech utterances and/or rustle noise concatenated together prior to forming the mixture.

While we can qualitatively test the performance of our algorithm using actual rustle-corrupted speech, to quantitatively evaluate performance using the metrics described in Section 3.2 requires mixtures with ground truth (i.e., the isolated speech and rustle used to create the mixture) available. This testing dataset is composed of speech from speakers not used to train the algorithm, as well as held out rustle noises that were distinct from those used during training. All testing set mixtures were 12 seconds in length and the SNR varied over the same -6 to $+9$ dB range.

3.2. Performance metrics

We quantitatively evaluate the performance of our algorithm in terms of separation performance and intelligibility. For separation performance, we use the SNR of the separated speech, typically referred to as source to distortion ratio (SDR) in the source separation literature [16]. For speech quality and intelligibility, we use the short-term objective intelligibility (STOI) metric proposed in [17]. The STOI algorithm returns a value in $[0, 1]$ range, with 1 representing the highest quality. However, we evaluate our rustle removal in terms of $\Delta\mathrm{STOI}$, which we define as the difference between the STOI score of the separated speech and that of the original noisy mixture, converted to a percentage.

3.3. Analysis of results

In this section we compare performance of our rustle removal algorithm for different network structures and their associated temporal context, as shown in Figure 1. Additionally, we investigate the impact of spectral resolution (i.e., the FFT size) used to create the spectrograms input and output by the network. All experiments use the Adadelta [18] optimizer and are trained for 20,000 mini-batches of 16 sequences each. Each sequence consists of 10 seconds of clean speech randomly mixed with segments of mic rustle.

Besides comparing objective measures, we have also performed informal listening tests with real-world speech signals having diverse SNR. They have shown a significant reduction in audibility of rustle. Some audio examples are available for download at *http://www.izotope.com/tech/aes_rustle*

3.3.1. FFT size

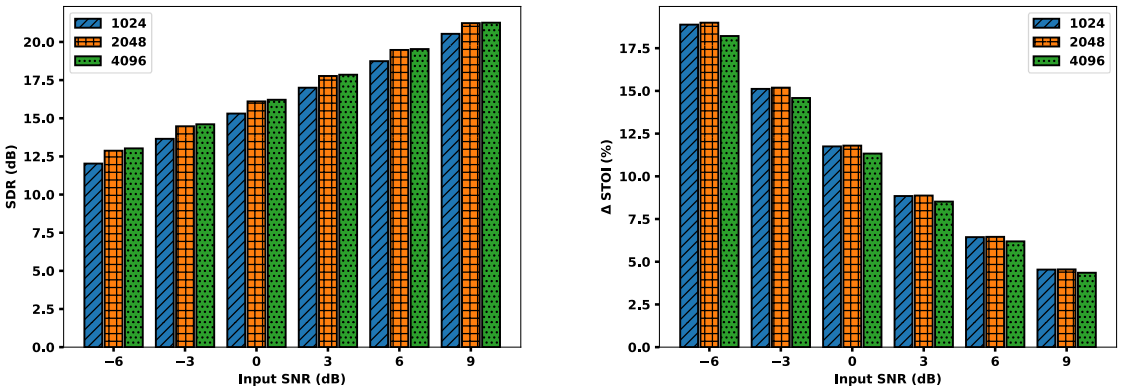
To determine an upper bound on source separation performance, we can use the so-called "oracle mask" which is the magnitude ratio mask computed using the ground truth isolated speech and rustle noise spectrograms. Figures 2(a) and (b) display the SDR and Δ STOI for FFT sizes of 1024, 2048, and 4096 (at 48 kHz sampling rate) as a function of input SNR. For all FFT sizes we used $4\times$ overlap and Hann windows. From Figure 2 we see that the FFT size of 4096 performs best in terms of SDR, but worst in terms of STOI.

Figure 3 repeats the same FFT size comparison, but this time evaluates testing set performance of a trained two-layer BLSTM network with 256 hidden units per layer. The SDR from Figure 3(a) exhibits the opposite trend with respect to increasing FFT size when compared to the oracle results from Figure 2(a), with 4096-point FFT leading to the lowest level performance. This discrepancy might be caused by the curse of dimensionality, as larger FFT sizes require more network parameters in the input and output layers. In terms of STOI performance for the trained BLSTM network shown in Figure 3(b), an FFT size of 2048 exhibits the best performance, while FFT sizes of 1024 and 4096 perform similarly, although the larger FTT size (4096) does show improvements at SNR of -6 dB. In terms of both the SDR and STOI results from Figures 2 and 3, the FFT size of 2048 appears to consistently demonstrate strong performance for both the oracle and trained BLSTM network.

3.3.2. Network structure

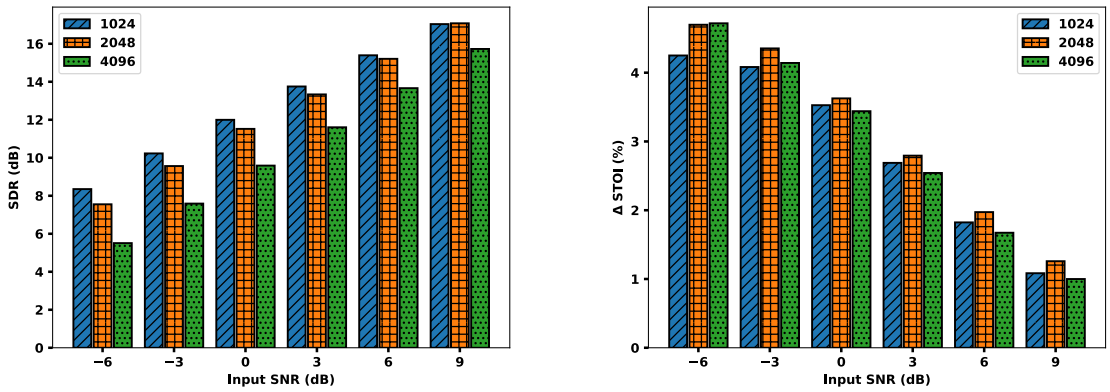
In this section we evaluate the feedforward, recurrent (LSTM), and bidirectional (BLSTM) architectures shown in Figure 1. All three architectures were designed to have a nearly equivalent number of parameters as shown in Table 1. For the feedforward architecture we used a context size of $c = 2$ frames, meaning that our network input is the concatenation of five frames. Because a single BLSTM layer has independent forward and backward layers, its complexity is comparable to a forward-only LSTM with four hidden layers.

Figures 4(a) and (b) display the SDR and STOI for the three different network architectures. The LSTM and BLSTM perform similarly in terms of SDR and much better than the feedforward



(a) SDR (b) ΔSTOI

Figure 2: Metrics of separated speech using the oracle (ground truth) ratio mask for different FFT sizes at different input SNR levels.

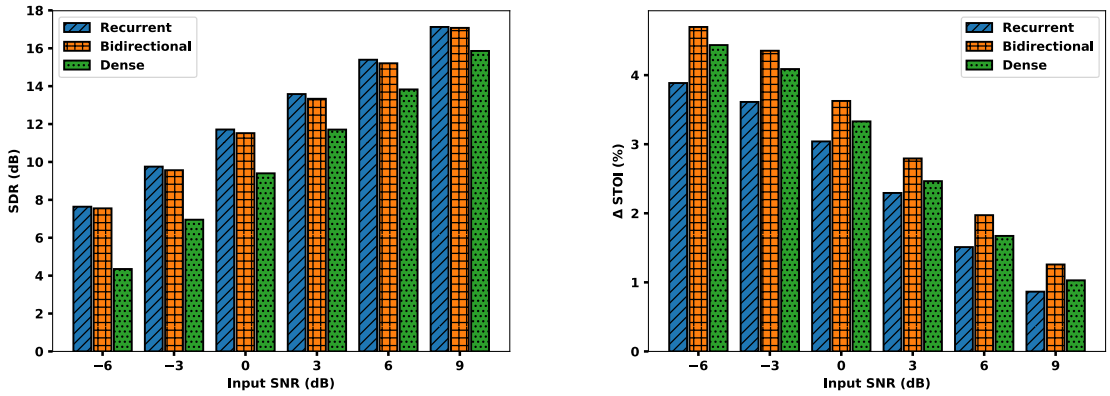


(a) SDR (b) ΔSTOI

Figure 3: Metrics of speech separated using BLSTM network for different FFT sizes at different input SNR levels.

	Input	Hid. 1	Hid. 2	Hid. 3	Hid. 4	Output
Feedforward	5125	512	512	512	512	1025
LSTM	1025	256	256	256	256	1025
BLSTM	1025	256	256	N/A	N/A	1025

Table 1: Layer sizes for different network configurations using 2048-point FFT. Sizes were chosen such that all architectures had approximately the same number of parameters.



(a) SDR (b) ΔSTOI

Figure 4: Metrics of separated speech comparing different network structures with FFT size of 2048 at different input SNR levels.

("dense") architecture, as shown in Figure 4(a). We can also interpret this result in terms of the amount of temporal context the network has available. Since the LSTM and BLSTM perform similarly, this could mean that future context is less important in terms of SDR. The recurrent architectures, however, exploit significantly more context than the feedforward architecture has available. In terms of the STOI shown in Figure 4(b), the BLSTM architecture performs best and the forward-only LSTM performs worst, demonstrating the importance of future context for intelligibility. Although the BLSTM architecture exhibits strong performance, it requires offline access for the backward pass over the data. A low-latency implementation becomes possible if BLSTM works on blocks of the audio signal or if a lookahead layer [19] is added to the forward-only LSTM architecture.

4. AMBIENCE PRESERVATION

The speech separation network trained on clean speech mixed with mic rustle seeks to optimally recover clean speech. In many real-life scenarios, input speech is corrupted with both mic rustle and some stationary (or quasi-stationary) noise (Figure 5(a)). In such cases our net trained for speech isolation produces excessive gating, i.e., attenuates stationary noise between sentences (Figure 5(b)). This can cause the separated speech to sound unnatural or overly processed, which was confirmed by our informal listening tests. The algorithm proposed in this section mitigates the problem by estimating the stationary noise floor and limiting the amount of spectral attenuation $\hat{\mathbf{m}}_t$ to ensure that the resulting signal $\hat{\mathbf{s}}_t$ does not have excessive gating (Figure 5(c)).

Because the algorithm adapts to the noise floor, it can be used for signals with low or high SNR. Its application is optional and often makes sense in the context of post-production, where preservation of the stationary noise floor ("room tone") is desirable.

4.1. Noise estimation

A simple adaptation algorithm is used to detect the quasi-stationary noise floor in speech. It operates on a magnitude spectrogram $\mathbf{y}_t$ of the input signal and computes magnitude estimates of the noise floor $\hat{\mathbf{n}}_t$ by applying a series of three filters: a Hann filter H, a sliding minimum filter M, and an asymmetric 1st order attack/decay filter E [20].

$$\hat{\mathbf{n}}_t = E(M(H(\mathbf{y}_t))) \tag{13}$$

The filters are independently applied to each frequency bin of the spectrogram along the time axis. The purpose of filter E is to quickly react to decays in the signal energy and slowly react to onsets of the signal energy. Its upward integration time (attack time) is set to 10000 ms, while its downward integration time (decay time) is set to 100 ms. The purpose of filter M is to keep noise floor estimates steady during speech utterances. Its window size is set to 2000 ms. The purpose of filter H is to prevent filter M from becoming trapped in spectrogram zeros. Its radius is set to 10 ms.

4.2. Limiting of attenuation

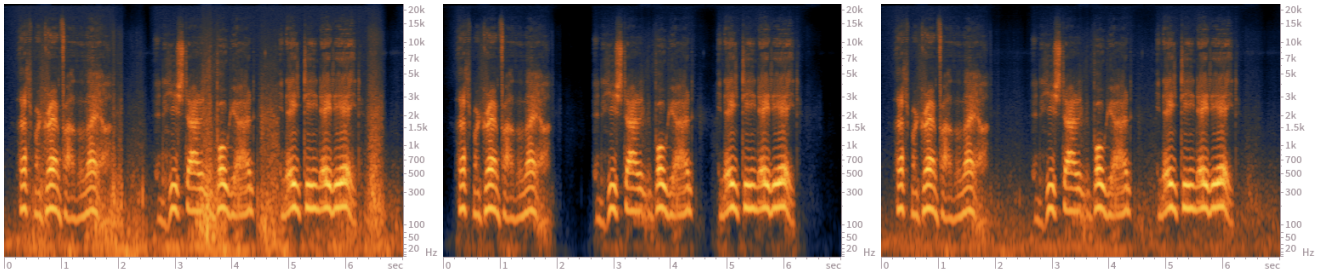
Gating is created when the resulting signal energy $\hat{\mathbf{s}}_t = \hat{\mathbf{m}}_t \odot \mathbf{y}_t$ falls below the noise floor $\hat{\mathbf{n}}_t$. To prevent this, we are limiting the spectral mask $\hat{\mathbf{m}}_t$ as follows:

$$\hat{\mathbf{m}}_t^{+} = \min\left\{1,\ \max\left\{\hat{\mathbf{m}}_t,\ \frac{\hat{\mathbf{n}}_t}{\mathbf{y}_t}\right\}\right\}. \tag{14}$$

Our noise floor estimate $\hat{\mathbf{n}}_t$ is quasi-stationary (slowly changing in time), so its distribution does not match the distribution of a typical noise power spectrum, which is random. When a quasi-stationary constraint (14) is applied to the mask and then to the signal (2), parts of the output signal obtain this unnatural distribution too. To improve naturalness of the distribution, we are applying a time-frequency smoothing to the mask $\hat{\mathbf{m}}_t^{+}$ using a "DFT thresholding" algorithm from [21]. This edge-adaptive smoothing also reduces "musical noise" artifacts resulting from processing the STFT spectrum. The updated processing formulas with smoothing of the mask are as follows:

$$\hat{\mathbf{m}}_t^{++} = \mathrm{Smooth}\left(\hat{\mathbf{m}}_t^{+}\right), \tag{15}$$

$$\hat{\mathbf{s}}_t = \hat{\mathbf{m}}_t^{++} \odot \mathbf{y}_t. \tag{16}$$



(a) Speech with rustle (b) De-rustle, formula (2) (c) De-rustle, formula (16)

Figure 5: Comparison of rustle attenuation without (b) and with (c) ambience preservation. Additional audio examples are available at *http://www.izotope.com/tech/aes_rustle*

5. CONCLUSIONS AND FUTURE WORK

In this paper we have described an approach for lavalier microphone rustle removal using deep neural networks, while maintaining natural sounding audio quality by supplementing the network output with spectral smoothing and stationary noise floor estimation. We also found a spectral resolution of around 20 Hz (FFT size of 2048 at 48 kHz) and bidirectional recurrent network architectures to provide the best performance for this specific speech separation application.

Bidirectional recurrent architectures (e.g., BLSTM) exhibited the overall best performance, but investigating low-latency bidirectional approximations for rustle removal is an important area for additional study. Exploring complex ratio masks [6] or time-domain Wavenet architectures [22] are other potentially interesting areas of future work.

6. REFERENCES

[1] P.C. Loizou, *Speech Enhancement: Theory and Practice*, CRC Press, Boca Raton, FL, 2013.

[2] P.S. Huang, M. Kim, M. Hasegawa-Johnson, and P. Smaragdis, "Joint optimization of masks and deep recurrent neural networks for monaural source separation," *IEEE/ACM Transactions on Speech and Language Processing*, vol. 23, pp. 2136–2147, Dec. 2015.

[3] Y. Wang, A. Narayanan, and D.L. Wang, "On training targets for supervised speech separation," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 22, pp. 1849–1858, 2014.

[4] Y. Xu, J. Du, L.-R. Dai, and C.-H. Lee, "A regression approach to speech enhancement based on deep neural networks," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 23, pp. 7–19, 2015.

[5] J. Chen, Y. Wang, S.E. Yoho, D.L. Wang, and E.W. Healy, "Large-scale training to increase speech intelligibility for hearing-impaired listeners in novel noises," *Journal of the Acoustical Society of America*, vol. 139, pp. 2604–2612, 2016.

[6] D.S. Williamson, Y.. Wang, and D.L. Wang, "Complex ratio masking for monaural speech separation," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 24, pp. 483–492, 2016.

[7] K. Cho, A. Courville, and Y. Bengio, "Describing multimedia content using attention-based encoder-decoder networks," *IEEE Transactions on Multimedia*, vol. 17, pp. 1875–1886, 2015.

[8] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 23, no. 8, pp. 1735–1780, 1997.

[9] H. Erdogan, J.R Hershey, S. Watanabe, and J. Le Roux, "Phase-sensitive and recognition-boosted speech separation using deep recurrent neural networks," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2015, pp. 708–712.

[10] M. Schuster and K.K. Paliwal, "Bidirectional recurrent neural networks," *IEEE Transactions on Signal Processing*, vol. 45, pp. 2673–2681, 1997.

[11] F. Weninger, J. Le Roux, J.R. Hershey, and B. Schuller, "Discriminatively trained recurrent neural networks for single-channel speech separation," in *IEEE GlobalSIP 2014 Symposium on Machine Learning Applications in Speech Processing*, 2014, pp. 577–581.

[12] G. Pirker, M. Wohlmayr, S. Petrik, and F. Pernkopf, "A pitch tracking corpus with evaluation on multipitch tracking scenario," in *Interspeech*, 2011, pp. 1509–1512.

[13] J. Barker, E. Vincent, N. Ma, H. Christensen, and P. Green, "The PASCAL CHiME speech separation and recognition challenge," *Computer Speech and Language*, vol. 27, pp. 621–633, 2013.

[14] G.J. Mysore, "Can we automatically transform speech recorded on common consumer devices in real-world environments into professional production quality speech? A dataset, insights, and challenges," *IEEE Signal Processing Letters*, vol. 22, pp. 1006–1010, 2015.

[15] P. Kabal, "TSP speech database," Tech. Rep., Department of Electrical & Computer Engineering, McGill University, Montreal, Quebec, Canada, 2002.

[16] E. Vincent, R. Gribonval, and C. Fevotte, "Performance measurement in blind audio source separation," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 14, pp. 1462–1469, 2006.

[17] C. Taal, R. Hendriks, R. Heusdens, and J. Jensen, "An algorithm for intelligibility prediction of time-frequency weighted noisy speech," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 19, pp. 2125–2136, 2011.

[18] M. D. Zeiler, "ADADELTA: an adaptive learning rate method," *Computing Research Repository*, 2012.

[19] C. Wang, D. Yogatama, A. Coates, T. Han, A. Hannun, and B. Xiao, "Lookahead convolution layer for unidirectional recurrent neural networks," in *Workshop Extended Abstracts of the 4th International Conference on Learning Representations*, 2016.

[20] A. Lukin, "Tips & tricks: fast image filtering algorithms," in *Proceedings of Graphicon'2007, Moscow, Russia*, 2007, pp. 186–189.

[21] A. Lukin and J. Todd, "Suppression of musical noise artifacts in audio noise reduction by adaptive 2D filtering," in *Audio Engineering Society Convention 123*, Oct 2007.

[22] A. van den Oord, S. Dieleman, H. Zen, K. Simonyan, O. Vinyals, A. Graves, N. Kalchbrenner, A. Senior, and Kavukcuoglu, "Wavenet: A generative model for raw audio," *CoRR abs/1609.03499*, 2016.

[23] D. Amodei, R. Anubhai, E. Battenberg, C. Case, J. Casper, B. Catanzaro, J. Chen, M. Chrzanowski, A. Coates, G. Diamos, and E. Elsen, "Deep speech 2: End-to-end speech recognition in English and Mandarin," in *Proceedings of The 33rd International Conference on Machine Learning*, 2016, pp. 173–182.

[24] K. Cho, B. van Merriënboer, C. Gülçehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, "Learning phrase representations using RNN encoder-decoder for statistical machine translation," in *Conference on Empirical Methods in Natural Language Processing*, 2014.

[25] B. Raj, T. Virtanen, S. Chaudhuri, and R. Singh, "Non-negative matrix factorization based compensation of music for automatic speech recognition," in *Interspeech*, 2010, pp. 717–720.

[26] D. Wang, "Time-frequency masking for speech separation and its potential for hearing aid design," *Trends in Amplification*, vol. 12, pp. 332–353, 2008.

[27] C. Raffel, B. McFee, E.J. Humphrey, J. Salamon, O. Nieto, D. Liang, and D.P.W. Ellis, "mir_eval: A transparent implementation of common MIR metrics," in *Proceedings of the 15th International Conference on Music Information Retrieval*, 2014.

[28] D. Stowell, D. Giannoulis, E. Benetos, M. Lagrange, and M. D. Plumbley, "Detection and classification of audio scenes and events," *IEEE Transactions on Multimedia*, vol. 17, pp. 1733–1746, 2015.

[29] A. Mesaros, T. Heittola, and T. Virtanen, "TUT database for acoustic scene classification and sound event detection," in *24th European Signal Processing Conference (EUSIPCO 2016)*, 2016.

A MICRO-CONTROLLED DIGITAL EFFECT UNIT FOR GUITARS

Geovani Cardozo Alves *

Departamento Acadêmico de Eletrotécnica,
Universidade Tecnologica Federal do Paraná
Curitiba, Brazil
geovani_ca@hotmail.com

Marcelo de Oliveira Rosa

Departamento Acadêmico de Eletrotécnica,
Universidade Tecnologica Federal do Paraná
Curitiba, Brazil
mrosa@utfpr.edu.br

ABSTRACT

Here we present a micro-controlled digital effect unit for guitars. Different from other undergraduate projects, we used high-quality 16-bit Analog-to-Digital (A/D) and Digital-to-Analog (D/A) converters operating at 48kHz that respectively transfer data to and from a micro-controller through serial peripheral interfaces (SPIs). We discuss the design decisions for interconnecting all these components, the project of anti-aliasing (low-pass) filters, and additional features useful for players. Finally, we show some results obtained from this device, and discuss future improvements.

1. INTRODUCTION

Analog guitar effects became very popular from 70's to 90's as an artifact that musicians could imprint their own personality touch in their sounds [1]. Once that the micro-controllers become popular among engineers, new and combined effects could be added into the so called digital effect units, which allow a single device to have multiple effects. Nowadays, powerful micro-controllers (with embedded DSP units) can execute sophisticated signal processing algorithms, creating configurable digital audio effect units.

Most of available micro-controlled boards for education purpose come with A/D converters (ADCs) operating at reasonable sampling rates but coding amplitudes at 12 bits. Usually they do not have any D/A converter (DAC) on-board. Since their use is focused on control applications, they usually come with PWM circuits, which are not suitable for audio applications.

Reasonable audio devices require sampling rates of 48kHz and sample resolution of 16bits/sample if we consider studio-quality recordings. Therefore, academic audio projects require the development of custom boards using a micro-controller and an external D/A converter at least, assuming that the micro-controller has an audio-oriented A/D converter.

As an undergraduate project, we envision a programmable digital effect unit that can be useful for students interested on signal and systems and digital signal processing: they will be able to develop their algorithms on tools like Matlab or Octave and convert them in compiled codes to be uploaded into these devices, obtaining real-time processing (at maximum of 48kHz). To achieve this goal, we conceived a device using three evaluation boards from Texas Instruments (TI), respectively dealing with the algorithms (a micro-controller) and with the A/D and D/A conversions. Our primary application is applying distortion effects over electric guitar sounds in real-time, although it can promptly adapted to other instruments.

Similar work was made by Young and Chih [2] using 16-bit converters with 48kHz but with a different micro-controller. In addition, Hasnain and Saleem [3] used another approach to their work on using re-programmable Matlab Simulink blocks of codes in order to generate the audio effects.

Here we will present the design aspects of an academic-oriented device, including the project of some analog filters and amplifier for signal conditioning (particularly avoid signal aliasing). Thus this paper is organized as follow: first we will describe the electronic boards and other components that we used to mount the device, particularly focusing on their connectivity. Next, we will present the design of low-pass filters for anti-aliasing purpose and D/A conversion, and linear amplifiers for signal condition, followed by details of how to implement signal processing algorithms (focusing on time CPU interruptions). Finally some results are presented from real use of the device (capture from digital oscilloscopes) along with conclusions and suggestions of improvements.

2. MATERIALS AND METHODS

Figure 1 presents a block diagram of the proposed device: The electric guitar signal is linearly amplified and filtered by a low-pass filter (LPF) to eliminate aliasing artifacts of the signal before it is sampled. After this signal conditioning, it is sampled by the external ADC in order to be properly read by the micro-controller.

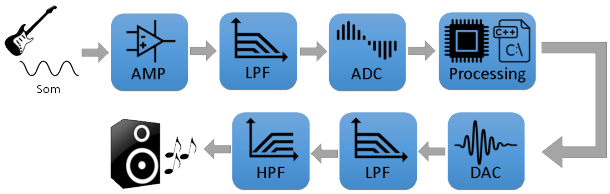


Figure 1: *Project's Block Diagram*

After digitally processed by the micro-controller, the signal passes through the external DAC and filtered by a low-pass filter (LPF) in order to be played. An additional HPF is used to remove the signal DC component since DAC generates analog signals with a fixed offset level equals to 2V.

Each component will be described in next sections.

2.1. Digital Processing Unit

Here we used the TI LaunchPad development kit (TIVA) that comes with the micro-controller TM4C1294NCPDT [4] shown in Figure 2. Such a kit has easy access to the micro-controller ports and comes with DIP switches and LEDs that can be programmed (we used them to allow its users to respectively choose a digital effect and to have a feedback of their choices). A motivation to use

* This work was part the author's undergraduate project.

this kit is that it has connectors to attach expansion boards named as BoosterPacks (both the A/D and D/A used here are expansion boards sold by TI).

It also has a Ethernet connector that can be used for data transfer that later will be used for direct-to-computer music recording and remote board configuration.

Figure 2: *TI's development kit for TM4C1294NCPDT (EK-TM4C1294XL)*

All codes were implemented, debugged, and uploaded to the kit in TI integrated development environment (Code Composer Studio - CCS).

2.2. External A/D and D/A Converters

Although the kit used here has a A/D converter, it only supports a bit depth of 12bits/sample, not meeting our requirements. Therefore we used an external A/D converter (ADC161S626) provided by TI [5] as an electronic board kit (a BoosterPack) that can easily be attached to the micro-controller board. Such an expansion board has an operational amplifier to offset the signal to a mid voltage reference. Its analog-to-digital conversion uses successive approximation register (SAR) architecture. The expansion board is shown in Figure 3.

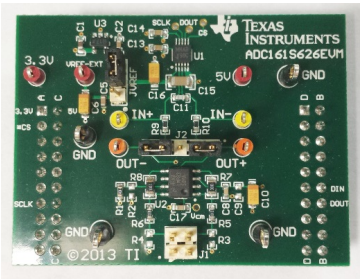


Figure 3: *ADC in TI's BoosterPack kit (ADC161S626EVM)*

After the signal is processed by the micro-controller, it is converted back into audio, from a digital form to a voltage signal. Another expansion board is used for this purpose once the micro-controller's kit does not have any digital-to-analog converter (DAC) in its circuitry. Since we required a DAC with 16-bit conversion resolution as a project specification, we chose DAC161S055, also provided by TI [6] in a BoosterPack (Figure 4). It has a resistor matrix topology and internal registers to setup its operational mode.

In this project, the DAC was configured to operate in write-through mode, which means it updates the voltage output as soon as data transfer is completed (other offered modes would require a delay between these two conversion steps). Therefore, we increased the available processing time between two successive sampling steps (time to execute the effects).

Figure 4: *DAC in TI's BoosterPack kit (DAC161S055EVM)*

While ADC samples are coded in two-complement binary representation (with the most significant bit for the math sign) in order to represent both positive and negative amplitudes, DAC samples represents only positive values since it produces only positive analog signals (a reference voltage is used to correctly understand the output signal). Therefore a simple integer math was required in order to obtain right conversions.

2.3. Analog Filters and Amplifier

Both filters and amplifier were based on operational amplifiers (op-amp) due to their simplicity. In both cases we powered the op-amps with 5V because this voltage was supplied by the USB port of the micro-controller board. In case of the amplifier, we added a biased voltage of 2.5V (a DC level) to the input signal since ADC and DAC require positive voltages. Consequently the outputs of all amplifiers bounce around $\pm 2.5\text{V}$ and the saturation is achieved for voltages exceeding 0 to 5V limits. To do that, the following circuit was implemented (Figure 5), where carefully chosen resistors and capacitors amplify only the AC part of the signal, ignoring its DC level.

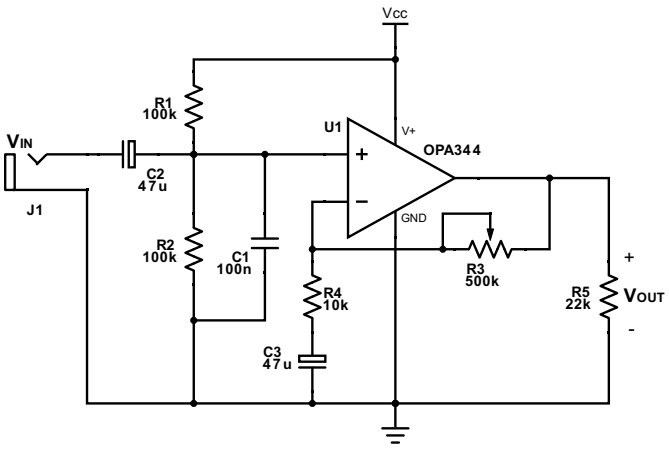


Figure 5: *Amplifier circuit*

The two LPFs in Figure 1 correspond to fourth order active low-pass filters - using two Sallen-Key topologies (quality factor $Q = 0.5$ and cut frequency $f_c = 24,405.14\text{Hz}$) in cascade - were built to work as anti-aliasing filters (the value of f_c is due to the use of electrical components that were commercially available).

Its resulting cutoff frequency was set to approximately 24kHz following Nyquist theorem [7].

TI OPA344 was the op-amp chosen for both amplifiers and LPFs, which is a low power single-supply rail-to-rail op-amp eligible for audio applications (also it comes in dual in-line package - DIP - which is suitable for breadboard testing and building academic circuits without any specific tool like the ones for SMD packages).

A passive HPF with a low cutoff frequency was designed to remove the DC signal component. A common RC passive topology was chosen with a resistor ($22\text{k}\Omega$) and a capacitor ($10\mu\text{F}$), resulting in a cutoff frequency of 0.72Hz.

Figures 6 and 7 shows, respectively, the frequency response of both LP and HP filters.

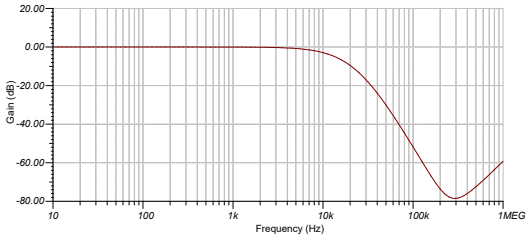


Figure 6: *Active Low-pass Filter Frequency Response*

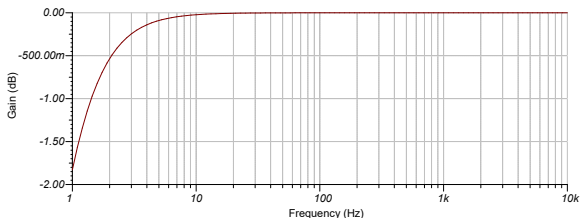


Figure 7: *Passive High-pass Filter Frequency Response*

2.4. Electrical Connections

Both converters use a synchronous serial interface (SSI) to communicate with the micro-controller. In such a data bus, a master device manages the communication while the slave ones answer back. Here we set the micro-controller as the master device and both converters as slave ones. To properly work four pins of master and slave devices should be used: the chip select pin (`CS`) to select the correct peripheral to send/receive data (one of the converters), the serial clock pin (`SCL`) to synchronize the data transfer, the synchronous serial transmitter (`SSTx`), and the synchronous serial receiver (`SSRx`) pins to send/receive bit streams between the devices. An interconnection diagram for SSI used here is illustrated in Figure 8. Programmatic, both ADC and DAC have internal FIFO queues to send and receive bit-oriented streams of data or commands.

The SSI clocks (or bit rate) for both ADC and DAC devices were set at their maximum values, 5MHz and 20MHz respectively. It was set higher than the required frequency for converting bits to voltage and vice-versa (24 bits/sample $\times 48.000$ samples/sec) in order to leave enough time for audio signal processing.

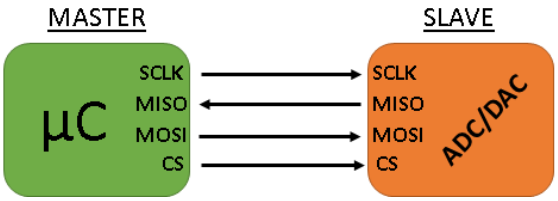


Figure 8: *Diagram of electrical interconnection between the micro-controller and both A/D and D/A converteres (Synchronous Serial Interface (SSI) Diagram)*

Electrical wires connect the analog filters (LPFs and HPFs) and amplifiers to the input of ADC board and output of the DAC board. We used a 6.3mm female J1 connector to plug an electric guitar in, and the same kind of connector is used to plug in an external speaker or a sound mixer.

2.5. Implementing Digital Distortions

Basically the distortion routines are implemented as a sequential procedure of acquiring digital samples from the ADC, processing them according to a predefined audio effect, and converting the result into an analog signal. Considering the sampling rate used here (48kHz), we coded these routines as part of a micro-controller timer interruption in order to minimize jitter effects on the output signal. Alternatively we considered to implement direct memory access (DMA) data transfer to speed up the process but we felt that our current implementation with timer interruptions was efficient enough for running some the digital distortions we describe here.

Therefore, each time this interruption is triggered, it executes the following steps:

1. Read a sample from ADC (waiting until the ADC release a 16-bit sample);
2. Process the sample;
3. Send the processed sample to the DAC (waiting until it finishes the conversion);

Naturally, the period between the execution of two successive timer interruptions was $(1/48000)\text{sec}$. There are different but almost fixed Δt's for running different audio effects, however the time interval between executions is constant (it means there should be low variable delays between input and output signals in current implementation).

The audio distortions are selected by user when he/she presses an specific button. It generates an interruption that alter a global counter/variable, which is used by the timer interruption code to sequentially select an audio distortion.

Prior to this endless procedure (meaning that our ADC and DAC never stop acquiring and generating audio signals), both micro-controller and external devices are properly configured.

We implemented four different audio effects: distortion (or saturation), delay, loop, and tremolo. The first one (the simplest effect) is the distortion. It adds harmonics to the output sound that make the sound look like an electric guitar played in a rock concert with its amplifiers saturating the sound levels. This effect basically clips the sound wave (regardless positive or negative amplitudes) and the amount of clipping defines how much "fuzzy" will the output sound be. The Algorithm 1 shows the basic operation of this effect: an if-then-else statement compares the input

value from ADC to the distortion limit set previously, and if the absolute magnitude of the sample is greater than the distortion limit value, then the output value will be the distortion limit, otherwise will be the value received at first.

Input: i^{th} sample from ADC: sample_to_adc
Output: i^{th} sample to DAC: sample_to_dac
if sample_to_adc >= distortion_limit **then**
  sample_to_dac = distortion_limit
else if sample_to_adc <= -distortion_limit **then**
  sample_to_dac = -distortion_limit
else
  sample_to_dac = sample_to_adc
end

Algorithm 1: Distortion implementation

The delay effect simply delays the sound signal by a fixed amount of time. It requires storing a quantity of samples in a vector/array. Here we used 1sec delay, which requires storing $48,000$ samples. To avoid moving data in order to store new samples, we logically implemented a circular buffer with a single variable to control the access to it. Algorithm 2 shows this implementation.

Input: i^{th} sample from ADC: sample_from_adc
Output: i^{th} sample to DAC: sample_to_dac
sample_to_dac = sound_array [i]
sound_array [i] = sample_from_adc
i ++
i = i % 48000

Algorithm 2: Delay implementation

The loop effect replays a pre-recorded signal in loop fashion, basically giving a base sound for musical arrangements. To do that, first an array of fixed size (here we used a size equivalent to 1sec) receives all samples from ADC (up to 48000 samples). Once the buffer is full, our routine starts to send all these stored samples to DAC indefinitely.

The user can set another pre-recorded signal by pressing a button which triggers an interruption where a variable called isRecording is set in order to enable the recording mode of Algorithm 2.

Input: i^{th} sample from ADC: sample_from_adc
Output: i^{th} sample to DAC: sample_to_dac
if isRecording **then**
  sound_array [i] = sample_from_adc
  i ++
  if i = $i_{\max}$ **then**
    i = 0
    isRecording = false
  end
else
  sample_to_dac = sound_array [i]
  i ++
  i = i % 48000
end

Algorithm 3: Loop implementation (Both i and isRecording are set by an interruption triggered by a button pressed)

The last effect is tremolo: it modulates the ADC signal according to a preset signal. Analog tremolos are implemented by a low-frequency oscillator (LFO) - whose frequency ranges from 0.5 to 10Hz - to vary the sound amplitude. Here we digitally implemented it using a 10000-samples array containing a 4.8Hz sinusoidal signal with amplitude equals to 0.25 and an offset of 0.75 (these values affect the way the input signal is altered). This array was generated in MATLAB and hardcoded in the tremolo routine.

The implementation consists of multiplying samples of this array by the samples from ADC. Therefore the amplitude of the input signals are attenuated to a maximum of 50% according to the preset sinusoidal signal used.

Input: i^{th} sample from ADC: sample_from_adc
Output: i^{th} sample to DAC: sample_to_dac
sample_to_dac = tremolo_array [i] $\times$
 sample_from_adc
i ++
i = i % 10000

Algorithm 4: Tremolo implementation

3. RESULTS

The resulting device is depicted in Figure 9. To demonstrate its usefulness, we first present the A/D and D/A conversions carried out by our prototype with no digital distortions and no analog filtering been applied to the signal except by the analog amplification. Two sinusoidal (narrow-band) signals were separately applied to the prototype input jack and the DAC output pin (therefore using no output LP and HP filters) was connected to an oscilloscope. Figures 10 and 11 shows this output signals for a 1kHz and 5kHz sinusoidal signals.

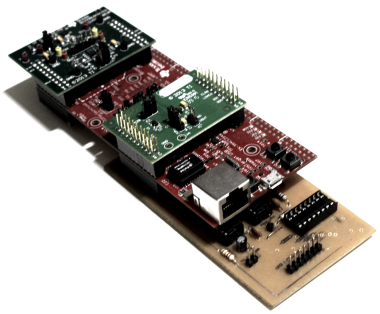


Figure 9: *Picture of the device*

The difference on voltage scale of each channel occurred because each converter had different numeric ranges: ADC works with 15-bit values plus one bit for signal (the most significant one) in two's complement notation, while DAC uses all 16-bits to represent positive output values. That led to an output value equals to the half of an input value in this no-distortion scenario.

The stair effect on channel 2 was given by the zero-order holder effect of DAC. Figure 11 shows that clearly. It also shows its influence on the signal frequency captured by the oscilloscope. Considering that we were focusing on building a guitar effect unit, such a problem may not be a big deal since an in-tune guitar has frequencies ranging from 80Hz up to 1200Hz and we were using high

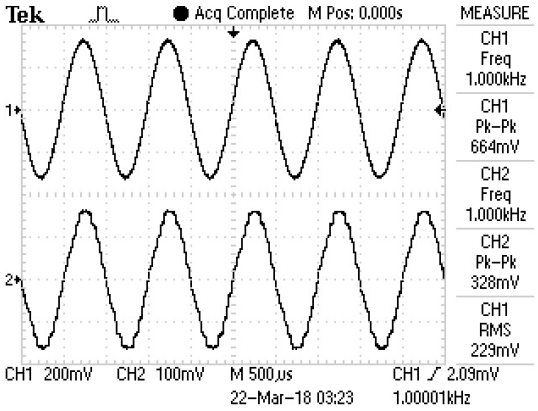


Figure 10: 1kHz *test sinusoid wave: channel 1 and 2 registers the input and output signals, respectively (this setting will be used in all following figures)*

clock rate to excite our DAC module. However, to confidently cope with any input signal, we used low-pass filters (Section 2.3 to finish the DA conversion, as shown in Figure 12. Note that these kind of filter impose a delay (in case of a 5kHz sinusoidal signal, it is about $52\mu s$).

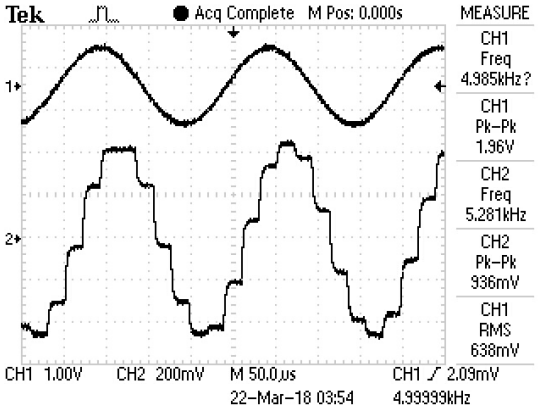


Figure 11: 5kHz *test sinusoid wave*

Figure 13 shows a guitar signal captured after directly connecting the guitar cable jack in an oscilloscope: its peak-to-peak amplitude does not exceed 200mV. Although it varies according to instrument technology, brand, and age, for example, the voltage amplitude never reaches 1V. Such voltage values demands a pre-amplifier for using sound systems as usually digital effect units requires. Our proposed amplifier at the input of the system allows some adjustments before applying analog filters and digital effects coded into the micro-controller.

To demonstrate the distortion effect, a sinusoid signal were applied: Figure 14 shows the resulting effect whose $distortion\_limit$ was set to 3000 which is equivalent to 0.92V after the analog-to-digital conversions.

For delay, loop, and tremolo effects, we played a few tones in an electric guitar connected to our device, which is altered by such effects. In all cases, the oscilloscope was set to capture 5sec of input (channel 1) and output (channel 2) signals after triggered.

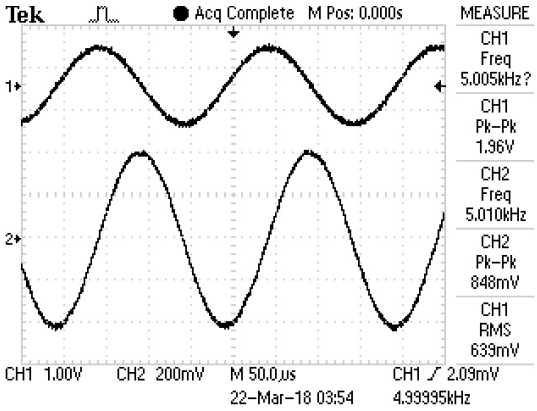


Figure 12: 5kHz *test sinusoid wave after the output LP filter*

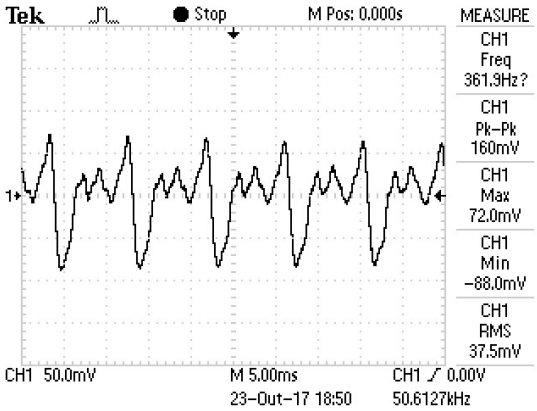


Figure 13: *Example of guitar signal*

Figure 15 shows the delay effect: channel 2 shows the delayed version of the input signal (delay of 1sec). In case of loop effect, one second of an input signal was previously recorded by the device (not shown here). After that, Figure 16 shows that an input signal captured by the device was ignored, and the recorded signal was repetitively reproduced by the device as its output signal.

Finally, the tremolo effect altered the input signal (channel 1) by modulating it with a sinusoidal signal. Channel 2 of Figure 17 shows the expected result. Due to the nature of the input signal, this modulation is more evident at the middle of the oscilloscope screen in this example.

4. CONCLUSIONS AND FUTURE IMPROVEMENTS

Here we present a digital audio effect for guitars that was implemented with a micro-controller and external digital converters - ADC and DAC - to operate at 48kHz. Additional circuitry for amplifiers and low pass filters were designed to cope with Nyquist limits, and a few digital effects were implemented to demonstrate the use of the unit. The use of timer interruptions to filter the input signal, sample-by-sample, before sending it to DAC minimized the jitter level. All the system requires a voltage source of $+5$V (0.47W), which can be powered through the USB connector of the micro-controller kit.

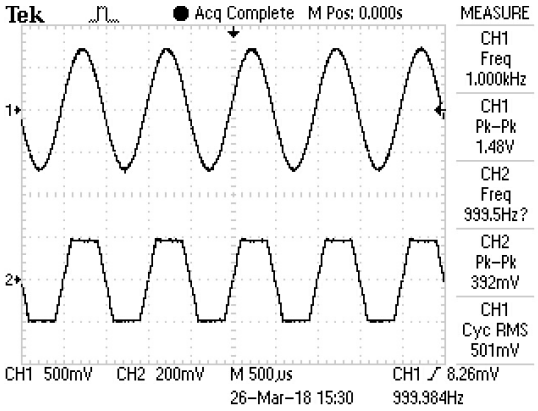


Figure 14: *Distortion result from a* 1kHz *sinusoid wave*

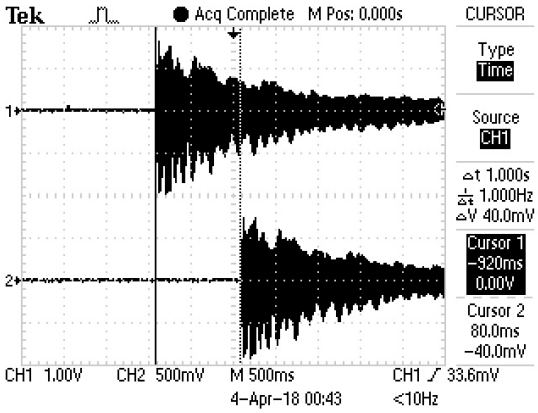


Figure 15: *Result (Channel 2) of applying the delay effect (*1sec*) over a signal produced by a electric guitar (Channel 1)*

Although the delays imposed by the analog LP filters, this academic prototype of a digital effect unit worked fine. Other effects can be readily implemented in order to have near real-time digital effects. Our next steps are:

- Add digital dithering to improve sound quality, specially for live guitar sounds;
- Reduce the jitter caused by lengthy complex digital filters by using two different timer interrupts (respectively for ADC and DAC procedures) and respective data buffers that operates in parallel;
- Create an internet (tcp/ip) server inside the micro-controller that transfers all sampled data to a remote client applications, allowing to create a virtual mixer with multiple channels (limited by the computer client capacity);
- Create an internet (tcp/ip) server inside the micro-controller that receives additional digital effects algorithms (and its configurations) from remote client applications, which allows the digital effect unit be remotely programmed.

5. ACKNOWLEDGMENTS

This work was funded by Fundação Araucária, Paraná, Brazil.

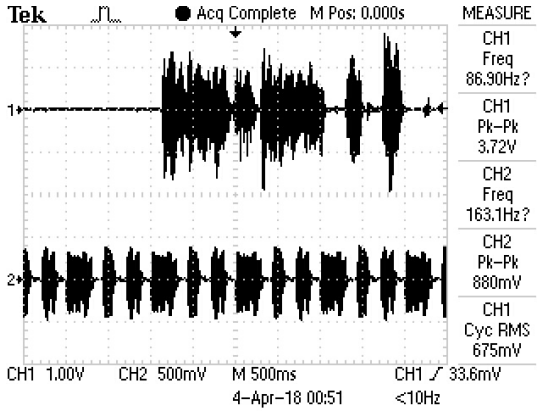


Figure 16: *Result (Channel 2) produced by a previous recorded sound from an electric guitar. The guitar keeps playing (Channel 1) but the loop, once recorded, wont change its output values*

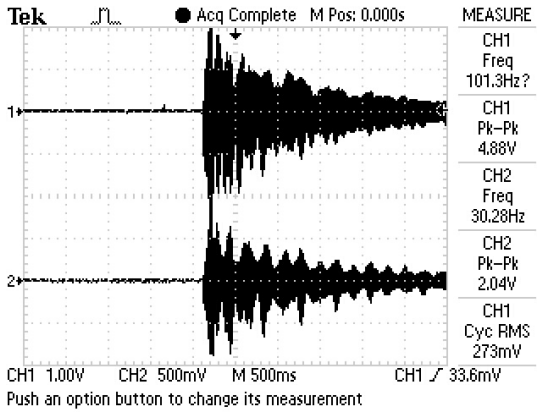


Figure 17: *Result (Channel 2) of applying the tremolo effect over a signal produced by a electric guitar (Channel 1)*

6. REFERENCES

[1] B. Tarquin, *Stomp on this! The Guitar Pedal Effects Guidebook*, Cengage Learning, 2015.

[2] C.-F. Yang and H.-Y. Chih, "An open source audio effect unit," in *IEEE Int. Conf. on Systems, Man, and Cybernetics (SMC)*. IEEE, 2016, pp. 638–643.

[3] S. Hasnain, A. Daruwalla, and A. Saleem, "A unified approach in audio signal processing using the tms320c6713 and simulink blocksets," in *2nd Int. Conf. on Computer, Control and Comm. (IC4)*. IEEE, 2009, pp. 1–5.

[4] Texas Instruments, *TIVA™ TM4C1294NCPDT Microcontroller*, 2014.

[5] Texas Instruments, *ADC161S626 16-Bit, 50 to 250 kSPS, Differential Input, MicroPower ADC*, 2008.

[6] Texas Instruments, *DAC161S055 Precision 16-Bit, Buffered Voltage-Output DAC*, 2010.

[7] A. Oppenheim, *Discrete-time signal processing*, Pearson Education India, 1999.

CREATING ENDLESS SOUNDS

Vesa Välimäki, Jussi Rämö, and Fabián Esqueda *

Acoustics Lab, Department of Signal Processing and Acoustics
Aalto University
Espoo, Finland
vesa.valimaki@aalto.fi

ABSTRACT

This paper proposes signal processing methods to extend a stationary part of an audio signal endlessly. A frequent occasion is that there is not enough audio material to build a synthesizer, but an example sound must be extended or modified for more variability. Filtering of a white noise signal with a filter designed based on high-order linear prediction or concatenation of the example signal can produce convincing arbitrarily long sounds, such as ambient noise or musical tones, and can be interpreted as a spectral freeze technique without looping. It is shown that the random input signal will pump energy to the narrow resonances of the filter so that lively and realistic variations in the sound are generated. For real-time implementation, this paper proposes to replace white noise with velvet noise, as this reduces the number of operations by 90% or more, with respect to standard convolution, without affecting the sound quality, or by FFT convolution, which can be simplified to the randomization of spectral phase and only taking the inverse FFT. Examples of producing endless airplane cabin noise and piano tones based on a short example recording are studied. The proposed methods lead to a new way to generate audio material for music, films, and gaming.

1. INTRODUCTION

Example-based synthesis refers to the generation of sounds similar to a certain sound but not identical. In audio, example-based synthesis solves a common problem, which we refer to as the small data problem. It is the opposite of the big data problem in which the amount of data is overwhelming and the challenge is how to find some sense of it. In the small data problem in audio processing, there may be only a few or even a single clean audio recording representing desirable sounds. It is usually unacceptable to only use that single sample in an application. For example, in various simulators, such as flight simulators [1] and working machine simulators [2], there is a need to produce a variety of sounds based on example recordings.

Previous related works have studied the synthesis of sound textures to expand the duration of example sounds. For some classes of sound, the concatenation and crossfading of samples can be quite successful. Fröjd and Horner have investigated such methods, which are related to granular synthesis [3]. They show that the method is particularly successful for the synthesis of seashore, car racing, and traffic sounds. Schwarz *et al.* compared several related approaches and showed that they perform slightly better than randomly chopping the input audio file into short segments [4]. Siddiq used a combination of granular synthesis and colored noise synthesis to produce for example the sound of running water based on modeling [5]. Both the grains and the spectrum of the background noise were extracted from a recording. Charles has also proposed a spectral freeze method, which uses a combination of spectral bins from neighboring frames to reduce the repetitive "frame effect" in the phase vocoder [6].

In this work, we use very high-order linear prediction (LP) to extract spectral information from single audio samples. The use of linear prediction has been common in audio processing for many years [7,8], but usually low or moderate prediction orders are used, such as about 10 for voice and between 10–100 for musical sounds. The use of a very high filter order is often considered overmodeling, which means that the predictive filter no longer approximates the spectral envelope, but it also models spectral details, such as single harmonics.

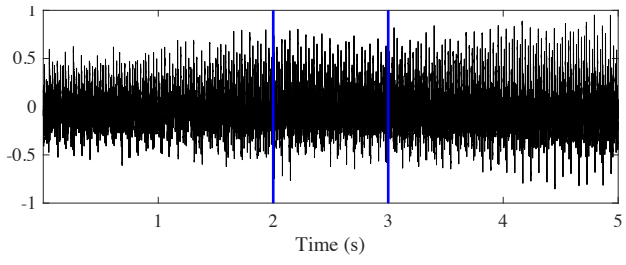
The idea and theory of utilizing higher-order LP is presented in Jackson *et al.* [9] and in Kay [10], where they studied the application of estimating the spectrum of sinusoidal signals in white noise. More recently, van Waterschoot and Moonen [11], and Giacobello *et al.* [12] have applied high-order linear predictors (order of 1024) to model the spectrum of synthetic audio signals consisting of a combination of harmonic sinusoids and white noise.

In this study we propose to use even higher orders than 1024 to obtain sufficiently accurate information, because we want to model multiple single resonances appearing in the example sounds. Obtaining high-order linear prediction filter estimates is easy in practice using Matlab, for instance. Matlab's `lpc` function uses the Levinson-Durbin recursion [13] to efficiently solve for the LP coefficients, and remarkably high prediction orders, such as 10,000 or more, are feasible. Previously, high-order linear prediction has been used for synthesis of percussive sounds [14] and for modeling of soundboard and soundbox responses of stringed musical instruments [15, 16].

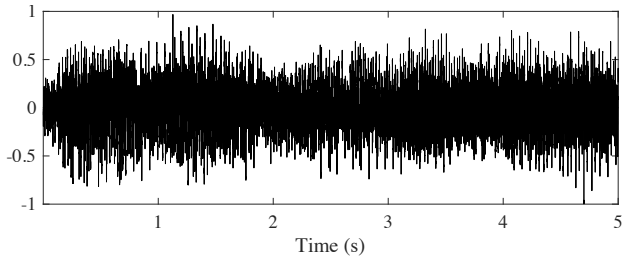
The computational cost of very high-order filtering used for synthesis is not of concern in offline generation of samples to be played back in a real-time application. However, in real-time sound generation, computational costs should be minimized. We show two ways to do so: one method replaces the white noise with velvet noise, and this leads to a simplified implementation of convolution. Another method uses the inverse FFT (fast Fourier transform) algorithm and produces a long buffer of output signal with one transformation. Neither of the methods use a high-order IIR filter, but they need its impulse response or a segment of the sound to be extended as the input signal.

This paper is organized as follows. Section 2 discusses the basic idea of analyzing a short sound example and producing a longer similar sound with life-like quality using filtered white noise. Section 3 discusses the use of velvet noise and Section 4 proposes an FFT-based method as two alternatives for the real-time imple-

* The work of Fabián Esqueda has been supported by the Aalto ELEC Doctoral School.



(a) Original waveform



(b) Synthetic waveform

Figure 1: *(a) Original airplane noise waveform and (b) a synthesized signal obtained with the LP method (P = 10,000) from the 1-second segment indicated with blue markers in (a).*

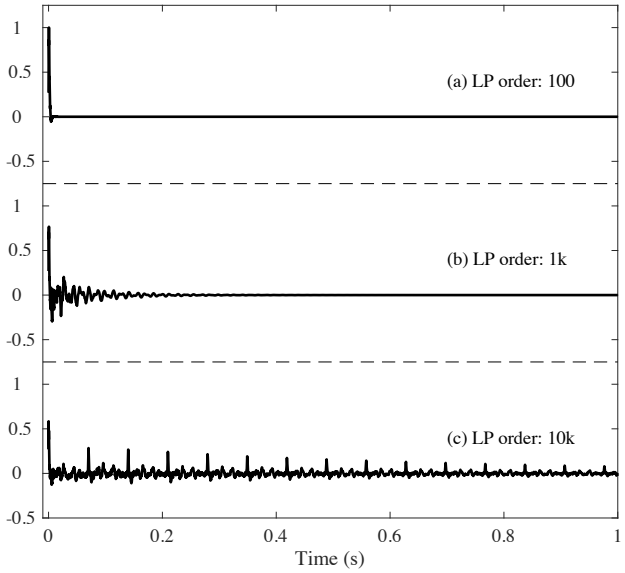


Figure 2: *Impulse responses of different order LP filters: (a) 100, (b) 1000, and (c) 10,000.*

mentation of the endless sound generator. Section 5 concludes this paper and gives ideas for further research on this topic.

2. EXTENDING STATIONARY SOUNDS

Various sounds, such as bus, road, traffic, and airplane cabin noises can be quite stationary, especially in situations where a bus is driving at a constant speed or a plane is cruising at a high altitude. Long sound samples like this are useful as background sounds in movies and games. There is also a need for sounds of this type when conducting listening tests evaluating audio samples in the presence of noise, such for evaluating headphone reproduction in heavy noise [17] or audio-on-audio interference in the presence of road noise [18].

In listening tests, controlled and stationary noises are often wanted, so that the noise signals themselves do not introduce any unwanted or unexpected results to the listening test. For example, if a short sample is looped, it may cause audible clicks each time the sample ends and restarts, or can lead to a distracting frozen-noise effect. Both irregularities can ruin a listening test. Another problem is that a recorded sample may not have a sufficiently long clean part in order to avoid looping problems. Noise recordings often include additional non-stationary audio events, such as braking/accelerating, turbulence, or noises caused by people moving, talking or coughing, which limit the length of the useful part of the sample.

These problems can be avoided by using the proposed high-order LP method. The idea is to use a short, clean stationary part of a sample (e.g. 0.5 to 1 s) to calculate an LP filter that models the frequency characteristics of the given sample. Figure 1(a) shows the waveform of a 5-second clip of airplane noise. The vertical blue lines indicate the selected clean 1-second stationary part which was used in the calculation of the LP filter.

An arbitrarily long signal can be synthesized by filtering white noise with the obtained LP filter. The resulting synthetic signal does not suffer from looping problems or include any unwanted non-stationary sound events which would degrade the quality of the signal. Figure 1(b) shows the resulting synthetic airplane noise, created by filtering 5 seconds of white noise with the LP synthesis filter calculated from the 1-second sample shown in Fig. 1(a) using prediction order of 10,000.

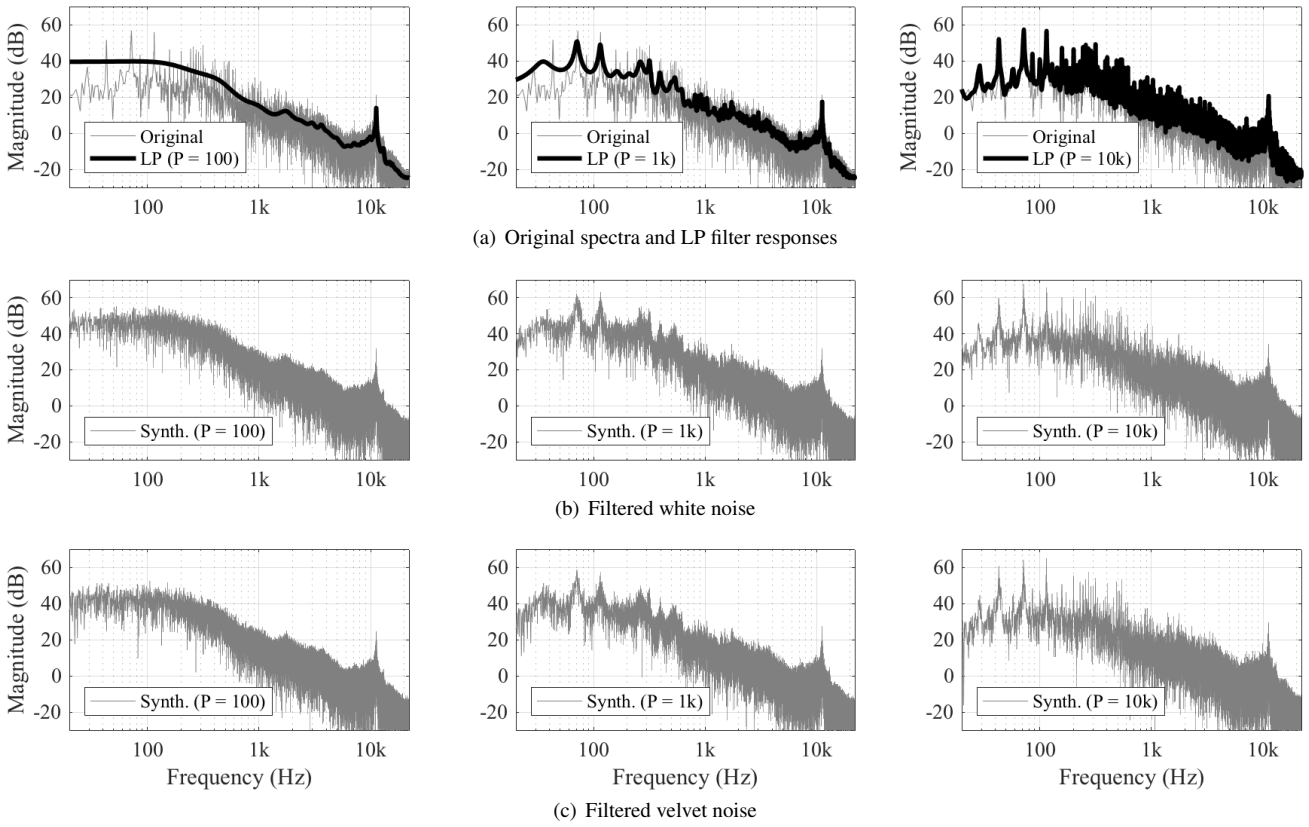
In this section, we study the synthesis of ambient noises and musical sounds using this approach. Additionally, we discuss how to change the pitch of the endless sounds.

2.1. Synthesis of Endless Stationary Audio Signals

All LP calculations in this work were done with Matlab using the built-in `lpc` function, which calculates the linear prediction filter coefficients by minimizing the prediction error in the least squares sense using the Levinson-Durbin recursion [13]. The determined FIR filter coefficients were then used as feedback coefficients to create an all-pole IIR filter, which models the spectrum of the original sample.

Figure 2 shows the calculated impulse responses of different order LP filters, where (a) is of the order of 100, (b) 1000, and (c) 10,000. As expected, the length of the impulse response increases with the LP filter order. The most interesting observation in Fig. 2 is the spiky structure of the impulse response in Fig. 2(c), where the order of the LP filter is 10,000.

Figure 3(a) shows the magnitude responses of the 1-second airplane noise sample (gray lines) from Fig. 1(a) and the magnitude response of different order (P) LP filters (black curves), i.e., from left to right the orders of 100, 1000, and 10,000 correspond to the impulse responses shown in Fig. 2. As can be seen in Fig. 3(a),



(a) Original spectra and LP filter responses

(b) Filtered white noise

(c) Filtered velvet noise

Figure 3: *Magnitude spectra of the original and synthesized airplane cabin noise. Subfigure (a) shows the magnitude spectra of an airplane cabin noise (gray lines) and magnitude responses of LP filters of different order* $P = 100, 1000,$ *and* $10,000$ *(black lines). Subfigures (b) and (c) show spectra of synthetic airplane noises created with white noise and velvet noise, respectively, using different LP filter orders.*

in order to model the low-frequency peaks of the original signal, the order P must be quite high; $P = 1000$ is not large enough to model the peak around 40 Hz, whereas $P = 10,000$ is.

Notice that in this case the order of the LP filter is very high and the filter is time-invariant, unlike in speech codecs in which the LP coefficients are updated every 20 ms or so. Thus, the whole synthesis of the sound can be conducted offline, using one large all-pole filter.

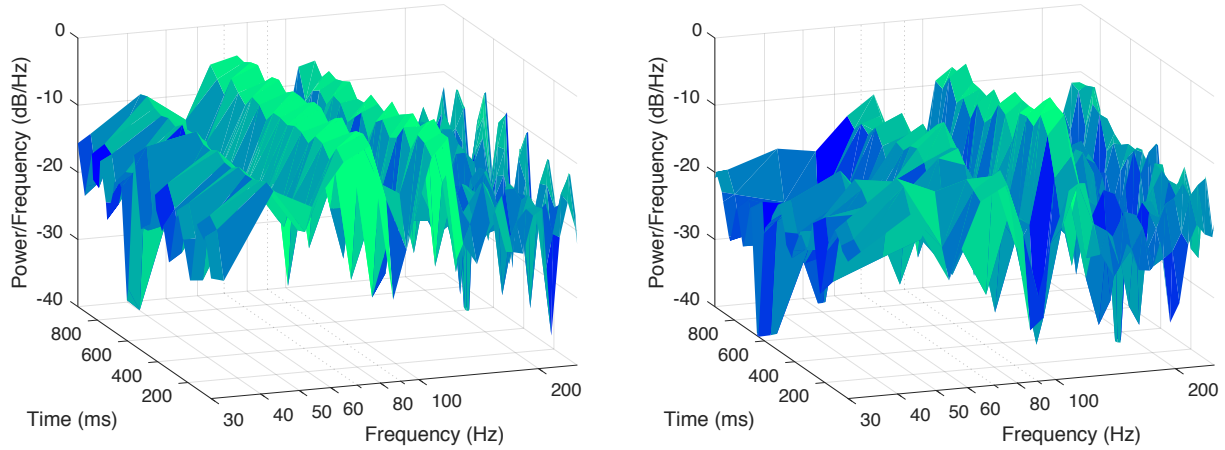
The ability of the high-order LP to capture the spectral details at low frequencies can be seen to help in the synthesis, as is shown in Fig. 3(b). In this figure, the magnitude spectra of the extended signals obtained by filtering a long white noise sequence with all-pole filters of different order are compared. It can be observed in Fig. 3(b) that using a low-order model ($P = 100$), spectral details do not appear at low and mid frequencies. However, when $P = 10,000$, the spectrum of the extended signal contains spikes even at low frequencies.

Surprisingly, although the LP filter is time-invariant, the resulting sounds are very realistic and contain lively variations. The explanation is that the white noise excites the sharp resonances of the LP filter randomly in time, making their energy fluctuate. This is illustrated in Figs. 4(a) and 4(b), which show the spectrograms of the original and synthesized airplane noise signals, respectively. As shown in the rightmost spectrogram, the signal amplitude at the resonances, excited by the white noise, is not constant and changes several dB over time. This can also be seen in Fig. 1(b), which shows the waveform of the synthesized airplane noise that is clearly fluctuating in time. In practice, the amplitude fluctuations are generally larger in the synthetic signals than in the original ones. This is not perceptually annoying, however, but rather appears to contribute to the naturalness of the extended sounds.

Furthermore, the spiky structure seen in the impulse response of the high-order LP filter, in Figure 2(c), creates natural sounding reverberance to the synthesized sound. Note that this feature is not found when the LP order is decreased to 1000, see Fig. 2(b), which otherwise sounds realistic. This implies that a fairly high LP order is required for best results.

The similarity between the magnitude response of the all-pole filter and the magnitude spectrum of the original signal suggest that it may be possible to use the original signal itself in the extension process. This idea was tested and was found to work very well: it is possible to use a short segment of the original signal, such as 0.5 s from a fairly stationary part, and use it as a filter for a white noise input. The resulting extended sound is very similar to the one obtained with high-order LP technique.

The extension technique can also be used to create tonal musical sounds using white noise as input. This has been tested with several musical signals. Figure 5 compares the spectrum of a short piano tone to that of a synthetic, extended version of the same signal. The LP filter order has been selected as 10,000 to capture the



(a) Original (b) Synthesized

Figure 4: *Spectrograms of (a) an original, and (b) LP modeled airplane noise ($P = 10,000$), from 30 Hz to 200 Hz for a 1-second sample, illustrating the fluctuation in low-frequency resonances.*

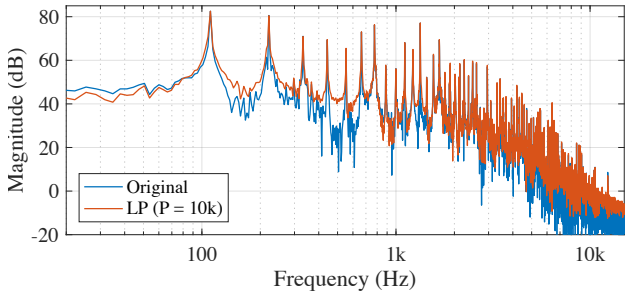


Figure 5: *Magnitude spectrum of a short piano tone (blue), and magnitude response of the LP filter (red) constructed based on that. The order P of the LP filter is 10,000.*

lowest harmonic peaks. It can be observed that the magnitude response of the filter is very similar to the spectrum of the piano tone. Listening confirms that the spectral details are preserved, and that the synthetic tone sounds similar to the original one, except that it is longer and that there are more amplitude fluctuations.

Instead of the standard LP method, it is possible to apply Prony's method or warped LP [19], for example, and hope to obtain good results with a lower model order. However, as the modeling and synthesis can be conducted offline, these options are not considered here. Instead, we will present other ideas for real-time processing in Sections 3 and 4.

The extension examples above are based on a mono signal. Pseudo-stereo signals are easily generated by repeating the extension with another white noise sequence, which is played at the other channel. This idea can be extended to more channels.

2.2. Pitch-Shifting Endless Sounds

It was found that the pitch of the extended signals can be changed easily using resampling. This is equivalent to playing the filter's impulse response at a different rate, when the output sample rate remains unchanged. A sampling-rate conversion technique can be used for this purpose.

For increasing the pitch, the sample rate of the impulse response must be lowered. Then, when the processed impulse response is convolved with white noise at the original sample rate, the pitch is increased. Similarly, the pitch of the extended sound can be lowered by increasing its sample rate and playing it back at the original rate.

This method does not require time-stretching, as the signal duration does not depend on the impulse response length. Notice that the impulse response will get shorter during downsampling and longer during upsampling, however. To better retain the original timbre, formant-preservation techniques can be used, but this topic is not discussed further in this paper.

3. REAL-TIME SYNTHESIS WITH VELVET NOISE

A direct time-domain implementation of the filtering of white noise with a very high-order all-pole filter is computationally intensive and can lead to numerical problems. It is safer for numerical reasons to evaluate the impulse response of the LP filter and convolve white noise with it. However, the computational complexity becomes even higher in this case, since there are generally more samples in the impulse response than there are LP prediction coefficients. The impulse response is often almost as long as the original signal segment to be processed. It is also possible to use the signal segment itself as the filter. To alleviate the computational burden for real-time synthesis, we suggest to use sparse white noise called velvet noise for synthesis.

Velvet noise refers to a sparse pseudo-random sequence consisting of sample values $+1$, -1, and 0 only. Usually more than 90% of the sample values are zero, however. Velvet noise has been originally proposed for artificial reverberation [20–23], where the input signal is convolved with a velvet-noise sequence. This is very efficient, because there are no multiplications, and the number of additions is greatly reduced in comparison to convolution with regular (non-sparse) white noise. Recent work also proposed the use of a short velvet-noise sequence for decorrelating audio signals [24, 25].

The convolution of an arbitrary input signal with a velvet-noise sequence can be implemented with a multitap delay line, as show in Fig. 6(a) [23]. The location and sign of each non-zero sample in the velvet noise determines one output tap in the multi-tap delay line. The sums of the signal samples at the locations of the positive and negative impulses in the velvet noise can be computed separately. Finally, the two sums are subtracted to obtain the output sample.

In the endless sound application considered in this paper, the role of the velvet noise is different than in the reverb or decorrelation application. Now, the velvet noise becomes the input signal, which is convolved with the short signal segment. The signal segment $x(n)$ can be stored in a buffer (table), and the taps of a multitap delay line, where the tap locations are determined by the velvet-noise sequence, move along it. This is illustrated in Fig. 6(b), which shows a time-varying multi-tap delay line in which the taps (read pointers) march one sample to the right at every sampling step. In this case, velvet noise can be generated in real time: every time a new velvet-noise frame begins, two random numbers are needed to determine the location and sign of the new tap. The oldest tap that reaches the end of the delay line is decimated. The computational efficiency of the proposed filtering of the velvet noise sequence is very high, as it is comparable to that of the standard velvet-noise convolution.

A velvet-noise signal with a density of 4410 samples per second (i.e., one non-zero impulse in a range of 10 samples) was used for testing this method. This corresponds to a 90% reduction in operations. Since velvet-noise convolution does not require multiplications but only additions, a total reduction of 95% is obtained w.r.t. standard convolution with white noise. In practice, the required velvet-noise density depends on the signal type. It is known that a lower density can sound smooth when the velvet noise is lowpass-filtered [20], which in this case corresponds to an input signal of lowpass type.

Figure 3(c) shows the magnitude spectra of extended signals obtained by filtering velvet noise, as described above. Comparison with Fig. 3(b) reveals that the results are very similar to those obtained by filtering regular white noise, which requires about 20 times more operations. The endless sound synthesis based on velvet-noise filtering can be executed very efficiently in real time, and additional processing, such as gain control or filtering, can be adjusted continuously. Below we propose another efficient method, which is based on FFT techniques.

4. ENDLESS SOUND SYNTHESIS USING INVERSE FFT

We propose yet another interesting technique for creating virtually endless sounds, which utilizes the concept of fast convolution [22, 26–28]. It is well known that frequency-domain convolution using the FFT becomes more efficient than the time-domain convolution when the convolved sequences are long. When two sequences of length N are convolved, the direct time-domain convolution takes approximately N^2 multiplications and additions whereas the FFT takes the order of $N \log(N)$ operations only [22, 29]. The difference in computational cost between these two implementations becomes significant even at fair FFT lengths, such as a few thousand samples.

The main point in the fast convolution is to utilize the convolution theorem [28, Ch. 11], which states that the time-domain convolution of two signals is equivalent to the point-wise multiplication of their spectra:

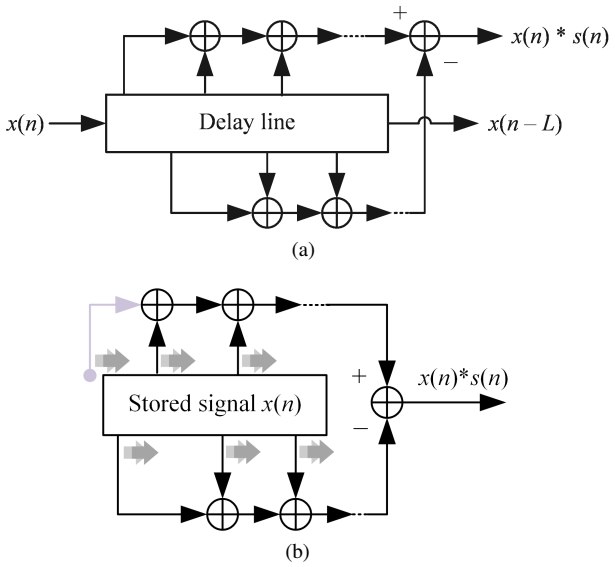


Figure 6: *(a) Convolution of an arbitrary signal $x(n)$ with a velvet-noise sequence $s(n)$ corresponds to a multi-tap delay line from which the output is obtained as the difference of two subsums. (b) Convolution of a short signal segment $x(n)$ with a velvet-noise signal can be implemented as a multi-tap delay line with moving output taps.*

$$v(n) * x(w) \leftrightarrow V(f)X(f), \tag{1}$$

where, in this application, $v(n)$ is a white noise signal and $x(n)$ is the signal segment (or the impulse response of the LP filter), and $X(f)$ and $V(f)$ are their Fourier transforms, respectively. Figure 7(a) shows a block diagram of the basic fast convolution method. Notice that the output is obtained by using the inverse FFT (IFFT).

The frequency-domain signals X and V can be written as

$$X = R_x e^{j\theta_x}, \tag{2}$$
$$V = R_v e^{j\theta_v}, \tag{3}$$

where R and θ are the magnitude and phase vectors of the two signals, respectively. Further, the multiplication of the frequency-domain signals can be written as

$$Y = VX = R_v e^{j\theta_v} R_x e^{j\theta_x} = R_v R_x e^{j(\theta_v + \theta_x)}. \tag{4}$$

By taking the IFFT of Y, one frame (N samples) of the convolved time-domain signal $y(n)$ is synthesized. As our aim is essentially to create a synthesized sound similar to the original but longer, we can apply zero padding to the short original sample, before taking the FFT, and use a white noise sequence of the same length.

Additionally, as it is known that the white noise has ideally a constant power spectrum and a random phase, the white noise can be produced directly in the frequency domain (instead of first creating it in the time domain and then transforming it to the frequency domain with the FFT). It is helpful to assume that the magnitude response of the short white noise sequence is flat, although this is not exactly true for short random signals. Siddiq used a

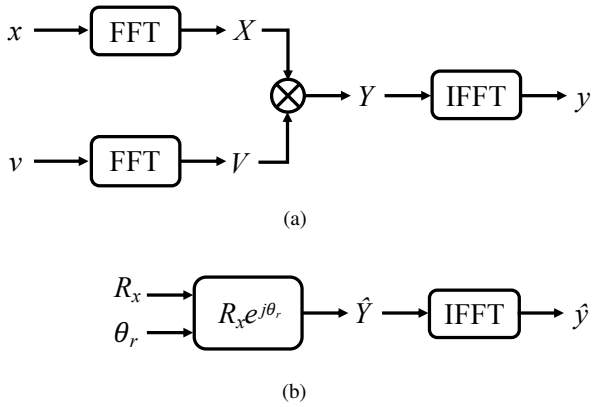


Figure 7: *(a) Regular fast convolution and (b) the proposed IFFT-based synthesis, where x is the signal segment to be extended, v is a white noise sequence, R_x is the magnitude of spectrum X, and θ_r is a randomized phase with values between $-\pi$ and π.*

similar approach to generate colored noise in granular texture synthesis [5].

Now, when we look at the last product in Equation (4), we can set the magnitude spectrum of the white noise to unity, so that the magnitude response R_x is left unchanged. Furthermore, as adding a random component to the original phase randomizes it, we may as well delete the original phase and replace it with a random one, resulting in

$$R_x e^{j(\theta_r+\theta_x)} \rightarrow R_x e^{j\theta_r}, \tag{5}$$

where θ_r is the randomized phase. Thus, the whole process of frequency-domain convolution is reduced to taking the FFT of the original signal segment (or impulse response), replacing its phase with random numbers while keeping the original magnitude, and taking the IFFT, as shown in Figure 7(b).

Stricktly speaking, in Figure 7(b) the polar coordinate inputs R_x and θ_r are transformed to Cartesian coordinates to construct $\hat{Y}$, an approximation of Y. By taking the IFFT, one frame of the time-domain waveform $\hat{y}(n)$ is obtained. Both signals R_x and θ_r can be constructed offline, R_x is the magnitude of the original sample, and θ_r is constructed as

$$\theta_r = [0,\ r,\ 0,\ -\tilde{r}], \tag{6}$$

where the two zeros in the phase vector are located at the DC and the Nyquist frequency, r contains uniformly distributed random values between $-\pi$ and π, and $\tilde{r}$ is r with reversed elements. Notice that the sign of phase values $\tilde{r}$ must be opposite to those or r, because they represent the negative frequencies. The length of both r and $\tilde{r}$ is $(N/2) - 1$, where N is the FFT length. Parameter N is chosen to be the same as the length of the zero-padded signal.

Note that with the technique described above and in Figure 7(b), R_x can be calculated directly as the FFT magnitude of the original signal, without the need of LP estimation. In fact, a high-order LP filter very closely imitates the magnitude spectrum of the signal segment. Figure 8(b) gives an example in which the same 1-second segment as in Fig. 1(a) has been employed. As can be seen, the produced signal fluctuates in a similar way as the one generated using filtering white noise with the all-pole filter.

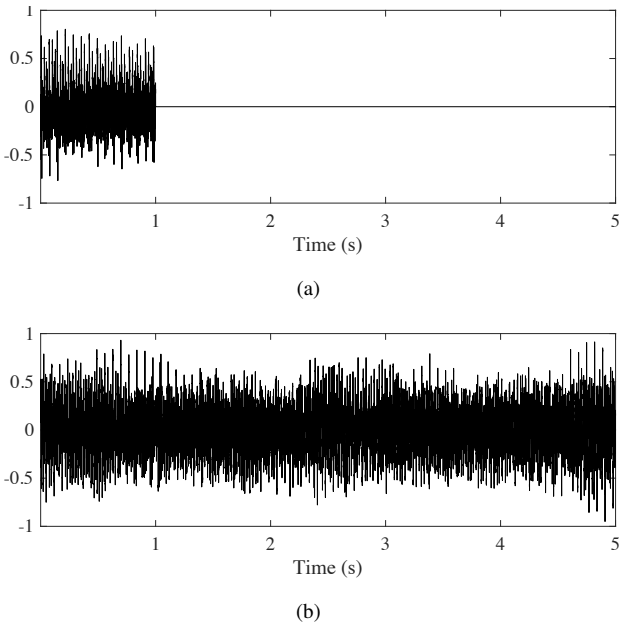


Figure 8: *(a) Original airplane noise segment (cf. Fig. 1(a)), which has been expanded with zero padding to a desired length. (b) Synthesized waveform obtained with the IFFT technique of Fig. 7(b).*

4.1. Concatenation Employing Circular Time

It is a remarkable fact that windowing or the overlap-add method are not necessary with the proposed IFFT synthesis technique. With this approach, copies of a long segment of the produced random-phase signal can simply be concatenated without introducing discontinuities at the junction points. This is a consequence of the fast convolution operation, where the time-domain representation is circular, and is therefore also called circular convolution [13, 27].

When the extended segment is long enough, such as 4 seconds or longer, it will be difficult to notice that it repeats [1]. The best option for endless sound synthesis thus appears to be to synthesize one long extended signal segment using the IFFT and then repeat it. However, if more than one extended segment is synthesized from the same input signal and they are concatenated, hoping to produce extra variation, they will usually produce clicks at the connection points. In this case a crossfade method would be needed to suppress the clicks. Naturally, this idea is not recommended, as it is much easier to produce only a single segment using IFFT and repeat it.

The next example illustrates the fact that the repetition of a single segment works fine. We use a 4000-sample segment of a piano tone as the input signal and apply the method of Fig. 7(b). The IFFT length N is 4096. Figure 9(a) shows two concatenated copies of this extended signal, leading to a signal of length 8192. Figure 9(b) zooms to the joint of the two copies, showing that there is no discontinuity, but that the end of the segment fits perfectly to its beginning.

[1] However, it has been shown in laboratory experiments that people can notice much longer repetitions in sound [30].

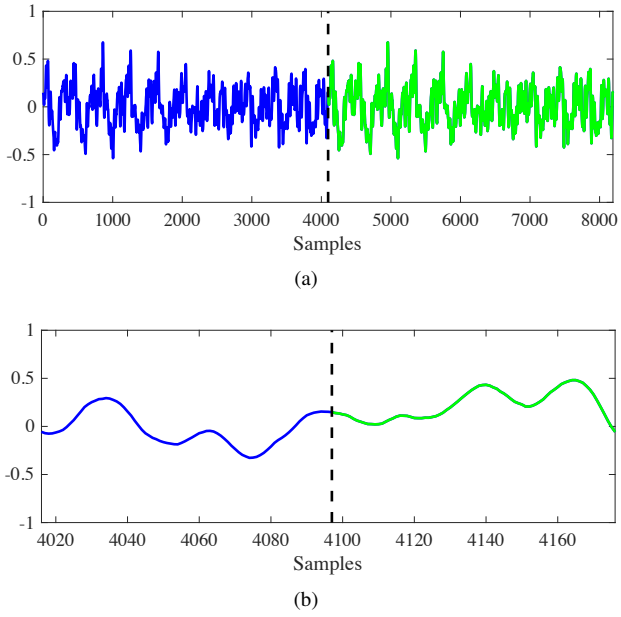


Figure 9: *Two concatenated copies of the same signal obtained with the proposed IFFT method, first copy plotted with blue line and the second with green line. Subfigure (a) shows the signals in their full length, and (b) zooms to the point where the signals are joined, illustrating the perfect fit of the junction point. The dashed vertical line indicates the beginning of the second copy of the signal.*

4.2. Comparison of Methods

So far there are three principally different methods for creating endless sounds: filtering of white noise with the LP-based all-pole filter, filtering a signal segment with velvet noise, and IFFT synthesis based on a signal segment. The filtering of regular white noise is the basic method, which also leads to the largest computational load, whereas the IFFT method is the most efficient one. Also the method based on filtering velvet noise is computationally efficient, and as it produces the output signal one sample at a time, it allows amplitude modulation or other modifications to be executed during synthesis. The filtering methods are suitable for low-latency application whereas the IFFT method is only suitable for synthesizing the signal in advance.

As a test case, we measured the time it takes to produce 1 minute of sound from a short signal segment using Matlab. For the first method, an LP filter of order 1000 was used, which produced an impulse response that could be truncated to the length 10,000 samples. The convolution of this filter impulse response with 2,646,000 samples ($60 \times 44{,}100$) of white noise took in average about 3.4 s. This is much less than 1 minute, so it should be easy to run the synthesis in real time.

For comparison, the IFFT of the length 2,646,000 produced the 1-minute segment of the extended signal at one go, and it took in average 0.14 s to compute[2]. Remarkably, practically the same result was obtained by producing 4.0 s of the extended signal with the IFFT in just about 0.0005 s, and by repeating it 15 times (at no extra cost!). As listeners do not generally notice the repetition over several seconds and as there are no clicks at the connection points, this produces equally good results as the longer IFFT synthesis.

[2]Matlab's FFT algorithm is fastest when the length is a power of 2, but 2,646,000 is not.

5. CONCLUSION AND FUTURE WORK

This paper has discussed the use of linear prediction and the inverse FFT for solving the small data problem in sampling synthesis. Useful methods were proposed to extend the duration of short example sounds to an arbitrary length. The first method employs high-order linear prediction to a selected short segment in the original recording. Surprisingly, the impulse response of the filter can be replaced with a short segment of the original sound signal.

A synthetic sound of arbitrary length may then be produced by filtering white noise with a segment of the original sound. Lively variations appear in the produced sound, as the random signal pumps energy to the narrow resonances contained in the signal's spectrum. These variations are shown to be generally larger in terms of amplitude variance than in the original sound, but they help to make the extended sound appear natural and non-frozen. Sound synthesis can take place offline so that during presentation the generated signal is played back from computer memory, like in sampling synthesis. In this case, the computational cost of running a large all-pole filter or long convolution is of no concern.

Alternatively, we proposed to reduce the computational cost for real-time synthesis by replacing the white noise signal with velvet noise or by generating the noisy extended signal using the inverse FFT from the original magnitude and a random phase spectrum. The IFFT-based method produces a long segment of the output signal at one time. Another unexpected result is that the segment produced by the IFFT method can be repeated by concatenating copies of itself without the need of windowing or crossfading. This property comes from the fact that the fast convolution, which is the basis of the proposed IFFT synthesis method, implements a circular convolution in the time domain.

Future work may consider the analysis of perceived differences in extended samples in comparison to the original recording. It would be desirable to find a method to control the fluctuations of resonances in the synthetic signal, although they are not annoying generally. It would also be of interest to consider formant-preserving pitch-shifting techniques, which could be used to build a sampling synthesizer based on the ideas proposed in this paper.

Audio examples related to this paper are available online at http://research.spa.aalto.fi/publications/papers/dafx18-endless/. The examples include synthetic signals obtained with different LP orders and IFFT lengths, and various sound types, such as the airplane cabin noise, the piano tone, a distorted guitar, and an excerpt taken from a recording by the Beatles.

6. REFERENCES

[1] H. Ploner-Bernard, A. Sontacchi, G. Lichtenegger, and S. Vössner, "Sound-system design for a professional full-flight simulator," in *Proc. Int. Conf. Digital Audio Effects (DAFx-05)*, Madrid, Spain, Sept. 2005, pp. 36–41.

[2] V. Mäntyniemi, R. Mignot, and V. Välimäki, "REMES final report," Tech. Rep., Science+Technology 16/2014, Aalto University, Helsinki, Finland, 2014, Available at https://aaltodoc.aalto.fi/handle/123456789/14705.

[3] M. Fröjd and A. Horner, "Sound texture synthesis using an overlap-add/granular synthesis approach," *J. Audio Eng. Soc.*, vol. 57, no. 1/2, pp. 29–37, Jan./Feb. 2009.

[4] D. Schwarz, A. Roebel, C. Yeh, and A. LaBurthe, "Concatenative sound texture synthesis methods and evaluation," in *Proc. 19th Int. Conf. Digital Audio Effects (DAFx-16)*, Brno, Czech Republic, Sept. 2016, pp. 217–224.

[5] S. Siddiq, "Morphing granular sounds," in *Proc. 18th Int. Conf. Digital Audio Effects (DAFx-15)*, Trondheim, Norway, Nov./Dec. 2015, pp. 4–11.

[6] J.-F. Charles, "A tutorial on spectral sound processing using Max/MSP and Jitter," *Computer Music J.*, vol. 32, no. 3, pp. 87–102, 2008.

[7] J. A. Moorer, "The use of linear prediction of speech in computer music applications," *J. Audio Eng. Soc.*, vol. 27, no. 3, pp. 134–140, Mar. 1979.

[8] P. R. Cook, *Real Sound Synthesis for Interactive Applications*, AK Peeters, Ltd., 2002.

[9] L. B. Jackson, D. W. Tufts, F. K. Soong, and R. M. Rao, "Frequency estimation by linear prediction," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, Tulsa, OK, USA, Apr. 1978, pp. 352–356.

[10] S. M. Kay, "The effects of noise in the autoregressive spectral estimator," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. 27, no. 5, pp. 478–485, Oct. 1979.

[11] T. van Waterschoot and M. Moonen, "Comparison of linear prediction models for audio signals," *EURASIP J. Audio, Speech, Music Process.*, vol. 2008, pp. 1–24, Dec. 2008.

[12] D. Giacobello, T. van Waterschoot, M. G. Christensen, S. H. Jensen, and M. Moonen, "High-order sparse linear predictors for audio processing," in *Proc. 18th European Signal Process. Conf.*, Aalborg, Denmark, Aug. 2010, pp. 234–238.

[13] L. B. Jackson, *Digital Filters and Signal Processing*, Kluwer, Boston, MA, USA, second edition, 1989.

[14] M. Sandler, "Analysis and synthesis of atonal percussion using high order linear predictive coding," *Appl. Acoust.*, vol. 30, no. 2–3, pp. 247–264, 1990.

[15] M. Karjalainen and J. O. Smith, "Body modeling techniques for string instrument synthesis," in *Proc. Int. Computer Music Conf.*, Hong Kong, Aug. 1996, pp. 232–239.

[16] F. v. Türckheim, T. Smit, and R. Mores, "String instrument body modeling using FIR filter design and autoregressive parameter estimation," in *Proc. Int. Conf. Digital Audio Effects (DAFx-10)*, Graz, Austria, Sept. 2010.

[17] J. Rämö and V. Välimäki, "Signal processing framework for virtual headphone listening tests in a noisy environment," in *Proc. Audio Eng. Soc. 132nd Conv.*, Budapest, Hungary, Apr. 2012.

[18] J. Francombe, R. Mason, M. Dewhirst, and S. Bech, "Elicitation of attributes for the evaluation of audio-on-audio interference," *J. Acoust. Soc. Am.*, vol. 136, no. 5, pp. 2630–2641, Nov. 2014.

[19] A. Härmä, M. Karjalainen, L. Savioja, V. Välimäki, U. K. Laine, and J. Huopaniemi, "Frequency-warped signal processing for audio applications," *J. Audio Eng. Soc.*, vol. 48, no. 11, pp. 1011–1031, Nov. 2000.

[20] M. Karjalainen and H. Järveläinen, "Reverberation modeling using velvet noise," in *Proc. Audio Eng. Soc. 30th Int. Conf. Intelligent Audio Environments*, Saariselkä, Finland, Mar. 2007.

[21] K.-S. Lee, J. S. Abel, V. Välimäki, T. Stilson, and D. P. Berners, "The switched convolution reverberator," *J. Audio Eng. Soc.*, vol. 60, no. 4, pp. 227–236, Apr. 2012.

[22] V. Välimäki J. D. Parker, L. Savioja, J. O. Smith, and J. S. Abel, "Fifty years of artificial reverberation," *IEEE Trans. Audio, Speech, and Lang. Processing*, vol. 20, no. 5, pp. 1421–1448, Jul. 2012.

[23] V. Välimäki, B. Holm-Rasmussen, B. Alary, and H.-M. Lehtonen, "Late reverberation synthesis using filtered velvet noise," *Appl. Sci.*, vol. 7, no. 483, May 2017.

[24] B. Alary, A. Politis, and V. Välimäki, "Velvet-noise decorrelator," in *Proc. Int. Conf. Digital Audio Effects (DAFx-17)*, Edinburgh, UK, Sept. 2017, pp. 405–411.

[25] S. J. Schlecht, B. Alary, V. Välimäki, and E. A. P. Habets, "Optimized velvet-noise decorrelator," in *Proc. Int. Conf. Digital Audio Effects (DAFx-18)*, Aveiro, Portugal, Sept. 2018, elsewhere in these proceedings.

[26] G. Stockham, Jr., "High speed convolution and correlation," in *Proc. Spring Joint Comput. Conf.*, Boston, MA, USA, Apr. 1966, pp. 229–233.

[27] D. Arfib, F. Keiler, U. Zölzer, V. Verfaille, and J. Bonada, "Time-frequency processing," in *DAFX: Digital Audio Effects, Second Edition*, U. Zölzer, Ed., pp. 219–278. Wiley, 2011.

[28] J. D. Reiss and A.P. McPherson, *Audio Effects: Theory, Implementation and Application*, CRC Press, Taylor & Francis Group, Boca Raton, FL, USA, 2015.

[29] J. O. Smith, *Spectral Audio Signal Processing*, Online book, http://ccrma.stanford.edu/˜jos/sasp/, 2011 edition, Accessed 23 March, 2018.

[30] R. M. Warren, J. A. Bashford, J. M. Cooley, and B. S. Brubaker, "Detection of acoustic repetition for very long stochastic patterns," *Perception & Psychophysics*, vol. 63, no. 1, pp. 175–182, Jan 2001.

AUTOENCODING NEURAL NETWORKS AS MUSICAL AUDIO SYNTHESIZERS

Joseph Colonel
The Cooper Union for the Advancement of Science and Art
NYC, New York, USA
colone@cooper.edu

Christopher Curro
The Cooper Union for the Advancement of Science and Art
NYC, New York, USA
curro@cooper.edu

Sam Keene
The Cooper Union for the Advancement of Science and Art
NYC, New York, USA
keene@cooper.edu

ABSTRACT

A method for musical audio synthesis using autoencoding neural networks is proposed. The autoencoder is trained to compress and reconstruct magnitude short-time Fourier transform frames. The autoencoder produces a spectrogram by activating its smallest hidden layer, and a phase response is calculated using real-time phase gradient heap integration. Taking an inverse short-time Fourier transform produces the audio signal. Our algorithm is light-weight when compared to current state-of-the-art audio-producing machine learning algorithms. We outline our design process, produce metrics, and detail an open-source Python implementation of our model.

1. INTRODUCTION

There are many different methods of digital sound synthesis. Three traditional methods are additive, subtractive, and frequency modulation (FM) synthesis. In additive synthesis, waveforms such as sine, triangle, and sawtooth waves are generated and added to one another to create a sound. The parameters of each waveform in the sum are controlled by the musician. In subtractive synthesis, a waveform such as a square wave is filtered to subtract and alter harmonics. In this case, the parameters of the filter and input waveform are controlled by the musician. Lastly, in FM synthesis the timbre of a waveform is generated by one waveform modulating the frequency of another. In this method, musicians control the parameters of both waveforms, and the manner in which one modulates the other.

Recently, machine learning techniques have been applied to musical audio sythesis. One version of Google's Wavenet architecture uses convolutional neural networks (CNNs) trained on piano performance recordings to prooduce raw audio one sample at a time [1]. The outputs of this neural network have been described as sounding like a professional piano player striking random notes. Another topology, presented by Dadabots, uses recurrent neural networks (RNNs) trained to reproduce a given piece of music [2]. These RNNs can be given a random initialization and then left to produce music in batches of raw audio samples. Another Google project, Magenta [3], uses neural network autoencoders (autoencoders) to interpolate audio between different instrument's timbres. While all notable in scope and ability, these models require immense computing power to train and thus strip musicians of full control over the tools.

In this paper, we present a new method for sound synthesis that incorporates deep autoencoders while remaining light-weight. This method is based off techniques for constructing audio-handling autoencoders outlined in [4]. We first train an autoencoder to encode and decode magnitude short-time Fourier transform (STFT) frames generated by audio recorded from a subtractive synthesizer. This training corpus consists of five-octave C Major scales on various synthesizer patches. Once training is complete, we bypass the encoder and directly activate the smallest hidden layer of the autoencoder. This activation produces a magnitude STFT frame at the output. Once several frames are produced, phase gradient integration is used to construct a phase response for the magnitude STFT. Finally, an inverse STFT is performed to synthesize audio. This model is easy to train when compared to other state-of-the-art methods, allowing for musicians to have full control over the tool.

This paper presents improvements over the methods outlined in [4]. First, this paper incorporates a phase construction method not utilized in [4], which allows for music synthesis through activating the autoencoder's latent space. The method presented in [4] requires an input time signal's phase response to construct a time signal at the output. Second, this work explores asymmetrical autoencoder design via input augmentation, which [4] did not. Third, this work compares the performance of several cost functions in training the autoencoder, whereas [4] only used mean squared error (MSE).

We have coded an open-source implementation of our method in Python, available at $github.com/JTColonel/canne\_synth$.

2. AUTOENCODING NEURAL NETWORKS

2.1. Mathematical Formulation

An autoencoder is typically used for unsupervised learning of an encoding scheme for a given input domain, and is comprised of an encoder and a decoder [5]. For our purposes, the encoder is forced to shrink the dimension of an input into a latent space using a discrete number of values, or "neurons." The decoder then expands the dimension of the latent space to that of the input, in a manner that reconstructs the original input.

We will first restrict our discussion to a single layer model where the encoder maps an input vector $x \in \mathbb{R}^d$ to the hidden layer $y \in \mathbb{R}^e$, where $d > e$. Then, the decoder maps y to $\hat{x} \in \mathbb{R}^d$. In this formulation, the encoder maps $x \rightarrow y$ via

$$y = f(Wx + b) \tag{1}$$

where $W \in \mathbb{R}^{(e \times d)}$, $b \in \mathbb{R}^e$, and $f(\cdot)$ is an activation function that imposes a non-linearity in the neural network. The decoder has a similar formulation:

$$\hat{x} = f(W_{\text{out}} y + b_{\text{out}}) \tag{2}$$

with $W_{\text{out}} \in \mathbb{R}^{(d \times e)}$, $b_{out} \in \mathbb{R}^d$.

A multi-layer autoencoder acts in much the same way as a single-layer autoencoder. The encoder contains $n > 1$ layers and the decoder contains $m > 1$ layers. Using equation 1 for each

mapping, the encoder maps $x \rightarrow x_1 \rightarrow \ldots \rightarrow x_n$. Treating x_n as y in equation 2, the decoder maps $x_n \rightarrow x_{n+1} \rightarrow \ldots \rightarrow x_{n+m} = \hat{x}$.

The autoencoder trains the weights of the W's and b's to minimize some cost function. This cost function should minimize the distance between input and output values.The choice of activation functions $f(\cdot)$ and cost functions relies on the domain of a given task.

2.2. Learning Task Description

In our work we train a multi-layer autoencoder to learn representations of musical audio. Our aim is to train an autoencoder to contain high level, descriptive audio features in a low dimensional latent space that can be reasonably handled by a musician. As in the formulation above, we impose dimension reduction at each layer of the encoder until we reach the desired dimensionality.

The autoencoding neural network used here takes 2049 points from a 4096-point magnitude STFT $s_n(m)$ as its target, where n denotes the frame index of the STFT and m denotes the frequency index. Each frame is normalized to $[0, 1]$.

The cost function used in this work is spectral convergence (SC) [6]:

$$C(\theta_n) = \sqrt{\frac{\sum_{m=0}^{M-1}(s_n(m) - \hat{s}_n(m))^2}{\sum_{m=0}^{M-1}(s_n(m))^2}} \tag{3}$$

where θ_n is the autoencoder's trainable weight variables,$s_n(m)$ is the original magnitude STFT frame, $\hat{s}_n(m)$ is the reconstructed magnitude STFT frame, and M is the total number of frequency bins in the STFT.

We fully discuss our decision to use SC in section 3.

2.3. Corpus

All topologies presented in this paper are trained using approximately 79,000 magnitude STFT frames, with an additional 6000 frames held out for testing and another 6000 for validation. This makes the corpus 91,000 frames in total. The audio used to generate these frames is composed of five octave C Major scales recorded from a MicroKORG synthesizer/vocoder across 80 patches. 70 patches make up the training set, 5 patches make up the testing set, and 5 patches make up the validation set. These patches ensured that different timbres were present in the corpus. To ensure the integrity of the testing and validation sets, the dataset was split on the "clip" level. This means that the frames in each of the three sets were generated from distinct passages in the recording, which prevents duplicate or nearly duplicate frames from appearing across the three sets.

By restricting the corpus to single notes played on a MicroKORG, the autoencoder needs only to learn higher level features of harmonic synthesizer content. These tones often have time variant timbres and effects, such as echo and overdrive. Thus the autoencoder is also tasked with learning high level representations of these effects. We have made our corpus available as both *.wav* files and as a *.npy* record. Furthermore, we provide a script that creates new corpora, formatted for training our autoencoder, given a *.wav* file.

3. NEURAL NETWORK CONSTRUCTION

3.1. Topology

A fully-connected, feed-forward neural network acts as our autoencoder. Refer to Figure 1 for an explicit diagram of the network architecture. Our decisions regarding activation functions, input augmentation, and additive biases are discussed below.

3.2. ReLU Activation Function

In order for training to converge, the rectified linear unit (ReLU) was chosen as the activation function for each layer of the autoencoder [7]. The ReLU is formulated as

$$f(x) = \begin{cases} 0 & , x < 0 \\ x & , x \geq 0 \end{cases} \tag{4}$$

This activation function has the benefit of having a gradient of either zero or one, thus avoiding the vanishing gradient problem [8].

Following [4], we found that using additive bias terms b in Equation 1 created a noise floor within the autoencoder, thus we chose to leave them out in the interest of musical applications.

3.3. Spectral Convergence Cost Function with L2 Penalty

As mentioned above SC (Eqn. 3) was chosen as the cost function for this autoencoder instead of mean squared error (MSE)

$$C(\theta_n) = \frac{1}{M}\sum_{m=0}^{M-1}(s_n(m) - \hat{s}_n(m))^2 \tag{5}$$

or mean absolute error (MAE)

$$C(\theta_n) = \frac{1}{M}\sum_{m=0}^{M-1}|s_n(m) - \hat{s}_n(m)| \tag{6}$$

The advantages of using SC as a cost function are twofold. First, its numerator penalizes the autoencoder in much the same way mean squared error (MSE) does. That is to say, reconstructed frames dissimilar from their input are penalized on a sample-by-sample basis, and the squared sum of these deviations dictates magnitude of the cost.

The second advantage, and the primary reason SC was chosen over MSE, is that its denominator penalizes the autoencoder in proportion to the total spectral power of the input signal. Because the training corpus used here is comprised of "simple" harmonic content (i.e. not chords, vocals, percussion, etc.), much of a given input's frequency bins will have zero or close to zero amplitude. SC's normalizing factor gives the autoencoder less leeway in reconstructing harmonically simple inputs than MSE or MAE. Refer to Figure 2 for diagrams demonstrating the reconstructive capabilities each cost function produces.

As mentioned in [4], we found that the autoencoder would not always converge when using SC by itself as the cost function. Thus, we added an L2 penalty to the cost function

$$C(\theta_n) = \sqrt{\frac{\sum_{m=0}^{M-1}(s_n(m) - \hat{s}_n(m))^2}{\sum_{m=0}^{M-1}(s_n(m))^2}} + \lambda_{l2}\|\theta_n\|_2 \tag{7}$$

Figure 1: Autoencoder Topology used. Each layer is fully-connected and feed-forward. The value above each layer denotes the width of the hidden layer.

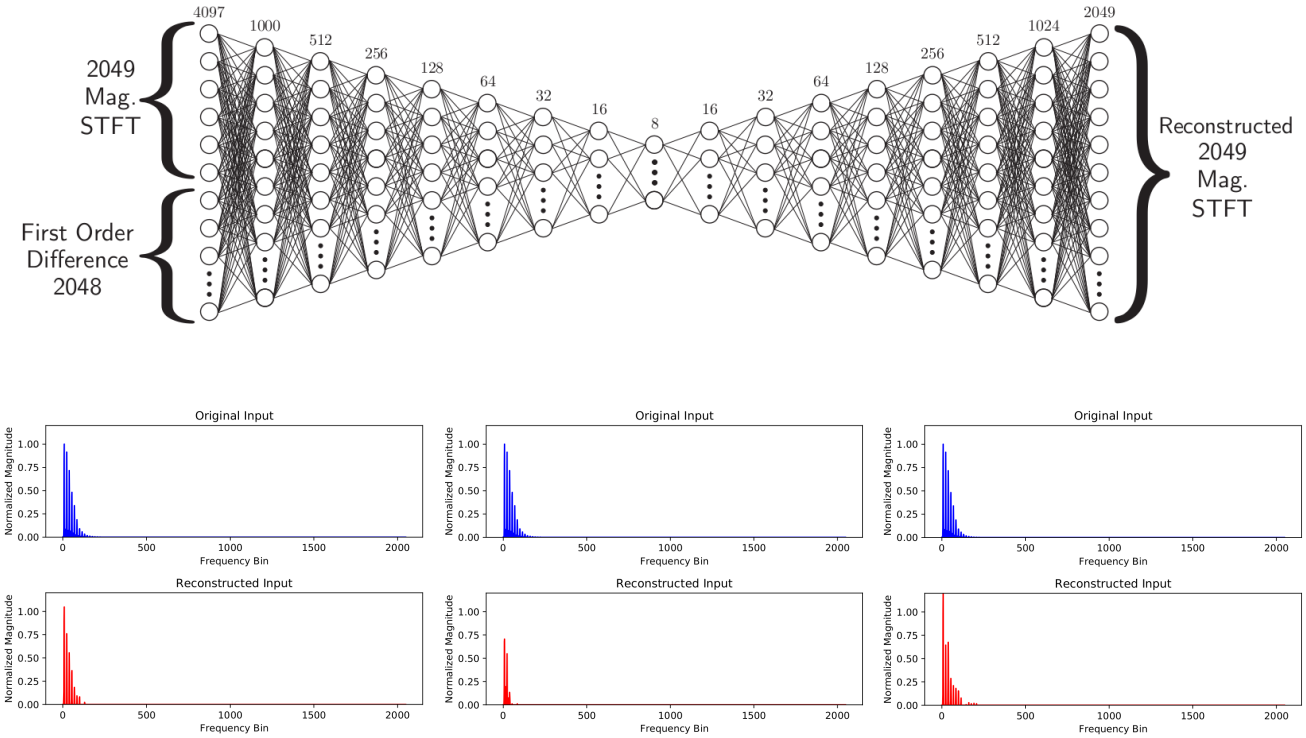


Figure 2: Sample input and recosntruction using three different cost functions: SC (left), MSE (center), and MAE (right)

where λ_{l2} is a tuneable hyperparameter and $\|\theta_n\|_2$ is the Euclidean norm of the autoencoder's weights [9]. This normalization technique encourages the autoencoder to use smaller weights in training, which we found to improve convergence. We set λ_{l2} to 10^{-20}. This value of λ_{l2} is large enough to prevent runaway weights while still allowing the SC term to dominate in the loss evaluation.

3.4. Input Augmentation

Despite these design choices, we still found the performance of the autoencoder to be subpar. To help the autoencoder enrich its encodings, we augmented its input with higher-order information. We tried augmenting the input with different permutations of the input magnitude spectrum's first-order difference,

$$x_1[n] = x[n+1] - x[n] \tag{8}$$

second-order difference,

$$x_2[n] = x_1[n+1] - x_1[n] \tag{9}$$

and Mel-Frequency Cepstral Coefficients (MFCCs).

MFCCs have seen widespread use in automatic speech recognition, and can be thought of as the "spectrum of the spectrum." In our application, a 512 band mel-scaled log-transform of $s_n(m)$ is taken. Then, a 256-point discrete-cosine transform is performed. The resulting aplitudes of this signal are the MFCCs. Typically the first few cepstral coefficients are orders of magnitude larger than the rest, and we found this to impede training. Thus before appending the MFCCs to our input, we throw out the first five cepstral values and normalize the rest to [-1,1].

3.5. Training Implementation

All audio processing was handled by the librosa Python library [10]. In this application, librosa was used to read *.wav* files sampled at 44.1kHz, perform STFTs of length 4096 with centered Hann window, hop length 1024 (25%), and write 16-bit PCM *.wav* files with sampling frequency 44.1kHz from reconstructed magnitude STFT frames.

The neural network framework was handled using TensorFlow [11]. All training used the Adam method for stochastic gradient descent with mini-batch size of 200 [12] for 300 epochs. ALl models were trained on an NVIDIA GeForce GTX Titan X GPU. A checkpoint file containing the trained weights of each autoencoder topology was saved once training was finished.

3.6. Task Performance/Evaluation

Table 1 shows the SC loss on the validation set after training. For reference, an autoencoder that estimates all zeros for any given input has a SC loss of 0.843.

As demonstrated, the appended inputs to the autoencoder improve over the model with no appendings. Our results show that while autoencoders are capable of constructing high level features

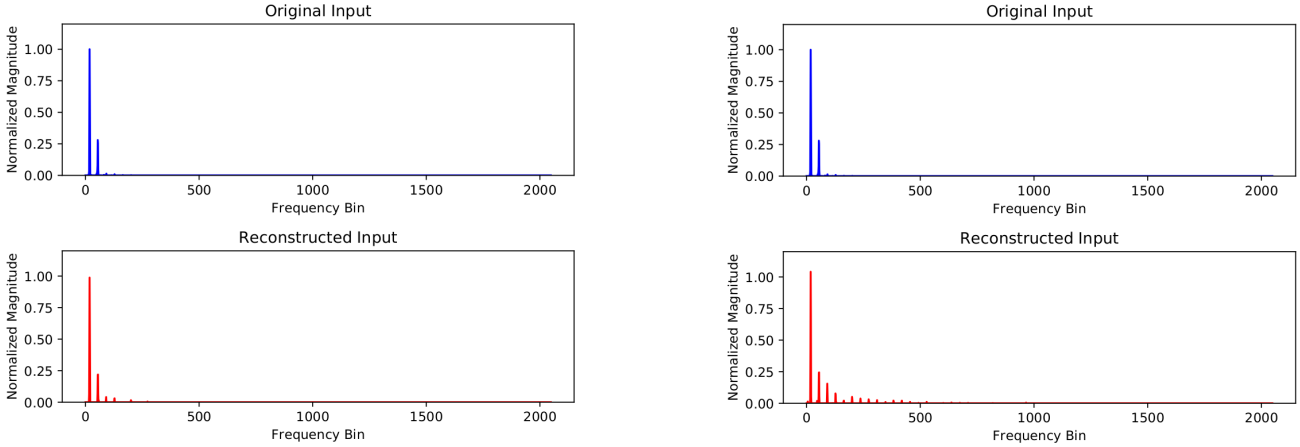


Figure 3: Sample input and reconstruction for the first-order appended model (left) and mfcc appended model (right)

Table 1: Autoencoder validation set SC loss and Training Time

Input Append	Validation SC	Training Time
No Append	0.257	25 minutes
1^{st} Order Diff	0.217	51 minutes
2^{nd} Order Diff	0.245	46 minutes
1^{st} and 2^{nd} Order Diff	0.242	69 minutes
MFCCs	0.236	52 minutes

from data unsupervised, providing the autoencoder with common-knowledge descriptive features of an input signal can improve its performance.

The model trained by augmenting the input with the signal's 1^{st} order difference (1^{st}-order-appended model) outperformed every other model. Compared to the 1^{st}-order-appended model, the MFCC trained model often inferred overtonal activity not present in the original signal (Figure 3). While it performs worse on the task than the 1^{st}-order-append model, the MFCC trained model presents a different sound palette that is valid for music synthesis. Options for training the model with different appending schemes are available in our implementation.

4. AUDIO SYNTHESIS

4.1. Spectrogram Generation

The training scheme outline above forces the autoencoder to construct a latent space contained in $\mathbb{R}^8$ that contains representations of synthesizer-based musical audio. Thus a musician can use the autoencoder to generate spectrograms by removing the encoder and directly activating the 8 neuron hidden layer. However, these spectrograms come with no phase information. Thus to obtain a time signal, phase information must be generated as well.

4.2. Phase Generation with RTPGHI

Real-time phase gradient heap integration (RTPGHI) [13] is used to generate the phase for the spectrogram. While the full theoretical treatment of this algorithm is outside the scope of this paper, we present the following synopsis.

The scaled discrete STFT phase gradient $\nabla\phi = (\phi_\omega, \phi_t)$ can be approximated by first finding the phase derivative in the time direction $\tilde{\phi}_{t,n}$

$$\tilde{\phi}_{t,n}(m) = \frac{aM}{2\gamma}(s_{log,n}(m+1) - s_{log,n}(m-1)) + 2\pi am/M \tag{10}$$

where $s_{log,n}(m) = log(s_n(m))$ and $\tilde{\phi}_{t,n}(0,n) = \tilde{\phi}_{t,n}(M/2,n) = 0$. Because a Hann window of length 4098 is used to generate the STFT frames, $\gamma = 0.25645 \times 4098^2$. Then, the phase derivative in the frequency direction is calculated using a first order difference approximation to estimate the phase $\tilde{\phi}_n(m)$ using the following algorithm

$$\tilde{\phi}_n(m) \leftarrow \tilde{\phi}_{n-1}(m) + \frac{1}{2}(\tilde{\phi}_{t,n-1}(m) + \tilde{\phi}_{t,n}(m)) \tag{11}$$

An inverse STFT (ISTFT) is then taken using the generated spectrogram and phase to produce a time signal.

An issue arises when using RTPGHI with this autoencoder architecture. A spectrogram generated from a constant activation of the hidden layer contains constant magnitudes for each frequency value. This leads to the phase gradient not updating properly due to the 0 derivative between frames. To avoid this, uniform random noise drawn from [0.999,1.001] is multiplied to each magnitude value in each frame. By multiplying this noise rather than adding it, we avoid adding spectral power to empty frequency bins and creating a noise floor in the signal.

5. PYTHON IMPLEMENTATION

5.1. CANNe

We realized a software implementation of our autoencoder synthesizer, "CANNe (Cooper's Autoencoding Neural Network)" in Python using TensorFlow, librosa, pygame, soundfile, and Qt 4.0. Tensorflow handles the neural network infrastructure, librosa and soundfile handle audio processing, pygame allows for audio playback in Python, and Qt handles the GUI.

Figure 4 shows a mock-up of the CANNe GUI. A musician controls the eight Latent Control values to generate a tone. The Frequency Shift control performs a circular shift on the generated

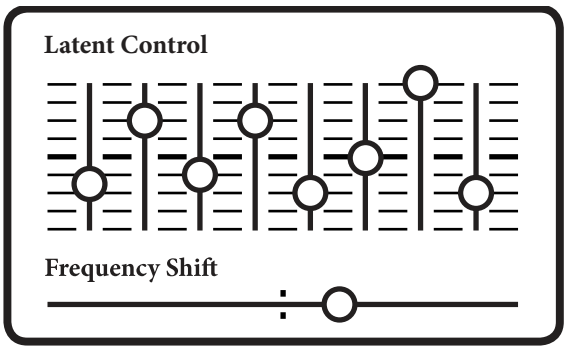


Figure 4: Mock-up GUI for CANNe.

magnitude spectrum, thus effectively acting as a pitch shift. It is possible, though, for very high frequency content to roll into the lowest frequency values, and vice-versa.

6. CONCLUSIONS

We present a novel method for musical audio synthesis based on activating the smallest hidden layer of an autoencoding neural network. By training the autoencoder to encode and decode magnitude short-time Fourier transform frames, the autoencoder is forced to learn high-level, descriptive features of audio. Real-time phase gradient heap integration is used to calculate a phase response for the generated magnitude response, thus making an inverse STFT possible and generating a time signal. We have implemented our architecture and algorithm in Python and host the open-source code at $github.com/JTColonel/canne\_synth$.

7. ACKNOWLEDGMENTS

We would like to thank Benjamin Sterling for helping to code early implementations of the CANNe GUI and Yonatan Katzelnik for helping design and structure our UI.

8. REFERENCES

[1] Aäron van den Oord, Sander Dieleman, Heiga Zen, Karen Simonyan, Oriol Vinyals, Alex Graves, Nal Kalchbrenner, Andrew W. Senior, and Koray Kavukcuoglu, "Wavenet: A generative model for raw audio," *CoRR*, vol. abs/1609.03499, 2016.

[2] "Generating black metal and math rock: Beyond bach, beethoven, and beatles," `http://dadabots.com/nips2017/generating-black-metal-and-math-rock.pdf`, Zack Zukowski and Cj Carr 2017.

[3] J. Engel, C. Resnick, A. Roberts, S. Dieleman, D. Eck, K. Simonyan, and M. Norouzi, "Neural Audio Synthesis of Musical Notes with WaveNet Autoencoders," *ArXiv e-prints*, Apr. 2017.

[4] Joseph Colonel, Christopher Curro, and Sam Keene, "Improving neural net auto encoders for music synthesis," in *Audio Engineering Society Convention 143*, Oct 2017.

[5] Pascal Vincent, Hugo Larochelle, Isabelle Lajoie, Yoshua Bengio, and Pierre-Antoine Manzagol, "Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion," *Journal of Machine Learning Research*, vol. 11, no. Dec, pp. 3371–3408, 2010.

[6] Nicolas Sturmel and Laurent Daudet, "Signal reconstruction from stft magnitude: A state of the art," .

[7] Vinod Nair and Geoffrey E Hinton, "Rectified linear units improve restricted boltzmann machines," in *Proceedings of the 27th international conference on machine learning (ICML-10)*, 2010, pp. 807–814.

[8] Sepp Hochreiter, Yoshua Bengio, Paolo Frasconi, and Jürgen Schmidhuber, "Gradient flow in recurrent nets: the difficulty of learning long-term dependencies," 2001.

[9] Anders Krogh and John A Hertz, "A simple weight decay can improve generalization," in *NIPS*, 1991, vol. 4, pp. 950–957.

[10] Brian McFee, Colin Raffel, Dawen Liang, Daniel PW Ellis, Matt McVicar, Eric Battenberg, and Oriol Nieto, "librosa: Audio and music signal analysis in python," in *Proceedings of the 14th python in science conference*, 2015, pp. 18–25.

[11] Martin Abadi, "Tensorflow: Learning functions at scale," *ICFP*, 2016.

[12] Diederik P. Kingma and Jimmy Ba, "Adam: A method for stochastic optimization," *CoRR*, vol. abs/1412.6980, 2014.

[13] Zdenek Pruša and Peter L Søndergaard, "Real-time spectrogram inversion using phase gradient heap integration," in *Proc. Int. Conf. Digital Audio Effects (DAFx-16)*, 2016, pp. 17–21.

AUDIO STYLE TRANSFER WITH RHYTHMIC CONSTRAINTS

Maciek Tomczak, Carl Southall and Jason Hockman

Digital Media Technology Lab (DMT Lab)
Birmingham City University
Birmingham, United Kingdom
{maciek, carl, jason}@bcu.ac.uk

ABSTRACT

In this transformation we present a rhythmically constrained audio style transfer technique for automatic mixing and mashing of two audio inputs. In this transformation the rhythmic and timbral features of both input signals are combined together through the use of an audio style transfer process that transforms the files so that they adhere to a larger metrical structure of the chosen input. This is accomplished by finding beat boundaries of both inputs and performing the transformation on beat-length audio segments. In order for the system to perform a mashup between two signals, we reformulate the previously used audio style transfer loss terms into three loss functions and enable them to be independent of the input. We measure and compare rhythmic similarities of the transformed and input audio signals using their rhythmic envelopes to investigate the influence of the tested transformation objectives.

1. INTRODUCTION

In the field of digital audio effects processing, creative transformations of musical audio refer to methods for automated manipulations of temporally-relevant sounds in time. These systems can be seen as part of a larger set of support systems to guide users when they lack inspiration, technical knowledge, musical capability as it relates to melody, harmony, rhythm, structure or style [1]. In recent years, the use of powerful machine learning algorithms, such as convolutional neural networks (CNN), have become an essential component in the development of such intelligent musical expert agents. A step in this direction has recently emerged as a research topic of audio style transfer.

1.1. Background

Audio style transfer (AST) methods use machine learning algorithms to modify the timbral characteristics of musical audio signals. AST was first attempted in [2, 3], which directly extended an algorithm proposed for images in [4]. In AST, a new output is synthesised by minimising the *content* loss with respect to the *content*-contributing audio input and the *style* loss with respect to one or more audio examples of a given *style*. The *content* loss is based on comparing the network activations of features derived from an audio spectrogram. The *style* loss matches the statistics of the Gram matrix (i.e., inner product between neural feature maps) activations in the higher levels of the network. In [5], the authors argue that *content* may refer to the underlying structure of the input music (e.g., note pitches, rhythm) and *style* can refer to timbres of instruments or genres.

Definitions and challenges of style transfer for music are presented in [6]. The appropriateness of the Gram matrix as a representation for *style* remains unclear for both music and images. This challenge is furthered by the ambiguous meaning of the term *style*, which is related to nearly all aspects of music. It has been suggested that the Gram matrix corresponds to a representation of musical timbre [5, 7]. To test the possibilities of creating rhythmically focused transformations varied according to different loss formulations we explore the use of the Gram matrix further and report on the suitability and shortcomings of this approach.

Approaches to AST can be divided into two categories: (1) time-frequency domain (i.e., spectrogram) based, where log-magnitudes of a short-time Fourier transform (STFT) are used as inputs to a CNN that performs the *style* transformation followed by a process of phase reconstruction; and (2) time-domain (i.e., raw audio) based, where the audio samples are directly optimised, removing the need for additional phase reconstruction.

The majority of AST research performs timbral transformation in the time-frequency domain, while preserving the rhythmic characteristics of the *content* recording. Grinstein et al. [5] introduced a spectral filtering method based on a sound texture model to improve the transformation of timbre from *style* directly onto a new audio initialised as *content* sound. The authors experimented with different pre-trained neural networks to aid their transformation. Similarly, Wyse [8] explored the effects of pre-trained weights from a network trained on an audio classification dataset for AST. The presented system appears to generate a more integrated transformation of *content* and *style* with the included pre-trained network. In [7] the authors provide an additional loss term that constrains the temporal envelope of the newly generated spectrogram to match that of the *style* recording. The motivation for the additional loss function was to better portray the temporal dynamics of the *style* recording and diminish the impact of the *content* recording. Audio style transfer was also used in the attempt to change the *style* of prosodic speech by [9]. The authors report success in transferring low-level textural features of the *content* but difficulty in transferring the high-level prosody such as emotion or accent of the *style* voice recording.

In addition to the above spectrogram based methods, AST systems have been proposed that can change rhythmic patterns of the input by applying the transformation directly on the raw audio. Mital [10] combines information from multiple discrete Fourier transform parts and presents them as different concatenated batches (layers) of a convolutional filter. Concatenated real, imaginary, and magnitude features are presented as producing the best results. Barry and Kim [11] implemented a parallel architecture that adds deep specialised networks with reduced frequency channels projected onto constant-Q transform basis, for key invariance capabilities, and Mel basis for representing longer rhythmic patterns. Their approach allows for longer temporal memory over the input features.

While the above methods are capable of timbral transforma-

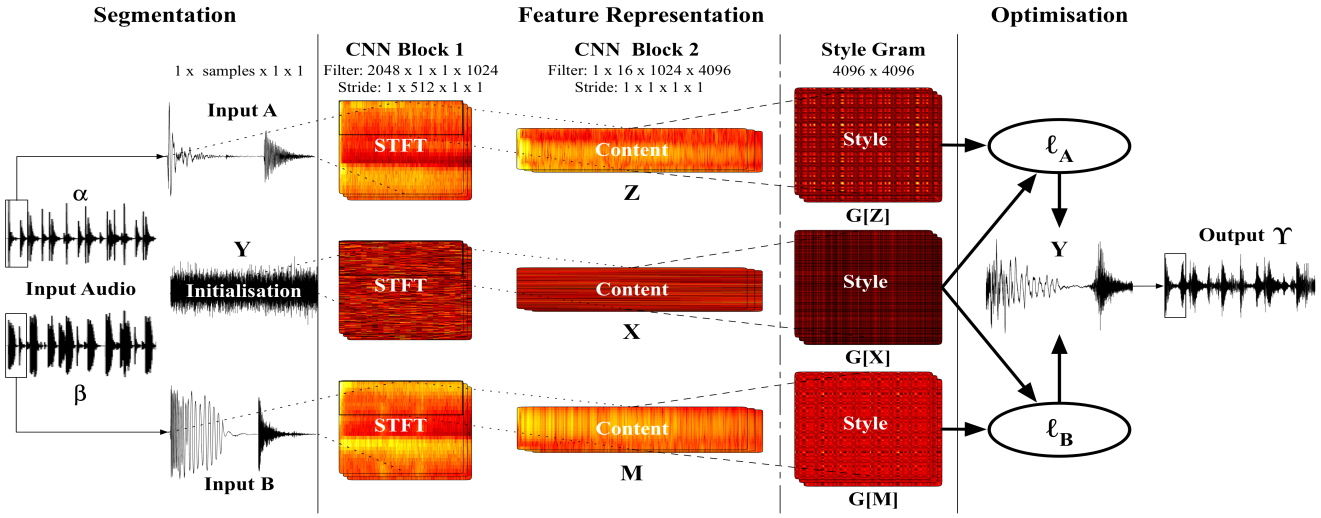


Figure 1: *Audio style transfer with rhythmic constraints in three stages: Segmentation, Feature Representation and Optimisation. A noise signal ϒ is iteratively transformed to represent the timbral and rhythmic characteristics of a user-defined mix between two input audio recordings A and B. Solid lines divide the three stages; dotted lines represent convolution, dashed lines represent style Gram computation and the vertical solid-dashed line represents a scaled exponential linear unit (SeLU) activation layer.*

tions, these modifications are not temporally restrictive and therefore do not constrain the elements in a metrically relevant manner. Alternatively, there have been several signal processing approaches to rhythmic transformations, including: percussive swing modification in polyphonic audio recordings [12]; rhythmic pattern manipulation of a drum loop to match that of another [13]; the rhythmic modification of an input polyphonic recording given the intra-measure structure of a model recording [14] and multi-song music mashup creation [15].

1.2. Motivation

In this paper we propose a system that extends the AST method to preserve the meter and the rhythmic structure of the chosen musical signal, while maintaining stylistic elements of both inputs. Our aim in the following is to transform two recordings such that their timbral and rhythmic patterns are merged together, with the presence of each being user-defined. To do this, we alter the original AST formulation to optimise the *style* representations of the input recordings simultaneously. To improve the creative application of this approach we constrain the transformation to act only on beat-length segments and test it on a small corpus of drum performances. This approach ensures that the transformation adheres to a larger rhythmic structure of the recordings with opportunities to generate new music that is both creative and realistic, as well as to uncover musical relationships of familiar audio samples that might otherwise have never been conceptualised.

The remainder of this paper is structured as follows: Section 2 presents our proposed method for AST with rhythmic constraints. Section 3 presents experiments undertaken and the results with discussion. We conclude with suggestions for future work in Section 4.

2. METHOD

Figure 1 presents an overview of our proposed system for AST. The system extends work by [11],[1] in which a noise signal Y is iteratively transformed to embody the timbral characteristics of a target associated with two audio recordings (α and β). In [11], the *content* refers to a network projection of input audio and *style* refers to a statistical representation of the feature map generated from previous layers of the network (as discussed in Section 2.3.1). We add to this kind of transformation through the integration of rhythmic constraints and with the addition of interchangeable loss terms with regards to both inputs.

The proposed model consists of three stages: (1) *segmentation*, where the two audio files (α and β) are divided into beat-length segments (A and B respectively); (2) *feature representation*, in which feature representations (Z, M and X) of A, B and Y are created using a CNN; and (3) *optimisation* in which Y is iteratively transformed to simultaneously match loss functions related to the feature representations of A and B. The resultant transformation Υ is a concatenation of the transformed beat-length segments Y.

2.1. Segmentation

Our motivation for the inclusion of segmentation in AST is to divide the inputs so that they adhere to a larger metrical structure during the transformation, while reducing the computation cost. In our experience, musically-interesting and rhythmically-stable transformations may be obtained when assessing beat-length audio segments. In order for input audio files to be processed by the proposed system, beat and downbeat positions must be first extracted. We compute segment boundaries using a state-of-the-art beat and downbeat tracking algorithm [16] included in the madmom Python

[1] https://github.com/anonymousiclr2018/Style-Transfer-for-Musical-Audio

library.[2] We then use the detected beat positions, starting from the first downbeat, as segment boundaries for A and B and generate the new noise segment Y using the same length.

2.2. Feature Representation

The aim of the feature representation stage is to project the input audio segments onto neural feature maps, which results in the creation of *content* and *style* matrices.

2.2.1. Content

To create the *content* matrices, the same two-stage process is performed in separate network branches for A, B and Y, where the weights of signal Y are initialised with random noise and matrices A and B contain input audio data as in [11]. First, feature maps are created by projecting the audio onto STFT bases. Then, the feature maps are projected further onto a larger number of channels as in [2, 5, 10, 11] to create the *content* representation.

The input audio (A, B and Y) is segmented into T frames using a Hanning window of n samples (n = 2048) with a $\frac{n}{4}$ hop-size. A frequency projection of each of the frames is then created with a single CNN layer that uses filters initialised with real and imaginary parts of the discrete Fourier transform resulting in a Tx$\frac{n}{2}$ spectrogram. We convert the created spectrogram to a log-magnitude representation. This transformation is represented in CNN Block 1 in Figure 1, where the filter size is nx1x1x$\frac{n}{2}$ with strides of 1x$\frac{n}{4}$x1x1.

CNN Block 2 (Figure 1) depicts neural feature computation from the STFT projections that becomes the *content* and can be understood as the low-level features of the input. The CNN architecture consists of a single convolutional layer with a filter size of 1xHxFxQ, where H is the number of time frames convolved with the filter, F is the number of frequency bins and Q represents the number of frequency channels that the input spectrogram will be projected onto. The filter size used in this implementation is 1x16x$\frac{n}{2}$x2n. We use a temporal receptive field (i.e., a contextual window modeled by each hidden state of the network) of 16 frames (~370ms) to capture acoustic information about instruments from a context longer than half beat length at 120 beats per minute (BPM). Each network is followed by a scaled exponential linear unit (SeLU) [17] activation layer, represented as vertical solid-dashed line in Figure 1, in place of standard rectified linear units (ReLU), as in [11]. This is done to increase the quality of the synthesised audio and reduce convergence time of the optimisation algorithm. For the rest of the paper, the *content* matrices for A, B and Y are termed Z, M and X respectively.

2.2.2. Style

Style can be understood as high-level information of the input neural features. To obtain a representation of the *style* of an input spectrogram, a Gram matrix G is used as in [4]. This feature space is designed to capture texture or intra-feature map statistics. For each *content* matrix (Z, M and X) G is calculated using the inner product:

$$G[X]_{ij} = \sum_k X_{ik} X_{jk}. \tag{1}$$

[2] https://github.com/CPJKU/madmom

2.3. Optimisation

2.3.1. Content and Style Loss Functions

In order to control the contributions of *content* and *style* from the two inputs, the total loss $\mathcal{L}$ is expressed as a sum of *content* ℓ_C and *style* ℓ_S loss functions for the input audio files A and B:

$$\mathcal{L} = \sigma\ell_C^A + \delta\ell_C^B + \theta\ell_S^A + \phi\ell_S^B, \tag{2}$$

where σ, δ, θ and ϕ are proportion parameters that add up to 1 and help configure loss preferences between the input recordings. The individual ℓ terms can be added and changed according to the transformation objective. The *content* loss ℓ_C is a squared error loss between the frame indices i and channels j of the transform *content* matrix X and the input audio *content* matrices (Z or M):

$$\ell_C^A = \frac{1}{2}\sum_{i,j}(X_{ij} - Z_{ij})^2, \tag{3}$$

$$\ell_C^B = \frac{1}{2}\sum_{i,j}(X_{ij} - M_{ij})^2. \tag{4}$$

The *style* loss ℓ_S is the sum of the squared difference between the transformed Gram matrix $G[X]$ and the input Gram matrices ($G[Z]$ or $G[M]$):

$$\ell_S^A = \frac{1}{Q^2}\sum_{i,j}(G[X]_{ij} - G[Z]_{ij})^2, \tag{5}$$

$$\ell_S^B = \frac{1}{Q^2}\sum_{i,j}(G[X]_{ij} - G[M]_{ij})^2. \tag{6}$$

The motivation for using the *style* loss as formulated above was to preserve the statistics about the convolutional representation over the entire input, while losing local information about where exactly different elements are.

2.3.2. Training

We use different combinations of *style* and *content* loss functions to shape the output of the transformation (Section 3.3). Following [11], we normalise the magnitudes of the gradients of loss terms to 1 to moderate the imbalances in weighting of either function. We use the limited-memory BFGS [18] gradient descent-based optimisation algorithm for its appropriateness in non-linear problems related to neural style transfer [4, 19]. Once initialised, the feature map representations of *content* and *style* from inputs A and B do not change throughout the training stage. In each gradient step the *content* and *style* activations are back-propagated all the way to the network output Y. Hence, only weights originating from Y are being manipulated during the optimisation process, while all feature representations remain unchanged for inputs A and B. The optimisation of the concerned weights is stopped after 500 iterations. An NVIDIA Tesla M40 computing processor was used for this project with an average of 3 seconds per algorithm iteration.

2.4. Implementation

Our system is implemented using the Tensorflow Python library.[3] The processing branches of A, B and Y are part of the same CNN in one Tensorflow computation graph. This means that the neural representations of the input time-domain audio Y can be optimised simultaneously in one stage.

[3] https://www.tensorflow.org/

Table 1: *Mean cosine similarities from 15 transformed target audio pairs. The cosine similarities are calculated between rhythmic envelopes extracted from full α, β and Υ audio files for loss functions $\mathcal{L}_1$, $\mathcal{L}_2$ and $\mathcal{L}_3$. Mean cosine similarity calculated from all α and β rhythmic envelopes in the experiment is 0.58.*

	$\mathcal{L}_1$	$\mathcal{L}_2$	$\mathcal{L}_3$
Υ to input α	0.32	0.60	0.43
Υ to input β	**0.37**	0.60	**0.52**

3. EXPERIMENTS

3.1. Experimental Setup

We test the rhythmic modification characteristic of our AST approach by assessing the rhythmic similarity of the transformed output to the input audio for three loss term combinations. To achieve this comparison, we generate rhythmic envelopes from the newly created audio files Υ and compare them to those of α and β.

3.2. Dataset

For this experiment we created 30 drum loops (mono .wav sampled at 22.05 kHz with 16-bit resolution) of 4 measures in length, which differ in rhythmic patterns consisting of various kick and snare drums. All transformation examples are created from 15 pairs of input drum loops to reduce computation cost. All drum loops have a fixed-tempo set to 120 BPM in $\frac{4}{4}$ meter. Our motivation for using a fixed-tempo of 120 BPM was to test how our transformation performs on already beat-synchronised inputs essential in the processes of mixing and mashing audio recordings together. The chosen tempo is typical for many genres in popular music as well, as it is the default tempo in various digital audio workstations used in music production. The drum loops used in our tests were generated with twelve different pattern styles defined by the Logic X Drummer virtual instrument.[4]

3.3. Rhythmic Similarity

To test the rhythmic constraints imposed by different transformation objectives within the AST technique, we compare the rhythmic similarity [15] of pairs of transformations. The rhythmic envelopes are calculated from the spectral difference function [20] of new audio Υ with inputs α or β. We calculate the rhythmic envelopes as the sum over frequency bins from the first-order difference between each adjacent magnitude spectra. The STFT parameters from Section 2.2 are used. The resulting rhythmic envelopes were normalised to range from 0 to 1. To determine the rhythmic similarity D between every pair of rhythmic envelopes R we calculate the cosine similarity as:

$$D_{\omega,\Upsilon} = \frac{R_\omega \cdot R_\Upsilon}{\|R_\omega\|\|R_\Upsilon\|}, \tag{7}$$

where ω can represent envelope of either α or β. Thus, the rhythmic similarity will be close to unity for very similar patterns and nearer to zero for dissimilar patterns. The mean of all D values is calculated across 15 transformation audio pairs per loss term formulation.

[4] `https://support.apple.com/kb/PH13070`

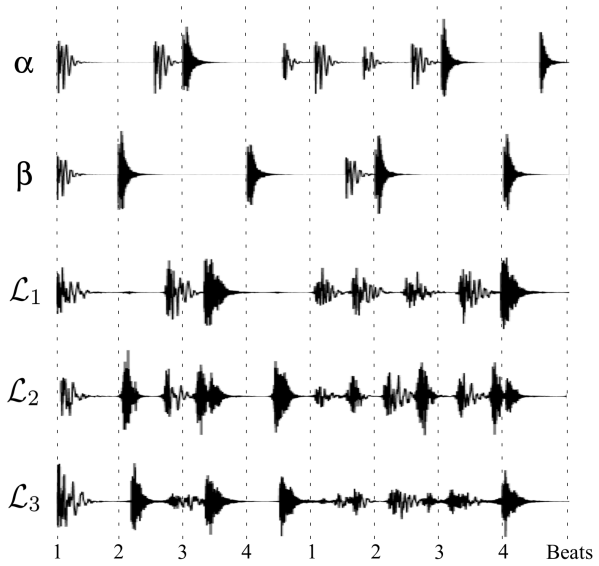


Figure 2: *Example transformations generated from three loss terms $\mathcal{L}_1$, $\mathcal{L}_2$ and $\mathcal{L}_3$ from input audio signals α and β.*

We test our approach with three objectives associated with combinations of loss terms with all proportion parameters σ, δ, θ and ϕ set to be equal:

Objective $\mathcal{L}_1$: $\ell_S^A + \ell_C^B$. In this objective we test the ability of our system to move acoustic events to create a rhythmically new performance that is more similar to β through the low-level information from the *content* loss.

Objective $\mathcal{L}_2$: $\ell_S^A + \ell_S^B$. In this objective we test a transformation that solely uses the *style* feature representations to mix high-level characteristics of both recordings. This transformation is akin to a mashup of both audio inputs.

Objective $\mathcal{L}_3$: $\ell_S^A + \ell_S^B + \ell_C^B$. This objective reinforces the mashup transformation with more low-level information from β.

3.4. Results and Discussion

The overall similarity results are summarised in Table 1. The cosine similarities of the transformations Υ compared with input β are higher for objectives $\mathcal{L}_1$ (0.37) and $\mathcal{L}_3$ (0.52), where the B *content* loss (ℓ_C^B) was used. When the B *content* loss was not used ($\mathcal{L}_2$) the transformation similarities to α and β are both 0.60. We believe this is due to both *style* losses having the same weighting, resulting in an equal mix of both inputs that creates a kind of rhythmic and timbral mashup. This is in agreement with the mean similarities of the α and β together (0.58). In addition, when larger proportions of the *content* loss are used the transformations are expected to be more similar to the corresponding *content* loss of the chosen input.

Figure 2 shows transformed waveforms of inputs α and β using the three different loss term combinations. In $\mathcal{L}_1$ the rhythmic pattern of α is recreated at different metrical positions that match the beat pattern of β (e.g., on beat 4 of the second measure). On beat 2 of the first measure the event from β does not appear in the

resulting transformation, while in objectives $\mathcal{L}_2$ and $\mathcal{L}_3$ the event is included. Similarly, the transformation in beat 4 of the first measure from $\mathcal{L}_1$ removes an event that is present in objectives $\mathcal{L}_2$ and $\mathcal{L}_3$ as an instrument from β. In this case a kick from α was transformed into a snare from β. The difference between the $\mathcal{L}_2$ and $\mathcal{L}_3$ transformations show the effect of the added *content* loss from B in that drum events that correspond to silences become attenuated in the resulting mashup of both recordings (e.g., beat 3 of the first measure).

Experimental transformations along with other examples are presented using the web-based audio player by [21] and can be found on the supporting website for this project.[5] The resultant audio examples acquired from loss terms $\mathcal{L}_1$, $\mathcal{L}_2$ and $\mathcal{L}_3$ are accompanied by transformation outputs from publicly available algorithms [11, 10, 2]. Our rhythmically-constrained transformation differs in that it is capable of generating new rhythmic patterns from both inputs while preserving the beat pattern of the chosen recording. Challenges faced by all AST transformations are the loss of phase information and the addition of noise, potentially due to the high-level representation of the *style* loss (i.e., Gram matrix).

As in other AST methods to date, we have used the Gram matrix as a representation for *style*, yet it remains questionable whether this feature representation is suitable for transformations based on high-level musical information. Briot and Pachet suggest that this technique presents challenges for audio due to anisotropy of the *content* representation [22]. Anisotropy signifies dependence on directions and here it refers to the nature of the audio spectrogram. In this time-frequency representation the dimensions do not correlate together in the same way a pixel would in an image. A pixel almost always corresponds to one object whereas in music multiple sources overlap causing inherent issues when using the Gram matrix to transform local changes in timbres.

3.5. Attempted Rhythmic Loss Terms

In addition to segmenting the audio and experimenting with different combinations of the existing loss terms, we also tested two new loss terms which aimed to aid the rhythmic aspects of AST. Both terms were formulated to minimise the cosine distance between rhythmic envelopes of the chosen input and the transformation. In the first loss term, each rhythmic envelope was calculated as the sum over frequency bins of the two spectrograms (i.e., feature representations from CNN Block 1 in Figure 1). In the second term, we created a new network branch for the chosen input where the resulting STFT projection was filtered with the first-order difference between each adjacent log-magnitude spectra to then create a detection function focused more on percussive events. The second loss term was formulated to minimise the cosine distance between rhythmic envelopes of the filtered input spectrogram and the transformation. Through informal listening we found that neither term improved the transformation in conjunction with $\mathcal{L}_1$ and $\mathcal{L}_2$ loss terms. The first loss term was causing generated drum events to lose their transient information, whereas the second term removed events created in the silent sections of the rhythmic envelope, while increasing amplitudes of drum event transients.

[5] https://maciek-tomczak.github.io/maciek.github.io/Audio-Style-Transfer-with-Rhythmic-Constraints

3.6. Additional Audio Inputs

In our rhythmic extension of AST we are able to create transformations using an arbitrary number of input recordings. In a music composition scenario, once the desired individual recordings are found, it is possible to create their combined transformation. One such purpose would be to mix multiple individual drum recordings together such as hi-hats, kicks and snares with the aim of creating their new rhythmic and timbral interpretation. However, with additional audio input signals the transformation becomes more difficult to control.

4. CONCLUSIONS AND FUTURE WORK

In this work we present a rhythmically constrained audio style transfer technique that explores different loss formulations. Our method utilises a time-domain approach to AST that acts on beat length segments of the input music signals. By constraining the transformation to shorter analysis segments that follow the metrical structure of the chosen input recording, we show that the resulting transformations sound rhythmically coherent, while reducing the computation cost. In the transformation the two input files are mixed together and allow the user to adjust the parameters of each loss term to experiment with the desired objective. The resulting transformation can be formulated as to replicate the exact spectral information of the input or to create a mashup.

Our attempt to measure the transformation similarities compared to their corresponding inputs shows differences in their rhythmic envelopes. From informal listening it can be heard that the beat detection does not need to be accurate for the transformation to produce rhythmically valid examples, however both inputs should have at least some rhythmic agreement when the *content* loss is used. In the case of the loss formulation that uses only the information about *content* and *style* of the inputs, the transformations are more different from both input files.

In future work, we intend to explore transformation objectives related to additional instrumentation and time scales, as well as, improving the phase reconstruction inherent in this kind of sound transformation.

5. REFERENCES

[1] Peter Knees, Kristina Andersen, Sergi Jordà, Michael Hlatky, Günter Geiger, Wulf Gaebele, and Roman Kaurson, "Giantsteps-progress towards developing intelligent and collaborative interfaces for music production and performance," in *Proceedings of the International Conference on Multimedia & Expo Workshops*. IEEE, pp. 1–4, 2015.

[2] Dmitry Ulyanov and Vadim Lebedev, "Audio texture synthesis and style transfer," 2016, Available at: https://tinyurl.com/ybgnsf9h.

[3] Davis Foote, Daylen Yang, and Mostafa Rohaninejad, "Do androids dream of electric beats?," 2016, Available at: https://tinyurl.com/yb5ww2tw.

[4] Leon A. Gatys, Alexander S. Ecker, and Matthias Bethge, "A neural algorithm of artistic style," *Computing Research Repository*, vol. abs/1508.06576, 2015.

[5] Eric Grinstein, Ngoc Q. K. Duong, Alexey Ozerov, and Patrick Pérez, "Audio style transfer," *Computing Research Repository*, vol. abs/1710.11385, 2017.

[6] Shuqi Dai, Zheng Zhang, and Gus G. Xia, "Music style transfer issues: A position paper," *arXiv preprint:1803.06841*, 2018.

[7] Prateek Verma and Julius O. Smith, "Neural style transfer for audio spectrograms," *Computing Research Repository*, vol. abs/1801.01589, 2018.

[8] Lonce Wyse, "Audio spectrogram representations for processing with convolutional neural networks," *Computing Research Repository*, vol. abs/1706.09559, 2017.

[9] Anthony Perez, Chris Proctor, and Archa Jain, "Style transfer for prosodic speech," Technical Report, Stanford University, 2017.

[10] Parag K. Mital, "Time domain neural audio style transfer," *Computing Research Repository*, vol. abs/1711.11160, 2017.

[11] Shaun Barry and Youngmoo Kim, "Style transfer for musical audio using multiple time-frequency representations," Unpublished article available at: https://tinyurl.com/y7nu7r9s, 2018.

[12] Fabien Gouyon, Lars Fabig, and Jordi Bonada, "Rhythmic expressiveness transformations of audio recordings: swing modifications," in *Proceedings of the Digital Audio Effects Workshop*, pp. 8–11, 2003.

[13] Emmanuel Ravelli, Juan P. Bello, and Mark Sandler, "Automatic rhythm modification of drum loops," *Signal Processing Letters, IEEE*, vol. 14, no. 4, pp. 228–231, 2007.

[14] Jason A. Hockman, Juan P. Bello, Matthew E. P. Davies, and Mark D. Plumbley, "Automated rhythmic transformation of musical audio," in *Proceedings of the International Conference on Digital Audio Effects*, pp. 177–180, 2008.

[15] Matthew E. P. Davies, Philippe Hamel, Kazutomo Yoshii, and Masataka Goto, "Automashupper: automatic creation of multi-song music mashups," *Audio, Speech, and Language Processing, IEEE/ACM Transactions on*, vol. 22, no. 12, pp. 1726–1737, 2014.

[16] Sebastian Böck, Florian Krebs, and Gerhard Widmer, "Joint beat and downbeat tracking with recurrent neural networks," in *Proceedings of the International Society for Music Information Retrieval*, pp. 255–261, 2016.

[17] Günter Klambauer, Thomas Unterthiner, Andreas Mayr, and Sepp Hochreiter, "Self-normalizing neural networks," in *Proceedings of the Conference on Neural Information Processing Systems*, pp. 972–981, 2017.

[18] Ciyou Zhu, Richard H. Byrd, Peihuang Lu, and Jorge Nocedal, "Algorithm 778: L-BFGS-B: Fortran subroutines for large-scale bound-constrained optimization," *ACM Transactions on Mathematical Software*, vol. 23, no. 4, pp. 550–560, 1997.

[19] Kun He, Yan Wang, and John Hopcroft, "A powerful generative model using random weights for the deep image representation," in *Proceedings of the Conference on Neural Information Processing Systems*, pp. 631–639, 2016.

[20] Simon Dixon, "Onset detection revisited," in *Proceedings of the International Conference on Digital Audio Effects*, pp. 133–137, 2006.

[21] Nils Werner, Stefan Balke, Fabian-Robert Stöter, Meinard Müller, and Bernd Edler, "Trackswitch.js: A versatile web-based audio player for presenting scientifc results," in *Proceedings of the Web Audio Conference*, 2017.

[22] Jean-Pierre Briot and François Pachet, "Music generation by deep learning-challenges and directions," *Computing Research Repository*, vol. abs/1712.04371, 2017.

PARAMETRIC SYNTHESIS OF GLISSANDO NOTE TRANSITIONS – A USER STUDY IN A REAL-TIME APPLICATION

Henrik von Coler
Audio Communication Group
TU Berlin
Germany
voncoler@tu-berlin.de

Moritz Götz
Audio Communication Group
TU Berlin
Germany

Steffen Lepa
Audio Communication Group
TU Berlin
Germany
steffen.lepa@tu-berlin.de

ABSTRACT

This paper investigates the applicability of different mathematical models for the parametric synthesis of fundamental frequency trajectories in glissando note transitions. Hyperbolic tangent, cubic splines and Bézier curves were implemented in a real-time synthesis system. Within a user study, test subjects were presented two-note sequences with glissando transitions, which had to be re-synthesized using the three different trajectory models, employing a pure sine wave synthesizer. Resulting modeling errors and user feedback on the models were evaluated, indicating a significant disadvantage of the hyperbolic tangent in the modeling accuracy. Its reduced complexity and low number of parameters were however not rated to increase the usability.

1. INTRODUCTION

Note transitions are an essential part of articulation and thus of expressive musical performances. On instruments with continuous excitation and a continuous frequency scale, such as the violin or the singing voice, glissando note transitions are thus of particular interest. A so called *Glissando* or *Portamento* mode has thus been implemented in many analog and digital synthesizers since their early days. Most devices allow the tuning of the transition time, some offer the selection of different trajectory functions. The comparison of different parametric models presented in this work is considered a step towards an extension of this established concept.

The topic of modeling fundamental frequency trajectories has been addressed in the disciplines of speech and music analysis / synthesis in the past. The main features of these glissando transitions can be expressed in terms of the fundamental frequency f_0 and short-term energy trajectories (RMS). 't Hart [1] compared straight lines and parabolas for modeling the fundamental frequency of speech syllables using modulated pulse trains. Simple straight-line segments were indistinguishable from parabolic ones in a listening test. For modeling the prosody of speech utterances, Hirst et al. [2] applied quadratic spline functions.

Battey [3] used third order Bézier splines to model trajectories of f_0, amplitude and spectral centroid for musicological analysis but also referred to the application in *expressive computer rendering*. Barbot et al. [4] compared the modeling accuracy of cubic B-splines and natural cubic splines for f_0 trajectories of speech syllables. Using 4 support points each, the B-splines achieved a lower RMS error. B-Spline and spline models were compared by Lolive et al. [5] for the use of modeling fundamental frequency in speech synthesis systems. Within a sinusoidal modeling approach, Hahn et al. [6] used B-splines to model the temporal trajectories of partial parameters. Ardaillon et al. [7] evaluated a parametric f_0 model based on B-splines within a concatenative singing voice synthesis system through listening tests.

Although the qualities of different trajectory models in terms of modeling error and perception have been investigated thoroughly in the past, little is known about the usability of these models in real-time applications. The nature of parameters is individual for each model and an increasing number of parameters might decrease the intuitiveness. Modeling precision and usability are hypothesized to be opposed. The simpler the model, the larger the modeling error but the easier the control. This work thus focuses on the usability of trajectory models with parametric control in a real-time application. Hyperbolic tangent, cubic splines and Bézier curves will be compared in a user experiment. The hyperbolic tangent offers just one parameter, cubic splines have been implemented with two and Bézier curves with three control parameters [8].

The remainder of this paper is organized as follows: In Section 2 the implemented models will be introduced. Section 3 presents the user study, followed by the results in Section 4 and their discussion in Section 5. A conclusion is presented in Section 6.

2. GLISSANDO MODELING

Glissando note transitions are the segment between two adjacent notes of different pitch, in which the fundamental frequency trajectory and the energy trajectory are continuous. The glissando segment is defined as the region between the stationary segments of the pitches f_1 and f_2, as shown in the idealized model in Fig. 1. The idealized fundamental frequency trajectory (b) of these regions is closely related to sigmoid curves whereas the idealized energy trajectory (a) remains constant.

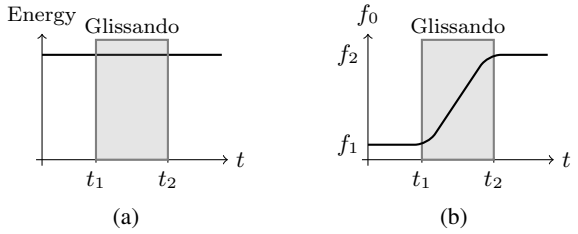


Figure 1: Idealized transition model for glissando articulation

For the calculation of the actual trajectories used in the experiment, an analysis of the fundamental frequency trajectory was performed with a hopsize of $L_{hop} = 256$ samples, respectively $2.7\,\mathrm{ms}$, using the YIN algorithm [9].

The resulting trajectories were subsequently modeled using hyperbolic tangent, cubic splines and Bézier curves. The fundamentals of these models and the resulting parameters will be outlined in the remainder of this section.

2.1. Hyperbolic Tangent

The hyperbolic tangent is defined as:

$$\tanh(x) = \frac{\sinh(x)}{\cosh(x)} \tag{1}$$

In order to make this basic function applicable for different intervals $\Delta f = f_2 - f_1$ and durations $\Delta t = t_2 - t_1$, the following parameters are added:

$$T(t) = c + d \tanh\left(\frac{t-a}{b}\right), \quad t, a, b, c, d \in \mathbb{R} \tag{2}$$

For a transition between two values f_1 and f_2 the parameters c and d must be:

$$\begin{aligned} d &= \frac{|f_1 - f_2|}{2}, \\ c &= \min(f_1, f_2) + d. \end{aligned} \tag{3}$$

Parameter a is depending on the time values. For a transition between the first value t_0 and the last value t_1 parameter a must be:

$$a = \frac{|t_1 - t_2|}{2}. \tag{4}$$

The resulting single parameter b, presented to the user in the study, controls the slope of the function by time-scaling. In Figure 2, hyperbolic tangent curves are plotted with different values for b.

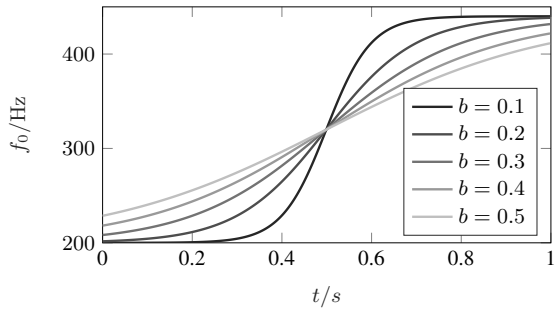


Figure 2: Hyperbolic tangent with different values for b

2.2. Cubic Splines

Splines are special functions for the piece wise interpolation by polynomials. A cubic spline S with n points $P_i = (x_i, y_i)$ is defined as:

$$\begin{aligned} S(x) := a_i + b_i(x - x_i) + c_i(x - x_i)^2 + d_i(x - x_i)^3, \\ x \in [x_i, x_{i+1}], a_i, b_i, c_i.d_i \in \mathbb{R}, i = 0, 1, ..., n - 1. \end{aligned} \tag{5}$$

Arbitrary points from the extracted f_0-trajectories can be used to get a polynomial representation of the curve. An equidistant 4-point model is used in the experiment. The x-values of the control points are thus fixed. In Figure 3, a natural cubic spline curve is plotted with four points. The outer points P_1 and P_2 are fully determined by the boundary conditions, so are the x-values for P_3 and P_4. Two remaining parameters – the y-values of P_3 and P_4 – are presented to the user in the experiment for controlling the trajectory.

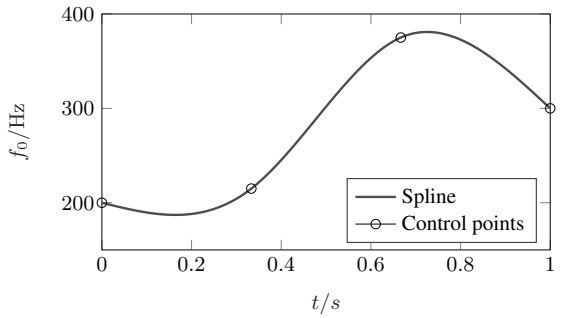


Figure 3: Example of a natural cubic spline with four control points

2.3. Bézier Curves

Bézier curves are controlled by a number of control points, of which only the start and end point lie on the curve itself. A Bézier curve $K(x)$ is defined by sum of Bernstein polynomials $B_i^n(x)$ and the control points P_i:

$$K(x) = \sum_{i=0}^{n} P_i B_i^n(x), n \in \mathbb{N} \tag{6}$$

For the application in the experiment the x-values of P_i have been set to be equally spaced:

$$P_{i,x} = \frac{i}{n}, \quad i = 0, 1, ..., n \tag{7}$$

Figure 4 shows an example of a Bézier curve with 5 control points. Since the outer points are fully determined by the boundary conditions (f_1 and f_2) and the x-values are predefined, the user is presented the three y-values of the inner control points as parameters in the experiment.

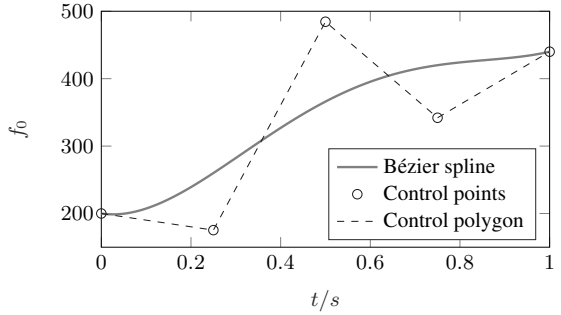


Figure 4: Bézier curve with five control points and control polygon

2.4. Modeling Accuracy

For evaluating the numerical modeling qualities, all 96 two-note sequences from the *TU-Note Violin Sample Library* [10, 11] with glissando transitions were used. This selection contains upward and downward glissandi at different positions and dynamics. The mean absolute error was applied to evaluate the deviation between original trajectories x_i and model estimates $\tilde{x}_i$ of length $N, \tilde{x}_i \in \mathbb{R}, n \in \mathbb{N}$:

$$\bar{\delta}_x := \frac{1}{N} \sum_{i=0}^{N-1} |x_i - \tilde{x_i}|. \tag{8}$$

Using $\bar{\delta}_x$, the best possible fit was calculated for all models, also evaluating different orders for splines and Bézier curves. The best parameter settings were found by calculating the model parameters related to a curve intersecting the original trajectory at the x-values of the control points. For the hyperbolic tangent this resulted in a modeling error of $\bar{\delta}_x = 0.083$. For the splines, an increase of the number of interpolation points lead to a monotonic decrease in modeling error (Table 1). For Bézier curves, a minimum error $\bar{\delta}_x = 0.0311$ was reached with 7 interpolation points (Table 2). A further increase lead to an increase of the error, which is likely to be caused by the fixed x-values of the control points and the method for finding the best parameter set. The numbers of control points chosen for the experiment are marked bold in Table 1 and 2.

Table 1: Minimum of mean absolute error for different spline orders

control points	mean of normalized $\bar{\delta}_x$	mean of $\bar{\delta}_x$ [Hz]
4	**0.0387**	**7.24**
5	0.0272	5.04
6	0.0205	3.66
7	0.0163	2.93
8	0.0145	2.59
9	0.0119	2.16
10	0.0109	1.98
11	0.0096	1.72
12	0.0094	1.67

Table 2: Minimum of mean absolute error for different Bézier orders

control points	mean of normalized $\bar{\delta}_x$	mean of $\bar{\delta}_x$ [Hz]
4	0.0539	10.21
5	**0.0394**	**7.26**
6	0.0358	6.61
7	0.0311	5.39
8	0.0325	5.74
9	0.0377	6.63
10	0.0379	6.67
11	0.0594	10.22
12	0.0805	13.73

3. USER STUDY

A user study was conducted to compare the usability of the three proposed trajectory models. Using a within-subject design, participants had to apply the three different models to reproduce seven sequences of two notes which are connected with a glissando. Errors between original and reproduction were evaluated alongside additional user feedback to obtain information on the real-time usability of the three models. The Bézier model was presented to the user with one tuning parameter, splines were used with two and Bézier curves with three parameters, respectively four and five support points.

3.1. Test System

The synthesis engine with the real-time trajectory modeling was programmed in C++, using the JACK API [12]. The runtime system was a Raspberry Pi 3 Model B Rev 1.2, running Raspbian GNU/Linux 9.1. A *Behringer U-Control UCA222* audio interface was used with a processing block size of 128 samples at a sampling rate of 48 kHz. A *Logilink* USB to MIDI Adapter was used for the MIDI input with a *Swissonic ControlKey 49*. Faders were routed to the parameters of the trajectory models to allow control by the participants. A pure sinusoidal synthesizer with fixed amplitude was implemented within the test system. A control surface for the user study which managed the handling of the trials, the input of the user data and configured the synthesis engine via MIDI was programmed in Pure Data [13].

Table 3: Stimuli employed in the seven tasks of the user study, stemming from the *TU-Note Violin Library* [10]

Item	note 1	note 2	length	direction
`TwoNote_DPA_18`	A3	D4	380 ms	up
`TwoNote_DPA_19`	D4	A3	320 ms	down
`TwoNote_DPA_65`	E5	B4	400 ms	down
`TwoNote_DPA_66`	B4	E5	485 ms	up
`TwoNote_DPA_113`	D4	G4	300 ms	up
`TwoNote_DPA_137`	A4	D5	700 ms	up
`TwoNote_DPA_186`	E6	B5	550 ms	down

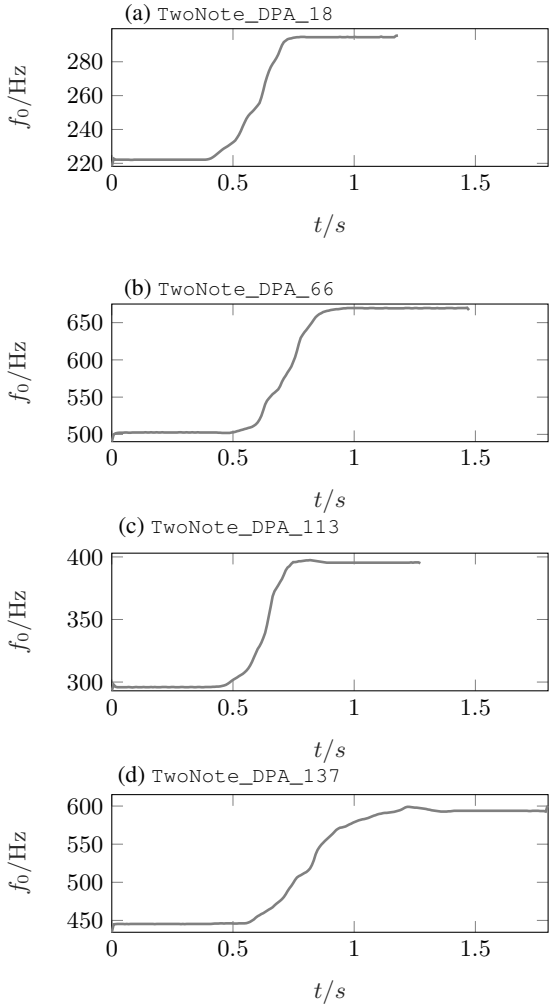


Figure 5: Fundamental frequency trajectories for upward glissando stimuli

3.2. Stimuli for Reproduction Tasks

Stimuli for the reproduction tasks were generated using the *TU-Note Violin Sample Library* [11, 10], which features two-note sequences with annotated glissando transitions. Four upward and three downward two-note sequences, listed in Table 3, were selected with different note frequencies, in order to cover the range of the instrument. The fundamental frequency trajectories of these seven sequences were extracted and are visualized in Figure 5 and Figure 6. These trajectories were then used to drive a simple sinusoidal synthesizer with a fixed amplitude, in order to exclude influences from features other than the fundamental frequency.

3.3. Participants

15 participants were recruited through the mailing list for students of the audio communication group at TU Berlin. 14 of them were male and one was female. Participants' mean age was 27.4 years with a standard deviation of 5.6 years. The majority of the participants were musically skilled: 60 % played an instrument on a regular basis for more than 6 years and also 66.67 % had ear training for more than one year.

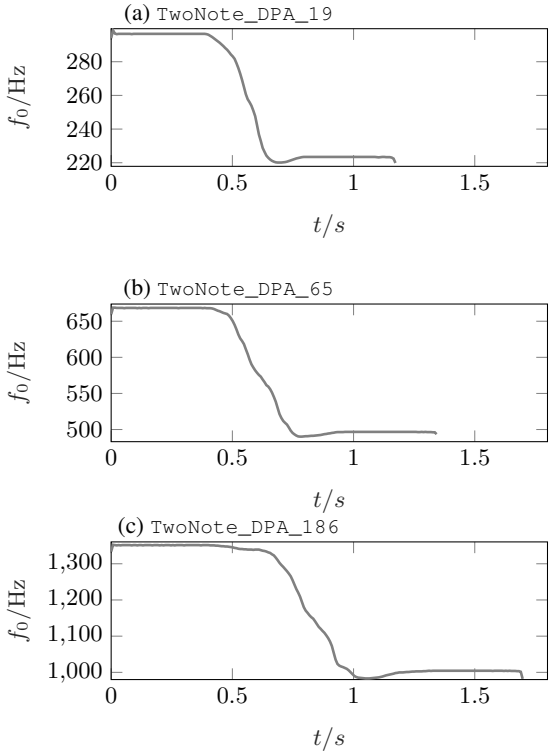


Figure 6: Fundamental frequency trajectories for downward glissando stimuli

3.4. Procedure

After an introduction to the test system and a free play period with all three trajectory models, each participant went through 21 experimental trials: Each of the 7 task stimuli had to be re-synthesized by the users using each of the three models in a fully randomized order. In every trial, the task stimulus could be played back as often as desired. Additional information on the current trial was shown on the graphical user interface, which included a number referring to the currently active trajectory model (1,2,3) and the starting and the ending note of the sequence.

Participants were then instructed to reproduce the sequence using the MIDI keyboard and the real-time synthesis engine. The length of the glissando was fixed for each stimulus, but the parameters of each model could be adjusted. Once the participants were satisfied with their settings, three questions about the just employed model and its parameters had to be answered using vertical continuous sliders (ranging from 0-100) on the graphical user interface. In the study the questions were in German, hence a translated version is shown in Table 4.

4. RESULTS

Since the resulting data is not normal distributed and the amount of 15 participants may be considered small, the non-parametric

Table 4: Rating scales asked after each single task

Question	0	100
Parameter changes were ...	not audible	clearly audible
The model allowed an easy adjustment of the stimulus	completely disagree	completely agree
The number of parameters in this model is ...	too low	too high

Friedman test has been chosen to evaluate each dependent variable, separately. The independent factor is the trajectory model with three levels. The dependent variables are the mean absolute modeling error in the reproduction of the task stimulus $\bar{\delta}_x$ as well as the scores from the three rating scales. All dependent variables have been averaged across the seven presented tasks.

4.1. Modeling Error

Box plots in Figure 7 show a higher modeling error for the hyperbolic tangent than for splines and Bézier curves. The results show a statistically significant difference in modeling error depending on the trajectory model, $\chi^2 = 7.600, p = 0.000$. A post hoc analysis was conducted using Wilcoxon signed-rank tests. Bonferroni correction resulted in a significance level of p < 0.017. Median (IQR) modeling errors for the hyperbolic tangent, Spline and Bézier model were .3377 (.3270 to .3621), .1230 (.0957 to .2042) and .1266 (.0782 to .1722), respectively. There was no significant difference between the Bézier and the Spline model (Z = -.795 , p = .427). The Hyperbolic tangent model, however, showed a significantly higher modeling error than the the Spline (Z = -3.408, p = .001) and the Bézier model (Z = -3.408, p = 0.001).

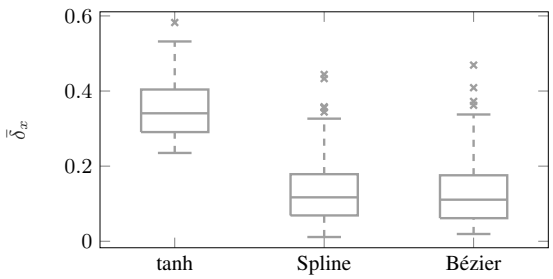


Figure 7: Boxplots for modeling error $\bar{\delta}_x$, averaged across tasks

4.2. Audibility of Parameter Changes

Results of the question whether parameter changes are audible are shown in Figure 8 as box plots, indicating a slightly better audibility for the hyperbolic tangent. Results of the Friedman test show a statistically significant difference in the audibility of parameter changes depending on the trajectory model, $\chi^2 = 6.218, p = .045$. Again, Wilcoxon signed-rank tests were used for a post hoc analysis with a Bonferroni correction, resulting in a significance level of p < 0.017. Median (IQR) of the rated audibility for the hyperbolic tangent, Spline and Bézier model were 86.1446 (77.9786 to 98.1354), 66.5328 (56.5978 to 87.6363) and 72.8342(56.3110 to 85.2744), respectively. The post hoc analysis, however, showed no significant difference between any of the models, neither between Bézier and the Spline model (Z = -.031 , p = .975) nor between Spline and hyperbolic tangent (Z = -2.166, p = .030) or Bézier and hyperbolic tangent (Z = -2.271, p = .023).

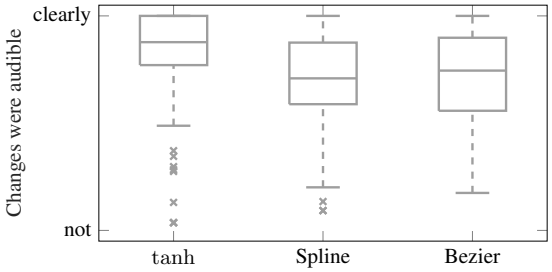


Figure 8: Box plots for question *audibility of parameter changes*

4.3. Ease of Adjustment

Figure 9 shows box plots for the responses to the question referring to the *ease of adjustment*. The Friedman Test showed no significant influence of the trajectory model on the perceived ease of adjustment, $\chi^2 = 2.533, p = .282$. Median (IQR) of the ease of adjustment for hyperbolic tangent, spline and Bézier model were 72.2318 (48.9673 to 93.2301), 77.3379 (57.8026 to 88.1813) and 71.7154 (45.1807 to 79.8336). No significant difference between any of the models, neither between Bézier and the Spline model (Z = -.909 , p = .363) nor between Spline and hyperbolic tangent (Z = -.057, p = .955) nor between Bézier and hyperbolic tangent (Z = -.795, p = .427).

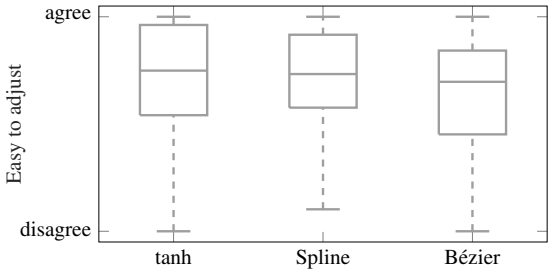


Figure 9: Box plots for the question *ease of adjustment*

4.4. Number of Parameters

Box plots for the question regarding the *number of parameters* are shown in Figure 10. There was a statistically significant difference in the rating whether the number of parameters was too low (0) or too high (100), depending on the number of provided parameters, $\chi^2 = 26.271, p = .000$. Median (IQR) of the response to the question for for hyperbolic tangent, spline and Bézier model were 29.1165 (19.8795 to 46.4429), 49.1968 (47.9920 to 50.4016) and 65.3758 (54.9340 to 67.6133). Results show a significant difference between one and two parameters (Z = -3.045 , p = .002) one and three parameters (Z = -3.408, p = .001) as well as two and three parameters (Z = -3.408, p = .001).

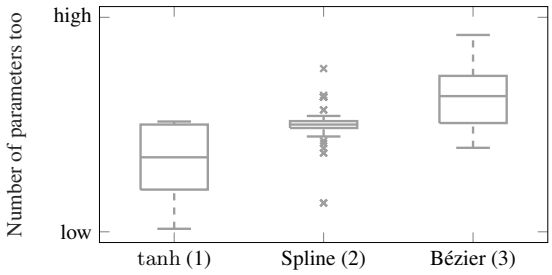


Figure 10: Box plots for question *Number of parameters*

5. DISCUSSION

The results show that the hyperbolic tangent leads to a larger modeling error than cubic splines and Bézier curves in the user experiment. Hence, the hyperbolic tangent is less suitable for synthesizing the glissando transitions presented in the sequences, regarding the mean absolute error. This relation could also be observed for the best model fits in the automated evaluation in Section 2.4, although the user experiment resulted in higher error rates.

Further, the results show a significant preference of two parameters, since this number is rated as neither too high, nor too low. This relation is presumably independent of the trajectory models and probably of a basic psychological nature, since two was the mean number of parameters presented to the users. Since the hyperbolic tangent was used with one, splines with two and Bézier with three parameters, these findings can not be interpreted, independently.

It would be conceivable that the hyperbolic tangent was easier to adjust by the participants. The *ease of adjustment*, however, was not influenced by the model or by the number of parameters. This justifies the use of more complex models and rejects the initial hypothesis that they could be more difficult to use.

6. CONCLUSION

The presented study could deliver first insights on the usability of hyperbolic tangent, cubic splines and Bézier curves for glissando modeling in a real-time scenario. Using the hyperbolic tangent resulted in the largest modeling errors, whereas an increased number of parameters for the other models did not reduce the usability. Thus, the use of such models can be considered justified.

Several aspects of this study could be subject to further, more detailed experiments. It would be of interest to investigate the factor *number of parameters* independently of the trajectory model. For reasons of feasibility, these aspects have been mixed in this study.

Since the errors for the seven trajectory types in the tasks have been averaged, the individual features of the glissandi were not evaluated. Studies using the glissando type (up, down) as independent variable might reveal more differences between the trajectory models.

Future research should incorporate other instruments, additional musicians and different musical content. The glissando transitions of the violin in this user study were of rather smooth nature. They contained no overshoots, unlike for example the singing voice, which might be easier to synthesize with Bézier curves. Different instruments may require other models.

Finally, the mean absolute error may not the ideal measure to evaluate the performance. It was nevertheless chosen as a first step towards a procedure. In fact, the perceived modeling accuracy is a more important factor in musical re-synthesis tasks. Thus, a combination of the presented study with a listening test can deliver further results.

7. REFERENCES

[1] Johan 't Hart. "F0 stylization in speech: Straight lines versus parabolas". In: *The Journal of the Acoustical Society of America* 90.6 (1991), pp. 3368–3370.

[2] Daniel Hirst and Robert Espesser. "Automatic Modelling Of Fundamental Frequency Using A Quadratic Spline Function". In: Travaux de l'Institut de Phonétique d'Aix (1993), pp. 75–85.

[3] Bret Battey. "Bézier spline modeling of pitch-continuous melodic expression and ornamentation". In: *Computer Music Journal* 28.4 (2004), pp. 25–39.

[4] Nelly Barbot, Olivier Boëffard, and Damien Lolive. "F0 stylisation with a free-knot b-spline model and simulated-annealing optimization". In: *Proceedings of the 9th European Conference on Speech Communication and Technology (Eurospeech)*. Lisbon, Portugal, 2005.

[5] Damien Lolive, Nelly Barbot, and Olivier Boëffard. "Comparing B-Spline and Spline Models for F0 Modelling". In: *International Conference on Text, Speech and Dialogue*. Springer. Brno, Czech Republic, 2006, pp. 423–430.

[6] Henrik Hahn and Axel Röbel. "Extended Source-Filter Model for Harmonic Instruments for Expressive Control of Sound Synthesis and Transformation". In: *Proceedings of the 16th International Conference on Digital Audio Effects (DAFx-13)*. Maynooth, Ireland, 2013.

[7] Luc Ardaillon, Gilles Degottex, and Axel Roebel. "A multi-layer F0 model for singing voice synthesis using a B-spline representation with intuitive controls". In: *Interspeech 2015*. Dresden, Germany, 2015.

[8] Moritz Götz. "Analysis and Synthesis of Control Parameters in Note Transitions". MA thesis. TU-Berlin, 2018.

[9] Alain de Cheveigné and Hideki Kawahara. "YIN, a Fundamental Frequency Estimator for Speech and Music". In: *The Journal of the Acoustical Society of America* 111.4 (2002), pp. 1917–1930.

[10] Henrik von Coler. "TU-Note Violin Sample Library – A Database of Violin Sounds with Segmentation Ground Truth". In: *Proceedings of the 21st International Conference on Digital Audio Effects (DAFx-18)*. Aveiro, Portugal, 2018.

[11] Henrik von Coler, Jonas Margraf, and Paul Schuladen. *TU-Note Violin Sample Library*. TU-Berlin, 2018. DOI: 10.14279/depositonce-6747.

[12] *JACK API Website*. 2018. URL: http://jackaudio.org/.

[13] Miller S Puckette. "Pure Data." In: *Proceedings of the International Computer Music Conference (ICMC)*. San Francisco, 1996, pp. 224–227.

TOWARDS MULTI-INSTRUMENT DRUM TRANSCRIPTION

Richard Vogl
Faculty of Informatics
TU Wien
Vienna, Austria
richard.vogl@tuwien.ac.at

Gerhard Widmer
Dept. of Computational Perception
Johannes Kepler University
Linz, Austria
gerhard.widmer@jku.at

Peter Knees
Faculty of Informatics
TU Wien
Vienna, Austria
peter.knees@tuwien.ac.at

ABSTRACT

Automatic drum transcription, a subtask of the more general automatic music transcription, deals with extracting drum instrument note onsets from an audio source. Recently, progress in transcription performance has been made using non-negative matrix factorization as well as deep learning methods. However, these works primarily focus on transcribing three drum instruments only: snare drum, bass drum, and hi-hat. Yet, for many applications, the ability to transcribe more drum instruments which make up standard drum kits used in western popular music would be desirable. In this work, convolutional and convolutional recurrent neural networks are trained to transcribe a wider range of drum instruments. First, the shortcomings of publicly available datasets in this context are discussed. To overcome these limitations, a larger synthetic dataset is introduced. Then, methods to train models using the new dataset focusing on generalization to real world data are investigated. Finally, the trained models are evaluated on publicly available datasets and results are discussed. The contributions of this work comprise: *(i.)* a large-scale synthetic dataset for drum transcription, *(ii.)* first steps towards an automatic drum transcription system that supports a larger range of instruments by evaluating and discussing training setups and the impact of datasets in this context, and *(iii.)* a publicly available set of trained models for drum transcription. Additional materials are available at http://ifs.tuwien.ac.at/~vogl/dafx2018.

1. INTRODUCTION

Automatic drum transcription (ADT) focuses on extracting a symbolic notation for the onsets of drum instruments from an audio source. As a subtask of automatic music transcription, ADT has a wide variety of applications, both in an academic as well as in a commercial context. While state-of-the-art approaches achieve reasonable performance on publicly available datasets, there are still several open problems for this task. In prior work [1] we identify additional information—such as bar boundaries, local tempo, or dynamics—required for a complete transcript and propose a system trained to detect beats alongside drums. While this adds some of the missing information, further work in this direction is still required.

Another major shortcoming of current approaches is the limitation to only three drum instruments. The focus on snare drum (SD), bass drum (BD), and hi-hat (HH) is motivated by the facts that these are the instruments *(i.)* most commonly used and thus with the highest number of onsets in the publicly available datasets; and *(ii.)* which often define the main rhythmical theme. Nevertheless, for many applications it is desirable to be able to transcribe a wider variety of the drum instruments which are part of a standard drum kit in western popular music, e.g., for extracting full transcripts for further processing in music production or educational scenarios. One of the main issues with building and evaluating such a system is the relative underrepresentation of these classes in available datasets (see section 2).

In this work we focus on increasing the number of instruments to be transcribed. More precisely, instead of three instrument classes, we aim at transcribing drums at a finer level of granularity as well as additional types of drums, leading to classification schemas consisting of eight and 18 different instruments (see table 1). In order to make training for a large number of instruments feasible, we opt for a single model to simultaneously transcribe all instruments of interest, based on convolutional and convolutional recurrent neural networks. Especially in the case of deep learning, a considerable amount of processing power is needed to train the models. Although other approaches train separate models for each instrument in the three-instrument-scenario [2, 3], for 18 instruments it is more feasible to train a single model in a multi-task fashion (cf. [4]). To account for the need of large volumes of data in order to train the chosen network architectures, a large synthetic dataset is introduced, consisting of 4197 tracks and an overall duration of about 259h.

The remainder of this paper is organized as follows. In section 2 we discuss related work, followed by a description of our proposed method in section 3. Section 4 provides a review of existing datasets used for evaluation, as well as a description of the new, large synthetic dataset. Sections 5 and 6 describe the conducted experiments and discuss the results, respectively. Finally, we draw conclusions in section 7.

2. RELATED WORK

There has been a considerable amount of work published on ADT in recent years, e.g., [5, 6, 7, 8, 9]. In the past, different combinations of signal processing and information retrieval techniques haven been applied to ADT. For example: onset detection in combination with *(i.)* bandpass filtering [10, 11], and *(ii.)* instrument classification [5, 6, 7]; as well as probabilistic models [8, 12]. Another group of methods focus on extracting an onset-pseudo-probability function (activation function) for each instrument under observation. These methods utilize source separation techniques like Independent Subspace Analysis (ISA) [13], Prior Subspace Analysis (PSA) [14], and Non-Negative Independent Component Analysis (NNICA) [15]. More recently, these approaches have been further developed using Non-Negative Matrix Factorization (NMF) variants as well as deep learning [1, 3, 16, 17].

The work of Wu et al. [18] provides a comprehensive overview of the publications for this task, and additionally performs in-depth evaluation of current state-of-the-art methods. Due to the large

Table 1: Classes used in the different drum instrument classification systems. Labels map to General MIDI drum instruments: e.g. bass drum: 35, 36; side stick: 37; etc. The mapping is available on the accompanying website.

number of classes 3	8	18	instrument name
BD	BD	BD	bass drum
SD	SD	SD	snare drum
		SS	side stick
		CLP	hand clap
		HT	high tom
	TT	MT	mid tom
		LT	low tom
		CHH	closed hi-hat
HH	HH	PHH	pedal hi-hat
		OHH	open hi-hat
		TB	tambourine
	RD	RD	ride cymbal
	BE	RB	ride bell
		CB	cowbell
	CY	CRC	crash cymbal
		SPC	splash cymbal
		CHC	Chinese cymbal
	CL	CL	clave/sticks

number of works and given the space limitations, in the remainder of this section, we will focus on work that is directly relevant with respect to the current state of the art and methods focusing on more than three drum instrument classes.

As mentioned, the state of the art for this task is currently defined by end-to-end activation function based methods. In this context, end-to-end implies using only one processing step to extract the activation function for each instrument under observation from a digital representation of the audio signal (usually spectrogram representations). Activation functions can be interpreted as probability estimates for a certain instrument onset at each point in time. To obtain the positions of the most probable instrument onsets, simple peak picking [19, 20, 1, 3, 2, 16, 15] or a language-model-style decision process like dynamic Bayesian networks [21] can be used. These methods can be further divided into NMF based and deep neural network (DNN) based approaches.

Wu et al. [16] introduce partially fixed NMF (PFNMF) and further modifications to extract the drum instrument onset times from an audio signal. Dittmar et al. [17] use another modification of NMF, namely semi adaptive NMF (SANMF) to transcribe drum solo tracks in real time, while requiring samples of the individual drum instruments for training. More recently, recurrent neural networks (RNNs) have successfully been used to extract the activation functions for drum instruments [19, 20, 2]. It has also been shown that convolutional (CNNs) [1, 3] and convolutional recurrent neural networks (CRNNs) [1] have the potential to even surpass the performance of RNNs.

The majority of works on ADT, especially the more recent ones, focus solely on transcribing three drum instrument (SD, BD, HH) [9, 19, 20, 1, 2, 3, 16, 8, 17, 7, 8]. In some works multiple drum instruments are grouped into categories for transcription [5] and efforts have been made to classify special drum playing techniques within instrument groups [22]. However, only little work exists which approach the problem of transcribing more than

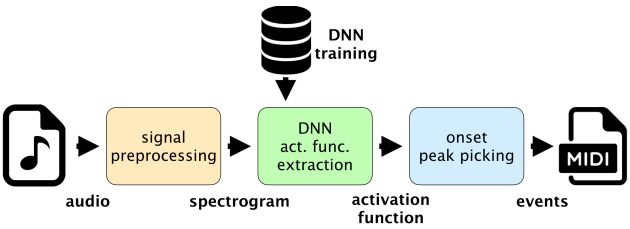


Figure 1: Overview of implemented ADT system using DNNs.

three individual drum instruments [15], furthermore, such a system has—to our knowledge—never been evaluated on currently available public drum transcription datasets.

In [6], a set of MIDI drum loops rendered with different drum samples are used to create synthetic data in the context of ADT. Using synthetic data was a necessity in the early years of music information retrieval (MIR), but due to the continuous efforts of creating datasets, this has declined in recent years. However, machine learning methods like deep learning, often requirer large amounts of data, and manual annotation in large volumes is unfeasible for many MIR tasks. In other fields like speech recognition or image processing, creating annotations is easier, and large amounts of data are commonly available. Using data augmentation can, to a certain degree, be used to overcome lack of data, as has been demonstrated in the context of ADT [20]. In [23] an approach to resynthesizes solo tracks using automatically annotated f0 trajectories, to create perfect annotations, is introduced. This approach could be applicable for ADT, once a satisfactory model for the full range of drum instruments is available. At the moment such annotations would be limited to the three drum instrument classes used in state-of-the-art methods.

3. METHOD

In this work, we use an approach similar to the ones introduced in [2] and [19], for drum transcription. As mentioned in the introduction, a single model trained in a multi-task fashion will be used. Creating individual models for each instrument is an option [2, 3], however, in the context of this work it has two downsides: First, training time will scale linearly with the amount of models, which is problematic when increasing the number of instruments under observation. Second, training multi-task models in the context of ADT can improve the performance [1]. Other state-of-the-art methods based on NMF [16, 17] are less suitable for a multi-task approach, since the performance of NMF methods is prone to degrade for basis matrices with higher rank.

Thus, the method proposed in [1] seems most promising for the goal of this work. We will only use CNNs and CRNNs, since simple RNNs do not have any advantage in this context. The implemented ADT system consists of three stages: a signal preprocessing stage, a DNN activation function extraction stage, and a peak picking post processing stage, identifying the note onset. The system overview is visualized in figure 1, and the single stages will be discussed in detail in the following subsections.

3.1. Preprocessing

During signal preprocessing, a logarithmic magnitude spectrogram is calculated using a window size of 2048 samples (@44.1kHz input audio frame rate) and choosing 441 samples as hop size for a

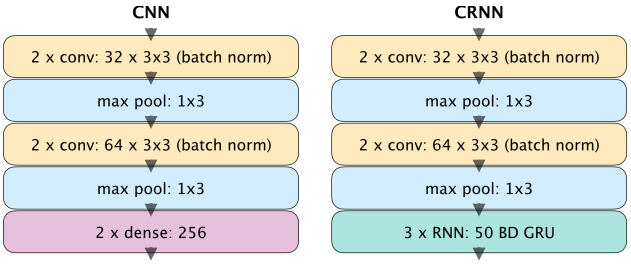


Figure 2: Architecture comparison between the CNN and CRNN used for activation function extraction.

100Hz target frame rate of the spectrogram. The frequency bins are transformed to a logarithmic scale using triangular filters in a range from 20 to 20,000 Hz, using 12 frequency bins per octave. Finally, the positive first-order-differential over time of this spectrogram is calculated and stacked on top of the original spectrogram. The resulting feature vectors have a length of 168 values (2x84 frequency bins).

3.2. Activation Function Extraction

The activation function extraction stage is realized using one of two different DNNs architectures. Figure 2 visualizes and compares the two implemented architectures. The convolutional parts are equivalent for both architectures, however, the dense output layers are different: while for the CNN two normal dense layers are used (ReLUs), in case of the CRNN two bidirectional RNN layers consisting of gated recurrent units (GRUs) [24] are used. As already noted in [1], GRUs exhibit similar capabilities as LSTMs [25], while being more easy to train.

The combination of convolutional layers which focus on local spectral features, and recurrent layers which model mid- and long-term relationships, has been found to be one of the best performing models for ADT [1].

3.3. Peak Picking

To identify the drum instrument onsets, a standard peak picking method introduced for onset detection in [26] is used. A peak at point n in the activation function $f_a(n)$ must be the maximum value within a window of size $m+1$ (i.e.: $f_a(n) = max(f_a(n-m), \cdots, f_a(n))$), and exceeding the mean value plus a threshold δ within a window of size $a+1$ (i.e.: $f_a(n) \geq mean(f_a(n-a), \cdots, f_a(n)) + \delta$). Additionally, a peak must have at least a distance of $w+1$ to the last detected peak n_{lp} (i.e.: $n - n_{lp} > w$,). The parameters for peak picking are the same as used in [1]: $m = a = w = 2$. The best threshold for peak picking is determined on the validation set. As observed in [3, 20, 1], appropriately trained DNNs produce spiky activation functions, therefore, low thresholds $(0.1 - 0.2)$ give best results.

3.4. Training and Evaluation

Training of the models is performed using *Adam* optimization [27] with mini-batches of size 100 and 8 for the CNNs and CRNNs respectively. The training instances for the CNN have a spectral context of 25 samples. In case of the CRNN, the training sequences consist of 400 instances with a spectral context of 13 samples. The DNNs are trained using a fixed learning rate ($l_r = 0.001$) with

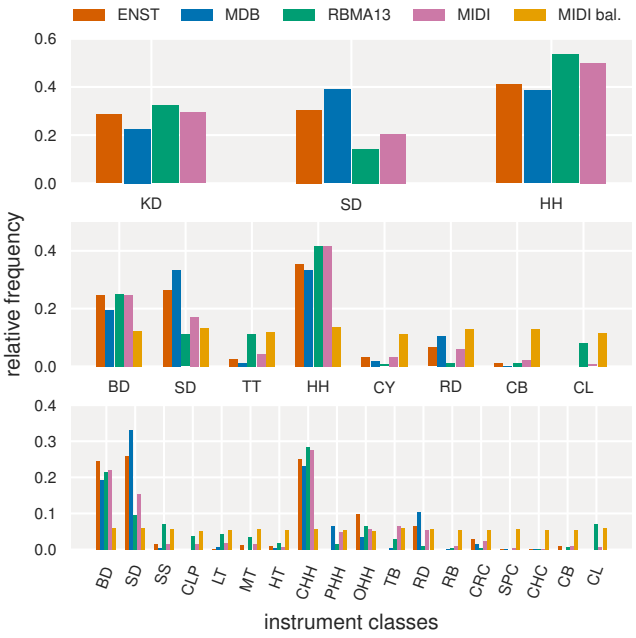


Figure 3: Label distributions of the different datasets used in this work.

additional refinement if no improvement on the validation set is achieved for 10 epochs. During refinement the learning rate is reduced ($l_r = l_r \cdot 0.2$) and training continues using the parameters of the best performing model so far.

A three-fold cross-validation strategy is employed, using two splits during training, while 15% of the training data is separated and used for validation after each epoch (0.5% in case of the large datasets, to reduce validation time). Testing is done on the third, during training unseen, split. Whenever available, drum solo versions of the tracks are used as additional training material, but not for testing/evaluation. The solo versions are always put into the same splits as their mixed counterparts, to counter overfitting. This setup is consistently used through all experiments, whenever datasets are mixed or cross-validated, corresponding splits are used.

For audio preprocessing, peak picking, and calculation of evaluation metrics, the madmom[1] python framework was used. DNN training was performed using Theano[2] and Lasagne[3]. For a more details on C(R)NN training and a comparison of their working principles in the context of ADT, we kindly refer the reader to our previous work [1] due to space limitations and a different focus of this work.

4. DATASETS

There are a number of publicly available datasets for ADT with varying size, degree of detail, and number of classes regarding the drum instrument annotations. As noted in the introduction, current state-of-the-art approaches limit the instruments under observation to the three most common ones (SD, BD, HH). This is done by ignoring other instruments like tom-toms and cymbals, as well as

[1] https://github.com/CPJKU/madmom
[2] https://github.com/Theano/Theano
[3] https://github.com/Lasagne/Lasagne

Table 2: F-measure (*mean*/*sum*) results of implemented ADT methods on public datasets for different class systems. The first line indicates state-of-the-art F-measure results in previous work using CNN and CRNN ADT systems in a three-class scenario.

CL	model	*ENST*	*MDB*	*RBMA13*
3	*SotA* [1]	*— / 0.78*	*— / —*	*— / 0.67*
3	CNN	0.75 / 0.77	0.65 / 0.72	0.53 / 0.63
	CRNN	0.74 / 0.76	0.64 / 0.70	0.55 / 0.64
8	CNN	0.59 / 0.63	0.68 / 0.65	0.55 / 0.44
	CRNN	0.65 / 0.70	0.68 / 0.63	0.55 / 0.50
18	CNN	0.69 / 0.49	0.76 / 0.47	0.62 / 0.31
	CRNN	0.75 / 0.67	0.77 / 0.55	0.64 / 0.39

grouping different play styles like closed, opened, and pedal hi-hat strokes. In order to investigate ways of generating a model which is capable to transcribe more than these three instruments, two classification systems, i.e., a medium and a large one, for drum instruments of a standard drum kit are defined. Table 1 shows the two sets of classes, which contain eight and 18 labels respectively, alongside with the classic three-class set used in state-of-the-art works and the mapping used between these classes.

In the following we discuss publicly available ADT datasets and their limitations, leading to the description of the large volume synthetic dataset introduced for training of our models.

4.1. ENST Drums (*ENST*)

The ENST Drums[4] dataset published by Gillet and Richard [28] in 2005, is commonly used in ADT evaluations. The freely available part of the dataset consists of single track audio recordings and mixes, performed by three drummers on different drum kits. It contains recordings of single strokes for each instrument, short sequences of drum patterns, as well as drum tracks with additional accompaniment (*minus-one* tracks). The annotations contain labels for 20 different instrument classes.

For evaluation, the *wet mixes* (contain standard post-processing like compression and equalizing) of the *minus-one tracks* were used. They make up 64 tracks of 61s average duration and a total duration of 1h. The rest of the dataset (single strokes, patterns) was used as additional training data.

4.2. MDB-Drums (*MDB*)

The MDB-Drums dataset[5] was published in [29] and provides drum annotations for 23 tracks of the Medley DB dataset[6] [30]. The tracks are available as drum solo tracks with additional accompaniment. Again, only the full mixes are used for evaluation, while the drum solo tracks are used as additional training data. There are two levels of drum instrument annotations, the second providing multiple drum instruments and additional drum playing technique details in 21 classes. Tracks have an average duration of 54 seconds and the total duration is 20m 42s.

[4] http://perso.telecom-paristech.fr/~grichard/ENST-drums/

[5] https://github.com/CarlSouthall/MDBDrums

[6] http://medleydb.weebly.com/

Table 3: F-measure results (*mean*/*sum*) of the implemented networks on synthetic datasets.

CL	model	*MIDI*	*MIDI 1%*	*MIDI bal.*
3	CNN	0.74 / **0.84**	0.70 / 0.79	— / —
	CRNN	0.74 / **0.84**	0.68 / 0.77	— / —
8	CNN	0.64 / 0.63	0.63 / 0.69	0.54 / 0.58
	CRNN	0.74 / **0.82**	0.69 / 0.73	0.58 / 0.70
18	CNN	0.66 / 0.39	0.65 / 0.39	0.59 / 0.18
	CRNN	0.73 / **0.70**	0.69 / 0.62	0.63 / 0.52

4.3. RBMA13 (*RBMA13*)

The RBMA13 datasets[7] was published alongside [1]. It consists of 30 tracks of the freely available 2013 Red Bull Music Academy Various Assets sampler.[8] The tracks' genres and drum sounds of this set are more diverse compared to the previous sets, making it a particularly difficult set. It provides annotations for 23 drum instruments as well as beat and downbeats. Tracks in this set have an average duration of 3m 50s and a total of 1h 43m.

4.4. Limitations of current datasets

A major problem of publicly available ADT datasets in the context of deep learning is the volume of data. To be able to train DNNs efficiently, usually large amounts of diverse data are used (e.g. in speech and image processing). One way to counter the lack of data is to use data augmentation (as done in [20] for ADT). However, data augmentation is only helpful to a certain degree, depending on the applicable augmentation methods and the diversity of the original data.

Given the nature of drum rhythms found in western popular music, another issue of ADT datasets is the uneven distribution of onsets between instrument classes. In case of the available datasets, this imbalance can be observed in figure 3. While it is advantageous for the model to adapt to this bias, in terms of overall performance, this often results in the trained models to never predict onsets for sparse classes. This is due to the number of potential false negatives being negligible, compared to the amount of false positives produced in the early stages of training. To counter a related effect on slightly imbalanced classes (BD, SD, HH in the three-class scenario), a weighting of the loss functions for the different classes can be helpful [20]. Nevertheless, a loss function weighting cannot compensate for the problem in the case of very sparse classes.

Since manual annotation for ADT is a very resource intensive task, a feasible approach to tackle these problems is to create a synthetic dataset using the combination of symbolic tracks, e.g. MIDI tracks, drum synthesizers and/or sampler software.

4.5. Synthetic dataset (*MIDI*)

For generating the synthetic dataset, a similar approach as in [6] was employed. Since the focus of this work is the transcription of multiple drum instruments from polyphonic music, full MIDI tracks of western popular music were used instead of MIDI drum loops. First, every MIDI track from a freely available online collection[9] was split into a drum and accompaniment track. Using

[7] http://ifs.tuwien.ac.at/~vogl/datasets/

[8] https://rbma.bandcamp.com/album/

[9] http://www.midiworld.com

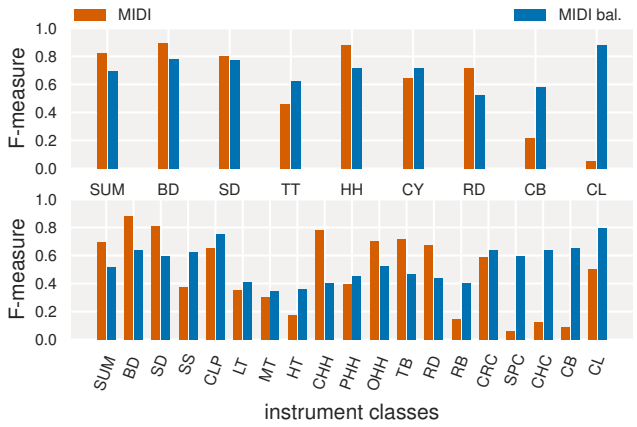


Figure 4: Instrument class details for evaluation results on *MIDI* and *MIDI bal.* for 8 and 18 instrument classes using the CRNN. First value (SUM) represents the overall sum F-measure results.

timidity++[10], the drum tracks were rendered utilizing 57 different drum SoundFonts[11]. The used SoundFonts were collected from different online sources, and great care was taken to manually check and correct the instrument mappings and overall suitability. They cover a wide range of drum sounds from electronic drum machines (e.g. TR808), acoustic kits, and commonly used combinations. The SoundFonts were divided into three groups for the three evaluation splits, to counter overfitting to drum kits. The accompaniment tracks were rendered using a full General MIDI SoundFont. Using the MIDI tracks, drum annotations as well as beat and downbeat annotations were generated. After removing broken MIDI files, very short (< 30s) as well as very long (> 15m) tracks, the set contains 4197 tracks with an average duration of 3m 41s and a total duration of about 259h. As with the other datasets, we only use the mixes for evaluation, while the drum solo tracks are used as additional train-only data.

Figure 3 shows that the general trend of the drum instrument class distribution is similar to the smaller datasets. This is not surprising since the music is of the same broad origin (western popular music). Since one of the goals of creating this dataset was to achieve a more balanced distribution, some additional processing is necessary. Due to the fact that we can easily manipulate the source MIDI drum files, we can change a certain amount of instruments for several tracks to artificially balance the classes. We did this for the 18 classes as well as for the 8 classes and generated two more synthetic datasets consisting of the same tracks, but with drum instruments changes so that the classes are balanced within their respective drum instrument class system. This was done in a way to switch instruments which have a similar expected usage frequency within a track, while keeping musicality in mind. Ideal candidates for this are CHH and RD: exchanging them makes sense from a musical standpoint, as well in terms of usage frequency. On the other hand, BD and CRC are close in expected usage frequency but switching them can be questionable from a musical standpoint, depending on the music genre. A full list of performed switches for the balanced versions can be found on the accompanying webpage.

[10] `http://timidity.sourceforge.net/`

[11] `https://en.wikipedia.org/wiki/SoundFont`

Table 4: F-measure results (*mean*/*sum*) for the CRNN model on public datasets when trained on different dataset combinations. The top part shows results for the 8 class scenario, while the bottom part shows results for the 18 class scenario. Whenever the *MIDI* set is mixed with real world datasets, only the 1% subset is used, to keep a balance between different data types.

8 instrument classes			
train set	*ENST*	*MDB*	*RBMA13*
all	0.61 / 0.64	0.68 / 0.64	0.57 / 0.52
MIDI	0.65 / 0.68	0.70 / 0.61	0.57 / 0.51
MIDI bal.	0.61 / 0.57	0.66 / 0.52	0.56 / 0.47
all+MIDI	0.58 / 0.62	0.67 / 0.57	0.57 / 0.52
all+MIDI bal.	0.61 / 0.64	0.68 / 0.56	0.56 / 0.51
pt MIDI	0.64 / **0.69**	0.72 / **0.68**	0.58 / **0.56**
pt MIDI bal.	0.61 / 0.63	0.72 / 0.67	0.58 / **0.56**
18 instrument classes			
train set	*ENST*	*MDB*	*RBMA13*
all	0.71 / 0.58	0.77 / 0.55	0.63 / 0.41
MIDI	0.73 / 0.61	0.77 / 0.53	0.64 / 0.39
MIDI bal.	0.70 / 0.52	0.76 / 0.45	0.63 / 0.35
all+MIDI	0.73 / 0.62	0.77 / 0.54	0.64 / 0.41
all+MIDI bal.	0.72 / 0.57	0.76 / 0.47	0.64 / 0.37
pt MIDI	0.74 / **0.67**	0.78 / **0.60**	0.64 / **0.47**
pt MIDI bal.	0.74 / 0.65	0.78 / 0.58	0.64 / 0.45

A downside of this approach is that the instrument switches may create artificial drum patterns which are atypical for western popular music. This can be problematic if the recurrent parts of the used CRNN architecture start to learn structures of typical drum patterns. Since these effects are difficult to measure and in order to be able to build a large, balanced dataset, this consequence was considered acceptable.

5. EXPERIMENTS

The first set of experiments evaluates the implemented ADT methods on the available public datasets, using the classic three drum instrument class labels, as well as the two new drum classification schemas with 8 and 18 classes, as a baseline. As evaluation measure primarily the F-measure of the individual drum instrument onsets is used. To calculate the overall F-measure over all instruments and all tracks of a dataset, two methods are used: First, the mean over all instruments' F-measure (=F-measure of track), as well as the mean over all tracks' F-measure is calculated (*mean*). Second, all false positives, false negatives, and true positives for all instruments and tracks are used to calculate a global F-measure (*sum*). These two values give insight into different aspects. While the *mean* value is more conservative for only slightly imbalanced classes, it is problematic when applied to sets containing only sparsely populated classes. In this case, some tracks may have zero occurrences of an instrument, thus resulting in a F-measure of 1.0 when no instrument is detected by the ADT system. In that case, the overall *mean* F-measure value for this instrument is close to 1.0 if it only occurs in a small fraction of tracks and the system never predicts it. On the other hand, the *sum* value will give a F-measure close to zero if the system never predicts an instrument, even for sparse classes—which is more desirable in this context.

The second set of experiments evaluates the performance of the ADT methods on the synthetic datasets, as well as a 1% subset

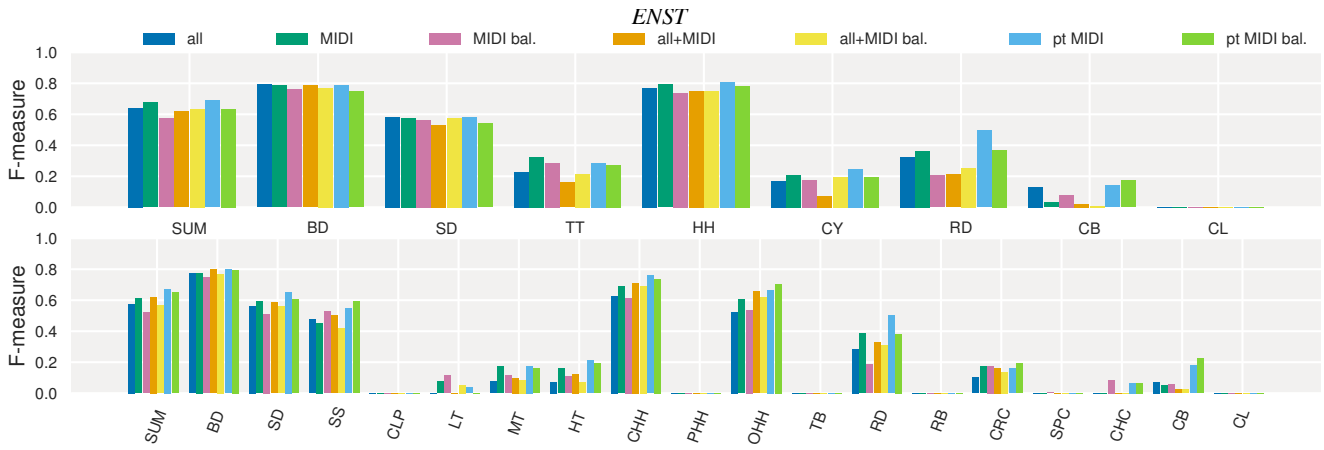


Figure 5: This figure shows F-measure results for each instrument, for both the 8 class (top) as well as the 18 class (bottom) scenarios, exemplary for the *ENST* dataset. Figures for other sets are found on the accompanying webpage (see sec. 7). The color of bars indicates the dataset or combinations trained on: *all*—three public datasets; *MIDI*—synthetic dataset; *MIDI bal.*—synthetic set with balanced classes; *all+MIDI*—three public datasets plus 1% split of synthetic dataset; *all+MIDI bal.*—three public datasets plus the 1% split of the balanced synthetic dataset; *pt MIDI* and *pt MIDI bal.*—pre-trained on the *MIDI* and *MIDI bal.* datasets respectively and fine tuned on *all*. The first set of bars on the left (SUM) shows the overall *sum* F-measure value.

for each of the instrument classification schemas. This will give insight in how the systems perform on the synthetic dataset and how relevant the data volume is for each of the schemas.

In the final set of experiments, models trained with different combinations of synthetic and real data will be evaluated. The evaluation will show how well models trained on synthetic data can generalize on real world data. Mixing the real world datasets with the symbolic data is a first, simple approach of leveraging a balanced dataset to improve detection performance of underrepresented drum instrument classes in currently available datasets. To be able to compare the results, models are trained on all of the public datasets (*all*), the full synthetic dataset (*MIDI*), the balanced versions of the synthetic dataset (*MIDI bal.*), a mix of the public datasets and the 1% subset of the synthetic dataset (*all+MIDI*), and a mix of the public datasets and a 1% subset of the balanced synthetic datasets (*all+MIDI bal.*). Additionally, models pre-trained on the *MIDI* and *MIDI bal.* datasets with additional refinement on the *all* dataset were included. We only compare a mix of the smaller public datasets to the other sets, since models trained on only one small dataset have the tendency to overfit, and thus generalize not well—which makes comparison problematic.

6. RESULTS AND DISCUSSION

The results of the first set of experiments is visualized in Table 2, which shows the 3-fold cross-validation results for models trained on public datasets with 3, 8, and 18 labels. The resulting F-measure values are not surprising: for the 3-class scenario the values are close to the reported values in the related work. Differences are due to slightly different models and hyper-parameter settings for training. As expected, especially the *sum* values drop for the cases of 8 and 18 classes. It can be observed, that the CRNN performs best for all sets in 18 class scenario and for two out of three sets for the eight class scenario.

Table 3 shows the results for models trained on synthetic datasets with 3, 8, and 18 labels. As expected, there is a tendency for the models trained on the 1% subset to perform worse, especially for the CRNN. However, this effect is not as severe as suspected. This might be due to the fact that, while different drum kits were used, the synthetic set is still quite uniform, given its size. The overall results for the balanced sets are worse than for the normal set. This is expected, since the difficulty of the balanced sets is much greater than for the imbalanced one (sparse classes can be ignored by the models without much penalty). Figure 4 shows a comparison of F-measure values for individual instruments classes when training on *MIDI* and *MIDI bal.* sets. The plot shows, that performance for underrepresented classes improves for the balanced set, which was the goal of balancing the set. A downside is that the performance for classes which have a higher frequency of occurrence in the *MIDI* dataset decreases in most cases, which contributes to the overall decrease. However, this effect is less severe in the 8 class case.

A general trend which can be observed, especially in the scenarios with more instrument class labels, is that CRNNs consistently outperform CNNs. Since this is true for all other experiments as well, and for reasons of clarity, we will limit the results for the next plots and tables to those of the CRNN model.

Table 4 shows the F-measure results for the CRNN model trained on different dataset combinations and evaluated on public datasets. In figure 5, a detailed look in the context of cross-datasets evaluation on instrument class basis for the *ENST* dataset is provided. As mentioned in section 5, results for models trained on only one public dataset are not included in this chart. While the performance for those is higher, they are slightly overfitted to the individual datasets and do not generalize well to other datasets, therefore a comparison would not be meaningful. Although an overall big performance improvement for previously underrepresented classes can not be observed, several interesting things are visible: *(i.)* both the models trained solely on the *MIDI* and the *MIDI bal.* datasets generalize surprisingly well to the real world dataset; *(ii.)* in some cases, performance improvements for underrepresented classes can be observed (e.g. for 18 classes: LT, MT, RD, CRC, CHC), when using the synthetic data; *(iii.)* bal-

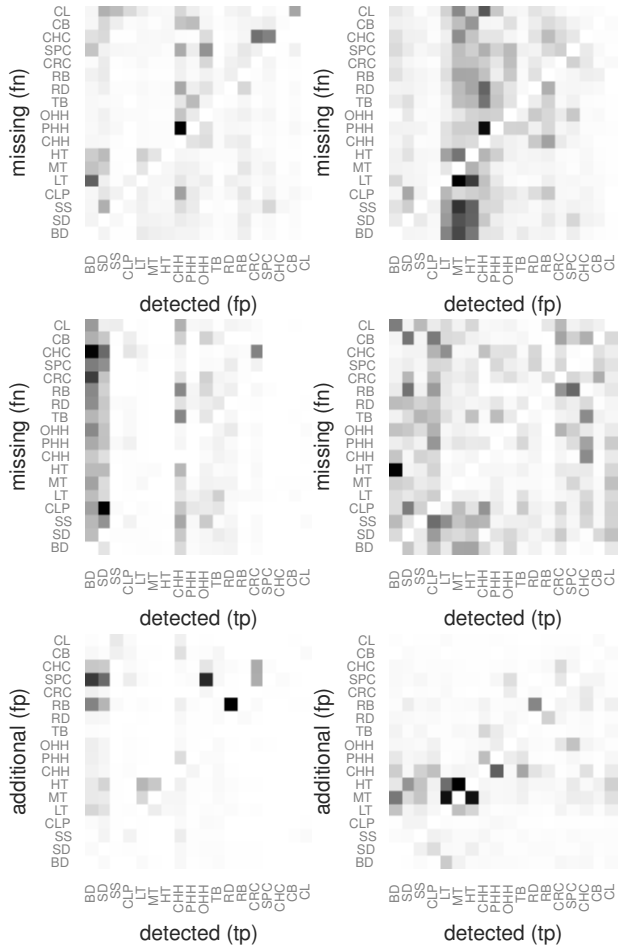


Figure 6: Left column shows matrices for *MIDI* set, right column shows matrices for *MIDI bal.* set, both for the 18 classes scenario. From top to bottom, the matrices display: classic confusions (fn/fp), masking by true positives (fn/tp), and positive masking (excitement—fp/tp).

ancing the instruments, while effective within the evaluation for the synthetic dataset, seems not to have a positive effect in the cross-dataset scenario and when mixing dataset; and *(iv.)* using pre-training on the *MIDI* set with refinement on the *all* set, seems to produce models which are better suited to detect underrepresented classes while still performing well on other classes.

To gain more insight into which errors the systems make when classifying within the 8 and 18 class systems, three sets of pseudo confusion matrices were created. We term them *pseudo* confusion matrices because one onset instance can have multiple classes, which is usually not the case for classification problems. These three pseudo confusion matrices indicate how often *(i.)* a false positive for another instrument was found for false negatives (classic confusions); *(ii.)* a true positive for another instrument was found for false negatives (onset masked or hidden); and *(iii)* a true positive for another instrument was found for a false positive (positive masking or excitement). Figure 6 shows examples of these matrices for the *MIDI* and *MIDI bal.* sets in the 18 class scenario. The images lead to intuitive conclusions: similar sounding instruments may get confused (BD/LT, CHH/PHH), instruments with energy over a wide frequency range mask more delicate instruments as well as similar sounds (HT/BD, CLP/SD), and similar sounding instruments lead to false positives (LT/MT/HT, RB/RD). Many of these errors may very well be made by human transcribers as well. This also strengthens the assumption that instrument mappings are not well defined: boundaries of the frequency range between bass drum, low, mid and high toms are not well defined, the distinction between certain cymbals is sometimes difficult even for humans, and different hi-hat sounds are sometimes only distinguishable given more context, like genre or long term relations within the piece.

To further improve performance, an ensemble of models trained on different datasets (synthetic and real, including balanced variants) can be used. However, experience shows that while these systems often perform best in real world scenarios and in competitions (e.g. MIREX), they give not so much insight in an evaluation scenario.

7. CONCLUSION

In this work we discussed a shortcoming of current state-of-the art automatic drum transcription systems: the limitation to three drum instruments. While this choice makes sense in the context of currently available datasets, some real world applications require transcription of more instrument classes. To approach this shortcoming, we introduced a new and publicly available large scale synthetic dataset with balanced instrument distribution and showed that models trained on this dataset generalize well to real world data. We further showed that balancing can improve performance for usually underrepresented classes in certain cases, while overall performance may decline. An analysis of mistakes made by such systems was provided and further steps into this directions were discussed. The dataset, trained models and further material are available on the accompanying webpage.[12]

8. ACKNOWLEDGEMENTS

This work has been partly funded by the Austrian FFG under the BRIDGE 1 project *SmarterJam* (858514), as well as by the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (ERC Grant Agreement No. 670035, project *CON ESPRESSIONE*).

9. REFERENCES

[1] Richard Vogl, Matthias Dorfer, Gerhard Widmer, and Peter Knees, "Drum transcription via joint beat and drum modeling using convolutional recurrent neural networks," in *Proc. 18th Intl. Soc. for Music Information Retrieval Conf. (ISMIR)*, Suzhou, CN, Oct. 2017.

[2] Carl Southall, Ryan Stables, and Jason Hockman, "Automatic drum transcription using bidirectional recurrent neural networks," in *Proc. 17th Intl. Soc. for Music Information Retrieval Conf. (ISMIR)*, New York, NY, USA, Aug. 2016.

[3] Carl Southall, Ryan Stables, and Jason Hockman, "Automatic drum transcription for polyphonic recordings using soft attention mechanisms and convolutional neural net-

[12] `http://ifs.tuwien.ac.at/~vogl/dafx2018`

works," in *Proc. 18th Intl. Soc. for Music Information Retrieval Conf. (ISMIR)*, Suzhou, China, Oct. 2017.

[4] Rich Caruana, "Multitask learning," in *Learning to learn*. Springer, 1998.

[5] Olivier Gillet and Gaël Richard, "Automatic transcription of drum loops," in *Proc. 29th IEEE Intl. Conf. on Acoustics, Speech, and Signal Processing (ICASSP)*, Montreal, QC, Canada, May 2004, vol. 4.

[6] Marius Miron, Matthew EP Davies, and Fabien Gouyon, "An open-source drum transcription system for pure data and max msp," in *Proc. 38th IEEE Intl. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, Vancouver, BC, Canada, May 2013.

[7] Kazuyoshi Yoshii, Masataka Goto, and Hiroshi G Okuno, "Drum sound recognition for polyphonic audio signals by adaptation and matching of spectrogram templates with harmonic structure suppression," *IEEE Trans. on Audio, Speech, and Language Processing*, vol. 15, no. 1, 2007.

[8] Jouni Paulus and Anssi Klapuri, "Drum sound detection in polyphonic music with hidden markov models," *EURASIP Journal on Audio, Speech, and Music Processing*, 2009.

[9] Chih-Wei Wu and Alexander Lerch, "Automatic drum transcription using the student-teacher learning paradigm with unlabeled music data," in *Proc. 18th Intl. Soc. for Music Information Retrieval Conf. (ISMIR)*, Suzhou, China, Oct. 2017.

[10] George Tzanetakis, Ajay Kapur, and Richard I McWalter, "Subband-based drum transcription for audio signals," in *Proc. 7th IEEE Workshop on Multimedia Signal Processing*, Shanghai, China, Oct. 2005.

[11] Maximos A. Kaliakatsos-Papakostas, Andreas Floros, Michael N. Vrahatis, and Nikolaos Kanellopoulos, "Real-time drums transcription with characteristic bandpass filtering," in *Proc. Audio Mostly: A Conf. on Interaction with Sound*, Corfu, Greece, 2012.

[12] Olivier Gillet and Gaël Richard, "Supervised and unsupervised sequence modelling for drum transcription," in *Proc. 8th Intl. Conf. on Music Information Retrieval (ISMIR)*, Vienna, Austria, Sept. 2007.

[13] Derry FitzGerald, Bob Lawlor, and Eugene Coyle, "Sub-band independent subspace analysis for drum transcription," in *Proc. 5th Intl. Conf. on Digital Audio Effects (DAFx)*, Hamburg, Germany, 2002.

[14] Andrio Spich, Massimiliano Zanoni, Augusto Sarti, and Stefano Tubaro, "Drum music transcription using prior subspace analysis and pattern recognition," in *Proc. 13th Intl. Conf. on Digital Audio Effects (DAFx)*, Graz, Austria, 2010.

[15] Christian Dittmar and Christian Uhle, "Further steps towards drum transcription of polyphonic music," in *Proc. 116th Audio Engineering Soc. Conv.*, Berlin, Germany, May 2004.

[16] Chih-Wei Wu and Alexander Lerch, "Drum transcription using partially fixed non-negative matrix factorization with template adaptation," in *Proc. 16th Intl. Soc. for Music Information Retrieval Conf. (ISMIR)*, Málaga, Spain, Oct. 2015.

[17] Christian Dittmar and Daniel Gärtner, "Real-time transcription and separation of drum recordings based on nmf decomposition," in *Proc. 17th Intl. Conf. on Digital Audio Effects (DAFx)*, Erlangen, Germany, Sept. 2014.

[18] Chih-Wei Wu, Christian Dittmar, Carl Southall, Richard Vogl, Gerhard Widmer, Jason Hockman, Meinhard Müller, and Alexander Lerch, "A review of automatic drum transcription," *IEEE Trans. on Audio, Speech and Language Processing*, vol. 26, no. 9, Sept. 2018.

[19] Richard Vogl, Matthias Dorfer, and Peter Knees, "Recurrent neural networks for drum transcription," in *Proc. 17th Intl. Soc. for Music Information Retrieval Conf. (ISMIR)*, New York, NY, USA, Aug. 2016.

[20] Richard Vogl, Matthias Dorfer, and Peter Knees, "Drum transcription from polyphonic music with recurrent neural networks," in *Proc. 42nd IEEE Intl. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, New Orleans, LA, USA, Mar. 2017.

[21] Sebastian Böck, Florian Krebs, and Gerhard Widmer, "Joint beat and downbeat tracking with recurrent neural networks," in *Proc. 17th Intl. Soc. for Music Information Retrieval Conf. (ISMIR)*, New York, NY, USA, 2016.

[22] Chih-Wei Wu and Alexander Lerch, "On drum playing technique detection in polyphonic mixtures," in *Proc. 17th Intl. Soc. for Music Information Retrieval Conf. (ISMIR)*, New York City, United States, August 2016.

[23] Justin Salamon, Rachel M Bittner, Jordi Bonada, Juan José Bosch Vicente, Emilia Gómez Gutiérrez, and Juan Pablo Bello, "An analysis/synthesis framework for automatic f0 annotation of multitrack datasets," in *Proc. 18th Intl. Soc. for Music Information Retrieval Conf. (ISMIR)*, Suzhou, China, Oct. 2017.

[24] Kyunghyun Cho, Bart van Merriënboer, Dzmitry Bahdanau, and Yoshua Bengio, "Learning phrase representations using rnn encoder-decoder for statistical machine translation," in *Proc. Conf. on Empirical Methods in Natural Language Processing (EMNLP)*, Doha, Qatar, Oct. 2014.

[25] Sepp Hochreiter and Jürgen Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, Nov. 1997.

[26] Sebastian Böck and Gerhard Widmer, "Maximum filter vibrato suppression for onset detection," in *Proc 16th Intl Conf on Digital Audio Effects*, Maynooth, Ireland, Sept. 2013.

[27] Diederik P. Kingma and Jimmy Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[28] Olivier Gillet and Gaël Richard, "Enst-drums: an extensive audio-visual database for drum signals processing," in *Proc. 7th Intl. Conf. on Music Information Retrieval (ISMIR)*, Victoria, BC, Canada, Oct. 2006.

[29] Carl Southall, Chih-Wei Wu, Alexander Lerch, and Jason Hockman, "Mdb drums – an annotated subset of medleydb for automatic drum transcription," in *Late Breaking/Demos, 18th Intl. Soc. for Music Information Retrieval Conf. (ISMIR)*, Suzhou, China, Oct. 2017.

[30] Rachel M Bittner, Justin Salamon, Mike Tierney, Matthias Mauch, Chris Cannam, and Juan Pablo Bello, "Medleydb: A multitrack dataset for annotation-intensive mir research.," in *Proc. 15th Intl. Soc. for Music Information Retrieval Conf. (ISMIR)*, Taipei, Taiwan, Oct. 2014, vol. 14.

STATIONARY/TRANSIENT AUDIO SEPARATION USING CONVOLUTIONAL AUTOENCODERS

Gerard Roma
CeReNeM
University of Huddersfield
Huddersfield, UK
g.roma@hud.ac.uk

Owen Green
CeReNeM
University of Huddersfield
Huddersfield, UK
o.green@hud.ac.uk

Pierre Alexandre Tremblay
CeReNeM
University of Huddersfield
Huddersfield, UK
p.a.tremblay@hud.ac.uk

ABSTRACT

Extraction of stationary and transient components from audio has many potential applications to audio effects for audio content production. In this paper we explore stationary/transient separation using convolutional autoencoders. We propose two novel unsupervised algorithms for individual and and joint separation. We describe our implementation and show examples. Our results show promise for the use of convolutional autoencoders in the extraction of sparse components from audio spectrograms, particularly using monophonic sounds.

1. INTRODUCTION

The problem of identifying transients in audio signals (especially musical audio) has received significant attention in the phase vocoder literature, given the difficulties posed by transients to sinusoidal models. As a consequence, a number of sines + transients + noise models were proposed [1, 2]. Transient and stationary components can in fact be related with general signal models prevalent in audio effects [3].

These models are often applied to monophonic sounds, but their application to broad polyphonic signals remains challenging. Meanwhile, researchers focusing on separation of polyphonic signals into their component sources have developed a similar separation task, often dubbed harmonic-percussive source separation (HPSS). This name obviously assumes the presence of harmonic and percussive components in audio. However, techniques employed for this task often do not actually take into account harmonicity of musical tones and instead focus on other aspects of typically harmonic components of polyphonic signals. Most algorithms are based, in one way or another, on the observation that percussive and harmonic components tend to form straight vertical and horizontal lines in the spectrogram. This property can be called the *anisotropic smoothness* [4]. Several works have been developed to exploit this using non-negative factorization algorithms [5, 6]. A very popular approach is to simply use a combination of two median filters [7].

Separation of audio into stationary and transient components, that is, without modeling sinusoids, was proposed in a recent study [8]. This perspective allows the application of ideas based on anisotropic smoothness to digital audio effects. In this sense, this task remains in an abstract domain related to signal models, which makes quantitative evaluation elusive.

In this paper, we propose two algorithms for transient/stationary separation using convolutional autoencoders (CAE). Autoencoders are neural network algorithms that purposely realize imperfect replicas of input signals based on some constraints. Thanks to current neural network programming libraries, such constraints can be specified directly into cost functions, without having to worry about their derivative. This provides a promising framework for experimenting with digital audio effects. In this paper we explore their use for transient/stationary separation by implementing anisotropic smoothness constraints from the HPSS literature in the cost functions.

2. CONVOLUTIONAL AUTOENCODERS

Autoencoders are neural network algorithms that try to reconstruct the input from a typically lower dimension hidden representation. The encoder is typically the combination of an affine transform with weights W and biases b with some non-linear activation σ:

$$h = \sigma(Wx + b). \tag{1}$$

Here, h is a hidden representation of x with dimensionality determined by the weight matrix. The decoder then performs the inverse operation to obtain a reconstruction y:

$$y = \sigma(W'h + b'). \tag{2}$$

This is typically accomplished using some variant of stochastic gradient descent (SGD) that learns the parameters W, b, W' and b' to minimize the some distance metric between x and y. Autoencoders have been extensively used in machine learning, usually not for the reconstruction itself but for learning useful features from data. The parameters that produce the hidden representation h are then used in other neural networks for e.g. image classification. In order to avoid that the algorithm learns to exactly copy the input to the output, which would not yeld useful features, the main strategies are choosing a lower dimensionality for h or adding some sparsity constraint to the cost function.

An analogy of traditional autoencoders with non-negative factorization (NMF) algorithms used for audio separation was proposed in [9] but evaluated only for the supervised case. Supervised neural networks used in separation of musical audio [10] can be seen as supervised autoencoders, in the sense that the output is the same shape as the input.

Traditional autoencoders, however, process data in one dimension and thus cannot be used to learn time-frequency patterns. The usual solution of stacking several spectral frames quickly degenerates into prohibitive computational costs.

Convolutional neural networks (CNNs) have become the standard algorithm for image classification and object recognition. They have also been shown to work for speech recognition [11] and audio classification [12]. In CNNs, the weights are typically square convolution kernels that are shared, i.e. each kernel is

convolved with the whole image. The resulting representation is downsampled with respect to the input image size (which can be further downsampled with pooling operations), but typically composed by multiple channels corresponding to each learnable kernel.

Convolutional autoencoders (CAEs) arose in this context, allowing the use of 2D convolution operations for learning features. In a CAE, the operation in Equation 1 is rewritten as :

$$h^i = \sigma(X * W^i + b^i), \tag{3}$$

where $*$ represents a 2D convolution operation. Here the input X is a matrix. The hidden representation is now a tensor, $h \in \mathbb{R}^{d,m,n}$, with d determined by the number of kernels (hence the index i for each convolution). In addition to the size of the input and the kernel, dimensions m and n can be affected by several parameters of the convolution, such as input padding and stride. Thus, h can be in all a higher-dimensional representation than the input, but the information has to be transmitted through the convolution with small (typically 5x5) kernels.

While these are conventional convolution layers used in CNNs, the particularity of CAEs is to introduce an upsampling convolution that allows restoring the original size in the decoder:

$$Y = \sigma(h^i * W'^i + b'^i). \tag{4}$$

This operation is informally called "deconvolution" [13], or more technically *fractionally strided convolution* [14], and it involves padding and re-shaping the kernels into a convolution matrix of a size that allows recovering the original size through convolution with the hidden representation. One interesting property of this architecture is that it can be used for images (in our case magnitude spectrograms) of arbitrary size.

Supervised networks with deconvolution decoders have recently started to appear in the source separation literature [15, 16]. In this paper, we explore the use of this architecture in an unsupervised setting for stationary / transient separation of audio. It is common in AEs and CAEs to implement restrictions in the loss function, in addition to the output being similar to the input. This feature can thus be used to devise new audio effects. In the case of CAEs, the loss function can take into account both the time and frequency dimensions and promote vertical or horizontal lines as commonly done for HPSS.

3. TRANSIENT / STATIONARY AUDIO SEPARATION

In this section we describe different loss functions that can be used to train a CAE. As noted, our approach consists of using the network to process a magnitude spectrogram. We define X to be such spectrogram (e.g. it has been obtained from some complex spectrogram C), and assume it to be a sum of two components:

$$X = X_t + X_s. \tag{5}$$

Here X_t represents the time-frequency bins associated with transients, and X_s the ones associated with stationary components. We regard this as a useful abstraction and not as a physical mixture, beyond the fact that musical sounds typically contain transients and steady tones. It is often useful to distinguish a noise component that is not associated with transients. While we do not model this component directly, we observe in Section 3.1 that a basic CAE can be used to remove background noise. In Section 3.2 we show a model that can be used to individually estimate X_t or X_s. This allows using a different time-frequency grid that may be more appropriate for each situation. In this case, a corresponding complex estimate can be recovered using the original phase, e.g. for a complex STFT:

$$\hat{C} = \hat{X} e^{\phi j}, \tag{6}$$

where $\hat{X}$ can either be $\hat{X}_t$ or $\hat{X}_s$, and ϕ is the phase of the original spectrogram.

On the other hand, for ensuring that X is recovered by the sum of both estimates, it may be convenient to estimate a soft mask, i.e:

$$M_t = \frac{\hat{X}_t}{\hat{X}_t + \hat{X}_s}, \tag{7}$$

$$M_s = \frac{\hat{X}_s}{\hat{X}_t + \hat{X}_s}, \tag{8}$$

using a common transform for both components. $\hat{C}$ is then obtained as $M_t \otimes C$ or $M_s \otimes C$, where $\otimes$ denotes the element-wise product. In Section 3.3 we describe a model for jointly obtaining $\hat{X}_t$ and $\hat{X}_s$ from the same spectrogram.

3.1. Basic CAE

A basic CAE implementation simply tries to recover the input. A suitable loss function would then be the mean square error (MSE) between the input X and the output Y:

$$L_{MSE} = \frac{1}{TF} \sum (X - Y)^2, \tag{9}$$

where T and F are the dimensions of the spectrogram. The goal of the algorithm is then to find an optimal set of kernels that allow this reconstruction through 2D convolutions. Using this function implies the danger of simply copying the input. It is easy to see that a convolution kernel with a single active weight would accomplish that. One common solution is to add a sparsity constraint on the hidden representation. However, here we are interested in the output (i.e. transient or stationary components) being sparser than the input. Promoting a sparse hidden representation does not directly accomplish that, because the decoder can try to learn to re-create the (non-sparse) input from the sparse hidden representation. Hence, we add a sparse penalty to the output directly:

$$L = L_{MSE} + \lambda_1 ||Y||_1, \tag{10}$$

where $||*||_1$ denotes the L1 norm. The parameter λ_1 then controls the sparsity of the output spectrogram. In early experiments with this model, we observed it may have interesting applications to denoising and dereverberation. In this sense, traditional autoencoders have been applied to speech enhancement [17]. It can also be used to implement more experimental effects. However our main goal in this work is the transient/stationary decomposition.

3.2. Individual extraction

Estimation of transient or stationary components from the input signal can be promoted by adding more terms to the loss function. We regard the difference across either the time or the frequency axis as a cost for estimating transient or stationary components respectively:

$$d_f = \frac{\sum_{t,f} (Y(t,f) - Y(t,f-1))^2}{{||Y||_2}^2}, \tag{11}$$

$$d_t = \frac{\sum_{t,f} (Y(t,f) - Y(t-1,f))^2}{||Y||_2{}^2}, \quad (12)$$

where $|| * ||_2$ denotes the L2 norm, and t and f are time and frequency indices. The loss for estimating either the transient or the stationary components is then computed by adding respectively $\frac{d_f}{d_t+\varepsilon}$ or $\frac{d_t}{d_f+\varepsilon}$ (where ε is a small number to prevent division by 0) to L_{MSE}:

$$L_S = L_{MSE} + \lambda_1 ||Y||_1 + \lambda_2 \frac{d_f}{d_t + \epsilon}, \quad (13)$$

$$L_T = L_{MSE} + \lambda_1 ||Y||_1 + \lambda_2 \frac{d_t}{d_f + \epsilon}, \quad (14)$$

Parameters λ_1 and λ_2 can here be mapped to user interface parameters: the first one defines the level of sparsity (i.e. how much magnitude will be lost in the process) and the second biases it towards the desired component.

3.3. Joint extraction

Estimating both transient and stationary components simultaneously has the potential advantage of allowing a more discriminative model that can use the input data to provide two estimates. The estimates can then be used to construct time-frequency masks as described in Equations 8 and 7. Here, the output of the autoencoder is a tensor $Y \in \mathbb{R}^{2,M,N}$ where M and N correspond to the original spectrogram size. For simplicity of notation, we denote $Y_t \in \mathbb{R}^{T,F}$ and $Y_s \in \mathbb{R}^{T,F}$ as the outputs of the CAE for transient and stationary components respectively. The MSE loss then needs to be rewritten as:

$$L_{MSE} = \frac{1}{TF} \sum (X - (Y_t + Y_s))^2. \quad (15)$$

The terms df and d_t can now be computed separately for Y_t and Ys respectively. The loss function for the CAE is then:

$$L_{ST} = L_{MSE} + \lambda_1 ||Y||_1 + \lambda_2 \frac{d_{t1}}{d_{f1} + \epsilon} + \lambda_3 \frac{d_{f2}}{d_{t2} + \epsilon}, \quad (16)$$

where d_{t1} / d_{f1} are computed from Y_s as in Equations 11 and 12, and d_{t2} / d_{f2} are equally computed from Y_t .

4. IMPLEMENTATION

In order to test the proposed approach, we implemented the CAE models described in Sections 3.1, 3.2 and 3.3, respectively denoted here as $cae1$, $cae2$ and $cae3$. The implementation was based on the pytorch library.[1] Figure 1 shows the layout that is common to the three models. We used 5x5 convolution kernels, which are widely used for images and have also been used for audio classification [12]. All networks were devised with 4 convolution kernels and one single hidden representation of 4 channels. Inputs to all convolutions were padded with 2 bins on each side and dimension. This means there was really no downsampling neither pooling, and the hidden representations had the same dimension of the input, which helped recovering the fine details of the input. Both the encoder and the decoder used Rectified Linear Units (ReLU) as activation functions. Initialization for weights connected to ReLUs is conventionally implemented as specified in [18]. However, we found that for this unsupervised setting, results could be unstable due to random initialization. On one hand, different initial balances between the components of the loss function could lead the network to fall into a local minimum. On the other, the network could end in a slightly different state for the same number of iterations even when converging to a stable solution. In order to make the networks predictable, we used a basic CAE trained to optimize only L_{MSE} to pre-initialize the weights. The proposed models were then used to *fine tune* the weights with the additional loss components. This had the side effect of choosing a random seed. It has been shown that pre-training is robust to changes in the random seed [19]. We verified that, for different pre-trained networks, our models would always converge to a stable solution. However, thinking about the use of the algorithm in an interactive effects processor, the predictability resulting from the use of a fixed random seed was also beneficial. All models were trained using the ADAM [20] variant of stochastic gradient descent (SGD). Like in [9], each spectrogram was used as a single batch, both for pre-training and fine tuning. For the pre-training, we used two different datasets, one composed of monophonic loops and one with polyphonic music signals. For dealing with monophonic sounds, the pre-training dataset was obtained by randomly sampling 100 loop sounds from the collection bundled with Apple's Logic Pro software. For dealing with polyphonic mixtures, the pre-training dataset was created by extracting one minute from each song in the test set (50 songs) of the DSD100 dataset.[2] For both the training and pre-training stages, the $weight\_decay$ parameter, available in pytorch, was used. This corresponds to an l_2 regularization in the weights, which is omitted in the formulation for clarity. A value of 0.01 was used for $cae1$ and $cae2$, while for $cae3$ a higher value of 0.5 helped prevent the weights getting biased towards one of the two outputs. All networks were trained for 100 epochs. Spectrograms were computed using 20 ms windows with 15 ms overlap except when noted. The code for the implementation can be obtained from `https://github.com/flucoma/DAFX-2018`.

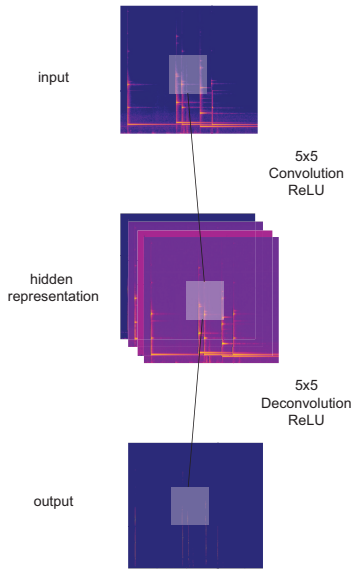
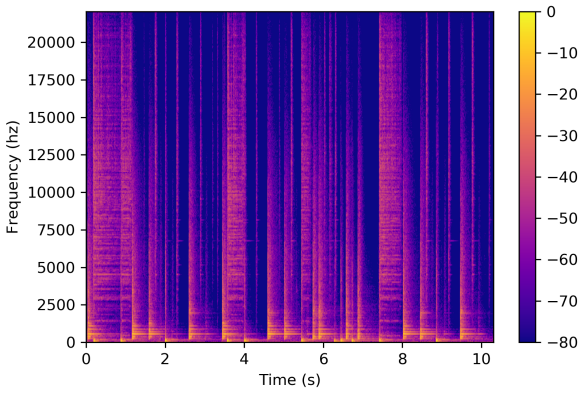


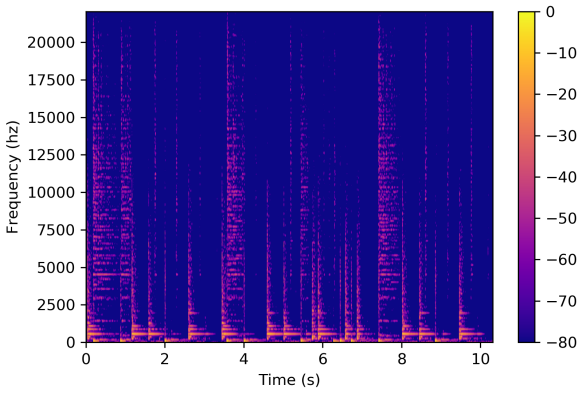
Figure 1: Convolutional autoencoder network structure

[1] `http://pytorch.org/`

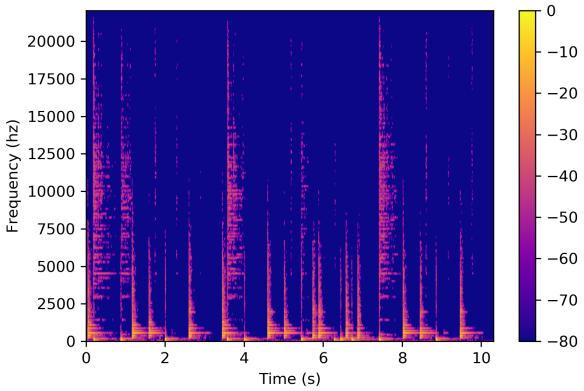
[2] `https://github.com/faroit/dsdtools`



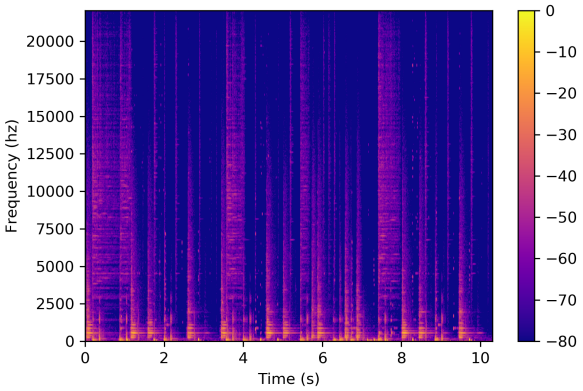
(a) Original drum loop sound



(b) Drum loop processed by cae1



(c) Drum loop with thresholded magnitude



(d) Drum loop remixed with cae1 variations

Figure 2: Using cae 1 on a drum loop

5. EXAMPLE RESULTS

In this section we show examples of the use of $cae1$, $cae2$ and $cae3$ as described in the previous sections. We first show a creative application of $cae1$ with a drum loop, then we analyze the separation into steady and transient components of $cae2$ and $cae3$ using a monophonic and a poyphonic sample. All audio examples can be listened in the companion web page for this paper: `http://www.flucoma.org/DAFX-2018/`.

The original drum loop is shown in Figure 2a. A sparse version obtained with $cae1$ is shown in Figure 2b. A lot of the resonance of the drums has been lost. For comparison, Figure 2c shows a version of the original with the same number of zero entries (around 94%) as the processed version (i.e. magnitude bins were sorted and zeroed below a threshold to obtain the same number of zeros). It seems that $cae1$ focuses more on the harmonics of the drums. We found this effect can be used for creative processing to obtain multiple variations of the same sample. As an example, Figure 2d shows an example where multiple copies using different values of λ_1 at different window and hop sizes have been mixed with the original.

We now focus on transient/stationary separation using $cae2$ and $cae3$. Figure 3 is a monophonic fragment of a glockenspiel melody from Freesound.org[3]. The original sound includes significant background noise. Figures 4a and 4b show the magnitude spectrograms of the separation with $cae2$. The background noise has been eliminated, and the stationary and transient components are clearly separated. When listening to the sounds, it can be noted

[3] `https://freesound.org/people/bbatv/sounds/332932/`

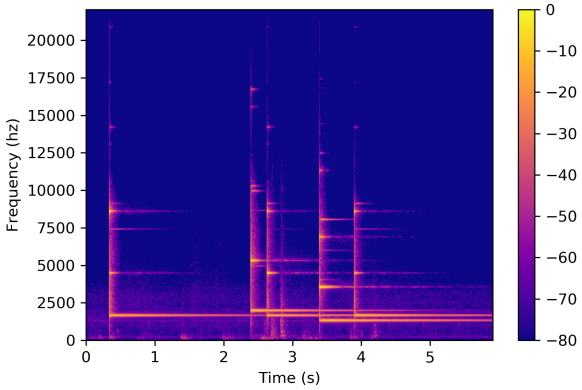
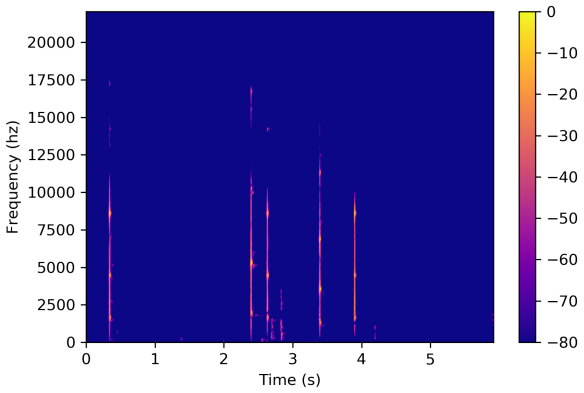
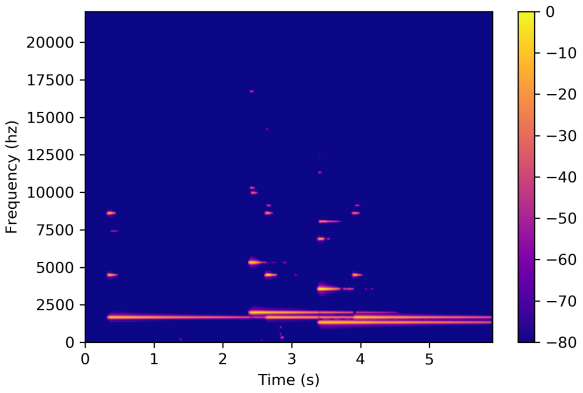


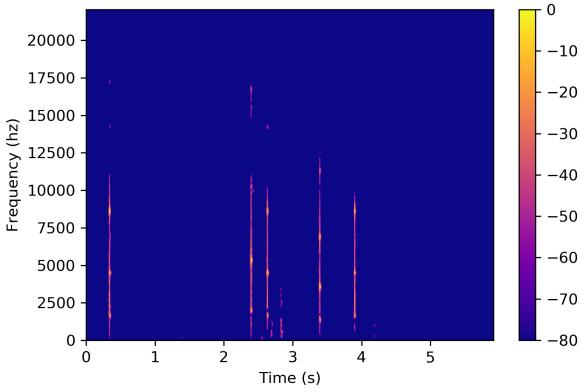
Figure 3: Original glockenspiel sound



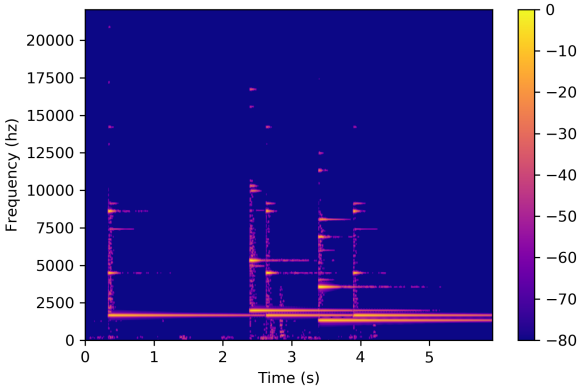
(a) Separation of glockenspiel transients with $cae2$.



(b) Separation of glockenspiel stationary components with $cae2$.



(c) Separation of glockenspiel transients with $cae3$.



(d) Separation of glockenspiel stationary components with $cae3$.

Figure 4: Using cae2 and cae3 on a glockenspiel sound

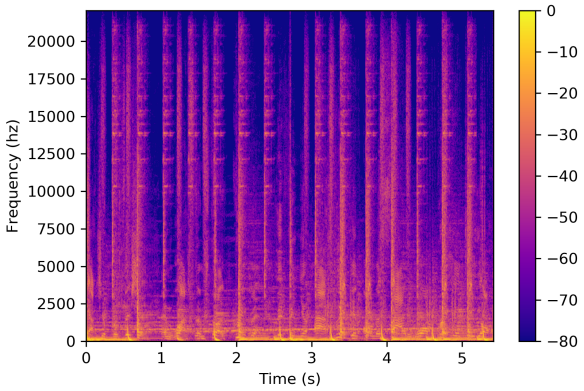


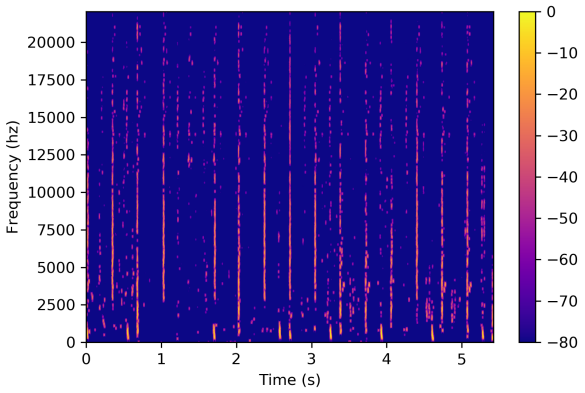
Figure 5: Original polyphonic mixture

that the transients still retain some of the pitch information but the duration is very short. In the stationary components, the attack has been clearly removed. The parameter values for the transient estimation were $\lambda_1 = 8\text{e-}4$, $\lambda_2 = 300$. For the stationary estimation, the values were $\lambda_1 = 4\text{e-}5$, $\lambda_2 = 10$. Figures 4c and 4d show the results with $cae3$. The spectrograms look also sparse, but the stationary components seem to show a stronger attack, which can be attributed to the use of the soft mask. When listening to the audio it can be noted that the attack is in fact very soft. In this case, the parameters were tuned to $\lambda_1 = 5\text{e-}5$, $\lambda_2 = 0.2$, $\lambda_3 = 0.3$.

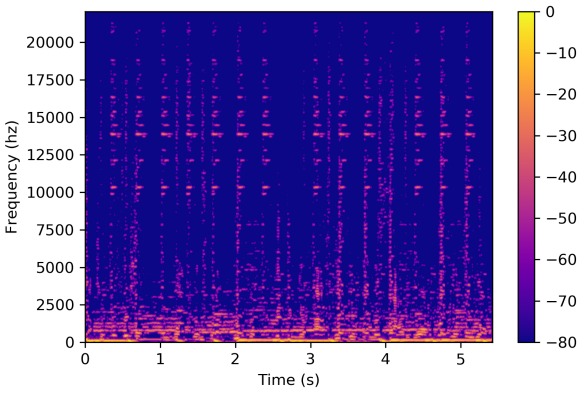
For both models, the strategy was to set first the target level of sparsity with λ_1 and then adjust the rest of parameters. However, we noted that the competition of both estimates in $cae3$ makes it more difficult to find appropriate values for the parameters.

Figures 5, 6a, 6b, 6c and 6d correspond to a hip hop music excerpt[4]. The separation is obviously more difficult. For both networks, the separation of transients produces noticeable musical noise. They are still good indicators of the downbeat of the rhythm. The stationary components in $cae2$ are biased towards the bass, which is salient and perhaps the only instrument producing steady tones. Contrastingly for $cae3$ the stationary part is remarkably more simlar to the mix, but with smoothed transients, which could be attributed to the joint estimation. The parameters for $cae2$ were $\lambda_1 = 1\text{e-}4$, $\lambda_2 = 100$ and $\lambda_1 = 4\text{e-}5$, $\lambda_2 = 50$ for transient and stationary components, and $\lambda_1 = 1\text{e-}4$, $\lambda_2 = 2$, $\lambda_3 = 6$ for the joint model $cae3$.

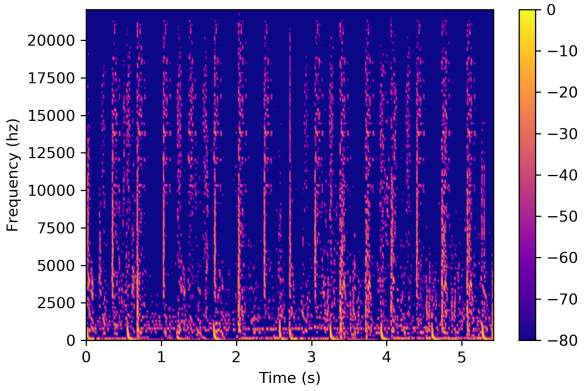
[4]The excerpt was extracted from *Attention* by Catburglaz, `http://catburglaz.com`



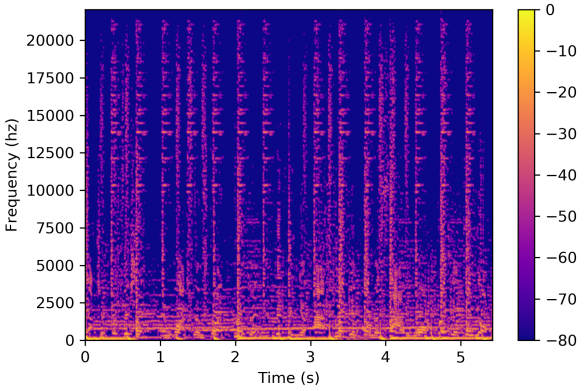
(a) Separation of transients in polyphonic mixture with $cae2$.



(b) Separation of stationary components in polyphonic mixture with $cae2$.



(c) Separation of transients in polyphonic mixture with $cae3$.



(d) Separation of stationary components in polyphonic mixture with $cae3$.

Figure 6: Using cae2 and cae3 with a polyphonic mixture

6. CONCLUSIONS

In this paper, we have explored the use of unsupervised convolutional autoencoders for audio transformation in the time-frequency domain. Specifically, we have shown that by programming custom loss functions they can be tuned to separate stationary and transient components. The results are encouraging, especially for monophonic sounds, while polyphonic mixtures are still challenging. One interesting aspect of this work is the possibility to control the learning process, producing different levels of sparseness and different qualities of transients and stationary components. This brings more flexibilty than HPSS approaches such as median filtering. Such flexibility is of particular interest to us as it presents opportunities for creative exploration: being able to tune processors by ear to fit aesthetically with the materials and the context in which they are used is a very important aspect of artistic interfaces. For future work, we plan to work on more useful mappings of the loss functions to user interface parameters.

7. ACKNOWLEDGEMENT

This project has received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (grant agreement n. 725899).

8. REFERENCES

[1] Scott N Levine and Julius O Smith III, "A sines+ transients+ noise audio representation for data compression and time/pitch scale modifications," in *Audio Engineering Society Convention 105*. Audio Engineering Society, 1998.

[2] Tony S Verma, Scott N Levine, and Teresa HY Meng, "Transient modeling synthesis: a flexible analysis/synthesis tool for transient signals.," in *Proceedings of the International Computer Music Conference (ICMC)*, 1997.

[3] Axel Roebel, "Between physics and perception: Signal models for high level audio processing," in *Proceedings of the International Conference on Digital Audio Effects (DAFx)*, 2010.

[4] H. Tachibana, N. Ono, H. Kameoka, and S. Sagayama, "Harmonic/percussive sound separation based on anisotropic smoothness of spectrograms," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 22, no. 12, pp. 2059–2073, Dec 2014.

[5] T. Virtanen, "Monaural sound source separation by non-negative matrix factorization with temporal continuity and sparseness criteria," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 15, no. 3, pp. 1066–1074, March 2007.

[6] Francisco Jesus Canadas-Quesada, Pedro Vera-Candeas, Nicolas Ruiz-Reyes, Julio Carabias-Orti, and Pablo Cabanas-Molero, "Percussive/harmonic sound separation by non-negative matrix factorization with smoothness/sparseness constraints," *EURASIP Journal on Audio, Speech, and Music Processing*, vol. 2014, no. 1, pp. 26, 2014.

[7] Derry Fitzgerald, "Harmonic/percussive separation using median filtering," in *Proceedings of the International Conference on Digital Audio Effects (DAFx)*, 2010.

[8] Kai Siedenburg and Simon Doclo, "Iterative structured shrinkage algorithms for stationary/transient audio separation," in *Proceedings of the International Conference on Digital Audio Effects (DAFx)*, 2017.

[9] Paris Smaragdis and Shrikant Venkataramani, "A neural network alternative to non-negative audio models," in *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2017, pp. 86–90.

[10] Aditya Arie Nugraha, Antoine Liutkus, and Emmanuel Vincent, "Multichannel music separation with deep neural networks," in *24th European Signal Processing Conference (EUSIPCO)*. IEEE, 2016, pp. 1748–1752.

[11] Ossama Abdel-Hamid, Abdel-rahman Mohamed, Hui Jiang, Li Deng, Gerald Penn, and Dong Yu, "Convolutional neural networks for speech recognition," *IEEE/ACM Transactions on audio, speech, and language processing*, vol. 22, no. 10, pp. 1533–1545, 2014.

[12] Shawn Hershey, Sourish Chaudhuri, Daniel PW Ellis, Jort F Gemmeke, Aren Jansen, R Channing Moore, Manoj Plakal, Devin Platt, Rif A Saurous, Bryan Seybold, et al., "Cnn architectures for large-scale audio classification," in *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2017, pp. 131–135.

[13] Matthew D Zeiler, Dilip Krishnan, Graham W Taylor, and Rob Fergus, "Deconvolutional networks," in *2010 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2010, pp. 2528–2535.

[14] Vincent Dumoulin and Francesco Visin, "A guide to convolution arithmetic for deep learning," *arXiv preprint arXiv:1603.07285*, 2016.

[15] Emad M Grais and Mark D Plumbley, "Single channel audio source separation using convolutional denoising autoencoders," in *IEEE Global Conference on Signal and Information Processing (GlobalSIP)*. IEEE, 2017, pp. 1265–1269.

[16] Andreas Jansson, Eric J. Humphrey, Nicola Montecchio, Rachel Bittner, Aparna Kumar, and Tillman Weyde, "Singing voice separation with deep U-Net convolutional networks," in *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, 2017, pp. 323–332.

[17] Xugang Lu, Yu Tsao, Shigeki Matsuda, and Chiori Hori, "Speech enhancement based on deep denoising autoencoder.," in *Interspeech*, 2013, pp. 436–440.

[18] K. He, X. Zhang, S. Ren, and J. Sun, "Delving deep into rectifiers: Surpassing human-level performance on imagenet classification," in *2015 IEEE International Conference on Computer Vision (ICCV)*, Dec 2015, pp. 1026–1034.

[19] Dumitru Erhan, Yoshua Bengio, Aaron Courville, Pierre-Antoine Manzagol, Pascal Vincent, and Samy Bengio, "Why does unsupervised pre-training help deep learning?," *Journal of Machine Learning Research*, vol. 11, no. Feb, pp. 625–660, 2010.

[20] Diederik P Kingma and Jimmy Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[21] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, 2012, pp. 1097–1105.

INCREASING DRUM TRANSCRIPTION VOCABULARY USING DATA SYNTHESIS

Mark Cartwright

Music and Audio Research Lab
New York University
New York, New York
mark.cartwright@nyu.edu

Juan Pablo Bello

Music and Audio Research Lab
New York University
New York, New York
jpbello@nyu.edu

ABSTRACT

Current datasets for automatic drum transcription (ADT) are small and limited due to the tedious task of annotating onset events. While some of these datasets contain large vocabularies of percussive instrument classes (e.g. ~20 classes), many of these classes occur very infrequently in the data. This paucity of data makes it difficult to train models that support such large vocabularies. Therefore, data-driven drum transcription models often focus on a small number of percussive instrument classes (e.g. 3 classes). In this paper, we propose to support large-vocabulary drum transcription by generating a large synthetic dataset (210,000 eight second examples) of audio examples for which we have ground-truth transcriptions. Using this synthetic dataset along with existing drum transcription datasets, we train convolutional-recurrent neural networks (CRNNs) in a multi-task framework to support large-vocabulary ADT. We find that training on both the synthetic and real music drum transcription datasets together improves performance on not only large-vocabulary ADT, but also beat / downbeat detection small-vocabulary ADT.

1. INTRODUCTION

Automatic Drum Transcription (ADT) is the task of creating a symbolic score of the percussion instrument events within an audio recording of a musical piece. It is a subtask within Automatic Music Transcription (AMT), which aims to create a symbolic score of all the events within a musical piece. With accurate AMT, tens of millions of musical recordings could be indexed, compared, recommended, mined, and studied at scale using familiar musical concepts like pitch, harmony, rhythm, meter, and tempo. Accurate ADT could also aid in the development of generative rhythm models, descriptive models of rhythmic style, and intelligent digital audio effects that are informed by transcription and style.

ADT researchers typically simplify this problem by focusing solely on detecting the onset time and instrument class of all the notes sounded by percussion instruments in the signal. And very often, they simplify it even further by limiting the problem to only the notes sounded by the bass drum (BD), snare drum (SD), and hi-hat (HH) [1, 2, 3, 4, 5, 6]—a limit that greatly decreasing the utility of these systems. This is due to the tedious and time consuming nature of annotating recordings, which results in ADT datasets that are small and limited, often consisting of just a couple of hours of audio [1, 7, 8, 9]. In addition, while some datasets have annotations of a large number of percussion instrument classes [8, 7], others are limited to the BD, SD, and HH classes [9, 1], and datasets that do have larger vocabularies have minimal examples of these extended classes. The recent successful ADT algorithms have used deep learning architectures incorporating forms of recurrent neural networks [5, 4, 1, 2]. While powerful, these models require training on many audio examples in order to generalize well, making it even more difficult to expand the vocabulary of ADT.

Our goal is to learn ADT models that support a large vocabulary of percussion sounds. To address this, we generated a synthetic dataset that is 126 times larger than four of the most popular drum transcription datasets combined. We constructed this dataset using a large collection of MIDI drum loops, a large collection of drum hit recordings, and non-rhythmic harmonic backgrounds. We then trained and evaluated a multi-task convolutional-recurrent neural network (CRNN) drum transcription model using both the synthetic data and existing real music datasets. However, there is a risk in training with synthetic data—it may not be reflective of the same distribution as "real music", and therefore it may not generalize to it. Despite this, our intuition behind synthesizing and training with such a dataset is that it would expose the model to a wide variety of plausible percussion timbres and rhythmic variations allowing it to generalize to more unseen data.

In this work, we investigate the utility of synthetic data for ADT. Furthermore, to make full use of the real music datasets on the ADT task, our model follows a multi-task learning paradigm. However, both data synthesis and multi-task learning are methods to mitigate the problem of data paucity. Therefore, we also investigate the utility of multi-task learning in ADT when used in combination with synthetic training data.

2. RELATED WORK

As in many domains, there has been a recent shift to solving automatic drum transcription (ADT) with deep learning [1, 2, 5, 4]. Earlier approaches to ADT often fell into one of three categories defined by Gillet and Richard: *segment and classify*, *match and adapt*, *separate and detect* [6]. These approaches often combined multiple machine learning techniques such as support vector machines (SVM), hidden markov models (HMM), and non-negative matrix factorization (NMF) [10, 11]. While these models could perform well on solo drums [11], they often failed in the presence of polyphonic music [12]. Recent approaches that use deep learning incorporate variants of recurrent neural networks (RNNs) and are more robust than their predecessors in the presence of polyphonic music [1, 2, 4, 5]. However, deep architectures require many audio examples to generalize, and therefore due to the limited amount of annotated data, these recent approaches have limited their vocabulary of drum voices to the commonly occurring *bass drum* (BD), *snare drum* (SD), and *hi-hat* (HH). Recently, Wu and Lerch proposed to address data paucity in ADT with a student-teacher learning paradigm that utilizes unlabeled audio data. However, they still limited their work to BD, SD, and HH.

The field of computer vision also has many tasks which do

not have enough annotated data to adequately train deep learning models. Some researchers have addressed this problem by generating synthetic datasets with 3D-modeling [13, 14]. Ros et al [13] addressed the problem of urban scene segmentation by generating images from synthetic worlds using a game development platform. They trained their models on a combination of real and synthetic data. Peng et al [14] addressed the problem of object detection using crowdsourced 3D CAD models to generate images. By using synthetic data, they were able to explicitly teach the model to learn invariances to specific transformations.

In machine listening, several researchers have used synthetic data to train models, including for percussion informatics tasks. Van Steelant et al [15] synthesized data using MIDI files and drum sample libraries for the percussive sound classification task. Helen and Virtanen [10] similarly synthesized data for the drum source separation task. Thompson et al [16] synthesized drum patterns for tackling the ADT task using a bar-level classification approach with mel-frequeny cepstral coefficients (MFCCs) and an SVM. While their method used a vocabulary of 6 percussion voices, it had difficulty detecting these voices in the presence of equally mixed polyphonic accompaniment (f-measure 0.48).

Recently, models incorporating deep multi-task learning [17] have achieved state-of-the-art performance on multiple music information retrieval tasks [1, 18]. McFee and Bello [18] used a structured representation of chord qualities along with a multi-task model for large-vocabulary chord recognition. Vogl et al [1] jointly trained a multi-task CRNN for ADT and beat tracking. While they obtained state-of-the-art results, they found minimal improvement jointly training the beat detection task, but they did see improvement when incorporating oracle beat features. Since they had a minimal amount of data with annotated beats, increasing the amount of training data may be helpful in this scenario. In this work, we build upon the results of Vogl et al and use a similar CRNN model for multi-task learning, but we train our model with an abundance of synthetic data.

3. METHODS

3.1. Task Definition

In this work, we define "small-vocabulary" drum transcription as the task of transcribing the onsets of 3 percussion voices: *bass drum* (BD), *snare drum* (SD), and *hi-hat* (HH). We define "large-vocabulary" drum transcription as the task of transcribing the onsets of the following 14 percussion voices which are commonly found in drum sample libraries: *bass drum*, *snare*, *snare (rim)*, *low tom*, *mid tom*, *high tom*, *open hi-hat*, *closed hi-hat*, *ride cymbal*, *crash cymbal*, *conga*, *hand clap*, *clave*, and *bell*.

3.2. Drum Transcription Datasets

To combat the problem of data paucity in large-vocabulary ADT, we use a combination of existing real music datasets (3.69 hours of data) along with a new synthetic dataset of 210k eight-second audio examples (467 hours). The tasks for which these datasets have annotations varies. Some have 3-voice drum annotations and others have annotations for more voices that must be mapped to our 14-voice task. Some of the datasets also have beat annotations. See Table 1 for details about the datasets.

Table 1: *Summary information on the datasets used in this work.*

	RBMA [1]	IDMT SMT [8]	ENST [9]	MDB [7]	SDDS (3.2.1)
Hours	1.67	0.51	1.28	0.23	467
Solo Drums	No	No	Yes	No	No
Onsets	24k	9k	25k	8k	14853k
14-voice	No	No	Yes	Yes	Yes
3-voice	Yes	Yes	Yes	Yes	Yes
Beat	Yes	No	No	No	Yes

3.2.1. Synthetic Drum Dataset (SDDS)

To generate synthetic data, we rendered 60k audio examples from a collection of MIDI drum loops using randomly selected drum samples from a sample library. These examples were then augmented by adding harmonic background noise, stochastic noise, and small pitch shifts, bringing the total number of audio examples to 210k.

The examples were constructed in the following manner. From a release of eight sample libraries[1], we collected all of the one-shot drum samples that were labeled as *bass drum*, *snare*, *snare (rim)*, *tom*, *open hi-hat*, *closed hi-hat*, *ride cymbal*, *crash cymbal*, *conga*, *hand clap*, *clave*, and *bell*. These samples were a mixture of both electronic- and acoustic-sounding drums. We did not have consistent specific tom labels, so we split the tom recordings based on the sum of their median pitch [19] rank and median spectral centroid rank into *low toms* (0–25 percentile), *mid toms* (35–65 percentile), and *high toms* (75–100 percentile). The largest, most diverse of the sample libraries[2] was set aside as the test/validate set, and the remaining libraries were used for the train set. The exception to this statement is the ride cymbals—the test set did not have any recordings labeled as ride cymbals. To compensate, we divided the ride cymbals between the two sets. All of the drum hit recordings were resampled to 44.1 kHz and peak RMS normalized. The pitch, spectral centroid, and RMS energy were computed on frames of size 1024 with 50% overlap. This overall process resulted in 3758 recordings in the train set and 2053 recordings in the test/validate set.

Next, 60k (50k train / 5k validate / 5k test) MIDI drum loops were sampled from the freely available Drum Percussion Midi Archive (800k)[3]—an archive of 800k MIDI drum loops scraped from public web sites, much like how the content of the Lakh MIDI dataset was acquired [20]. We sampled these loops by randomly selecting one measure from a random MIDI file in the collection with the constraints that the measure was less than 3 seconds long and contained at least 3 percussion instruments. The measure was then looped and processed to be 8 seconds in duration with the first downbeat occurring at a random offset from the beginning of the file. Each MIDI loop was then rendered to audio by creating a separate "track" for each percussion voice, randomly selecting a drum sample for each percussion voice, and placing them in the track "monophonically" (i.e. drum hits for the same voice never overlapped) at each note onset time. The training set loops were rendered with the training set of drum samples, and both the validate and test loops were rendered with the testing/validation set of

[1] `http://musician.givegetwin.com/drum-heaven/`
[2] *Wave Alchemy - Drum Tools 01 Deluxe*
[3] `https://goo.gl/GhV7pc`

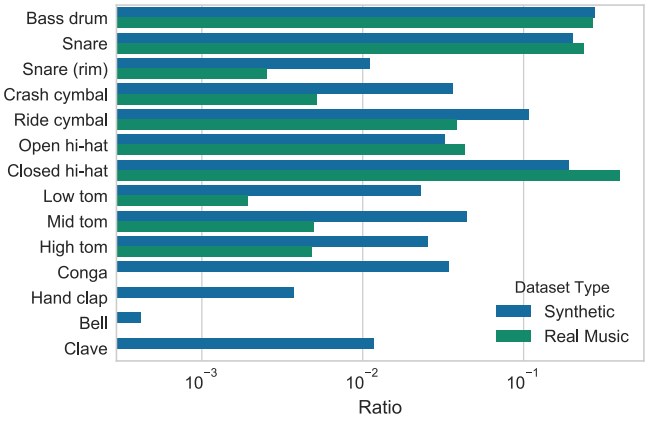


Figure 1: *Distribution of drum onsets in both the Real music (i.e., RBMA13, ENST-Drums, IDMT-SMT, MDB-Drums) and Synth (i.e., SDDS) dataset groups. Note the log-scale of the x-axis.*

drum samples. When rendering the loops, we scaled the amplitude of each drum sample by MIDI velocity as recommended in [21]:

$$r = 10^{r_{\text{dB}}/20} \tag{1}$$

$$b = \frac{v_{\max}}{(v_{\max} - 1)\sqrt{r}} - \frac{1}{v_{\max} - 1} \tag{2}$$

$$m = \frac{1 - b}{v_{\max}} \tag{3}$$

$$a = \begin{cases} (mv + b)^2 & \text{if } v > 0 \\ 0 & \text{otherwise} \end{cases} \tag{4}$$

where v is the note's MIDI velocity in the range $[0, v_{\max}]$ with $v_{\max} = 127$, r_{dB} is the output amplitude range in dB (set to 60 dB), and a is the resulting amplitude. All percussion voice tracks were mixed together with equal weights.

To generate non-rhythmic background accompaniment without discernible onsets, we selected 20 recordings (10 train, 10 test) containing harmonic instruments from MedleyDB [22]. Each recording was "smeared" in time by processing it with the Pysox [23] reverberator both forward and backward with randomly selected reverberation settings in a range to produce long-tail reverberation. This was repeated 12 times for each recording, each time pitched up an additional semitone, to produce 120 training backgrounds and 120 test/validate backgrounds. All files were trimmed to 30 seconds.

Using the MUDA data augmentation library [24], we then augmented the rendered training set of drum loops by a factor of four by varying both the background and the pitch, each with two variants. The goal of augmentation was to help the model learn invariances to different backgrounds and small changes in pitch. For each file, we selected two random 8 s background segments with random mixing coefficients in the range $[0.01, 0.5]$, and we selected two random semitone pitch shifts sampled from $N(0, 0.05)$. Lastly, to add robustness to noise, we also added white noise with a random mixing coefficient in the range $[0.01, 0.1]$. We repeated this process for the training and validation sets, but we only used one variant of each augmentation step, which resulted in only one processed recording per drum loop. The unprocessed drum loops were discarded. The augmentation process increased the size of the training set from 50k to 200k, while both the testing and validation sets remained at 5k. We will refer to this dataset as the Synthetic Drum Dataset (SDDS).

3.2.2. *Real Music Datasets*

We also trained and evaluated on four standard drum transcription datasets: RBMA13 [1], IDMT-SMT [8], ENST-Drums [9], and MDB-Drums [7].

RBMA13 [1] is a dataset of 27 fully-produced music tracks in the genres of electronic dance music (EDM), singer-songwriter, and fusion-jazz. It contains both annotated drum onsets and beat / downbeats. While there are several classes of percussion sounds in the recordings, only the BD, SD, and HH are annotated. We used the dataset's 3 predefined cross-validation splits.

IDMT-SMT [8] is a dataset of solo drum recordings consisting of BD, SD, and HH sounds from acoustic drum kits (10 different kits), drum synthesizers, and drum sample libraries. The dataset contains both recordings of isolated, single drum sounds and also recordings of rhythmic patterns using multiple drums. In this work, we only used the recordings of the rhythmic patterns. We randomly split the dataset into 3 cross-validation splits.

ENST-Drums [9] is a dataset of recordings from 3 drummers with different drum kits. It contains drum onset annotations for 20 different classes of percussion sounds. Of these 20 classes, we mapped 11 down to 10 of the classes in our vocabulary. The 9 remaining classes that were out of our vocabulary (e.g., *brush sweep*, *Chinese ride cymbal*) were ignored. While the dataset also contains many solo drum recordings, we only used the subset of recordings with accompaniment. The accompaniment and drums were summed together to create polyphonic mixtures. We used the splits of the drummers for our 3 cross-validation splits.

MDB-Drums [7] is a set of 23 fully-produced music tracks from the MedleyDB dataset [22]. It contains 6 classes of percussion sounds (*bass drum*, *snare drum*, *hi-hat*, *tom*, *cymbal*, *other*) with 21 subclasses (e.g., *snare drum: drag*, *open hi-hat*). Of these 21 subclasses, we mapped 17 down to 9 of the classes in our vocabulary and ignored the remaining 4 classes. We used the dataset's three predefined cross-validation splits.

3.3. Multi-Task Model

In this work, we used a convolutional-recurrent neural network (CRNN) model, very similar to the current state-of-the-art ADT model published in [1]. The model is constructed of 4 blocks of components as described in Table 2 and visualized in Figure 2.

We want to fully utilize the ADT and beat annotations in the real music datasets, but some datasets have 3-voice annotations and others have a larger number of voices that we reduced to our 14-voice vocabulary. Rather than trying to map the 3-voice annotations *up* to 14-voices and dealing with class assignment ambiguity, we instead treat it as two separate tasks—3-voice transcription and 14-voice transcription—and map the 14-voice annotations *down* to 3-voices (i.e., grouping the specific snare and hi-hat annotations, keeping bass drum as is, and ignoring the rest). To support all of these tasks, we designed our model as a multi-task with different outputs and losses for three tasks: 14-voice drum transcription, 3-voice drum transcription, and beat / downbeat detection.

The model receives two feature types as inputs in the **input block**. The first is the log-magnitude, log-frequency short-time Fourier transform (Logf-STFT). To compute this feature, we first

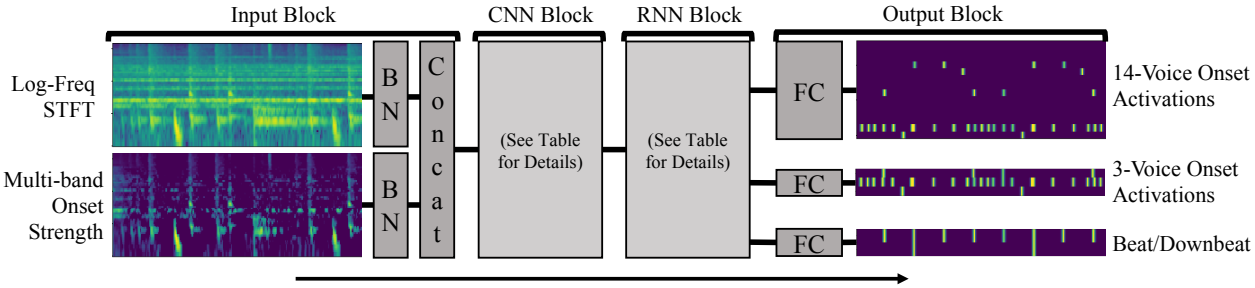


Figure 2: *High-level architecture of multi-task convolutional-recurrent neural network model. See Table 2 for details.*

Table 2: *Detailed architecture of multi-task convolutional-recurrent neural network model. Components that occur in parallel at the same model depth are presented in the same row. The parentheses on the right-hand side of each cell indicate the output size for that component.*

In Block	**Logf-STFT** (799, 64)	**Logf-Onset** (799, 64)	
	BatchNorm (799, 64)	BatchNorm (799, 64)	
	Stack (799, 64, 2)		
CNN Block	32 (3x3) Conv (799, 64, 32)		
	32 (3x3) Conv (799, 64, 32)		
	BatchNorm (799, 64, 32)		
	ReLU (799, 64, 32)		
	30% Dropout (799, 64, 32)		
	64 (3x3) Conv (799, 64, 64)		
	64 (3x3) Conv (799, 64, 64)		
	BatchNorm (799, 64, 64)		
	ReLU (799, 64, 64)		
	30% Dropout (799, 64, 64)		
	64 (1x64) Conv (799, 1, 64		
	BatchNorm (799, 1, 64)		
	ReLU (799, 1, 64)		
RNN Block	(-6:+6) Context Windowing (799, 832)		
	64 BLSTM (799, 128)		
	64 BLSTM (799, 128)		
	64 BLSTM (799, 128)		
Out Block	14 FC (799, 14)	3 FC (799, 3)	2 FC (799,2)
	14-voice	**3-voice**	**Beats**

Figure 3: *Round robin sampling for training. S is the synthetic dataset and RN are the real datasets: 1:RBM-13, 2:IMDT-SMT, 3:ENST, 4:MDB*

resample the audio to 22050 Hz and peak normalize it. We then compute the linear-frequency STFT on 1024-sample frames with a ~10 ms (221 sample) hop size. The magnitudes of the linearly-spaced frequency bins are then grouped into log-spaced bins using triangular frequency-domain filters—8 octaves of 8 bins per octave, starting at 40 Hz (i.e, 64 bins). We then log-scale these features. The second input is a multi-band onset signal computed from the Logf-STFT features before we log-scale magnitude. For each frequency bin, we compute the difference function between the current frame and the mean of the previous 22 frames. We then half-wave rectify and log-scale the signal. In the model, these features are batch-normalized [25] and concatenated on top of each other, creating a 2-channel input to the CNN block.

The purpose of the **CNN block** is to model the timbral characteristics of drum onsets. Described in detail in Table 2, this block consists of 3 stacks of convolutions, batch-normalization, rectified linear unit (ReLU) activations, and dropout [26] ($rate = 0.3$) layers. Each of the first two stacks uses 2 layers of 3×3 convolution filters, padded so the output is the same size as the input. The final stack uses one layer of 1×64 convolution filters (without padding) and does not include dropout. This block maintains the temporal resolution of the input.

The purpose of the **RNN block** is to model the temporal dynamics of drum onsets. This block consists of a stack of three bidirectional long short-term memory (BLSTM) [27] components. To more explicitly model the context, the input to the first BLSTM is padded and each temporal frame is concatenated with the 6 previous and 6 subsequent frames.

Each output frame of the RNN block is fed into three task-specific fully-connected (FC) layers in the **output block**: 14-voice transcription, 3-voice transcription, and beat / downbeat detection. In multi-task learning, this architecture is described as "hard parameter sharing" in which tasks share parameters for several layers followed by task-specific layers [17]. The model outputs and training data are encoded as multi-label binary activations with the same temporal resolution as the input (see Figure 2)—i.e, for each temporal frame, a class (i.e, percussion voice or beat / downbeat) bin is 1 if it contains an onset event, and 0 otherwise. The loss for each output is computed using binary cross-entropy.

3.4. Training

For the experiments in this paper, our model was implemented in Keras [28] and was trained on 8 s examples. We optimized using Adam [29] with gradients clipped at 1.0, a learning rate of 0.001, $\beta_1 = 0.9$, $\beta_2 = 0.999$, and $decay = 0$. We used batch sizes

of 8 and an "epoch" size of 2000. We reduced the learning rate on plateau (patience=5), and we trained for a maximum of 100 epochs with early stopping set to a patience of 25 epochs.

3.4.1. Task weights

With multiple outputs and loss functions, the optimizer minimizes a weighted combination of the losses:

$$L = \gamma_0 L_0 + \gamma_1 L_1 + \gamma_2 L_2 \tag{5}$$

where γ_i are the loss weights. Tasks with sparse output targets can achieve a low loss by simply predicting constant output. Therefore, if losses are equally weighted, training heuristics such as early-stopping and learning rate reduction will be dominated by tasks with denser outputs. In our training data, the distribution of the 14-voice transcription task is much sparser than the others, resulting in a small loss. To remedy this, we weighted the tasks by the inverse estimated entropy of their event activity distribution:

$$p_i = \frac{n_{events}}{n_{classes} \times n_{timesteps}} \tag{6}$$

$$\gamma_i = \left(-p_i \log(p_i) - (1 - p_i) \log(1 - p_i)\right)^{-1} \tag{7}$$

We estimated γ_i from the empirical distribution of events in the training set for each task, and weighted the task losses 0.53, 0.16, and 0.31 respectively for 14-voice transcription, 3-voice transcription, and beat / downbeat detection.

In addition, a task loss was masked for training examples without annotations for that particular task. This enables us to train a multi-task model with incomplete data.

3.4.2. Sampling

To have a numerically stable loss when training with incomplete data, the data has to be sampled such that all tasks are represented in each batch. Therefore, a simple random sampling procedure is not adequate. To accommodate this constraint, we utilize a round-robin sampling procedure using Pescador [30] as shown in Figure 3. This sampling procedure cycles through the real music datasets. Each time it samples from a real music dataset, it randomly selects an 8 s time interval from the dataset. Each time it samples from the synthetic music dataset, it randomly selects an 8 s example from the dataset. When training with both real and synthetic data, every other sample is from the synthetic dataset. This helps prevent overfitting due to the large quantity of synthetic data. The round-robin sampling procedure ensure that all active datasets will be present in a single batch, and therefore all tasks will be present as well. However, since the datasets vary in size, examples in smaller datasets will be presented more frequently to the model. This sampling procedure was used in both training and the calculation of the validation loss. However, when testing, we did not use this procedure—outputs were predicted for each example once, using the entire duration of the signal, i.e., the full duration of a music track rather than a 8 s time interval.

3.5. Experiments

To evaluate the effectiveness of our synthetic training data, we trained our model with three variations of training data:

1. **Real**: the real music dataset group (RBMA13, IDMT-SMT, ENST, MDB-Drums)
2. **Synth**: the synthetic dataset (SDDS)
3. **Real + Synth**: and the combination of both the real music dataset group and the synthetic dataset.

To balance the task in each batch, we used the round robin sampling scheme described in Section 3.4.2. When training with the real music datasets, we used the cross-validation splits as noted in Section 3.2.2 for training. To do so, we grouped the corresponding splits of the datasets together, e.g. the first splits from each dataset were grouped together when determining training and validation sets. For validation and testing, we further partitioned the data, using 25% of each split for validation and 75% for testing. In contrast, the synthetic dataset had one large training set and smaller validation and test sets. For consistency with the *Real* and *Real + Synth* variants, training with the *Synth* variant was also repeated three times with different random samples.

Furthermore, to investigate if the addition of large amounts of synthetic data could benefit from a larger model, we also varied the capacity of the model. We trained a "small" model as described in Table 2, and we also trained a "large" model. In the large model, convolution layers that originally had 32 filters were increased to 128 filters, and the number of units in the BLSTM components was increased from 64 to 256.

As noted earlier, we trained our models with a multi-task learning paradigm to make use of the *Real* datasets that only have small-vocabulary annotations. However, both data synthesis and multi-task learning can be viewed as methods to mitigate the problem of data paucity. To investigate how multi-task learning affects the ADT task in combination with data synthesis, we also trained single task models for comparison. These models used the small-capacity configuration and were only trained on *Real + Synth* data. We again evaluated them on *Real* and *Synth* data separately.

For each variation, we trained three models, one for each validation split. For each split, we selected the model with the lowest validation loss for evaluation. To evaluate our model outputs, we estimated the locations of onsets and beats from the output activations using the peak picking method described in [31], in which an output activation sample is selected as a peak if it meets the following criteria:

1. $x(n) = max(x(n - m_{pre}), \ldots, x(n + m_{post}))$
2. $x(n) \geq mean\left(x(n - o_{pre}), \ldots, x(n + o_{post})\right) + \delta$
3. $n - n_{lastOnset} > w$

where $x(n)$ is an output activation, n is the index of the current sample, $n_{lastOnset}$ is the index of the last identified onset, δ is a threshold parameter, w is the minimum number of samples between onsets, and m_{pre}, m_{post}, o_{pre}, o_{post} are sample offset values that define the window over which the max and $mean$ functions are computed. We tuned peak parameters for each model and task combination using a randomized search with 500 iterations scored on the validation set.

The resulting models were separately evaluated on both the test set of their corresponding split's real music dataset group and of the synthetic dataset. We used a 50 ms evaluation window in all experiments.

4. RESULTS

Table 3 presents the results of the experiments described in Section 3.5. Each value in the table is a mean metric (f-measure, precision, and recall) averaged over the test sets for the three cross-validation splits. For the small capacity variants evaluated on the

Table 3: *Mean model performance (over CV splits) for all three tasks evaluated on both real music and synthetic datasets while varying the distribution of training data, model capacity, and tasks.* ***F****: f-measure,* ***P****: precision,* ***R****: recall. Bold items indicate best performance per column for each group of training variants. *Asterisk indicates best task performance on Real data across all variants.*

Learning Paradigm	Capacity	Eval. Data	Train. Data	14-voice Trans.			3-voice Trans.			Beat Detect.		
				F	P	R	F	P	R	F	P	R
Multi-task	Small	Real	Real	0.58	0.55	0.61	0.69	0.64	0.74	0.62	0.68	0.58
			Synth	0.43	0.45	0.42	0.61	0.57	0.67	0.61	0.59	0.64
			Real + Synth	**0.68**	**0.67**	***0.70***	***0.77***	***0.77***	**0.77**	**0.74**	**0.74**	**0.74**
		Synth	Real	0.47	0.51	0.43	0.61	0.60	0.62	0.58	0.56	0.60
			Synth	**0.74**	**0.76**	0.72	**0.85**	**0.84**	0.86	**0.70**	0.68	**0.75**
			Real + Synth	0.70	0.64	**0.77**	0.84	0.81	**0.87**	0.69	**0.72**	0.68
	Large	Real	Real	0.63	0.63	0.63	0.72	0.69	0.76	0.57	0.59	0.55
			Synth	0.44	0.42	0.45	0.62	0.58	0.68	0.63	0.58	0.69
			Real + Synth	**0.70**	**0.73**	**0.68**	**0.76**	**0.74**	**0.78**	***0.75***	***0.75***	***0.75***
		Synth	Real	0.48	0.51	0.46	0.63	0.64	0.63	0.60	0.55	0.65
			Synth	**0.77**	**0.78**	**0.77**	**0.87**	**0.86**	**0.87**	**0.75**	0.68	**0.84**
			Real + Synth	0.72	0.71	0.74	0.83	0.81	**0.87**	0.69	**0.72**	0.69
Single-task	Small	Real	Real + Synth	*0.72*	*0.74*	0.69	-	-	-	-	-	-
		Synth	Real + Synth	0.69	0.68	0.71	-	-	-	-	-	-
		Real	Real + Synth	-	-	-	*0.77*	0.74	*0.81*	-	-	-
		Synth	Real + Synth	-	-	-	0.82	0.78	0.87	-	-	-
		Real	Real + Synth	-	-	-	-	-	-	0.64	0.59	0.69
		Synth	Real + Synth	-	-	-	-	-	-	0.65	0.61	0.71

Real test set, the model trained on only *Synth* data performs worse than the model trained on only *Real* data—e.g., f-measure 0.43 vs 0.58 for 14-voice transcription. This is true across all three tasks, though the effect on beat detection is minimal. However, when models are trained on both the *Real* and *Synth* data together, we see a large improvement in performance on all three tasks—e.g., f-measure 0.68 for 14-voice transcription. This trend is present in all three tasks. This implies that both *Real* and *Synth* are useful and complementary when training ADT models. We speculate that the *Synth* data teaches the model a wider variety of percussion timbres, whereas the *Real* data teaches the model to ignore instruments not relevant to the ADT task. This conjecture seems to hold with respect to the models' performance when evaluated on the *Synth* data—the models trained on only the *Synth* perform better or equal to the models trained on both *Synth* and *Real* data.

To further understand the models' performance on the 14-voice transcription task, we also evaluated class-specific performance (see Figure 4). We find that when trained on only *Real* data, the model only predicts 3 classes with f-measure performance above 0.5—*bass drum* (0.68), *snare drum* (0.61), and *closed hi-hat* (0.62). Except for the *open hi-hat* (f-measure 0.11), the model fails to predict all other classes. When *Synth* data is added to the training set, the f-measure performance improves for the *bass drum* (0.73), *snare drum* (0.74), *closed hi-hat* (0.74), and *open hi-hat* (0.55). The model now also has non-zero f-measure for several classes on which the *Real*-trained model completely failed to predict—e.g., *crash cymbals* (0.08), *ride cymbals* (0.45), *low toms* (0.09), *mid toms* (0.05), *high and toms* (0.04). Unfortunately, the performance on all of these classes is quite low with the exception of the ride cymbals. Interestingly, the precision on the tom classes is much higher than the recall—while this may have several causes, one possible cause could be improperly tuned peak-picking parameters, indicating that we should investigate class-specific parameters. There are also several classes on which the model still completely fails—*snares (rim)*, *congas*, *hand claps*, *bells*, *claves*. However, if we revisit the class distribution of drum onsets for the *Real* data in Figure 1, we see that class performance is closely correlated to the data's class distribution. While this could be caused by high-variance performance estimates due to the rarity of these onset events in the evaluation data, if this were the case, one might expect the model to have a high performance for at least one such classes. For comparison, if we look at the model's performance when evaluated on *Synth* data, we see a significant increase in performance for all of these classes except for the *hand claps* and *bells*, which have the lowest representation in the *Synth* data. Therefore while the *Synth* training data may expose the model to a wider variety of timbres, we still need to expose the model to classes of interest in a musical context with other instruments—with its current architecture, the model does not seem to generalize this ability across classes.

From Table 3, we also see that the large capacity models have roughly the same performance as the small models (sometimes slightly worse, sometimes slightly better) when evaluated on *Real* data. Whereas, the large capacity models trained and evaluated on *Synth* data see a small but consistent boost in performance from the higher capacity. Therefore, while increasing model capacity may help when training and evaluating on an abundance of synthetic data from the same distribution, it does not seem to improve model performance when evaluated on real music data.

Lastly, the models trained separately on the 14-voice and 3-voice transcription tasks typically performed very similarly to their multi-task counterparts (f-measure ± 0.02) with the single-task 14-voice model performing a bit better on the *Real* data (f-measure 0.72 vs 0.68). In contrast, the single-task beat detection model

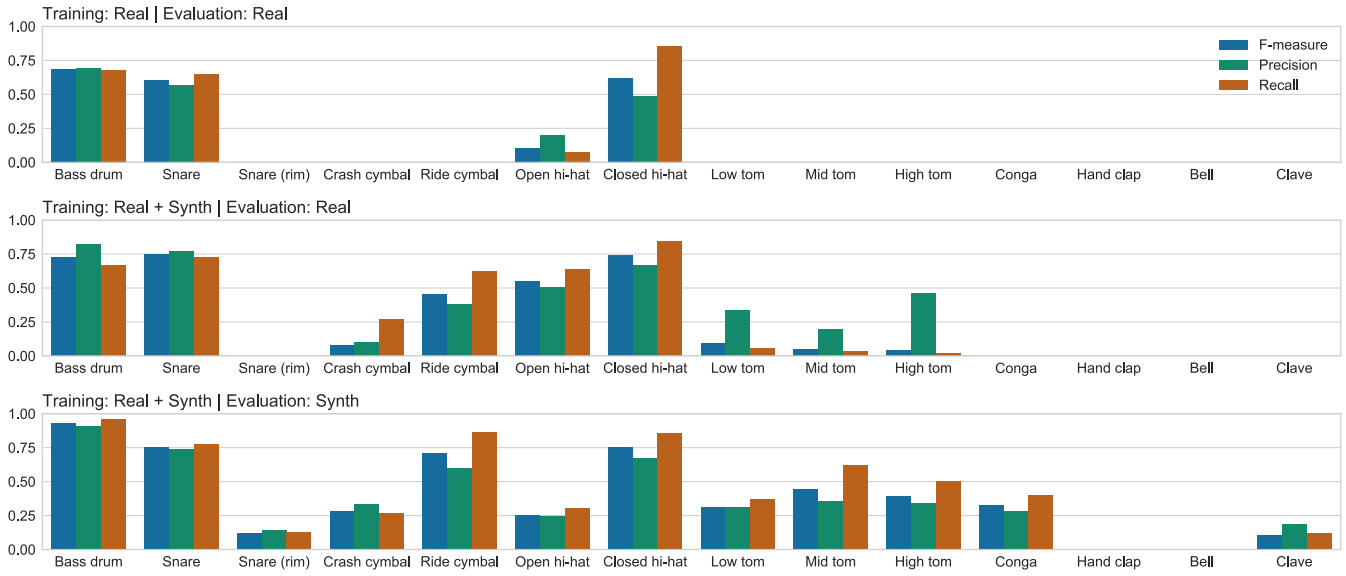


Figure 4: *Mean model performance (over CV splits) for 14-voice transcription performance broken down by class.* ***Top****: Model trained and evaluated on the real music datasets (RBM-13, IDMT-SMT, ENST, MDB).* ***Middle:*** *Model trained on both the real music dataset group and synthetic dataset (SDDS), and evaluated on the real music datasets.* ***Bottom:*** *Trained on both the real music datasets and synthetic dataset, and evaluated on the synthetic dataset.*

performed worse than its multi-task counterpart when evaluated on the *Real* data (f-measure 0.64 vs 0.74). Furthermore, we had the least amount of annotated *Real* data for the beat detection task (only RBMA-13 has beat / downbeat annotations). Thus, while multi-task learning in conjunction with synthetic data does seem to considerably aid on some tasks (e.g., beat / downbeat detection), this is not true for all tasks, and for 14-voice drum transcription it seems to actually hinder performance. These results seem to indicate that the benefit of synthetic data possibly overwhelms the benefit of multi-task learning for the ADT task.

5. CONCLUSION

In this work, we addressed the problem of data paucity for large-vocabulary automatic drum transcription (ADT) by generating a large synthetic dataset. We found that training with synthetic data can improve performance not only on ADT but also on beat detection. Improvements were observed on both 3-voice and 14-voice transcription tasks. On the 14-voice task, training with synthetic data increased performance for five classes on which the model without synthetic data training failed completely. Unfortunately, there is still a lot of room for improvement for 14-voice drum transcription. For synthetic data to help, it needs to be utilized in conjunction with real music data. In fact, it seems that it may need at least some minimum amount of annotated real music data in each class of interest, a problem that is exacerbated by the class imbalance in real music ADT training data. In our experiments we did not investigate what this minimum threshold is. However, these results imply that determining this minimum and focusing efforts to annotate up to this minimum for classes of interest could be a reasonable next step for improving large-vocabulary drum transcription. Another reasonable next step could be to resynthesize new percussion tracks with a variety of timbres for annotated datasets that have separate accompaniment and percussion tracks (e.g,, MDB-Drums). We also investigated the benefits of multi-task learning for ADT in combination with training on synthetic data. In our experiments, we trained both single task and multi-task models, and we found that multi-task learning potentially harmed the ADT task, but greatly benefited our auxiliary beat / downbeat detection task.

While the distribution of the full SDDS dataset is prohibitive due to its size, we have made both a small portion of the data available for download along with the trained multi-task and single-task models with highest performance on the *Real* datasets. Furthermore, we have also released a Python package to generate a similar dataset given collections of drum samples and MIDI files.[4]

In summary, data synthesis is a promising approach to combat the problem of data paucity in ADT and to increase the vocabulary size of ADT systems, but additional work is needed to investigate how to further improve performance on rare percussion classes. In addition, multi-task learning can also be a powerful tool to take advantage of limited training data, but its benefit is not consistent across tasks when combined with synthetic data. In future work, we hope to further investigate when training on auxiliary tasks is beneficial in music information retrieval.

6. REFERENCES

[1] Richard Vogl, Matthias Dorfer, Gerhard Widmer, and Peter Knees, "Drum transcription via joint beat and drum modeling using convolutional recurrent neural networks," in *Proc. of the Int'l Society for Music Information Retrieval Conf.*, 2017, pp. 150–157.

[4] https://github.com/mcartwright/dafx2018_adt

[2] Richard Vogl, Matthias Dorfer, and Peter Knees, "Drum transcription from polyphonic music with recurrent neural networks," in *Proc. of the Int'l Conf. on Acoustics, Speech and Signal Processing*. 2017, pp. 201–205, IEEE.

[3] Chih-Wei Wu and Alexander Lerch, "Automatic drum transcription using the student-teacher learning paradigm with unlabeled music data," in *Proc. of the Int'l Society for Music Information Retrieval Conf.*, 2017, pp. 613–620.

[4] Carl Southall, Ryan Stables, and Jason Hockman, "Automatic drum transcription for polyphonic recordings using soft attention mechanisms and convolutional neural networks," in *Proc. of the Int'l Society for Music Information Retrieval Conf.*, 2017, pp. 606–612.

[5] Carl Southall, Ryan Stables, and Jason Hockman, "Automatic drum transcription using bi-directional recurrent neural networks," in *Proc. of the Int'l Society for Music Information Retrieval Conf.*, 2016, pp. 591–597.

[6] Olivier Gillet and Gaël Richard, "Transcription and separation of drum signals from polyphonic music," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 16, no. 3, pp. 529–540, 2008.

[7] Carl Southall, Chih-Wei Wu, Alexander Lerch, and Jason Hockman, "Mdb drums–an annotated subset of medleydb for automatic drum transcription," in *Int'l Society for Music Information Retrieval Conf. Late-breaking and Demo Papers*, 2017.

[8] Christian Dittmar and Daniel Gärtner, "Real-time transcription and separation of drum recordings based on nmf decomposition," in *DAFx*, 2014, pp. 187–194.

[9] Olivier Gillet and Gaël Richard, "Enst-drums: an extensive audio-visual database for drum signals processing," in *ISMIR*, 2006, pp. 156–159.

[10] Marko Helen and Tuomas Virtanen, "Separation of drums from polyphonic music using non-negative matrix factorization and support vector machine," in *Proc. of the European Signal Processing Conf.* 2005, pp. 1–4, IEEE.

[11] Olivier Gillet and Gaël Richard, "Automatic transcription of drum loops," in *Proc. of the Int'l Conf. on Acoustics, Speech, and Signal Processing*. 2004, vol. 4, pp. iv–iv, IEEE.

[12] Olivier Gillet and Gaël Richard, "Drum track transcription of polyphonic music using noise subspace projection," 2005, pp. 92–99.

[13] German Ros, Laura Sellart, Joanna Materzynska, David Vazquez, and Antonio M Lopez, "The synthia dataset: A large collection of synthetic images for semantic segmentation of urban scenes," in *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition*, 2016, pp. 3234–3243.

[14] Xingchao Peng, Baochen Sun, Karim Ali, and Kate Saenko, "Learning deep object detectors from 3d models," in *Proc. of the Int'l Conf. on Computer Vision*. 2015, pp. 1278–1286, IEEE.

[15] Dirk Van Steelant, Koen Tanghe, Sven Degroeve, Bernard De Baets, Marc Leman, Jean-Pierre Martens, and T De Mulder, "Classification of percussive sounds using support vector machines," in *Proc. of the annual machine learning conference of Belgium and The Netherlands, Brussels, Belgium*. 2004, pp. 146–153, Citeseer.

[16] Lucas Thompson, Matthias Mauch, and Simon Dixon, "Drum transcription via classification of bar-level rhythmic patterns," in *Proc. of the Int'l Society for Music Information Retrieval Conf.*, 2014, pp. 187–192.

[17] Sebastian Ruder, "An overview of multi-task learning in deep neural networks," *arXiv preprint arXiv:1706.05098*, 2017.

[18] Brian McFee and Juan P Bello, "Structured training for large-vocabulary chord recognition," in *Proc. of the Int'l Society for Music Information Retrieval Conf.*, 2017, pp. 188–194.

[19] Alain De Cheveigné and Hideki Kawahara, "Yin, a fundamental frequency estimator for speech and music," *The Journal of the Acoustical Society of America*, vol. 111, no. 4, pp. 1917–1930, 2002.

[20] Colin Raffel, *Learning-based methods for comparing sequences, with applications to audio-to-midi alignment and matching*, Columbia University, 2016.

[21] Roger B Dannenberg, "The interpretation of midi velocity," in *Proceedings of the International Computer Music Conference*, 2006, pp. 193–196.

[22] Rachel M Bittner, Justin Salamon, Mike Tierney, Matthias Mauch, Chris Cannam, and Juan P Bello, "Medleydb: A multitrack dataset for annotation-intensive mir research," in *Proc. of the Int'l Society for Music Information Retrieval Conf.*, 2014, vol. 14, pp. 155–160.

[23] Rachel Bittner, Eric Humphrey, and Juan P Bello, "Pysox: Leveraging the audio signal processing power of sox in python," in *Int'l Society for Music Information Retrieval Conf. Late-Breaking and Demo Papers*, 2016.

[24] Brian McFee, Eric J Humphrey, and Juan P Bello, "A software framework for musical data augmentation," in *Proc. of the Int'l Society for Music Information Retrieval Conf.* 2015, pp. 248–254, Citeseer.

[25] Sergey Ioffe and Christian Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," *arXiv preprint arXiv:1502.03167*, 2015.

[26] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *The Journal of Machine Learning Research*, vol. 15, no. 1, pp. 1929–1958, 2014.

[27] Sepp Hochreiter and Jürgen Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[28] François Chollet, "Keras," `https://github.com/fchollet/keras`, 2015.

[29] Diederik P Kingma and Jimmy Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[30] Brian McFee, Eric Humphrey, and Christopher Jacoby, "Pescador," `https://github.com/pescadores/pescador`, 2016.

[31] Sebastian Böck, Florian Krebs, and Markus Schedl, "Evaluating the online capabilities of onset detection methods," in *ISMIR*, 2012, pp. 49–54.

AUTOMATIC DRUM TRANSCRIPTION WITH CONVOLUTIONAL NEURAL NETWORKS

C. Jacques

Analysis-Synthesis team,
STMS-UMR 9912, IRCAM, Sorbonne University, CNRS
Paris, France
celine.jacques@ircam.fr

A. Roebel

Analysis-Synthesis team,
STMS-UMR 9912, IRCAM, Sorbonne University, CNRS
Paris, France
axel.roebel@ircam.fr

ABSTRACT

Automatic drum transcription (ADT) aims to detect drum events in polyphonic music. This task is part of the more general problem of transcribing a music signal in terms of its musical score and additionally can be very interesting for extracting high level information e.g. tempo, downbeat, measure. This article has the objective to investigate the use of Convolutional Neural Networks (CNN) in the context of ADT. Two different strategies are compared. First an approach based on a CNN based detection of drum only onsets is combined with an algorithm using Non-negative Matrix Deconvolution (NMD) for drum onset transcription. Then an approach relying entirely on CNN for the detection of individual drum instruments is described. The question of which loss function is the most adapted for this task is investigated together with the question of the optimal input structure. All algorithms are evaluated using the publicly available ENST Drum database, a widely used established reference dataset, allowing easy comparison with other algorithms. The comparison shows that the purely CNN based algorithm significantly outperforms the NMD based approach, and that the results are significantly better for the snare drum, but slightly worse for both the bass drum and the hi-hat when compared to the best results published so far and ones using also a neural network model.

1. INTRODUCTION

Automatic music transcription is the task of describing a music signal in a symbolic form - a score - that contains all of the necessary information to replay the same music. Every event in a piece of music has to be characterized by musically relevant parameters like the pitch, time position, duration, and the instrument. Accordingly, the problem of music transcription can be divided into different challenges: onset detection, f0-estimation and instrument recognition. While the problem is considered as solved for monophonic signals, it is more challenging for polyphonic ones. The additivity of signals and the overlapping of partials of different notes make the task more and more complex as the number of sources increases.

A piece of music is generally performed by harmonic and percussive instruments. These instruments have different features. The spectrogram of a note is sparse in frequency, and a harmonic note has relatively few constraints with respect to its duration. On the contrary, a drum event covers a continuous part of the spectrum, but has a specific temporal response. Accordingly, different features are used to transcribe the different events. In this article we will focus on the automatic transcription of parts of the drum kit.

Automatic drum transcription is still a challenge today. Several methods have been proposed in literature and most of them can be categorised into two families: segment and classify or separate and detect. The first category segments the audio and then tries to describe what the audio segment contains. The second one separates different instruments and tries to detect onsets in the different channels.

In 2009, Paulus et al. proposed a method based on Hidden Markov Model (HMM) network in [1]. Recently, different deep learning methods have been proposed. Vogl et al. use a Recurrent Neural Network (RNN) which provides an activation function for the drum instrument (bass drum, snare drum and hi-hat) in [2]. The first study to use CNN for drum transcription has been performed in [3].

These different methods can be compared easily as most of them have been evaluated on the same database, the ENST drum database [4]. In light of the results, most DNN approaches seem to lag behind those using Hidden Markov Models (HMM) such as proposed in [1].

Automatic onset detection, which consists in locating the onsets of musical events in a piece of music, is an important initial step for efficient transcription. Onset detection is frequently used as a preprocessing step for more refined transcription, as used recently in [5] for piano transcription, and in [6] for drum transcription. A successful detection of all onsets significantly reduces the processing time of the subsequent transcription algorithm which does not need to be run over the complete signal.

There exists a large multitude of approaches that have been developed for the onset detection problem. Bello et al. provide a rather extensive overview of the various methods in [7]. The methods generally are variations of the following approach: after a pre-processing step, which highlights some properties of the signal facilitating the subsequent detection stage, the so called Onset Detection Function (ODF) is calculated. The local maxima of the ODF with a value above a threshold (which is a parameter of the algorithm) are then retained as onsets. Elowsson in [8] for example used the spectral flux, which is the difference of energy between the actual temporal frame and the previous one, to calculate the ODF. Many other approaches to calculate the ODF have been discussed in the literature.

Recently, onset detection methods based on deep learning have shown very good results. While some works aim to improve peak picking from an onset detection function as in [9], others use RNN (Recursive Neural Network) as in [10] to create the ODF. In 2014, Schlüter et al. investigated using CNN (Convolutional Neural Network) [11] for the onset detection task, and according to MIREX 2017[1] the CNN based onset detection can now be considered as state of the art. In [11] it is shown that the weights of the kernels of the convolutive layers that are used to detect percussive and

[1] http://nema.lis.illinois.edu/nema_out/mirex2017/results/aod/summary.html

harmonic onsets are rather different. This observation seems to suggest that these networks may not only be able to detect onsets, but to detect onsets for specific classes of instruments.

If we compare the CNN architecture used by Schlüter in [11] for general purpose onset detection and by Wang in [5] for piano onset detection, we find that the overall structure is very similar. However, they do not use the same data structure. Similarly how the RGB channels are accounted for in image processing, Schlüter uses as input three mel band spectrograms with the same number of bands but calculated from different STFT representations. On the contrary Wang uses just one constant Q spectrogram with a much larger number of bands.

The following paper aims to investigate the use of CNN for drum transcription. Two different approaches will be considered. First, we will use a CNN based onset detection as an initial step for subsequent drum transcription based on a recent method using non-negative matrix deconvolution [6]. Here we will introduce the new idea of a detection of qualified onsets meaning onsets fulfilling additional criteria - for example onsets belonging to percussive events or drum instruments. In developing the qualification of onsets further we will investigate a CNN based drum transcription where the CNN are trained to detect individual drum instruments. The later system has strong resemblance to the approach proposed in [3]. However, instead of training a multi label system that detects multiple instruments at the same time, we will separate the systems into individual drum instrument detectors. That allows us to investigate the optimal input representation for the different instruments. Instead of using the magnitude spectrogram data directly [3], we will use single and multi channel[2] mel band spectrogram data that has been introduced successfully for onset detection in [11]. We will compare two different cost functions. We will evaluate the final system using the ENST-Drums drummer that was left aside during training. That allows to compare our results with the various evaluations performed so far on the ENST-Drums database. We notably compare with results in [1] that to our knowledge are the best results reported so far. We also evaluate the available model of Southall [3] on the three drummers from ENST-Drums.

The article is organised as follows: Section 2 introduces the neural network and the different parameters to be compared, Section 3 shortly summarizes the NMD algorithm, Section 4 describes the experimental results, and finally Section 5 summarizes the conclusions and describes future work.

2. ONSET DETECTION AND COMPARISON OF CONFIGURATIONS

2.1. The CNN network

The model we use to compare different configurations is very similar to the one in [11, 5] and is represented in Figure 1. We summarize here the architecture of the network.

The input data contains mel frequency spectrogram data. The subsequent layers are alternating stacks of convolutional layers with ReLU activations and max-pooling layers. It finally ends with a fully connected layer of ReLU units and an output layer containing either a sigmoid unit or a linear unit. The output layer provides the ODF. The method then follows the standard approach to detect local maxima and uses a fixed threshold of 0.5 for the detection of onset in a given frame, which significantly simplifies the algorithmic design.

The feature maps at the output of these layers can be seen as a convolution between the input and a filter kernel. Usually in computer vision, the convolution is achieved with square filter. In time-frequency representation, the two dimensions represent two different quantities. As the aim of the network is to find changes over time dimension, it can be more interesting to use narrow rectangular filters frequency-wise and the max-pooling operations performed only on the frequency axis.

Following [11] we apply dropout with 50% drop out probability at the output of the first fully connected layer, to reduce overfitting during the training.

2.2. Parameter comparison

2.2.1. Loss function

The loss function used to direct the optimization of the neural network measures the divergence between the predicted value - the output of the network - and the target label. For onset detection, cross-entropy is commonly used because the task of detecting an onset in a frame has some relations to a binary classification task: frames containing an onset are marked as 1 and frames without onsets are marked as 0.

We note however, that the resemblance of the target ODF with a probability is only partially followed. As the CNN model is smooth in all parameters, the ODF function produced is smooth as well. Accordingly, a Dirac-impulse is difficult to produce, and therefore, similarly to [11], we will construct the target function by means of placing a sequence of three ones centered at the annotated onset. Broadening the target labels has the beneficial effect of increasing the pressure on the network to correctly represent the target labels, and at the same time reduces the problems of incoherent label positions. In our experiments we have seen that broadening the labels leads to reduced training times and slightly improved results. The use of a CNN as onset detector does not require the ODF to be confined to $[0, 1]$. This fact motivates us to compare two different cost functions combined with two corresponding output activation functions. On the one hand there is the binary cross entropy together with sigmoid activation function, and on the other hand the linear (ReLU) output unit with MSE loss function. We will discuss the results of the use of these two loss functions in the experimental section.

2.2.2. Input data structure

Kelz et al. in [12] compare the importance of hyper-parameters for piano transcription and they rank some hyper-parameters in respect to relative importance. The data representation is the second most important hyper-parameter. As a matter of fact, several different data representations are used as input data throughout the literature.

Schlüter et al. in [11] use three log-magnitude mel band spectrograms obtained with different time-frequency resolutions. They process the short time Fourier transform (STFT) with a hop size of 10 ms and window sizes of 23 ms, 46 ms and 93 ms. As the spectrograms must have the same size, they filter the spectrogram with an 80-band mel filter bank covering the band from 27.5 Hz to 16 kHz. We will subsequently denote this representation as multi

[2] the term channel will be used for the feature channels of a deep network in the following and has nothing to do with the channels of stereo audio signal

[3] `https://github.com/CarlSouthall/ADTLib`

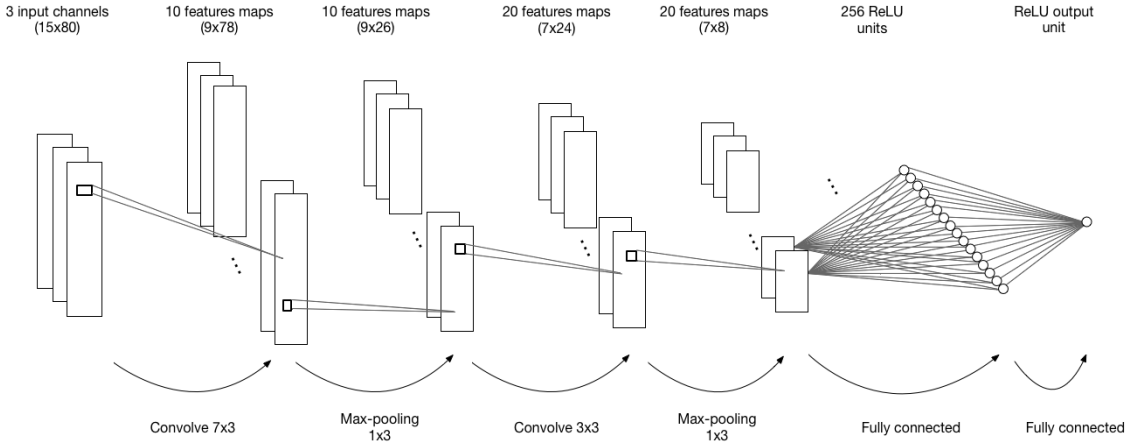


Figure 1: *Convolutional neural network used for this work.*

channel mel spectrogram (MCMS) where the term channel refers to the feature channel of a DNN.

On the contrary, Wang in [5] uses a single constant Q transform spectrogram. In this paper, we compare different data representations. We feed the eight networks with spectrograms with different resolution. Two STFT are processed with two different window sizes, 0.064 ms and 0.125 ms. Then for each spectrogram four mel spectrograms are calculated with triangular filters to compare four numbers of mel-bands: 116, 174, 231 and 289. We compare these mel spectrograms calculated from an individual STFT with the input representation proposed by Schlüter.

3. APPLICATION TO DRUM TRANSCRIPTION

We will use the CNN presented in this article in two ways to perform automatic drum transcription: combined with an ADT algorithm or alone.

As mentioned in the introduction we will investigate qualified onset detection with CNN with the objective to use these qualified onsets in the context of drum transcription. By "qualified onsets", we mean onsets that are created by one of the three targeted parts of a drum kits (hi-hat, bass drum and snare drum), either in collection (onset of any of these instruments) or individually.

In the first case, to achieve drum transcription, we combine the onset detection with a second stage to determine which of the three instruments have generated the onset. In the second case CNNs will perform the complete transcription task.

3.1. Combination of onsets detector with a drum transcription algorithm

The NMD algorithm for drum transcription we will use in the following is detailed in [6]. It decomposes the time-frequency representation of the audio signal into a convolution of a dictionary containing patterns of instruments and a matrix of activations.

For percussive instruments, the temporal response is a significant characteristic. This is the reason for using a dictionary of two-dimensional time-frequency patterns. These are previously learned from isolated events of each instrument.

The dictionary contains only patterns from drum instruments (hi-hat, snare drum and bass drum). But the drum transcription is processed on polyphonic music with harmonic instruments. To avoid the activation of drum patterns by other events, the decomposition includes some patterns in the dictionary dedicated to representing the non percussive part of the signal, which we call the background.

To reduce the computational costs, a prior knowledge of the onset position is given to the algorithm. An external algorithm, e.g. [8], feeds the transcription algorithm with the onsets that it detected. The transcription algorithm focuses on the parts of the signal that are around these positions. At these positions several instruments are likely to play. In that case, the segment study enables to separate them.

For each segment, the NMD algorithm aims to approach the studied spectrogram by activating some patterns from the dictionary. In order to, the dictionary of patterns, called W, and activations H are usually updated iteratively. For our algorithm, only background patterns in W are updated but all activations are concerned by the updating step. The update rules are calculated by minimizing a cost function, here the Itakura-Saïto divergence. As the background patterns are very flexible, they could in principle represent all parts of the signal under study. Therefore, it is important to penalize the algorithm for the use of background patterns. To this end the objective function

$$C = D_{IS}(V|\sum_t W^t H^{t\rightarrow}) + \lambda_{seg} P(H), \tag{1}$$

used for the decomposition contains a regularization term $P(H)$ that penalizes the use of background patterns. λ_{seg} is a weight to give more or less importance to the penalization.

To keep the non-negativity property, we use multiplicative updates:

$$W^p_{lb} \leftarrow W^p_{lb} \frac{\sum_n \frac{V_{ln} H_{p,n-b}}{V^2_{est_{ln}}}}{\sum_n \frac{H_{p,n-b}}{V_{est_{ln}}}} \tag{2}$$

$$H_{pn} \leftarrow H_{pn} \frac{\sum_f \sum_t \left(W^t_{fp} \frac{V_{f(n+t)}}{(V_{est}{}_{f(n+t)})^2} \right)}{\sum_f \sum_t \left(\frac{W^t_{fp}}{V_{est}{}_{f(n+t)}} \right) + \lambda_{hseg} \mathbb{1}_{p \in bg}} \tag{3}$$

$$\tag{4}$$

with p the number of patterns, l the frequency bin and n the time frame and with bg designating the background.

Once all segments are analyzed, we follow the procedure described in [6] to adapt the detection thresholds that are applied to the activation to retain onsets of the targeted instruments.

3.2. Using CNN to transcribe drum parts

We also can use CNN described in section 2.1 to transcribe one of the targeted instrument. Instead of training the network to detect qualified onsets, we train three individual networks so that each of them detects only one instrument.

4. RESULTS

4.1. Datasets

4.1.1. RWC dataset

The training database used to adapt the CNN is extracted from the Real World Computing (RWC) music database [13]. This database contains annotated polyphonic music of different styles in MIDI format. We choose two genres, Pop and Jazz and pick only pieces where drums are present. For Jazz, there are 34 pieces of music and 100 for the Pop database. Each piece was generated with three different publicly available MIDI sound fonts: FluidR3_GM, GuGS_1.47 and HQOrchestralSFCollv2.1.2. The training database finally contains 102 jazz pieces and 300 pop pieces. In addition, we add recordings of a the single targeted instrument. These recordings are given in SMT-Drums.

For some of the experiments, evaluation is performed using a small hold out test set containing four pieces from the synthetic RWC database described above.

4.1.2. ENST-Drums dataset

The ENST-Drums database [4] is composed of different multichannel recordings from three drummers on three different drum kits. For each drummer, the data set provides individual hits and phrases, individual soli which are more complex than the phrases and longer tracks played without scores but with an accompaniment. For these longer tracks, called 'minus-one', the accompaniment is provided with two mixes: "dry" where minimal effects are added and "wet" with effects and compression. The "wet" mix sounds closer to commercial recordings than "dry" mix does and we use the "wet" mix for the following evaluation.

We use the 'minus-one' tracks mixed with the synchronized accompaniment. As in [1], scaling factors are applied to the different parts: 2/3 for the drums and 1/3 for the accompaniment. The data set also provides the ground truth annotations for each percussive instrument. The test database contains 64 tracks (21 for two drummers and 22 for the last one) which last between 30 s and 75 s.

The evaluation is performed by using the drummer cross validation procedure on the ENST-Drums database [4].

4.2. Evaluation criteria

To evaluate the algorithms, the detected onsets are compared to the ground truth onsets. A detected onset is considered correct if the absolute time difference with the associated ground truth onset does not exceed 30 ms. We denote by Tp the true positives, correctly detected onsets, by Fp false positives, detected onsets which are not in ground truth annotations and by Fn false negatives, onsets present in ground truth annotation but not detected by the algorithm.

Several measures are calculated from these values. The precision P gives the part of detected onsets which is relevant and the recall R gives the part of relevant onsets which is selected. They are defined as:

$$P = \frac{Tp}{Tp + Fp} \quad R = \frac{Tp}{Tp + Fn} \tag{5}$$

The F-measure is a compromise between recall and precision:

$$F = \frac{2PR}{P + R} \tag{6}$$

4.3. Evaluation of onset detection for drum instruments

In the first part of the evaluation we will study the performance of the CNN onset detection algorithm for detecting specific onsets. In our case, this means the onsets of any of the targeted percussive instruments. The goal of this first step is to prepare the subsequent integration of the CNN onset detection algorithm as preprocessing step into the NMD drum transcription algorithm.

Following a general practice, we will evaluate the detection of the three main instruments of the percussive part: bass drum (BD), snare drum (SD) and the hi-hat (HH). These three instruments are predominant in popular music and are representative of the rhythmic feel in music.

4.3.1. Loss function

As discussed before we compare two loss functions along with adequate changes in the output activation function: binary cross entropy with sigmoid output units and mean square error with ReLU output units. Several networks are trained with different configurations. We study four numbers of mel-bands (116, 174, 231 and 289) and two sizes of STFT window (0.064 s and 0.125 s). We also give the results for the MCMS input data configuration detailed in 2.2.2. The networks are trained and evaluated to detect the onsets of any of the three targeted percussive onsets (hi-hat, bass drum and snare drum) in the RWC database detailed in section 4.1.1.

In Figures 2 and 3, the results obtained with binary cross entropy are consistently outperforming those that are obtained with mean square error. For the following comparison, we will therefore focus on the binary cross entropy. We note that the onset prediction performance of only the three target percussive instruments is encouraging with F-measure above 90% for all configurations. There is no apparent and significant difference between any of the input data structures.

4.3.2. Evaluation the influence of data structure on the ENST-Drums database

To improve the relevance of the evaluation for real world sounds we will now evaluate CNN drum detection approach on the recordings of the ENST-Drums database [4]. The evaluation follows the common three-fold cross-validation scheme with the three configurations of the 3 drummers of the ENST-Drums database 4.1.2. The networks are trained on two drummers of the dataset and tested on the remaining one. We use all pieces available in the

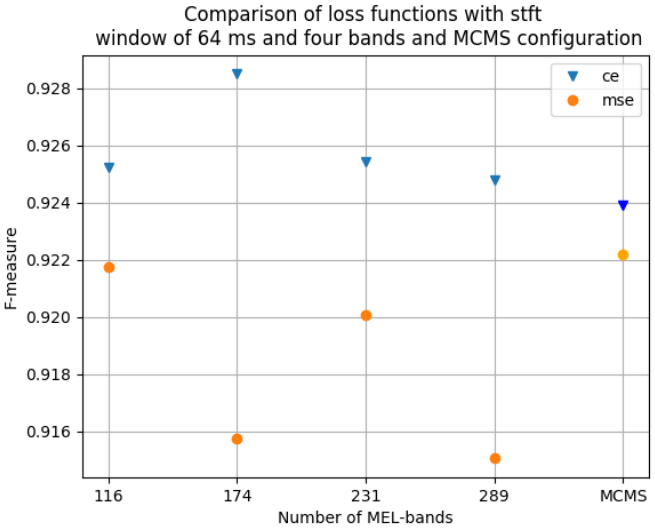


Figure 2: *Comparison of loss functions on RWC database: binary cross entropy and mean square error, for STFT window 0.064 s.*

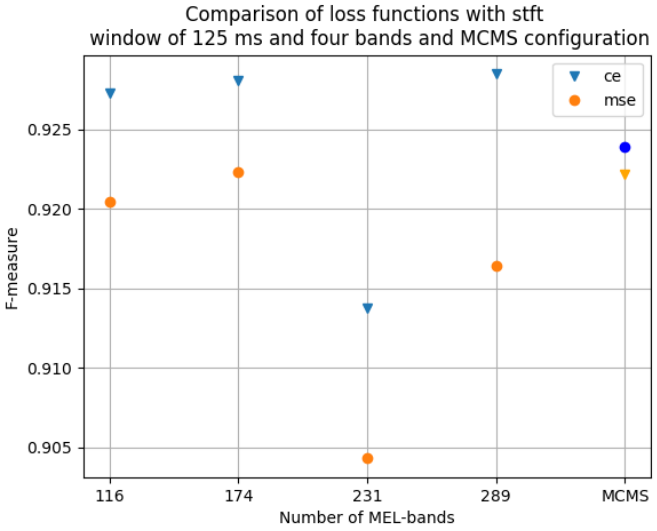


Figure 3: *Comparison of loss functions on RWC database: binary cross entropy and mean square error, for STFT window 0.125 s.*

data set for the learning phase, during which the evaluation is performed over the minus-one of the same drummers to determine the optimal result for drum detection (according to the F-measure). Then we test the generalization on the third drummer who was not used during training.

We compare the different data input configurations: two STFT window sizes 64 ms and 125 ms and four numbers of mel-bands 116, 174, 231 and 289 and the MCMS input representation. The Figure 4 averages the results over the three experiments for the detection of all percussive onset.

We notice that contrary to the evaluation with the RWC database, for the ENST-Drums database the use of the MCMS format (three spectrograms) seems to provide a significantly better results, improving the performance from 91.5% F-measure for the best single channel mel-band spectrogram to nearly 93.5% for the MCMS. While the MCMS representation was equivalent with the individual spectrogram formats on the RWC database, it is significantly better for the ENST-Drums database. That suggests the conclusion that the multiple time resolutions in the different channels of the MCMS lead to improved robustness of the final detection.

It is interesting to see to what extent the training of MCMS detector on specific onsets (the main three percussive instruments) does change its performance. To this end we use the MCMS detector provided by Schlüter in the madmom package [14]. We evaluate the two detectors on a different hold out test set of the RWC database and we find that the specific onset detector significantly improves the detection performance in F-measure from 86.8% for the general purpose onset detector to 93.2% for the percussive onset detector.

4.4. Application to drum transcription

Characterizing detected onsets might be advantageous for drum transcription. We investigate here two uses of the MCMS format for the drum transcription task. The first method combines the drum onset detector based on CNN with the ADT algorithm based on NMD described in 3.1. The CNN gives the drum onset positions and the NMD algorithm studies the segments around these positions to determine which percussive instruments provided the onset. The second one uses three individual CNNs. Each CNN is trained to detect one of the three main percussive instruments.

4.4.1. Drum onset detector combined to automatic drum transcription algorithm

Given the rather high performance of the drum onset detection algorithm, we are interested in seeing the effect of the specific onset detection when combined with an NMD based drum transcription algorithm [6]. We evaluate the performance on ENST-Drums dataset and present in Table 1 the average results on the three cross-validation experiments. We compare the obtained results with Paulus' and Southall's results. We evaluate the models by transcribing the drum parts for the 'minus one' pieces of ENST-Drums and perform the mean over the three drummers.

Table 1: *Results of transcription on three-fold cross validation.*

Methods	Metric	BD	SD	HH
HMM+	P(%)	80.2	66.3	84.7
MLLR [1]	R(%)	81.5	45.3	82.8
	F(%)	80.8	53.9	83.6
Soft Attention+	P(%)	98.5	88.2	67.8
mechanisms [15]	R(%)	62.2	40.1	87.9
	F(%)	72.0	53.7	76.4
NMD fed by	P(%)	79.6	68.8	72.6
drum onset	R(%)	64.7	43.9	67.1
detected by CNN	F(%)	68.9	52.6	68.3

Feeding the drum onsets to the NMD algorithm does not enable it to reach Paulus's or Southall's results.

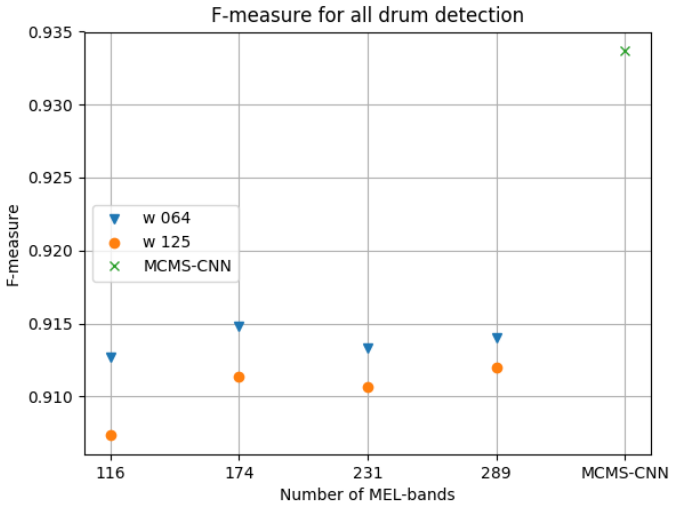


Figure 4: *Comparison different input configurations on percussive onset detection task on ENST-Drums dataset.*

4.4.2. *Individual CNNs trained on each drum instrument*

In a final experiment, motivated by the very good performance of the CNN based drum detection algorithm, we evaluate the CNN specific onset detectors trained to detect the individual drum instrument events. We therefore perform the complete transcription of an individual instrument. We focus this last experiment on the MCMS network which had the best performance in the previous experiments. Three independent CNNs are involved in this experiment. Each network is trained to detect only one of the three main percussive instrument. They are evaluated with the three-fold cross validation on the ENST-Drums database. The results averaged over the three folds of the cross validation are given in Table 2. The results of drum onset detection in 'all drums' are also displayed. They are obtained with the CNN trained to detect qualified onsets (without distinction between instrument) for our method. Southall's model does not provide those results.

Table 2: *Results of drum transcription per instrument on three-fold cross validation.*

Methods	Metric	BD	SD	HH	Percus.
HMM+	P(%)	80.2	66.3	84.7	79.0
MLLR [1]	R(%)	81.5	45.3	82.8	70.9
	F(%)	80.8	53.9	83.6	74.7
Soft Attention+	P(%)	98.5	88.2	67.8	-
mechanisms [15]	R(%)	62.2	40.1	87.9	-
	F(%)	72.0	53.7	76.4	-
CNN with	P(%)	77.5	57.9	71.0	93.7
MCMS	R(%)	75.0	67.0	89.7	93.0
configuration	F(%)	76.2	62.1	79.3	93.4

We notice that the CNN provides comparatively good results for the snare drum, for which it obtains 8pts more in F-measure than Paulus' method. But it also loses 4pts for the two other instruments, the bass drum and hi-hat. Our model is better than Southall's method.

Comparing the bass-drum results between the HMM and CNN methods we can find an explanation for the reduced performance in Table 3, which displays the results of the individual folds of the cross evaluation experiment. While the detection of drummer 3 and drummer 2 are performing very satisfyingly, the recall of drummer 1 is particularly low. Listening to the bass drum signals of the different drummers reveals that the bass drum signal of drummer 1 is clearly different from the two other drummers. Its energy is significantly lower in comparison to the bass drum signals of drummers 2 and 3. We have tried to counter this difference by means of using different mixes when training the network, without achieving any improvement. One can also observe that the bass drum signal of drummer 1 contains a much less pronounced onset, which may constitute another explanation for the low recall. Here, the high specificity of the CNN leads to an over-fitting of the training signals, which in turn reduces the recall for drummer 1. Although Southall's model seems to encounter the same problem, the HMM model displayed in Table 2 apparently does not have the same issue with drummer 1. It may indicate that the CNN model we chose and which worked very well for the general drum detection task, is too complex.

Table 3: *Results of bass drum transcription on three-fold cross validation.*

Train drummers	Eval drummer	P	R	F
1 and 2	3	82.5	96.7	89.1
2 and 3	1	75.1	36.7	45.0
3 and 1	2	74.6	98.1	84.8

An other idea to improve detection of bass drum played by drummer 1 is to normalize over time only. It highlights the sudden changes of energy which can be characteristic of onsets. However, this kind of normalization modify the relation of energy between the frequency bands. But as the energy of bass drum is located in low frequency bands, the networks is able to correctly detect the onsets. In fact, for drummer 1, the F-measure on drummer 1 for bass drum raises 67.7 % instead of 45%. The results for the other drum instruments and for the percussive instruments are given in Table 4. We compare the results with [1] and [15].

Table 4: *Results of drum transcription per instrument on three-fold cross validation with normalization over time.*

Methods	Metric	BD	SD	HH	Percus.
HMM+	P(%)	80.2	66.3	84.7	79.0
MLLR [1]	R(%)	81.5	45.3	82.8	70.9
	F(%)	80.8	53.9	83.6	74.7
Soft Attention+	P(%)	98.5	88.2	67.8	-
mechanisms [15]	R(%)	62.2	40.1	87.9	-
	F(%)	72.0	53.7	76.4	-
CNN with	P(%)	84.0	54.2	71.9	93.8
MCMS config.	R(%)	80.7	68.1	86.6	91.7
and tnorm	F(%)	81.5	59.4	77.8	92.7

The F-measure is slightly better for bass drum and significantly better for snare drum than F-measures obtained with the method of HMM. The detection of percussive onsets is also largely more effective. Although normalization over time degrades a little bit the F-measure for snare drum and hi-hat detection in comparison with our model, it is much better for bass drum detection.

5. CONCLUSIONS

In this paper, we investigated different new approaches to the use of Convolutional Neural Networks for automatic drum transcription. We compared different loss functions and input representations. We found that the best results are obtained with the MCMS representation of the input data, namely three log-magnitude spectrograms with three different STFT window sizes: 23, 46 and 93 ms filtered into 80 mel frequency bands. We trained the network for the detection of percussive onsets, achieving very good detection performance well above 90% in F-measure. The combination of specific onset detectors based on CNN with a drum (bass drum, snare drum and hi-hat) transcription algorithm based on Non-negative Matrix Deconvolution did not lead to competitive performances.

Finally, we trained three individual CNNs: each of them detecting one of the three percussive instruments (bass drum, snare drum and hi-hat). The results obtained are significantly better than the results obtained with the NMD, which leads us to believe that the use of CNN for drum transcription has more potential than the use of a non-negative decomposition. We conjecture that the main reason for the better results is the fact that the CNN is trained with an objective function (the ODF) that is much closer to the final task than the objective function used in the NMD training. Further investigation is required to compare the single label detector proposed in the present paper with the multi label detector. While the single label detector may have the advantage of specializing more on the specific instrument, it also may be the reason for the over-fitting observed notably during the bass drum detection of drummer 1 of the ENST-Drums database.

6. REFERENCES

[1] Jouni Paulus and Anssi Klapuri, "Drum sound detection in polyphonic music with hidden markov models," *EURASIP Journal on Audio, Speech, and Music Processing*, 2009.

[2] Richard Vogl, Matthias Dorfer, and Peter Knees, "Drum transcription from polyphonic music with recurrent ,eural networks," *Proceedings of the 17th International Society for Music Information Retrieval Conference (ISMIR)*, 2016.

[3] Carl Southall, Ryan Stables, and Hockman Jason, "Automatic drum transcription for polyphonic recordings using soft attention mechanisms and convolutional neural networks," *Proceedings of the 18th International Society for Music Information Retrieval Conference (ISMIR)*, 2017.

[4] Olivier Gillet and Gaël Richard, "Enst-drums: an extensive audio-visual database for drum signals processing," *Proceedings of the 7th International Society for Music Information Retrieval Conference (ISMIR)*, 2006.

[5] Qi Wang, Ruohua Zhou, and Yonghong Yan, "A two stage approach to note-level transcription of a specific piano," *Applied Science*, 2017.

[6] Axel Roebel, Jordi Pons, Marco Liuni, and Mathieu Lagrange, "On automatic drum transcription using non-negative matrix deconvolution and itakura-saito divergence," *Proceedings IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 414–418, 2015.

[7] Juan Pablo Bello, Laurent Daudet, Samer Abdallah, Chris Duxbury, Mike Davies, and Mark B. Sandler, "A tutorial on onset detection in musical signals," *IEEE Transactions on Speech and Audio Processing*, vol. 13, no. 5, pp. 1035–1047, 2005.

[8] Anders Elowsson and Anders Friberg, "Modelling perception of speed in music audio," *Proceedings of the Sound and Music Computing Conference*, 2013.

[9] Matija Marolt, Alenka Kavcic, and Marko Provosnik, "Neural networkd for note onset detection in piano music," *Proceedings of the International Computer Music Conference (ICMC)*, 2002.

[10] Sebastian Böck, Andreas Artz, Florian Krebs, and Markus Shedl, "Online real-time onset detection with recurrent neural networks," *Proceedings of the 15th International Conference on Digital Audio Effects (DAFx)*, September 2012.

[11] Jan Schlüter and Sebastian Böck, "Improved musical onset detection with convolutional neural networks," *Proceedings IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2014.

[12] Rainer Kelz, Matthias Dorfer, Filip Korzeniowski, Sebastian Böck, Andreas Artz, and Ghehard Widmer, "On the potential of simple framewise approaches to piano transcription," *Proceedings of the 17th International Society for Music Information Retrieval Conference (ISMIR)*, 2016.

[13] Masataka Goto, Hiroki Hashigichi, Takuichi Nishimura, and Ryuichi Oka, "RWC music database: Popular, classical and jazz music databases.," *Proceedings of the 3rd International Society on Music Information Retrieval Conference (ISMIR)*, vol. 2, pp. 287–288, 2002.

[14] Sebastian Böck, Filip Korzeniowski, Jan Schüter, Florian Krebs, and Gerhard Widmer, "Madmom: a new python audio and music signal processing library," *Late-Breaking Demo Session of the 17th International Society for Music Information Retrieval Conference (ISMIR)*, 2016.

[15] Carl Southall, Nicholas Jillings, Ryan Stables, and Jason Hockman, "Adtweb: An open source browser based automatic drum transcription system," *Proceedings of the 18th International Society for Music Information Retrieval Conference (ISMIR)*, 2017.

OPTIMIZED VELVET-NOISE DECORRELATOR

Sebastian J. Schlecht

International Audio Laboratories Erlangen *
Erlangen, Germany
Sebastian.Schlecht@audiolabs-erlangen.de

Benoit Alary †

Acoustics Lab, Dept. of Signal Processing and Acoustics
Aalto University, Espoo, Finland
Benoit.Alary@aalto.fi

Vesa Välimäki

Acoustics Lab, Dept. of Signal Processing and Acoustics
Aalto University, Espoo, Finland
Vesa.Valimaki@aalto.fi

Emanuël A. P. Habets

International Audio Laboratories Erlangen *
Erlangen, Germany
Emanuel.Habets@audiolabs-erlangen.de

ABSTRACT

Decorrelation of audio signals is a critical step for spatial sound reproduction on multichannel configurations. Correlated signals yield a focused phantom source between the reproduction loudspeakers and may produce undesirable comb-filtering artifacts when the signal reaches the listener with small phase differences. Decorrelation techniques reduce such artifacts and extend the spatial auditory image by randomizing the phase of a signal while minimizing the spectral coloration. This paper proposes a method to optimize the decorrelation properties of a sparse noise sequence, called velvet noise, to generate short sparse FIR decorrelation filters. The sparsity allows a highly efficient time-domain convolution. The listening test results demonstrate that the proposed optimization method can yield effective and colorless decorrelation filters. In comparison to a white noise sequence, the filters obtained using the proposed method preserve better the spectrum of a signal and produce good quality broadband decorrelation while using 76% fewer operations for the convolution. Satisfactory results can be achieved with an even lower impulse density which decreases the computational cost by 88%.

1. INTRODUCTION

In multichannel reproduction systems as well as binaural reproduction, the decorrelation of signals is key in controlling the spatial extent of a reproduced sound source. With decorrelation we aim to reduce the cross-correlation of the reproduction signals. For instance, when reproducing a mono source on headphones, the spatial image is perceived in the center of the head. Decorrelation can extend the width of the auditory image such that it appears originating from a larger area. Fully decorrelated signals may even be perceived as separate auditory events [1]. Common applications of decorrelation include controlling the spatial extent, spatial audio coding, sound distance simulation, coloration reduction and headphone externalization [2–5]. This paper focuses on decorrelation methods suitable for controlling the perceived spatial extent of a sound source.

Decorrelation may be achieved by randomizing the phase of a signal while maintaining its magnitude spectrum. In [2], Kendall proposed a decorrelation filter based on 20–30 ms sequences of white noise. Shorter decorrelation filters can preserve the quality of the transients and prevent a reverberation effect [2]. Indeed, since high frequencies have shorter wavelengths, randomizing their phases can produce a noticeable smearing effects on short transient signals if the delays are too long. Unfortunately, limiting the length of a filter will limit its ability to decorrelate low frequencies, since long wavelengths require long delays to alter their phase significantly. This duality illustrates the challenge of designing a good broadband decorrelator that can compromise between preserving the transients and low-frequency decorrelation. This is the reason why most modern decorrelation methods operate in the time-frequency domain and restrict the phase variation based on the wavelength of various frequency bands [6].

Laitinen et al. proposed to apply a random delay within perceptually motivated bounds at each frequency band [7]. Although this method can lead to audible artifacts in stereo reproduction, these artifacts are less perceivable in multichannel reproduction. An alternative and common method is to decompose the signal into transient and non-transient signals, and apply the decorrelation only to the non-transient signal. For time-domain methods, finite impulse response (FIR) filters are applied with the fast convolution technique which can be computationally prohibitive for long filters in multiple decorrelation stages of multichannel systems. Alternatively, infinite impulse response (IIR) filters such as single or cascaded allpass filters, which guarantee a flat magnitude response, are computationally efficient [2, 8, 9]. However, if the group delay of the filter becomes too large, higher-order allpass filters can cause an undesired chirping effect [10].

Karjalainen and Järveläinen proposed velvet-noise sequences (VNSes), i.e., sparse series of uniformly distributed ± 1s, as a perceptually smoother alternative to Gaussian white noise [11, 12]. At only a fraction of the computational cost of dense FIR filters, VNSes are suitable for artificial reverberation [13,14] and approximation of room impulse responses [11, 15–19]. Short VNSes were proposed as an effective decorrelation method, although it suffered from spectral coloration [20]. In this work, we present a method to optimize the decorrelation properties of VNSes without altering the computational cost. We also conduct a formal listening test to evaluate the new method and to compare it with previous methods.

This paper is organized as follows. In Sec. 2, we review vel-

* The International Audio Laboratories Erlangen are a joint institution of the Friedrich-Alexander-Universität Erlangen-Nürnberg (FAU) and Fraunhofer Institut für Integrierte Schaltungen IIS.

† This work was supported by the Academy of Finland (ICHO project, grant no. 296390).

vet noise and its time and frequency-domain representations. Section 3 proposes an optimization technique for VNSes to minimize spectral coloration. Section 4 proposes a selection process to improve the decorrelation in sets of sequences. Section 5 presents the listening tests we conducted to evaluate the proposed method.

2. VELVET NOISE

2.1. Velvet-Noise Sequences

For a given density N_d and sampling rate f_s, the average spacing between two impulses in a VNS is

$$T_\mathrm{d} = f_\mathrm{s}/N_\mathrm{d}, \tag{1}$$

which is called the grid size [12]. The total number of impulses is

$$M = L_\mathrm{s} T_\mathrm{d}, \tag{2}$$

where L_s is the total length in samples. The sign of each impulse is

$$\sigma(m) = 2\,\lfloor r_1(m) \rceil - 1, \tag{3}$$

where $\lfloor \cdot \rceil$ denotes the rounding operation to the closest integer and $0 \le m \le M-1$ is the integer impulse index, and $r_1(m)$ is a uniformly distributed random number between 0 and 1. The impulse location is

$$\tau(m) = \begin{cases} 0 & \text{for } m = 0 \\ \lceil T_\mathrm{d}(m - 1 + r_2(m)) \rceil & \text{for } m > 0, \end{cases} \tag{4}$$

where $\lceil \cdot \rceil$ is the ceil operation to the next higher integer and $0 < r_2(m) \le 1$ is a uniformly distributed random number.

Exponentially decaying impulse gains have been found to improve the sharpness of transients and therefore the quality of the overall decorrelation [20]. The positive gain of each impulse is

$$\gamma(m) = e^{-\tau(m)\alpha}, \tag{5}$$

where $\alpha > 0$ denotes the slope of the exponential decay

$$\alpha = \frac{-\ln 10^{-L_\mathrm{dB}/20}}{L_\mathrm{s}}, \tag{6}$$

where L_dB is the target total decay in dB. The exponentially decaying velvet noise is denoted EVN_M, where M indicates the total number of impulses. In this work, we consider modifications to the EVN_M by allowing deviations from the exponential pulse gains (5) to improve the sequence's magnitude response. We refer to this non-exponential sequences as optimized velvet noise OVN_M obtained using the method described in Sec. 3.

Since velvet noise is the sum of single delayed impulses, the impulse response $h(n)$ of the resulting sparse FIR filter with M coefficients that are unequal to zero, is given by

$$h(n) = \sum_{m=0}^{M-1} \sigma(m)\gamma(m)\delta(n - \tau(m)), \tag{7}$$

where δ denotes the Kronecker delta function and n denotes the time index in samples. An input signal x can be decorrelated by convolution with the impulse response h. For this, we take advantage of the sparsity of the sequence. By storing the VNS as a series of non-zero elements, all mathematical operations involving zero can be skipped [17, 19]. For a sequence with a density of a 1000 impulses per second, which has been found sufficient for decorrelation [20], and a sample rate of 44.1 kHz, the zero elements represent 97.7% of the sequence. Therefore, given a sufficiently sparse sequence, time-domain convolution can be more efficient than a fast convolution using the FFT for an equivalent white-noise sequence [20]. Furthermore, this sparse time-domain convolution offers the benefit of being latency-free.

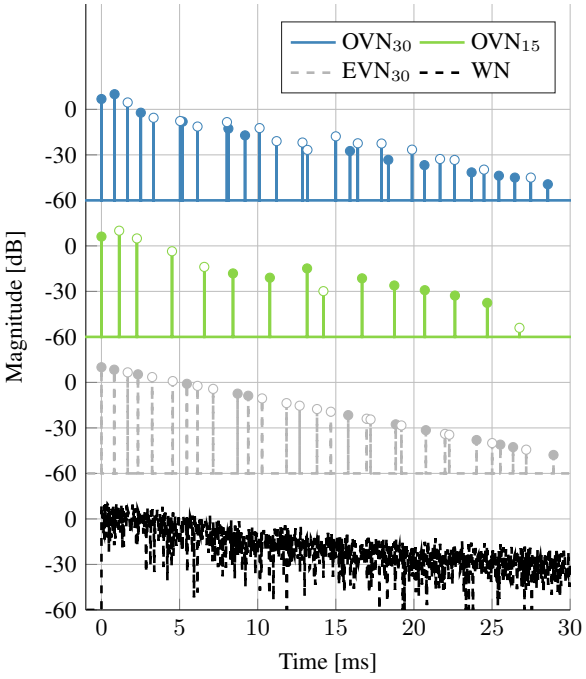


Figure 1: *Decorrelator sequences in the time domain: white noise WN, exponential velvet noise EVN_{30}, and two optimized velvet-noise sequences OVN_{15} and OVN_{30}. Positive impulses are indicated by • and negative gains by ∘ (except for WN).*

For comparison, we use an exponentially decaying Gaussian white noise sequence WN, with the same envelope as given in (5). The spectral coloration, i.e., non-flatness of the magnitude response, of the WN is reduced by replacing its magnitude response with a constant number, and re-synthesizing the time-domain sequence using the inverse Fourier transform.

Figure 1 depicts four decorrelation sequences: OVN_{30}, OVN_{15}, EVN_{30}, and WN. The total length of each sequence is 30 ms such that the VNS sequences have an impulse density of 1 and 0.5 impulse per ms, respectively. The total decay is $L_\mathrm{dB} = -60$ dB. However, the impulse response of WN decays only by about -30 dB in total, because of the spectral post-processing of WN. Convolution with OVN_{30} according to (7) uses 76% fewer operations than the fast convolution with WN, whereas OVN_{15} decreases the computational cost by 88% [20].

2.2. Velvet Noise in Frequency Domain

In addition to the time-domain formulation given in [20], we formulate the z-domain transfer function of the velvet noise. This formulation can be generalized to continuous impulse locations which is critical for the optimization procedure in Sec. 3. The cor-

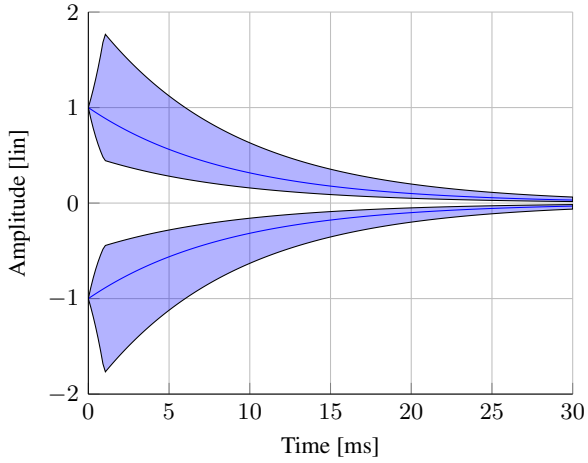


Figure 2: *Constraint on the optimized impulse gain γ over time. The solid blue line indicates the exponential decay as defined in* (5) *with $L_{\mathrm{dB}} = -60$ dB. The shaded blue area indicates the range of the optimized impulse gain with ± 6 dB and the enforced normalization of the first pulse to ± 1.*

responding z-domain transfer function of (7) is

$$H(z) = \sum_{m=0}^{M-1} \sigma(m)\gamma(m) z^{-\tau(m)} = \sum_{m=0}^{M-1} H_m(z), \qquad (8)$$

where $H_m(z)$ indicates the transfer function of the mth impulse. The magnitude response of the mth impulse is

$$|H_m(e^{\imath\omega})| = \gamma(m), \qquad (9)$$

where ω is the frequency in radians and $\imath = \sqrt{-1}$. The corresponding unwrapped phase response is

$$\angle H_m(e^{\imath\omega}) = \begin{cases} -\omega\tau(m) & \text{for } \sigma(m) = 1 \\ \pi - \omega\tau(m) & \text{for } \sigma(m) = -1, \end{cases} \qquad (10)$$

where $\angle$ denotes the radian angle of a complex number. The phase response formulation in (10) generalizes directly to continuous impulse locations $\widetilde{\tau}(m)$. The corresponding single impulse and summed transfer functions are denoted $\widetilde{H}_m$ and $\widetilde{H}$, respectively. The continuous formulation plays a critical role in the optimization process presented in the following section as it allows continuous modification of both impulse location and impulse gain.

3. MAGNITUDE RESPONSE OPTIMIZATION

A central challenge in decorrelation is the coloration caused by a non-flat magnitude response of the decorrelator. This section is concerned with modifying the impulse locations $\tau(m)$ and impulse gains $\gamma(m)$ of a VNS to improve the flatness of its magnitude response $|H(e^{\imath\omega})|$. In the following subsections, we describe: i) heuristic constraints on the velvet-noise parameters; ii) the objective function; iii) the optimization process; and iv) the performance results.

3.1. Parameter Constraints

In the following, we impose heuristic constraints on the time location $\tau(m)$ and gain $\gamma(m)$ of the impulses of the velvet noise. An even distribution of impulses over time is desirable to ensure a smooth time-domain response [20]. Therefore, the impulse locations should not exceed the boundaries defined in (4).

An impulse with a long delay and a large gain is perceived as an echo, so it degrades the perceptual quality of decorrelated transients. The exponential decay of impulse gains over time as defined in (5) effectively minimizes the time-domain smearing of transients signals [20]. Nonetheless, small deviations from the exponential decay may be marginal for the perception. Informal experiments determined an appropriate range of ± 6 dB deviation from the exponential decay, which corresponds to a multiplicative gain factor χ up to 2. To enforce a normalization of the impulse gains, we set the first impulse gain to be ± 1. Later for evaluation purposes, all sequences are normalized to the same energy. Figure 2 depicts the constraints on the impulse gain γ over time. The positive and negative impulse gain ranges in Fig. 2 are not connected such that a continuous optimization process cannot change the impulse sign σ.

3.2. Objective Function

We establish the objective function as to represent the perceived quantity of coloration of the decorrelator. In this work, we employ a third-octave smoothing of the magnitude response in dB between 20 Hz to 20 kHz [21]. The magnitude response is sampled at logarithmically spaced frequencies

$$\boldsymbol{f}_{\log}(k) = e^{\boldsymbol{f}_{\mathrm{lin}}(k)}, \qquad (11)$$

where $\boldsymbol{f}_{\mathrm{lin}} = [\ln(20), \ldots, \ln(f_{\mathrm{s}}/2)]$ is a linearly spaced $1 \times K$ vector and K is the number of frequency points. The corresponding frequencies in radian are $\boldsymbol{\omega}_{\log} = \frac{2\pi}{f_{\mathrm{s}}} \boldsymbol{f}_{\log}$. The rectangular smoothing kernel $\boldsymbol{\kappa}$ for a third-octave smoothing is then given by

$$\boldsymbol{\kappa}(k) = \begin{cases} \frac{1}{2\kappa_w + 1} & \text{for } |k| < \kappa_w \\ 0 & \text{otherwise,} \end{cases} \qquad (12)$$

where the kernel width κ_w is defined by

$$\frac{\kappa_w}{K} \frac{\ln(f_{\mathrm{s}}/2)}{\ln(20)} = \frac{1}{6}. \qquad (13)$$

The third-octave smoothed magnitude response $\mathcal{H}$ is then

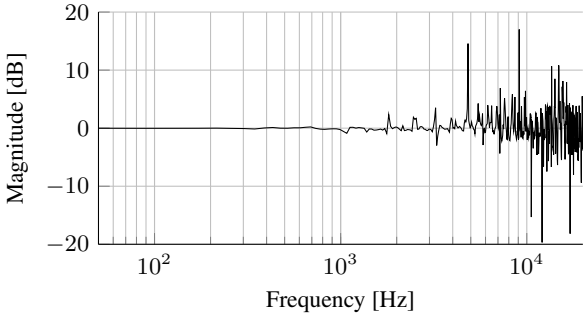
$$\mathcal{H}(k) = \left(\boldsymbol{\kappa} * 20 \log \left| H\left(e^{\imath \boldsymbol{\omega}_{\log}(k)} \right) \right| \right), \qquad (14)$$

where $*$ denotes the convolution operation. The objective function $\mathcal{L}$ is given by the root mean squared error (RMSE) of the smoothed magnitude response

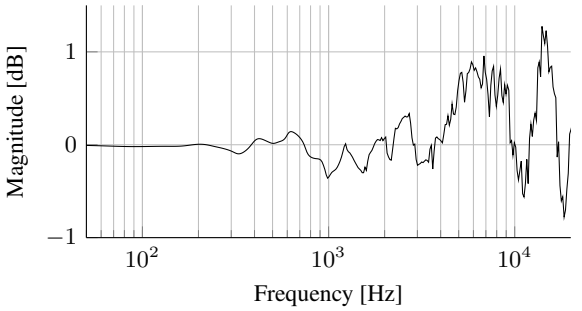
$$\mathcal{L}(\tau, \gamma) = \sqrt{\frac{1}{K} \sum_{k=0}^{K-1} \left(\mathcal{H}(k) - \overline{\mathcal{H}} \right)^2}, \qquad (15)$$

where $\overline{\mathcal{H}} = \sum_{k=0}^{K-1} \mathcal{H}(k)/K$ is the mean smoothed magnitude response. The proposed optimization problem is then

$$\begin{aligned} &\min_{\tau,\gamma} \mathcal{L}(\tau, \gamma) \\ &\text{subject to } \tau(0) = 0 \text{ and } \gamma(0) = 1 \\ &\qquad T_{\mathrm{d}}(m-1) < \tau(m) \leq T_{\mathrm{d}}\, m \\ &\qquad e^{-\tau(m)\alpha}/\chi \leq \gamma(m) < \chi e^{-\tau(m)\alpha}, \end{aligned} \qquad (16)$$

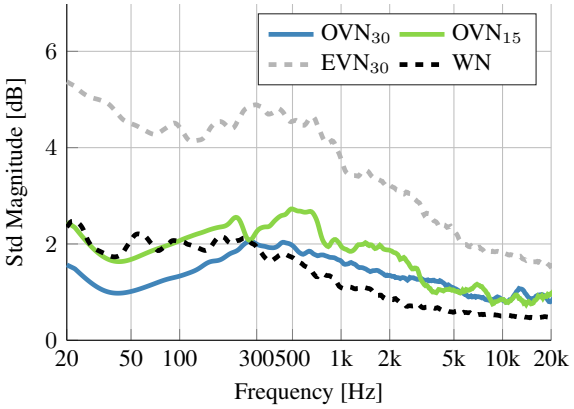


(a) Magnitude response error without smoothing.

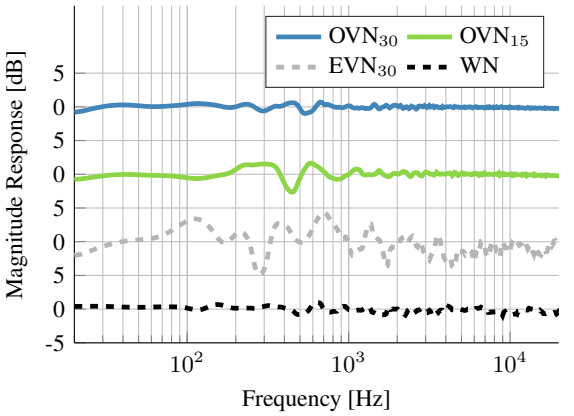


(b) Magnitude response error with third-octave smoothing.

Figure 3: *Magnitude responses error of an EVN$_{30}$ between a continuous impulse location $\widetilde{\tau}$ and the closest integer impulse location $\lfloor\widetilde{\tau}\rceil$. The error between the non-smoothed magnitude responses in Fig. 3a increases with frequency up to 20 dB. However, for the third-octave smoothed response in Fig. 3b the error is within 1.3 dB.*



(a) Standard deviation on the smoothed magnitude response for 500 sequences.



(b) Smoothed magnitude response of the best sequence, i.e., with the lowest objective function value, out of 500 sequences.

Figure 4: *Performance evaluation of the proposed optimization process by comparing 500 sequences of the four decorrelator types: WN, EVN$_{30}$, OVN$_{30}$, and OVN$_{15}$.*

where the possible gain deviation $\chi = 2$ and the impulse sign σ is a random, but fixed parameter in the objective function $\mathcal{L}$.

3.3. Optimization Process

The optimization problem (16) is a constrained, non-linear and non-convex problem such that the optimal solution, i.e., the global minimum, is generally difficult to find. However, local minima can be attained by various gradient descent algorithms. Here we employ a variant of the interior-point method [22]. The initial point is given by a randomly generated EVN according to (4) and (5).

To allow gradual changes of all parameters during optimization, we employ the continuous impulse location $\widetilde{\tau}$ in the objective function

$$\min_{\widetilde{\tau},\gamma} \mathcal{L}(\widetilde{\tau}, \gamma). \tag{17}$$

The corresponding integer impulse location solution is then given by $\tau = \lfloor\widetilde{\tau}\rceil$. In the following, we evaluate the error introduced by the continuous impulse location solution.

The continuous impulse location $\widetilde{\tau}$ introduces a phase error of the single impulse transfer function in (10). The maximum impulse location error is

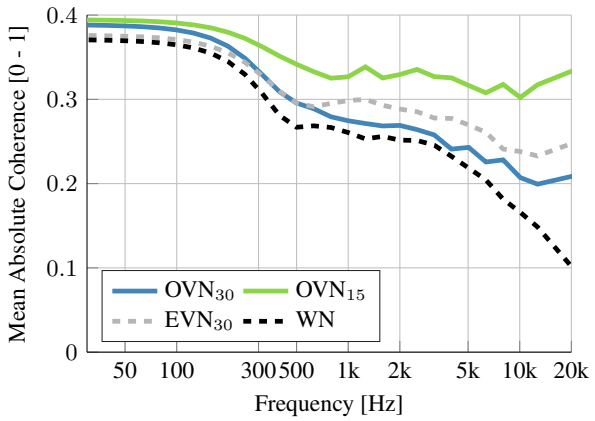
$$|\widetilde{\tau}(m) - \lfloor\widetilde{\tau}(m)\rceil| \leq 0.5. \tag{18}$$

Consequently, the maximum phase error between the continuous and the closest integer transfer function is

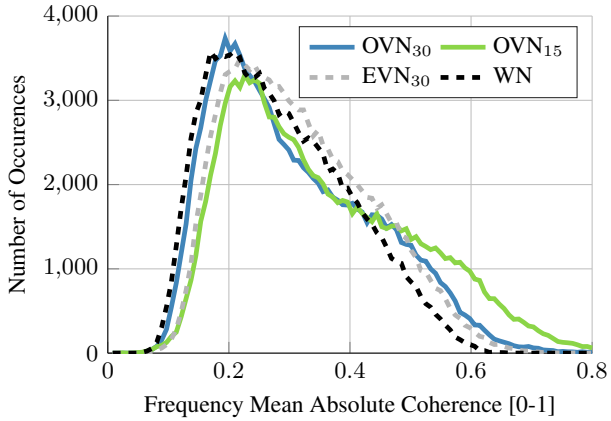
$$\left|\angle\widetilde{H}_m(e^{\imath\omega}) - \angle H_m(e^{\imath\omega})\right| \leq \omega/2 \tag{19}$$

such that the maximum phase error increases linearly with frequency. The phase error of the single impulse transfer function H_m results in a magnitude error of the full sequence transfer function H.

Figure 3a depicts the magnitude response error of an EVN$_{30}$ between a continuous impulse location $\widetilde{\tau}$ and the closest integer impulse location $\lfloor\widetilde{\tau}\rceil$. Whereas the magnitude error is below 1 dB for frequencies below 1 kHz, the error increases up to 20 dB for high frequencies. In Fig. 3b, the magnitude response error of the same two sequences are shown with third-octave smoothing. The maximum error over the complete frequency range stays below 1.3 dB. Similarly, Karjalainen and Järveläinen observed that increasing the time resolution beyond 44.1 kHz, does not improve velvet noise [11]. Hence, the proposed optimization using continuous impulse locations which are then rounded to the nearest integers introduces only minor deviations in the magnitude response.



(a) Mean absolute coherence.



(b) Distribution of frequency mean absolute coherence.

Figure 5: *Evaluation of the absolute coherence between over all sequence pairs of the 500 randomly generated sequences of four decorrelator types: WN, EVN$_{30}$, OVN$_{30}$, and OVN$_{15}$.*

3.4. Results

In this subsection, we compare the magnitude response of four decorrelation sequence types: WN, EVN$_{30}$, OVN$_{30}$ and OVN$_{15}$. The total length of the sequences is 30 ms and the total decay is $L_{\mathrm{dB}} = -60$ dB. We generated 500 sequences for each decorrelation filter type. For the optimized sequence types, the initial sequences are EVN$_{15}$ and EVN$_{30}$, respectively, which were randomly generated. As convergence is not guaranteed, the optimization algorithm was limited to 60 iteration steps to comply with a time limit of 30 s. The mean absolute change in impulse location between the initial point and the local minima is 11 to 12 samples. The mean absolute gain deviation from the exponential decay is about 3 to 4 dB.

Figure 4a depicts the standard deviation of the smoothed magnitude response over 500 sequences. The EVN$_{30}$ has the largest standard deviation over all frequencies indicating a relatively poor flatness of the magnitude response. The largest deviation is in the low frequencies with 5.3 dB, which decays with frequency to 1.5 dB. The standard deviation of the WN is similar in shape to the EVN$_{30}$ with the largest deviation of 2.3 dB in the low frequencies and a minimum of 0.5 dB in the high frequencies. The standard deviations of the optimized sequences OVN$_{30}$ and OVN$_{15}$ are similar to WN for high frequencies, but is considerably lower for low frequencies. The minimum standard deviation at around 30 Hz is 1 dB and 1.6 dB, respectively, and by this up to 2.5 times lower than WN and up to 4 times lower than EVN$_{30}$. The low standard deviation of the OVN$_{30}$ implies a successful minimization of the objective function (16).

Figure 4b depicts the smoothed magnitude response for the best sequences, i.e., with the lowest objective function value, out of all 500 sequences. The magnitude responses confirm the trends of the standard deviation, as shown in Fig. 4a. The best sequence demonstrates that optimization can yield sequences with less than a 1-dB maximum deviation from the mean magnitude. Despite the large standard deviation in the low frequencies, the best sequences have rather flat magnitude responses at low frequencies.

4. SET OF DECORRELATOR SEQUENCES

In many applications, a set of decorrelators is required such that each pair of decorrelation filters is as "different" as possible. In the following, we measure the difference using the coherence and present a method to choose a low-coherence set of multiple decorrelators. When a mono signal is required to be decorrelated to $N_{\mathcal{D}}$ channels, we need $N_{\mathcal{D}}$ decorrelation sequences where each pairwise coherence is minimal.

4.1. Coherence

The effectiveness of decorrelation can be measured with the cross-correlation in different frequency bands, called coherence. Normally, a broadband decorrelator is more effective at higher frequencies than at lower, which is a result of the effective length of a decorrelation filter. Indeed, a longer filter will exhibit stronger decorrelation for longer wavelengths, but will also create potentially perceivable artifacts when the input signal contains transients. To study the decorrelation behavior on a frequency-dependent scale, we use a third-octave filterbank. The signals for the jth band are denoted as a_j and b_j and the normalized correlation coefficient as

$$\rho_{a,b}^{(j)} = \frac{\sum_n a_j(n) b_j(n)}{\sqrt{\sum_n a_j^2(n) \sum_n b_j^2(n)}}, \tag{20}$$

where $1 \leq j \leq J$, and J is the number of third-octave bands. Between 20 Hz and 20 kHz, we have $J = 30$. A lower absolute value indicates a more effective decorrelation such that we are mainly interested in the absolute correlation $\left|\rho_{a,b}^{(j)}\right|$. To summarize the broadband effectiveness of the decorrelation, we use the frequency mean absolute coherence

$$\overline{|\rho_{a,b}|} = \frac{1}{Q} \sum_{j=1}^{J} \left|\rho_{a,b}^{(j)}\right|. \tag{21}$$

Note that the sparse impulse locations of two velvet noise sequences rarely coincide such that the classic broadband decorrelation is ill-defined and (21) is preferred instead.

In the following, we evaluate the coherence between the 500 generated sequences of each decorrelation type explained in Sec. 3. Since the coherence is symmetric, there are $500 \times 499/2 = 124,750$ different pairs of sequences. Figure 5a depicts the mean absolute coherence for each third-octave band over all sequence

Table 1: *Best pair of optimized velvet noise OVN_{30} found with the proposed method. The gains γ are given with a factor of 10.*

m	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
$\tau_a(m)$	1	46	91	134	175	182	239	271	351	359	407	484	531	536	581
$\gamma_a(m)$	4.71	7.37	-3.72	1.46	1.12	-1.84	0.64	-0.54	-0.64	1.08	-0.32	0.24	0.21	-0.49	0.14
$\tau_b(m)$	1	5	78	125	172	219	234	271	318	381	403	460	531	575	583
$\gamma_b(m)$	4.11	-3.91	5.58	4.30	-2.96	2.02	-0.61	-1.34	1.15	-0.93	0.81	-0.37	-0.26	0.16	0.14
m	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
$\tau_a(m)$	651	669	731	797	829	851	890	961	984	1027	1074	1130	1175	1232	1246
$\gamma_a(m)$	0.18	-0.14	-0.09	-0.08	-0.08	0.07	0.05	0.04	-0.04	0.02	0.02	0.01	-0.01	0.01	-0.01
$\tau_b(m)$	663	703	737	791	809	881	902	950	999	1041	1083	1135	1177	1216	1258
$\gamma_b(m)$	0.10	-0.19	0.07	0.06	0.05	0.05	-0.06	-0.04	0.03	0.02	-0.02	0.01	-0.01	-0.01	-0.01

Table 2: *Best pair of optimized velvet noise OVN_{15} found with the proposed method. The gains γ are given with a factor of 10.*

m	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
$\tau_a(m)$	1	51	101	200	291	372	476	581	627	736	827	913	998	1089	1180
$\gamma_a(m)$	4.80	-7.51	-4.18	-1.58	-0.48	0.29	0.21	0.43	-0.08	0.20	0.12	0.08	0.05	0.03	-0.01
$\tau_b(m)$	1	10	140	215	279	365	485	579	668	756	836	892	1005	1071	1192
$\gamma_b(m)$	6.10	-2.94	6.63	-1.05	-2.88	-0.46	-0.28	-0.68	-0.36	0.06	0.04	-0.09	0.02	0.01	-0.02

pairs. For all four decorrelator types, the absolute coherence decreases with frequency due to the effective length of the decorrelator. The maximum absolute coherence at low frequencies is between 0.35 and 0.4 and the minimum absolute coherence of 0.1 and 0.33 at high frequencies. The coherence is generally slightly larger for EVN_{30} and OVN_{15} due to the systematic exponential gain, and higher sparsity, respectively. Since coherence is not modeled in the optimization process in Sec. 3, it is expected to have little influence on the overall coherence.

Figure 5b depicts the distribution of the frequency mean absolute coherence $\overline{|\rho_{a,b}|}$ over all pairs. The difference between the four decorrelation types is small, as expected, and a frequency mean absolute coherence of around 0.19 to 0.22 is most frequent. However, there are sequence pairs with rather large coherence values up to 0.8 suggesting poor decorrelation performance. In the next subsection, we present methods to choose a set of decorrelation sequences with low pairwise coherence.

4.2. Choosing Set of Decorrelators

Although the mean absolute coherence is typically between 0.19 and 0.22, the coherence of a set of sequences can be improved by a selection process. More formally, the goal is to find a set $\mathcal{D}$ of $N_\mathcal{D}$ sequences such that

$$\min_{\mathcal{D}} \sum_{a,b\in\mathcal{D}} \overline{|\rho_{a,b}|}. \tag{22}$$

Let us consider the coherence matrix, i.e., all pairwise frequency mean absolute coherences, to be the adjacency matrix of an undirected graph. The minimization problem (22) then corresponds to finding the *thinnest* $N_\mathcal{D}$-subgraph. By taking the negative of the coherence matrix, this problem is equivalent to the better known *densest* $N_\mathcal{D}$-subgraph problem [23]. Although finding the optimal solution is NP-hard, greedy algorithms can be applied to yield an approximative solution [24]. In this contribution, however, we are mainly concerned with pairs of sequences to allow decorrelated stereo reproduction. Thus, (22) is merely the minimum entry of the coherence matrix. Although, the frequency mean absolute coherence peaks around 0.2 in Fig. 5b, sequence pairs with coherence as low as 0.05 can be found for all decorrelator types.

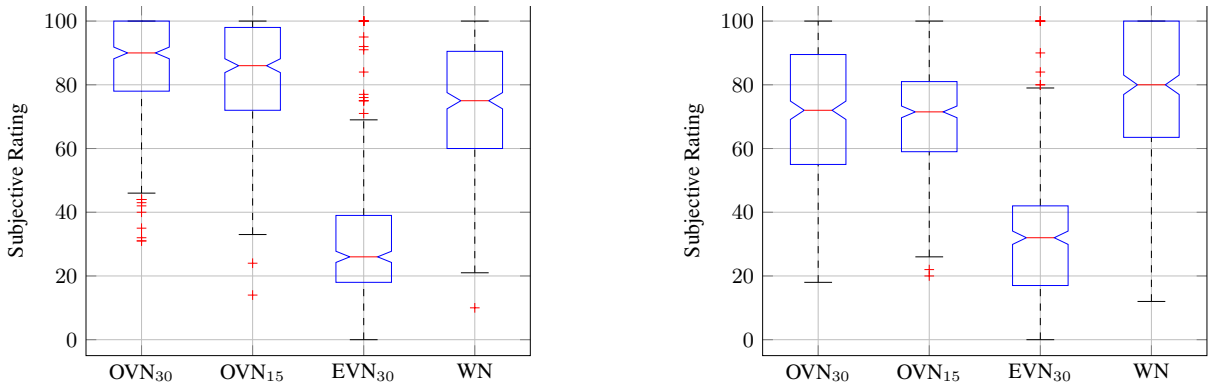
In the choice of the optimal set of decorrelators, the lowest coherence pairs are not necessarily those which have flat magnitude responses. To account for the coloration of the single sequences, we introduce a penalty term for (22):

$$\min_{\mathcal{D}} \sum_{a,b\in\mathcal{D}} (1-\lambda)\overline{|\rho_{a,b}|} + \lambda\,\mu(\mathcal{L}_a + \mathcal{L}_b), \tag{23}$$

where $\mathcal{L}_a$ and $\mathcal{L}_b$ are the objective functions (15) of sequences a and b, λ is the weighting factor, and μ is the normalization factor to balance the two objective functions with $\lambda = 0.5$. The balance is optimal if the distributions of $\overline{|\rho_{a,b}|}$ and $\mu(\mathcal{L}_a + \mathcal{L}_b)$ overlap maximally. In this work, this is achieved by $\mu = 0.1$. The larger λ, the more emphasis is put on magnitude flatness rather than a low coherence value. Tables 1 and 2 give the best decorrelation pairs we have found through our proposed method with $\lambda = 0.8$. These sequences were evaluated via a formal listening test, as explained in the next section.

5. PERCEPTUAL EVALUATION

We conducted two formal listening tests to evaluate the perceived quality of the decorrelation filters obtained using the proposed method. The first test assessed the coloration introduced by the decorrelators via comparison of the processed signal to the unprocessed signal. The second test evaluated the effectiveness of the decorrelators to extend the auditory source width and overall quality. The tests were conducted in special listening booths built for sound isolation and high-quality reproduction over headphones. The test interface was based on a MUSHRA-type web interface with a subjective rating scale from 0 to 100 allowing seamless switching between test conditions and looping of short sections.



(a) Coloration test with mono reproduction.

(b) Stereo quality test with stereo reproduction.

Figure 6: *Results of two listening tests of four decorrelator types: OVN_{30}, OVN_{15}, EVN_{30}, and WN. In each box, the central red line indicates the median, and the bottom and top edges of the box indicate the 25th and 75th percentiles, respectively. The whiskers extend to the most extreme data points not considered outliers, and the outliers are plotted individually using the + symbol. The box notches indicate the confidence intervals, i.e., two medians are significantly different at the 95% confidence level if their intervals do not overlap.*

Each test page compared six conditions: OVN_{30}, OVN_{15}, EVN_{30}, WN, anchor, and reference. For each decorrelation type, we chose four decorrelator instances. Each test page was repeated once during the test. In total, 4 instances $\times$ 2 trials $\times$ 4 input signals $= 32$ test pages were presented for each test[1].

Each listening test was participated by 11 listeners (10 males and 1 female) who were all aged between 24 and 34. Due to the long test time, few participants performed both tests on the same day. Four different input signals were convolved with the decorrelation sequences: drums, guitar, singing, and speech. The order of the test conditions was individually randomized. From the difference between the identical trials, the test-retest reliability could be computed. The cross-correlation coefficient between the first and second trial was 0.96 suggesting that most participants were able to give consistent ratings.

5.1. Coloration Test

The first listening test evaluated how much the decorrelation filters distort the input signal. The input signal was convolved with a single decorrelation filter, and the difference to the unprocessed signal was rated by the participants. In MUSHRA terminology, the unprocessed mono signal was the reference, and the input signal processed with a lowpass filter having a 3.5-kHz cutoff frequency was the anchor. The resulting mono signals were reproduced on both headphone channels. The main coloration was expected to be caused by the change in timbre and smearing of transients.

The four decorrelation instances were selected out of the 500 sequences which were generated in Sec. 3. For OVN_{30} and OVN_{15}, we selected the four best sequences according to spectral flatness as defined in (15). The EVN_{30} sequences were selected as the initial sequences of the OVN_{30}, i.e., the original random sequence before the optimization to emphasize the improvement gained by the proposed method. The WN sequences were generated randomly and spectrally flattened, as described in Sec. 2.

Figure 6a shows the resulting subjective rating of the coloration test. The median ratings for OVN_{30}, OVN_{15}, EVN_{30}, and WN are 90, 86, 26, and 75, respectively. All pairwise comparisons of the confidence interval suggests that the medians are significantly different at the 95% confidence level. The superior rating of both optimized velvet-noise sequences suggests a substantial reduction in spectral coloration compared to EVN_{30}, and this demonstrates the effectiveness of the optimization method and the corresponding objective function (15). Furthermore, both OVN_{30} and OVN_{15} were rated slightly superior to WN suggesting that they are valid alternatives.

5.2. Stereo Quality Test

The second listening test evaluated the effectiveness of the decorrelators in extending the auditory source width and the overall spatial quality. The input signal was convolved with a decorrelation filter for each channel (left and right) and the participants were asked to rate the perceived width, localization at the center, and overall quality. In this test, no ideal reference could be defined, so the unprocessed mono signal was provided only for guidance. The lowpass filtered mono signal was given as the anchor. The resulting stereo signal was reproduced on the left and right headphone channels. Once again, we selected the sequences from the generated set as in the coloration test. For OVN_{30} and OVN_{15}, we selected the four best sequence pairs according to the rating function (23) and weighting factor $\lambda = 0.8$. Tables 1 and 2 present the top-rated sequence pairs. The EVN_{30} sequence pairs were selected as the initial optimization sequences of the OVN_{30} pairs. The WN sequence pairs were generated randomly according to Sec. 2.

Figure 6b shows the resulting subjective rating of the auditory source width test. The median ratings for OVN_{30}, OVN_{15}, EVN_{30}, and WN are 72, 71, 32, and 80, respectively. Pairwise comparison of the confidence interval suggests that the EVN_{30} and WN medians are significantly different at the 95% confidence level. No significant difference between OVN_{30} and OVN_{15} was found. Here again, a superior rating was given to the optimized sequences over the EVN_{30}, which is expected due to the perceptible coloration of the EVN_{30} found in the coloration test. A slightly inferior rating was given to the optimized methods compared to WN. This may be a result of our pair selection process favoring a flat spectrum over

[1] Audio examples are available at `https://www.audiolabs-erlangen.de/resources/2018-DAFx-VND`.

low coherence. Nonetheless, these results suggest that OVN_{30} and OVN_{15} are valid alternatives to WN, since they can yield reduction in the computational cost without affecting significantly the overall sound quality.

6. CONCLUSION

We have proposed an optimization method to improve the perceived quality of velvet-noise decorrelators. The original method EVN employed short, sparse, and exponentially decaying sequences, which were generated randomly [20]. The proposed method OVN attempts to improve such sequences by allowing small deviations in the impulse gains and timings. The optimization maximizes the spectral flatness within given heuristic constraints. A continuous impulse location formulation facilitates simultaneous modifications of gains and times. Furthermore, we proposed a method to select a set of minimally correlated sequences according to a coherence metric. An additional weighting factor allows user-defined control over the trade-off between coherence and spectral flatness.

Two formal listening tests were conducted to evaluate possible coloration as well as the auditory source width and overall stereo quality. The subjective ratings show a substantial improvement of the proposed method against the original and perceptually satisfactory decorrelation. While convolving a signal with velvet noise can be performed using as much as 88% less operations compared with WN, the objective ratings as well as the subjective ones confirms that the proposed OVN method is a good alternative to the WN decorrelation, when it is possible to pre-compute sets of optimal sequences.

7. ACKNOWLEDGMENT

Part of this research was conducted in March 2018, when Dr. Sebastian Schlecht visited the Aalto Acoustics Lab for one week.

8. REFERENCES

[1] J. Blauert, *Spatial Hearing: The Psychophysics of Human Sound Localization*, MIT press, 1997.

[2] G. S. Kendall, "The decorrelation of audio signals and its impact on spatial imagery," *Computer Music J.*, vol. 19, no. 4, pp. 71–87, 1995.

[3] G. Potard and I. Burnett, "Decorrelation techniques for the rendering of apparent sound source width in 3D audio displays," in *Proc. DAFX-04*, Naples, Italy, Oct. 2004, pp. 280–284.

[4] C. Faller, "Parametric multichannel audio coding: synthesis of coherence cues," *IEEE Trans. Audio, Speech, and Lang. Processing*, vol. 14, no. 1, pp. 299–310, Jan. 2006.

[5] V. Pulkki and J. Merimaa, "Spatial impulse response rendering II: Reproduction of diffuse sound and listening tests," *J. Audio Eng. Soc.*, vol. 54, no. 1/2, pp. 3–20, Jan./Feb. 2006.

[6] M. Bouéri and C. Kyriakakis, "Audio signal decorrelation based on a critical band approach," in *Proc. AES 117th Conv.*, San Francisco, CA, USA, Oct. 2004.

[7] M. Laitinen, F. Kuech, S. Disch, and V. Pulkki, "Reproducing applause-type signals with directional audio coding," *J. Audio Eng. Soc.*, vol. 59, no. 1/2, pp. 29–43, Jan./Feb. 2011.

[8] E. Kermit-Canfield and J. Abel, "Signal decorrelation using perceptually informed allpass filters," in *Proc. DAFx-16*, Brno, Czech Republic, Sept. 2016, pp. 225–231.

[9] E. K. Canfield-Dafilou and J. S. Abel, "A group delay-based method for signal decorrelation," in *Proc. AES 144th Conv.*, Milan, Italy, May 2018.

[10] V. Välimäki, J. S. Abel, and J. O. Smith, "Spectral delay filters," *J. Audio Eng. Soc.*, vol. 57, no. 7/8, pp. 521–531, Jul./Aug. 2009.

[11] M. Karjalainen and H. Järveläinen, "Reverberation modeling using velvet noise," in *Proc. AES 30th Int. Conf.: Intelligent Audio Environments*, Saariselkä, Finland, Mar. 2007.

[12] V. Välimäki, H.-M. Lehtonen, and M. Takanen, "A perceptual study on velvet noise and its variants at different pulse densities," *IEEE Trans. Audio, Speech, and Lang. Processing*, vol. 21, no. 7, pp. 1481–1488, July 2013.

[13] P. Rubak and L. G. Johansen, "Artificial reverberation based on a pseudo-random impulse response," in *Proc. AES 104th Conv.*, Amsterdam, The Netherlands, May 1998.

[14] P. Rubak and L. G. Johansen, "Artificial reverberation based on a pseudo-random impulse response II," in *Proc. AES 106th Conv.*, Munich, Germany, May 1999.

[15] J. Vilkamo, B. Neugebauer, and J. Plogsties, "Sparse frequency-domain reverberator," *J. Audio Eng. Soc.*, vol. 59, no. 12, pp. 936–943, Dec. 2012.

[16] S. Oksanen, J. Parker, A. Politis, and V. Välimäki, "A directional diffuse reverberation model for excavated tunnels in rock," in *Proc. IEEE ICASSP*, Vancouver, Canada, May 2013, pp. 644–648.

[17] B. Holm-Rasmussen, H.-M. Lehtonen, and V. Välimäki, "A new reverberator based on variable sparsity convolution," in *Proc. DAFx-13*, Maynooth, Ireland, Sept. 2013, pp. 344–350.

[18] V. Välimäki, J. D. Parker, L. Savioja, J. O. Smith, and J. S. Abel, "More than 50 years of artificial reverberation," in *Proc. AES 60th Int. Conf.*, Leuven, Belgium, Feb. 2016.

[19] V. Välimäki, B. Holm-Rasmussen, B. Alary, and H.-M. Lehtonen, "Late reverberation synthesis using filtered velvet noise," *Appl. Sci.*, vol. 7, no. 483, May 2017.

[20] B. Alary, A. Politis, and V. Välimäki, "Velvet-noise decorrelator," in *Proc. DAFx-17*, Edinburgh, UK, Sept. 2017, pp. 405–411.

[21] F. E. Toole, *Sound Reproduction: The Acoustics and Psychoacoustics of Loudspeakers and Rooms*, Focal Press, Burlington, MA, USA, 2008.

[22] J. Nocedal and S. J. Wright, *Numerical Optimization*, Springer Series in Operations Research and Financial Engineering. Springer Science & Business Media, New York, NY, USA, Jan. 1999.

[23] U. Feige, D. Peleg, and G. Kortsarz, "The dense k-subgraph problem," *Algorithmica*, vol. 29, no. 3, pp. 410–421, Mar. 2001.

[24] M. Charikar, "Greedy approximation algorithms for finding dense components in a graph," in *Proc. Int. Work. Approx. Algor. Comb. Optim.*, Berlin, Germany, 2000, pp. 84–95.

SURROUND SOUND WITHOUT REAR LOUDSPEAKERS: MULTICHANNEL COMPENSATED AMPLITUDE PANNING AND AMBISONICS

Dylan Menzies

Institute of Sound and Vibration Research,
University of Southampton
d.menzies@soton.ac.uk

Filippo Maria Fazi

Institute of Sound and Vibration Research,
University of Southampton

ABSTRACT

Conventional panning approaches for surround sound require loudspeakers to be distributed over the regions where images are needed. However in many listening situations it is not practical or desirable to place loudspeakers some positions, such as behind or above the listener. Compensated Amplitude Panning (CAP) is a method that adapts dynamically to the listener's head orientation to provide images in any direction, in the frequency range up to $\approx$ 1000 Hz using only 2 loudspeakers. CAP is extended here for more loudspeakers, which removes some limitations and provides additional benefits. The new CAP method is also compared with an Ambisonics approach that is adapted for surround sound without rear loudspeakers.

1. INTRODUCTION

Amplitude panning is a method for producing a spatial audio image in which 2 or more waves combine coherently at the listener position, each carrying the same signal but independent gains. For some choices of plane wave directions and gains the listener perceives an image, or phantom source, from a definite direction, a phenomena known as summing localisation [1]. The direction of the image can be varied continuously by varying the gains.

Below $\approx$ 1000Hz the perception of image direction is mainly determined by the Interaural Time Difference (ITD) cue. In this frequency range, a central stereo image, produced by panning with 2 loudspeakers, is unstable. If the listener faces straight ahead the image is also straight ahead. As the listener turns away from this direction the image moves in the direction of the listener, as illustrated in Fig. 1 [2, 3, 4]. A typical scene contains multiple

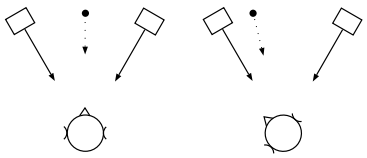


Figure 1: The black dot indicates the direction of the image when 2 loudspeakers each have the same signal, for different head directions.

images in different directions, so at any moment images that are not directly ahead of the listener or inline with a loudspeaker will be distorted. The distortion is greater when the angle between the loudspeakers, viewed from the listener, is increased. For example the listener can approach a stereo pair until the loudspeakers are 180° apart. In this position an image panned to the centre would be completely unstable. Producing consistent ITD cues when the head rotates, otherwise known as *dynamic ITD cues*, is important for localisation [1, 5, 6, 7].

The change in the panned image direction when the head is rotated is caused by the ITD cue not matching that of a static source for each head angle. Compensated Amplitude Panning (CAP), is an extension of conventional panning methods in which the ITD cues are corrected by modifying the gains to take account of the head orientation of the listener [8]. Tracking the listener accurately in real-time with low latency is a challenging requirement for this system. However suitable tracking technology is progressing very rapidly, driven by a wide range of applications.

CAP has been developed for 2 loudspeaker reproduction (Stereo-CAP). This produces more stable images than conventional stereo across the front stage. Further more, the method can produce images in any direction, because ITD is reproduced accurately in any case. Dynamic ITD cues generated by small head movements allow the resolution of front-back ambiguities, and elevation.

To cover the full bandwidth CAP can be combined with high frequency reproduction methods. CAP requires only 2 loudspeakers that are capable of driving the ITD frequency range, while the high frequency range can be driven using smaller and lighter loudspeakers, that are practical to use in higher numbers. Energy based panning, or *Vector Base Intensity Panning (VBIP)* [9] can be combined with Stereo-CAP to provide a very stable full bandwidth front stage. Stereo-CAP provides low frequency coverage elsewhere, which is useful for immersive ambience and reverberation. High frequency coverage can also be provided in all directions using *transaural cross-talk cancellation* [10, 11]. Cross-talk cancellation systems generally perform poorly at low frequencies because the inverse transfer function is then ill-conditioned. CAP can take over in this range, and has the advantage of not requiring calibration for the listener's head diameter.

An extension to Stereo-CAP for near-field images has been made by matching the low frequency ILD (Inter-aural Level Difference) to that of a near source. This is possible using complex panning gains realized with a 1st order filter [12].

For a low frequency spherical head model, [8], the condition that the ITD and ILD cues match with the target plane wave can be formulated as

$$\hat{\boldsymbol{r}}_R \cdot (\hat{\boldsymbol{r}}_I - \boldsymbol{r}_V) = 0 \tag{1}$$

where $\hat{\boldsymbol{r}}_I$ is the direction of the image, $\hat{\boldsymbol{r}}_R$ is the inter-aural axis, and $\boldsymbol{r}_V$ is the Makita vector that represents the sound field at low frequencies [13]. If the field is produced by panning, the waves at the listener can be approximated as plane waves provided the listener is not so close to the loudspeakers that near-field cues are significant. In this case the Makita vector is given by

$$\boldsymbol{r}_V = \frac{\sum g_i \hat{\boldsymbol{r}}_i}{\sum g_i} \tag{2}$$

where g_i are the gains of the source signal *at the listener*, and $\hat{\boldsymbol{r}}_i$ are the direction vectors of the loudspeakers relative to the listener [8]. The gains at the loudspeakers are compensated for the variable distance to the loudspeakers. Since the wave amplitude falls by $1/r$ the compensated loudspeaker gains are $r_i g_i$. Also delays are introduced to the loudspeaker feeds so that the signals at the listener are in phase. These compensations depend on accurate knowledge of the ambient speed of sound, as well as the distances.

Combining (1) and (2), and normalising the total gain, which determines the overall level, leads to expressions for Stereo-CAP gains,

$$g_1 = \frac{\hat{\boldsymbol{r}}_R \cdot (\hat{\boldsymbol{r}}_I - \hat{\boldsymbol{r}}_2)}{\hat{\boldsymbol{r}}_R \cdot (\hat{\boldsymbol{r}}_1 - \hat{\boldsymbol{r}}_2)} \quad g_2 = \frac{\hat{\boldsymbol{r}}_R \cdot (\hat{\boldsymbol{r}}_I - \hat{\boldsymbol{r}}_1)}{\hat{\boldsymbol{r}}_R \cdot (\hat{\boldsymbol{r}}_2 - \hat{\boldsymbol{r}}_1)} \tag{3}$$

These panning laws were tested objectively by calculating the resulting cues at different frequencies for a KEMAR dummy head [8]. The perceived directional error was then calculated and found to be within a Minimum Audible Angle (MMA) [14] for a wide range of target images and head orientations. Subjective tests were carried out to evaluate the stability of images in all directions. Dynamic head tracking was used to allow natural unrestricted listening. The tests showed that images between loudspeakers were improved, and further more steady images could now be created in directions away from the loudspeakers.

It is helpful to visualise the 3-dimensional vectors in the solution. Fig. 2 shows a plan view of these vectors. This is called a *Makita diagram* here since each point on this diagram corresponds to a value of $\boldsymbol{r}_V$, rather than a point in 3-dimensional space. The

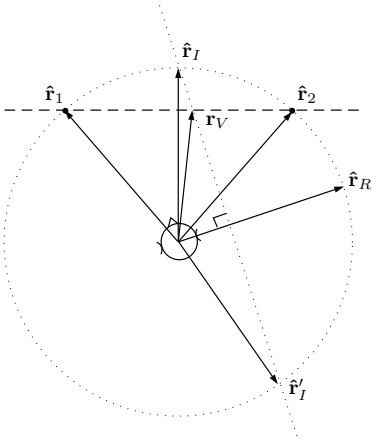


Figure 2: Makita diagram for Stereo CAP, in plan view, for a listener facing towards left of centre of the stereo array. The Makita vector is to the right of centre in order to keep the image central. Shown are loudspeaker directions $\hat{\boldsymbol{r}}_1$, $\hat{\boldsymbol{r}}_2$ the inter-aural direction $\hat{\boldsymbol{r}}_R$, image direction $\hat{\boldsymbol{r}}_I$ and Makita vector $\boldsymbol{r}_V$

dotted circle is a cross section through a sphere of radius 1. A point $\boldsymbol{r}_V$ on the circle or sphere corresponds to a plane wave, such as that from a distant loudspeaker or source. The dotted line represents a plane perpendicular to the page containing all the values of $\boldsymbol{r}_V$ of sound fields that produce an image $\hat{\boldsymbol{r}}_I$. The image is not unique, since there is a circle of consistent images, where the plane intersects with the sphere, the *cone of confusion*. The dashed line shows the values of $\boldsymbol{r}_V$ that can be produced by panning using the 2 loudspeakers. Where the plane and line cross is the single value of $\boldsymbol{r}_V$ that can produce the image using stereo panning. The method is valid whatever the direction of the image, even if it is behind or above.

The panning gains are positive for values of $\boldsymbol{r}_V$ between $\hat{\boldsymbol{r}}_1$ and $\hat{\boldsymbol{r}}_2$. Outside this region, one of the gains is negative, and there is cancellation of the pressure at the listener. The cancellation implies the sum of gain magnitudes $\sum |g_i|$ is greater than the sum of gains $\sum g_i$. Since the reproduction error due to each gain generally accumulates, then for given $\sum g_i$ the total error increases as the sum of gain magnitudes $\sum |g_i|$, and degree of cancellation. Reproduction error is due to inaccuracies in the head model, the audio hardware, and the tracking of the listener and loudspeakers.

If the listener faces towards the side, the plane and line become close to parallel, and the denominators vanish. The gains become large and polarised and the error increases. The common gain in the denominators can be limited, however this will reduce the perceived image level.

Introducing another loudspeaker between the existing pair would introduce more freedom for controlling $\boldsymbol{r}_V$,and the singular case can be avoided. Solutions for more than 2 loudspeakers are developed in the remainder of this article.

2. SOLUTIONS WITH MORE THAN 2 LOUDSPEAKERS

A Makita diagram with 3 loudspeakers is illustrated Fig. 3. Provided the loudspeakers direction vectors are distinct, then the producible values of $\boldsymbol{r}_V$ cover a plane containing $\hat{\boldsymbol{r}}_1, \hat{\boldsymbol{r}}_2, \hat{\boldsymbol{r}}_3$. The corresponding gains are positive for $\boldsymbol{r}_V$ in the triangular region inside these points, the *convex hull* of the points, and at least one gain is negative for each point outside. Two image examples are shown, each with a head superimposed to show the head orientation in order to simplify the picture in Fig. 2. The image direction and head orientation define the plane of permitted $\boldsymbol{r}_V$ values indicated by the dotted line. If the dotted and dashed planes intersect then

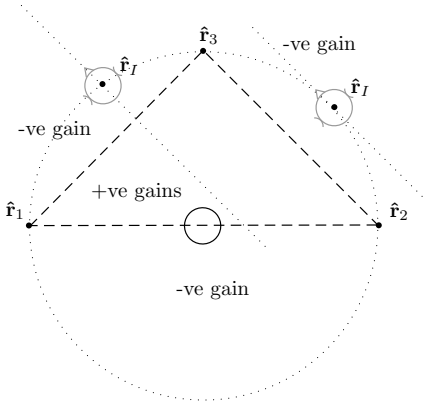


Figure 3: Makita diagram for CAP with 3 loudspeakers, in plan view. Shown are loudspeaker directions $\hat{\boldsymbol{r}}_1$, $\hat{\boldsymbol{r}}_2$, $\hat{\boldsymbol{r}}_3$, and two images $\hat{\boldsymbol{r}}_I$, each with associated head orientations.

there are possible solutions along the line of intersection. There are no solutions only when the planes are parallel and separated, which only happens when the inter-aural axis is perpendicular to

the loudspeaker plane, ie when one ear is pointing directly up. Different strategies can be considered for selecting from the possible solutions:

Localised energy : It is natural to try and localise loudspeaker energy in the directions where images are. In high frequency panning this reduces image spread, and makes images more compatible for multiple listeners is different locations. In the low frequency ITD range image spread is perceived much less, provided the cues are consistent, because the cues only contain directional information. The image on the left side in Fig. 3 has a localised solution where the dotted line crosses the dashed line between $\hat{\boldsymbol{r}}_1$ and $\hat{\boldsymbol{r}}_3$. The gain is zero for the other loudspeaker $g_2 = 0$. This is similar to a pairwise panning arrangement. However for the image on the right side there are no positive solutions. Solutions are possible with negative gain and either $g_2 = 0$ or $g_3 = 0$, but they are not localised to the target image. To move continuously between these solutions when the head rotates requires non-zero gain from all loudspeakers.

Least radiated energy: The energy radiated, $\sum r_i^2 g_i^2$, drives room reverberance that interferes with the direct signal at the listener. Reducing this energy reduces interference, and also the maximum power required from the loudspeakers. Although the precedence effect mitigates the localisation error caused by reverberance, it is desirable to minimise the reverberance because of its overall effect. A minimum energy solution will generally be spread over all the available loudspeakers. However, as explained above, spreading is not a primary concern in the low frequency ITD range.

Least direct energy: CAP may produce gains with opposite sign, and cancellation of pressures at the listener. As with the case of Stereo-CAP, cancellation implies the sum of gain magnitudes $\sum |g_i|$ is greater than $\sum g_i$, and the total reproduction error is increased. The energy sum $\sum g_i^2$ provides a measure of total error that captures the incoherent addition of errors, and is convenient to optimise. Minimising this quantity will minimise the reproduction error due to the direct signal. The solutions for least radiated energy and least cancellation error could be combined to give partial weight to each strategy. These solutions are the same when the distances r_i are equal. Note that $r_V >> 1$ implies cancellation and $\sum g_i^2 >> 1$, however $\sum g_i^2 >> 1$ is also possible for $r_V = 1$, for example in the case of Ambisonics.

Ambisonic: If the image direction $\hat{\boldsymbol{r}}_I$ is restricted to the plane containing the loudspeaker directions, then there is a solution $\boldsymbol{r}_V = \hat{\boldsymbol{r}}_I$ that is independent of head orientation. This is equivalent to Ambisonic panning based on mode matching of the sound field to first order [3, 15]. The low frequency cues depend only on the first order approximation. It is unusual to consider mode matching for full surround without loudspeakers behind the listener. Mathematically this is possible, but it is not immediately clear how well conditioned it is, and how much direct energy is needed.

2.1. Least energy solution

From the above discussion, the most useful solutions for general images are for the least radiated energy and the least direct energy. These solutions can be found analytically. This is shown first for the least radiated energy case. The least direct energy solution is then a special case of this.

Substituting (2) in (1) and multiplying by $\sum g_i$ gives the constraint

$$\sum g_i \, (\hat{\boldsymbol{r}}_R \cdot \hat{\boldsymbol{r}}_i) = \hat{\boldsymbol{r}}_R \cdot \hat{\boldsymbol{r}}_I \tag{4}$$

The summation range for the index i is omitted here and in the following. A second condition is needed to fix the level of the perceived image to a non-zero value, without which the gains would be minimized to zero. This is achieved by specifying the the incident pressure at the listener, which ensures the binaural signals will match those of a planewave with the same incident pressure. For a normalised level,

$$\sum g_i = 1 \tag{5}$$

The 2 constraints (4) and (5) can be combined to produce an alternative for constraint (4),

$$\sum g_i \, \alpha_i = 0 \, , \;\; \alpha_i = \; \hat{\boldsymbol{r}}_R \cdot (\hat{\boldsymbol{r}}_i - \hat{\boldsymbol{r}}_I) \tag{6}$$

where α_i is defined here for convenience. Using constraints (5) and (6) simplifies the gain formulae that will be derived. The least energy problem can be stated by minimising the total energy radiated by the loudspeakers,

$$\text{argmin}_{\{g_i\}} \sum (r_i g_i)^2 \tag{7}$$

subject to the previous constraints (6) and (5). This function and the conditions are smooth, so a closed solution is sought using Lagrange multipliers. The Lagrangian is

$$\mathcal{L} = \sum (r_i g_i)^2 - \lambda_1 \sum g_i \alpha_i - \lambda_2 (\sum g_i - 1) \tag{8}$$

with multipliers λ_1, λ_2. Setting partial derivatives by the unknown parameters to zero, $\partial\mathcal{L}/\partial g_i = 0, \partial\mathcal{L}/\partial\lambda_1 = 0, \partial\mathcal{L}/\partial\lambda_2 = 0$, produces $n + 2$ constraints, including the original 2 constraints, where n is the number of loudspeakers.

$$2 r_i^2 g_i - \lambda_1 \alpha_i - \lambda_2 \;\; = \;\; 0 \, , \; i = 1 \, .. \, n \tag{9}$$

$$\sum g_i \alpha_i \;\; = \;\; 0 \tag{10}$$

$$\sum g_i \;\; = \;\; 1 \tag{11}$$

From (9) the gains can be written

$$g_i = \frac{\lambda_1 \alpha_i + \lambda_2}{2 r_i^2} \tag{12}$$

Substituting the gains into (10),

$$\begin{aligned} \sum \frac{\lambda_1 \alpha_i + \lambda_2}{2 r_i^2} \alpha_i &= 0 \\ \lambda_1 \sum \frac{\alpha_i^2}{r_i^2} + \lambda_2 \sum \frac{\alpha_i}{r_i^2} &= 0 \\ \lambda_1 \gamma + \lambda_2 \beta &= 0 \end{aligned} \tag{13}$$

where $\beta = \sum \frac{\alpha_i}{r_i^2}$ and $\gamma = \sum \frac{\alpha_i^2}{r_i^2}$ are defined for convenience. Substituting the gains into (11),

$$\begin{aligned} \sum \frac{\lambda_1 \alpha_i + \lambda_2}{r_i^2} &= 2 \\ \lambda_1 \beta + \eta \lambda_2 &= 2 \end{aligned} \tag{14}$$

where $\eta = \sum \frac{1}{r_i^2}$. (13) and (14) can be solved simultaneously to find λ_1 and λ_2,

$$\lambda_1 = \frac{-2\beta}{\gamma\eta - \beta^2} \tag{15}$$

$$\lambda_2 = \frac{2\gamma}{\gamma\eta - \beta^2} \tag{16}$$

The resulting optimal gains are found by substituting into (12),

$$g_i = \frac{\gamma - \beta\alpha_i}{r_i^2(\gamma\eta - \beta^2)} \tag{17}$$

These gains are inexpensive to evaluate, which allows them to be updated frequently when the listener moves. The compensated loudspeaker gains are $r_i g_i$. A global gain factor can be added to set the reproduction level. The least direct energy solution can be found by setting all the loudspeaker distances $r_i = 1$. The least energy solution using 2 loudspeakers has to be identical to Stereo-CAP, because there can be only one solution. This can also be checked algebraically by simplifying (17) for the case $n = 2$. Like Stereo-CAP, it is possible to extend the least energy solution for near-field images, although this is not shown here.

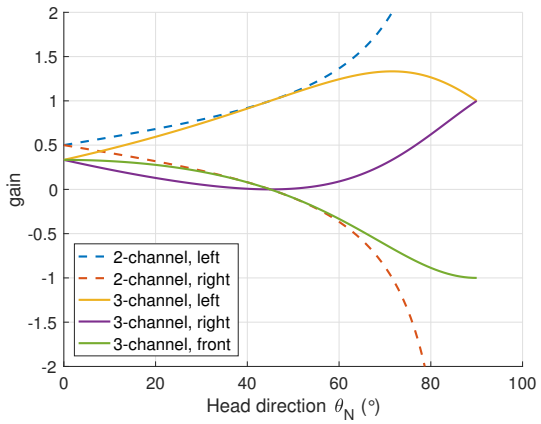


Figure 4: Gains for Stereo-CAP and 3-way CAP for an image at 180° azimuth, and a range of head directions.

The plots shown in Fig. 4 compare the gains produced by the Stereo-CAP system with the least energy 3-way CAP system. Head direction is varied, and the image is directly behind. The Stereo loudspeakers are directly to the left and right. The 3-way system has loudspeakers in these positions and an extra one directly in front, the same as Fig. 3. When the listener turns to the side the Stereo-CAP gains become large, whereas the 3-way CAP gains have magnitudes similar to the total gain $\sum g_i = 1$.

Adding a 4th loudspeaker that is not coplanar with the others, for example above the front loudspeaker in the example shown in Fig. 3, increases the space of $\boldsymbol{r}_V$ that can be produced by panning, from a plane to the whole 3-dimensional Makita space. The panning gains are all positive for points inside the convex hull described by the 4 loudspeaker direction vectors, and at least one gain is negative for each point outside this region. The intersection of the whole space with the plane described by the ITD constraint is always non empty, so there are no singular configurations.

The multichannel solution can be used with any number of loudspeakers. While an advantage of the CAP system is that it requires only a few loudspeakers, more loudspeakers can be added to progressively reduce the radiated energy. Effectively this is beam forming focused on the listener.

The subjective performance of least energy CAP with more than 2 loudspeakers can be inferred from the objective and subjective results for the 2-channel case [8]. These results show that an upper bound for the subjective localisation error can be given that depends only on the total gain energy $\sum g_i^2$. From this the given 3-channel case the total energy is sufficiently low, across all common configurations of image and listener, so that the inferred error is within an MMA. This also implies reverberant interference is at least as low as the 2-channel test cases, for which reverberance could be heard but did not affect image localisation.

2.2. Ambisonic solution

In the mode-matched Ambisonic approach, the aim is to produce an image by reproducing the associated sound field. To produce an accurate low frequency ITD cue it is enough to reproduce pressure and velocity, forming the 1st order of approximation. The first order problem can be written in terms of the variables used in this article by combining (2) and (5) into a single matrix equation,

$$\begin{bmatrix} 1 & 1 & 1 \,.. \\ \hat{\boldsymbol{r}}_1 & \hat{\boldsymbol{r}}_2 & \hat{\boldsymbol{r}}_3 \,.. \end{bmatrix} \begin{bmatrix} g_1 \\ g_2 \\ g_3 \\ . \\ . \end{bmatrix} = \begin{bmatrix} 1 \\ \boldsymbol{r}_V \end{bmatrix} \tag{18}$$

Or, abreviated,

$$\mathbf{R}\mathbf{g} = \mathbf{s} \tag{19}$$

The least energy solution, where it exists, is given using the pseudoinverse

$$\mathbf{g} = \mathbf{R}^+\mathbf{s} \tag{20}$$

$\mathbf{R}^+$ is the Ambisonic decoding matrix. For the example shown in Fi. 4, with 3 loudspeakers and an image behind,

$$\mathbf{s} = \begin{bmatrix} 1 \\ -1 \\ 0 \\ 0 \end{bmatrix}, \quad \mathbf{R} = \begin{bmatrix} 1 & 1 & 1 \\ 0 & 0 & 1 \\ 1 & -1 & 0 \\ 0 & 0 & 0 \end{bmatrix} \tag{21}$$

$$\mathbf{R}^+ = \begin{bmatrix} 1/2 & -1/2 & 1/2 & 0 \\ 1/2 & -1/2 & -1/2 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}, \quad \mathbf{g} = \begin{bmatrix} 1 \\ 1 \\ -1 \end{bmatrix} \tag{22}$$

The ordering of loudspeakers here is left, right then centre. Although there are no rear loudspeakers, the target image can be produced without excessive gains or cancellation in this case. However if the listener position is set further back, so that the loudspeakers are separated by smaller angles relative to the listener, then the gains for rear images increase rapidly in size and there is more cancellation. For example if the left and right loudspeakers are positioned closer at $-30°, +30°$ then the gains producing a rear image are

$$\mathbf{g} = \begin{bmatrix} 7.46 \\ 7.46 \\ -13.93 \end{bmatrix} \tag{23}$$

Using CAP the gains in this case are small when the listener is facing forward,

$$\mathbf{g} = \begin{bmatrix} 0.33 \\ 0.33 \\ 0.33 \end{bmatrix} \tag{24}$$

Assuming equal loudspeaker distances, the total energy radiated by the Ambisonic array is 916 times greater than that for CAP. The CAP gain magnitudes generally increase smoothly as the listener turns their head to the side, and are equal to the Ambisonic gains when the listener faces directly to the sides.

Adding a 4th loudspeaker that is not coplanar with the others, for example above the centre loudspeaker in the example shown in Fig. 3, allows gains to be produced for any image direction, using the Ambisonic method. Comparatively high gains are required when the loudspeakers are positioned more closely, as for the 3 loudspeaker case.

3. CONCLUSION

Using 2 loudspeakers and with full 6-degrees-of-freedom head tracking, position and orientation, it was previously shown possible to create low frequency images in any direction, although excessive gain is required for some listener orientations . Here it was shown that with 3 loudspeakers all images directions can be reproduced with moderate gain except for a small range of orientations that are practically unimportant. Alternatively, taking an Ambisonic approach with position tracking, 3 frontal loudspeakers can reproduce horizontal images, and 4 loudspeakers can reproduce images in any 3D direction. Ambisonics does not require orientation tracking. As loudspeaker separation is reduced Ambisonics suffers from rapidly increasing gains and cancellation for forward head directions and rear images, whereas in this case CAP gains are lower and remain low as separation is reduced. The overall CAP energy can be reduced further by increasing the number of loudspeakers. In the light of the objective and subjective results for the 2-channel case, the multichannel CAP gains mean the localisation accuracy is within an MMA for all common configurations for the test array considered.

The most recent real-time implementation of the multichannel CAP system is based on an extensive and flexible C++ / Python framework for spatial sound rendering, called the *Versatile Interactive Software Rendering framework (VISR)*. It is planned to make this publicly available in due course.

4. ACKNOWLEDGMENT

This work was supported by the Engineering and Physical Sciences Research Council (EPSRC) "S3A" Programme Grant EP/L000539/1, and the BBC Audio Research Partnership. No new data was created in this work.

5. REFERENCES

[1] Jens Blauert, *Spatial hearing*, Cambridge, MA: MIT Press, 1997.

[2] Benjamin Bernfeld, "Attempts for better understanding of the directional stereophonic listening mechanism," in *Audio Engineering Society Convention 44*, March 1973, number C-4.

[3] Michael Anthony Gerzon, "General metatheory of auditory localisation," in *92nd Audio Engineering Society Convention, Vienna*, 1992, number 3306.

[4] Ville Pulkki, "Compensating displacement of amplitude-panned virtual sources," in *Audio Engineering Society Conference: 22nd International Conference: Virtual, Synthetic, and Entertainment Audio*, Jun 2002.

[5] Hans Wallach, "On sound localization," *J. Acoust. Soc. Am*, vol. 10, pp. 270–274, 1939.

[6] Hans Wallach, "The role of head movements and vestibular and visual cues in sound localization.," *Journal of Experimental Psychology*, vol. 27, no. 4, pp. 339, 1940.

[7] Bosun Xie and Dan Rao, "Analysis and experiment on summing localization of two loudspeakers in the median plane," in *Audio Engineering Society Convention 139*. Audio Engineering Society, 2015.

[8] Dylan Menzies, Marcos F. Simon Galvez, and Filippo Maria Fazi, "A low frequency panning method with compensation for head rotation," *IEEE Trans. Audio, Speech, Language Processing*, vol. 26, no. 2, February 2018.

[9] Jean-Marie Pernaux, Patrick Boussard, and Jean-Marc Jot, "Virtual sound source positioning and mixing in 5.1 implementation on the real-time system genesis," in *Proc. Conf. Digital Audio Effects (DAFx-98)*. Citeseer, 1998, pp. 76–80.

[10] Bishnu S. Atal and Manfred R Schroeder, "Apparent sound source translator," Feb. 22 1966, US Patent 3,236,949.

[11] Ole Kirkeby, Philip A. Nelson, and Hareo Hamada, "Virtual source imaging using the stereo dipole," in *Audio Engineering Society Convention 103*, Sep 1997.

[12] Dylan Menzies and Filippo Maria Fazi, "Spatial reproduction of near sources at low frequency using adaptive panning," in *Proc. TecniAcustica, Valencia*, October 2015.

[13] Y Makita, "On the directional localization of sound in the stereophonic sound field," *E.B.U Review*, vol. A, no. 73, pp. 102–108, 1962.

[14] Allen William Mills, "On the minimum audible angle," *The Journal of the Acoustical Society of America*, vol. 30, no. 4, pp. 237–246, 1958.

[15] J. Daniel, "Spatial sound encoding including near field effect," in *Proc. AES 23nd International Conference, Helsinger, Denmark*, 2003.

A FEEDBACK CANCELING REVERBERATOR

Jonathan S. Abel, Eoin F. Callery, and Elliot K. Canfield-Dafilou

Center for Computer Research in Music and Acoustics,
Stanford University, Stanford, CA 94305 USA
abel|ecallery|kermit@ccrma.stanford.edu

ABSTRACT

A real-time auralization system is described in which room sounds are reverberated and presented over loudspeakers. Room microphones are used to capture room sound sources, with their outputs processed in a canceler to remove the synthetic reverberation also present in the room. Doing so suppresses feedback and gives precise control over the auralization. It also allows freedom of movement and creates a more dynamic acoustic environment for performers or participants in music, theater, gaming, and virtual reality applications. Canceler design methods are discussed, including techniques for handling varying loudspeaker-microphone transfer functions such as would be present in the context of a performance or installation. Tests in a listening room and recital hall show in excess of 20 dB of feedback suppression.

1. INTRODUCTION

Real-time virtual acoustic/auralization systems have been made possible by advances in signal processing and acoustics measurement. Computational methods for simulating reverberant environments are well developed, and these auralization systems process sound sources according to impulse responses encapsulating the acoustics of the desired space and render them over loudspeakers in the venue or through headphones [1, 2]. In live and recoding settings, close mics or contact mics are commonly used for acoustic instruments and voice to avoid feedback. Such mic'ing can be cumbersome and can affect or restrict performances. In virtual, augmented, or mixed reality settings, the immersive audio possibilities are similarly restricted by the use of headphones. Moreover, in all of these situations, unless the locations of the sound sources are tracked, movement in the virtual space will not be reflected in the experienced auralization.

In recent years, several systems which use room microphones and loudspeakers have been developed to create virtual reverberant auralizations. These include products by Meyer Sound [3] and Lexicon [4] as well as the system designed by Woszczyk [5] at McGill. See [6] for a more extensive review. In such systems, a number of approaches have been used to suppress feedback, including adaptive notch filtering to detect and suppress individual frequencies as they initiate feedback [7], frequency shifting the synthesized acoustics [8], varying the synthesized acoustics over time [4], and decorrelating the various auralization impulse responses [9, 10]. Such processing compromises the original dry signals. In addition, to provide the needed control and to achieve the best possible performance, these systems are typically built from the ground up using proprietary hardware and software. Accordingly, they do not take advantage of existing loudspeaker and microphone arrays already present on site. Ultimately, this makes these systems expensive, involving significant alteration to the installation site, and requiring prolonged calibration and tuning.

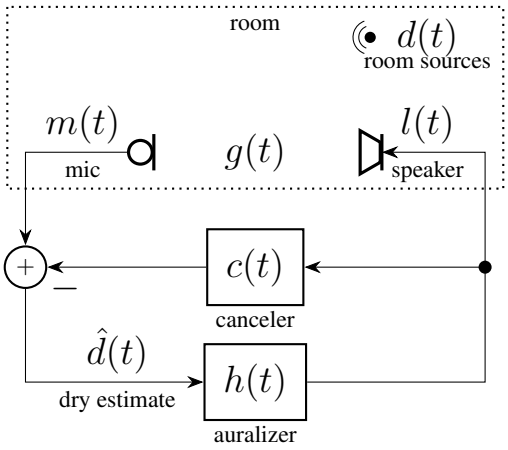


Figure 1: Feedback Canceling Auralization System. Room sounds are convolved with an auralization impulse response $h(t)$, generating simulated acoustics $l(t)$ which are projected into the room via a loudspeaker. A room microphone captures both room sounds and simulated acoustics $m(t)$, and is processed according to measurements of the loudspeaker-microphone transfer function to remove the simulated acoustics, thus leaving an estimate of the room sounds $\hat{d}(t)$ to be auralized.

Here, we present a system for real-time auralization that uses standard room microphones and loudspeakers, and employs signal processing tools to cancel the feedback, thus eliminating the need for close or contact microphones. The cancellation method described here is similar to the adaptive noise cancellation approach developed by Widrow [11] for removing unwanted additive noise from a signal. In that approach, a reference signal, which is correlated with the unwanted noise, is used to estimate and subtract the unwanted noise from the primary signal. Related literature also includes echo cancellation and dereverberation [12–14].

The system we describe can also be integrated into existing speaker arrays as it does not requires proprietary hardware and can be implemented using inexpensive and readily available software. The system is designed to be easy to configure and straightforward to calibrate. The ease of use and mobility afforded by not requiring close mic'ing creates opportunities for dynamic artistic experiences for performers and audiences in disciplines such as music, theater, dance, and emerging digital art forms [15]. For example, in virtual, augmented, and mixed reality scenarios, the system allows users to dispense with headphones for more immersive virtual acoustic experiences.

In the sequel, the system and cancellation processing are described. Example applications and a performance analysis follow.

2. AURALIZATION SYSTEM

We begin by describing the auralization system, which is similar to the recording processing described in [16].

Referring to Fig. 1, a room microphone captures contributions from room sound sources $d(t)$ and synthetic acoustics produced by the loudspeaker according to its applied signal $l(t)$, with t being the discrete time sample index. One can impart the sonic characteristic of a space, $h(t)$, on the room sounds $d(t)$ through convolution,

$$l(t) = h(t) * d(t)\,. \tag{1}$$

Many auralization systems work this way, using fast, low-latency convolution methods to save computation [17–19]. The difficulty is that the room source signals $d(t)$ are not directly available. As described above, the room microphones also pick up the synthesized acoustics, and would cause feedback if the room microphone signal $m(t)$ were reverberated without additional processing.

Here, we auralize an estimate of the dry signal $\hat{d}(t)$, formed by subtracting from the microphone signal $m(t)$ an estimate of the synthesized acoustics. Assuming the geometry between the loudspeaker and microphone is unchanging, we have

$$d(t) = m(t) - g(t) * l(t), \tag{2}$$

where $g(t)$ is the impulse response between the loudspeaker and microphone. Here, we design an impulse response $c(t)$, which approximates the loudspeaker-microphone response, and use it to form an estimate of the "dry" signal $\hat{d}(t)$,

$$\hat{d}(t) = m(t) - c(t) * l(t). \tag{3}$$

This is shown in the signal flow diagram Fig. 1: the synthetic acoustics are canceled from the microphone signal $m(t)$ to estimate the room signal $\hat{d}(t)$, which is then reverberated.

2.1. Canceler Design

The question then becomes how to design the canceling filter $c(t)$. A measurement of the impulse response $g(t)$ provides an excellent starting point, though there are time-frequency regions over which the response is not well known due to measurement noise (typically affecting the low frequencies) or changes over time due to air circulation or performers, participants, or audience members moving about the space (typically later in the impulse response). In regions where the impulse response is not well known, the cancellation should be reduced so as to not introduce additional reverberation.

Here, we choose the cancellation filter impulse response $c(t)$ to minimize the expected energy in the difference between the actual and estimated room microphone loudspeaker signals. For simplicity of presentation, for the moment let us assume that the loudspeaker-microphone impulse response is a unit pulse,

$$g(t) = g\,\delta(t), \tag{4}$$

and that the impulse response measurement $\tilde{g}(t)$ is equal to the sum of the actual impulse response and zero-mean noise with variance σ_g^2. Consider a canceling filter $c(t)$ which is a windowed version of the measured impulse response $\tilde{g}(t)$,

$$c(t) = w\,\tilde{g}\,\delta(t)\,. \tag{5}$$

In this case, the measured impulse response $\hat{g}(t)$ is scaled according to a one-sample-long window w. The expected energy in the difference between the auralization and cancellation signals at time t is:

$$\mathbb{E}\left[(g\,l(t) - w\,\tilde{g}\,l(t))^2\right] = l^2(t)\left[w^2\sigma_g^2 + g^2(1-w)^2\right]. \tag{6}$$

Minimizing the residual energy over the window w, we find

$$c^*(t) = w^*\,\tilde{g}\,\delta(t), \quad w^* = \frac{g^2}{g^2 + \sigma_g^2}, \tag{7}$$

a Wiener-like weighting of the measured impulse response. When the loudspeaker-microphone impulse response magnitude is large compared with the impulse response measurement uncertainty, the window w will be near 1, and the cancellation filter will approximate the measured impulse response. By contrast, when the impulse response is poorly known, the window w will be small—roughly the measured impulse response signal-to-noise ratio—and the cancellation filter will be attenuated compared to the measured impulse response. In this way, the optimal cancellation filter impulse response is seen to be the measured loudspeaker-microphone impulse response, scaled by a compressed signal-to-noise ratio (CSNR).

Typically, the loudspeaker-microphone impulse response $g(t)$ will last hundreds of milliseconds, and the window that scales the measured impulse response will preferably be a function of time t and frequency f so as to account for changes in impulse response variance over time and frequency. Denote by $\tilde{g}(t, f_b), b = 1, 2, \ldots N$ the measured impulse response $\tilde{g}(t)$ split into a set of N discrete frequency bands f_b using a filterbank such that the sum of the band responses is the original measurement,

$$\tilde{g}(t) = \sum_{b=1}^{N} \tilde{g}(t, f_b). \tag{8}$$

In this case, the canceler response $c^*(t)$ is the sum of measured impulse response bands $\tilde{g}(t, f_b)$, scaled in each band by a corresponding window $w^*(t, f_b)$. Expressed mathematically,

$$c^*(t) = \sum_{b=1}^{N} c^*(t, f_b), \tag{9}$$

where

$$c^*(t, f_b) = w^*(t, f_b)\,\tilde{g}(t, f_b), \tag{10}$$

$$w^*(t, f_b) = \frac{g^2(t, f_b)}{g^2(t, f_b) + \sigma_g^2(t, f_b)}. \tag{11}$$

We suggest using the measured impulse response bands $\tilde{g}(t, f_b)$ as stand-ins for the actual impulse response bands $g(t, f_b)$ in computing the optimal window $w^*(t, f_b)$. In addition, repeated measurements of the impulse response $g(t, f_b)$ could be made, with the measurement mean used for $g(t, f_b)$, and the variation in the impulse response measurements as a function of time and frequency used to form $\sigma_g^2(t, f_b)$. We also suggest smoothing $g^2(t, f_b)$ over time and frequency in computing $w(t, f_b)$ so that the window is a smoothly changing function of time and frequency.

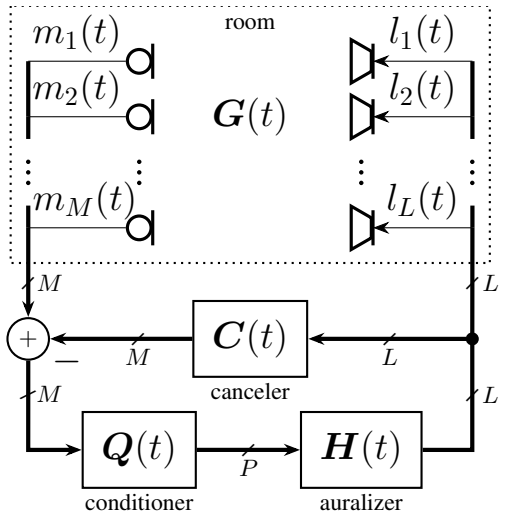


Figure 2: Canceling auralizer using multiple loudspeakers and microphones. Multiple loudspeakers and microphones can be accommodated in this auralizer architecture by estimating the matrix of loudspeaker-microphone transfer functions, $\mathbf{G}(t)$. Additionally, the room sound estimates may be processed using beamforming or other techniques before being diffused about the space.

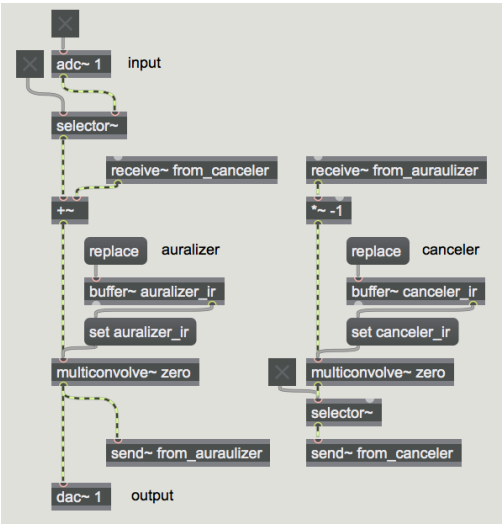


Figure 3: Max/MSP patch showing one possible implementation of the auralization system.

2.2. Multiple Microphones and Speakers

In the presence of L loudspeakers and M microphones, a matrix of loudspeaker-microphone impulse responses is measured, and used in subtracting auralization signal estimates from the microphone signals. Stacking the microphone signals into an M-tall column $\boldsymbol{m}(t)$, and the loudspeaker signals into an L-tall column $\boldsymbol{l}(t)$, our cancellation system becomes

$$\boldsymbol{l}(t) = \boldsymbol{H}(t) * \hat{\boldsymbol{d}}(t)\,, \tag{12}$$

$$\hat{\boldsymbol{d}}(t) = \boldsymbol{m}(t) - \boldsymbol{C}(t) * \boldsymbol{l}(t)\,, \tag{13}$$

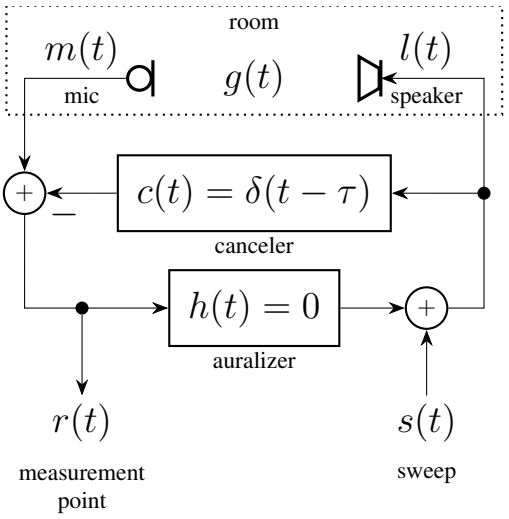


Figure 4: Canceling Auralizer Calibration. The cancellation processing $c(t)$ may be determined by measuring the impulse response between the loudspeaker and microphone, simultaneously with the response through $c(t)$.

where $\boldsymbol{H}(t)$ is the matrix of auralizer filters and $\boldsymbol{C}(t)$ the matrix of canceling filters. As in the single-speaker single-microphone case, the canceling filter matrix is the matrix of measured impulse responses, each windowed according to its respective CSNR.

Moreover, a conditioning processor, $\boldsymbol{Q}$, can be inserted between the microphones and auralizers,

$$\boldsymbol{l}(t) = \boldsymbol{H}(t) * \boldsymbol{Q}\left(\hat{\boldsymbol{d}}(t)\right)\,, \tag{14}$$

$$\hat{\boldsymbol{d}}(t) = \boldsymbol{Q}\left(\hat{\boldsymbol{d}}(t)\right) - \boldsymbol{C}(t) * \boldsymbol{l}(t)\,, \tag{15}$$

as seen in Fig. 2. This processor could serve several functions. First, $\boldsymbol{Q}$ could act as the weights of a mixing matrix to determine how the microphones signals are mapped to the auralizers, and subsequently, the loudspeakers. For example, it might be beneficial for microphones that are on one side of the room to send the majority of their energy to loudspeakers on the same side of the room, as could be achieved using a B-format microphone array and Ambisonics processing driving the loudspeaker array. Another use could be for when the speaker array and auralizers are used to create different acoustics in different parts of the room. The processor $\boldsymbol{Q}$ could also be a beamformer or other microphone array processor to auralize different sounds differently according to their source position. In such a situation, $\boldsymbol{Q}$ could change the dimensionality of the M microphone signals into P signals which are then auralized. It is worth noting that depending on the purpose, $\boldsymbol{Q}$ could a matrix of weights, a matrix of convolutions, a combination of the two, or other processor.

3. IMPLEMENTATION AND EVALUATION

3.1. MaxMSP Implementation and System Calibration

The signal flow of Fig. 2 is straightforward to implement in any number of environments. A Max/MSP implementation of a single-

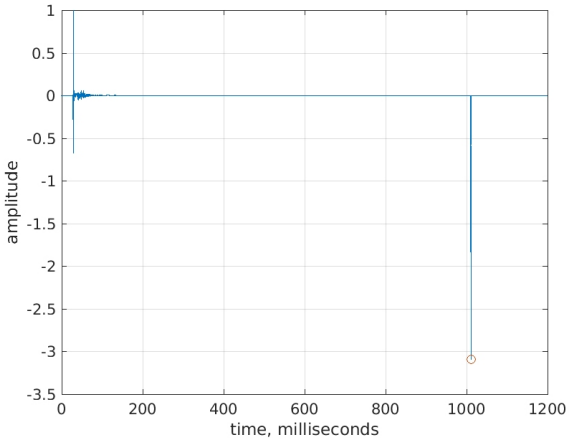


Figure 5: Cancellation Processing Design. The cancellation processor reproduces the impulse response between the loudspeaker and microphone, accounting for the scaling and delay experienced through the canceler convolution.

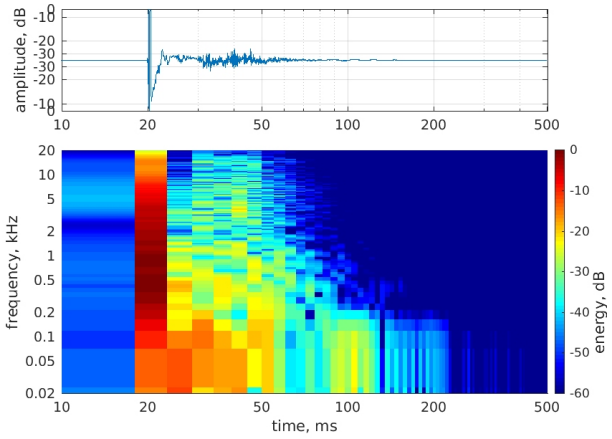


Figure 6: Example Cancellation Impulse Response. A cancellation impulse response $c(t)$ (top) and its associated spectrogram (bottom) are shown for the Listening Room at CCRMA, Stanford University, configured with a ceiling-mounted full-range loudspeaker and hanging omnidirectional microphone.

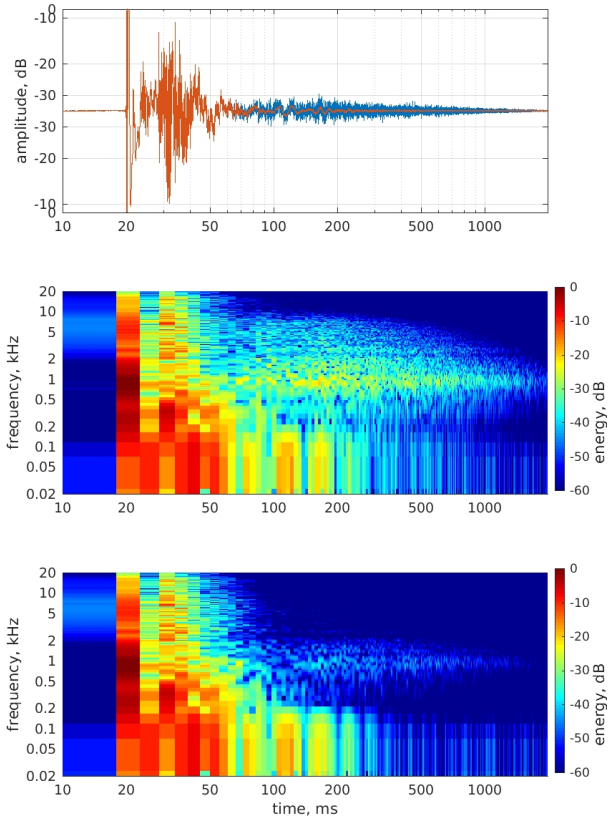


Figure 7: Canceling Auralizer Room Impulse Response. A sine sweep from a separate loudspeaker in the room was used to measure the impulse response between a room source and the canceling reverberator system microphone input (top, blue), and system room source estimate (top, orange). The corresponding spectrograms are also shown (middle and bottom). Note that the room impulse response contains both the "dry" room response and the "wet" synthesized room acoustics (Memorial Church at Stanford University), while the estimated room source response shows a substantially drier signal.

microphone, single-loudspeaker canceling auralizer is shown in Fig. 3. We use [20] for fast convolution.

To calibrate the system, the canceler impulse response $c(t)$ was set to a delayed pulse and the impulse response of the system configured as shown in Fig. 4 was used to determine the scaling and delay through the Max/MSP patch and to measure the loudspeaker-microphone transfer function. An example result, using a Sennheiser MKH 20-P48 omnidirectional microphone placed about 50 cm from an Adam A8X full-range loudspeaker is shown in Fig. 5. To find $c(t)$, the measured impulse response $\tilde{g}(t)$ is shifted and scaled according to the amplitude and arrival time of the $c(t) = \delta(t - \tau)$ pulse. An example canceler impulse response is shown in Fig. 6. Finally, note that an optimal window may be applied according to the discussion above by making a number of measurements, and estimating the variance of the measured impulse responses as a function of time and frequency.

3.2. Performance Evaluation

It is useful to anticipate the effectiveness of the virtual acoustics cancellation in any given microphone. Substituting the optimal windowing (7) into the expression for the canceler residual energy (6), the virtual acoustics energy in the canceled microphone signal is expected to be scaled by a factor of

$$\nu = \frac{\sigma_g^2}{g^2 + \sigma_g^2}, \tag{16}$$

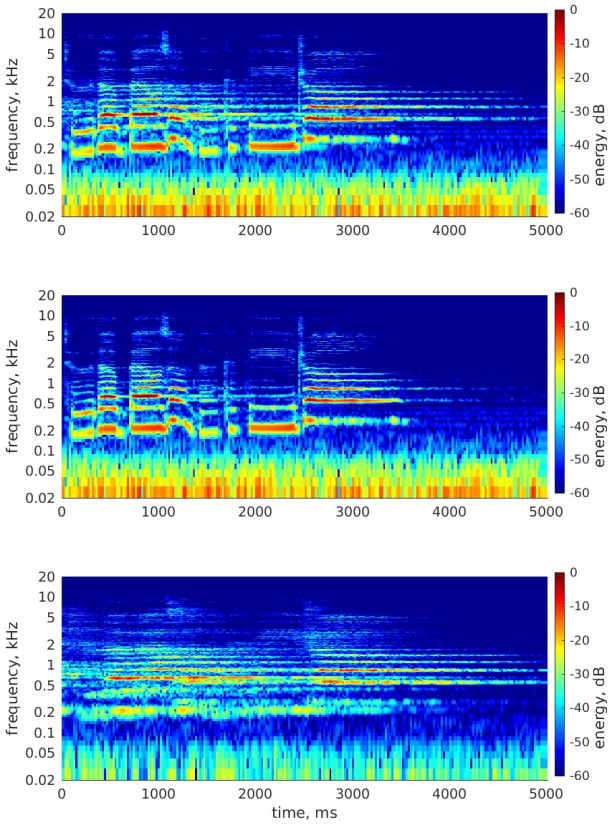


Figure 8: Canceling Auralizer Example. A dry source, Suzanne Vega's "Tom's Diner," was played in the CCRMA Listening Room, configured with the canceling auralizer described here. Spectrograms are shown for the microphone signal (top), the room signal estimate (middle), and the synthetic acoustics projected into the room (bottom). The room signal estimate contains little of the synthetic reverberation, and is effectively a mix of the dry Suzanne Vega track and low-frequency ventilation noise present in the room.

compared to that in the original microphone signal. Note that the reverberation-to-signal energy ratio is improved in proportion to the measurement variance for accurately measured signals, $\sigma_g^2 \ll g^2$. By contrast, when the impulse response is inaccurately measured, the reverberation-to-signal energy ratio is nearly unchanged, $\nu \approx 1$.

To evaluate the performance of the system, we implemented several versions of the system shown in Fig. 2 with one–two microphones and one–four loudspeakers in the CCRMA Listening Room and CCRMA Stage recital hall at Stanford University. We used a single loudspeaker source, playing exponentially swept sinusoid test signals and Suzanne Vega's "Tom's Diner" as dry program material. This was selected as it often used to test reverberators and makes for a repeatable test signal.

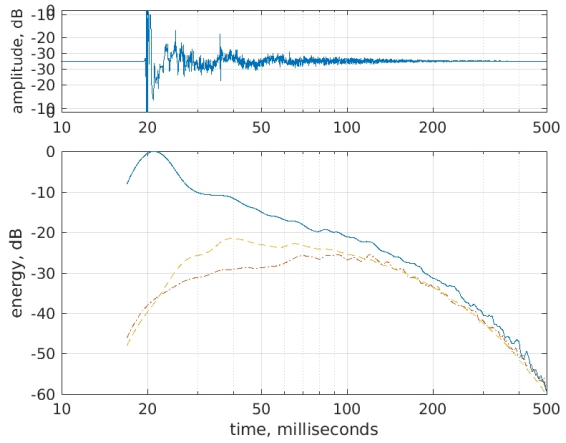


Figure 9: Room Impulse Response Variation. The mean room impulse response (top) formed from 1145 sine sweep responses measured between a loudspeaker and microphone mounted in the CCRMA Stage, a 120-seat recital hall at Stanford University that was unoccupied during the measurement. The impulse response energy, smoothed over a 10-millisecond-long Hanning window, is also shown (bottom, solid), along with the smoothed sample standard deviation (bottom, dashed). The smoothed sample standard deviation is also shown for a set of 75 measurements made with a dozen occupants near the loudspeaker and microphone, and in different positions for each measurement (bottom, dotted). Note that the impulse response variation is smallest relative to the impulse response energy near the beginning of the impulse response, and that the variation for the occupied room is modestly larger as the room becomes mixed.

In a first test, the impulse response of the room with the system active is measured. As seen in Fig. 7, the room impulse response contains both the "dry" room response and the "wet" synthesized room acoustics of Memorial Church at Stanford University. The 4.5 s reverberation time is plainly visible. Also shown in Fig. 7 is the system dry signal estimate, $\hat{d}(t)$. Compared to the virtual room impulse response, the canceler produces a substantially dry signal, canceling in excess of 30 dB of the simulated reverberation.

Fig. 8 shows the response of the system to a dry source, Vega's "Tom's Diner." Spectrograms are shown for the microphone signal, the room signal estimate, and the synthetic acoustics projected into the room. Note that the room signal estimate contains little of the synthetic reverberation, and is effectively a mix of the dry Suzanne Vega track, and low-frequency ventilation noise present in the room. As expected, the room response shows the imprint of the Memorial Church acoustics, as added by the system.

To better understand the practical performance of the system, repeated measurements of the loudspeaker-microphone response were made at the CCRMA Stage in unoccupied and occupied conditions. Fig. 9 shows the mean room impulse response and the impulse response energy, smoothed over a 10-millisecond-long Hanning window. The sample standard deviation is shown separately for the unoccupied and occupied conditions. The impulse response variation is smallest relative to the impulse response energy near the beginning of the impulse response. Also, the variation for the occupied room is modestly larger as the room becomes mixed. As

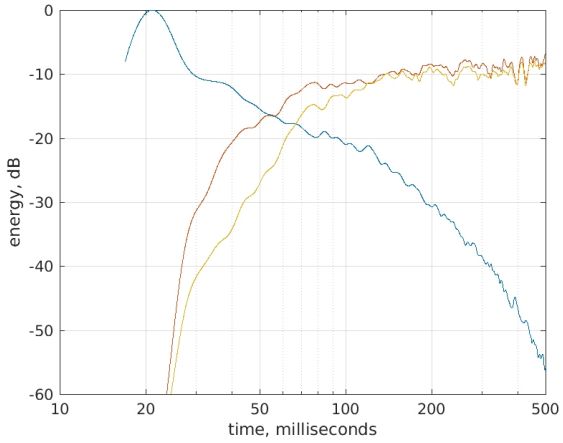


Figure 10: Canceler Performance Example. The smoothed energy of the mean loudspeaker-microphone impulse response is shown (blue), as is the residual energy of suppressed loudspeaker signals for the unoccupied (yellow) and occupied (orange) rooms. Note that the cancellation is most effective at the impulse response start, during which there is little variation, cf. Fig. 9. Note also that the occupied room has a slightly larger residual energy as the room is becoming well mixed.

seen in Fig. 10, the canceler residual energy is small near the beginning of the response, and increases relative to the decreasing impulse response energy throughout the response, consistent with the notion that the beginning of the impulse response shows little variation. As seen in Fig. 11, the canceler residual energy is also small for frequencies below about 2 kHz. Over the speech band of 200 Hz–3200 Hz, the residual simulated acoustics energy present in the room signal estimate $\hat{d}(t)$ was 16.4 dB for the occupied CCRMA Stage with moving participants, and 24.3 dB for the unoccupied CCRMA Stage.

Finally, we present an example of the ability of the system to suppress feedback resulting from creating a wet synthetic acoustic environment. Fig. 12 shows a spectrogram of a recording of the canceling auralizer simulating Memorial Church, along with the spectrogram of the same recording, but with the canceler component of the system switched off, and then switched back on. Note the rapid build up and subsequent suppression of feedback with the temporary removal of the cancellation processing.

4. CONCLUSIONS AND FUTURE WORK

In this paper, we have shown a real-time auralization system capable of generating multiple auralizations while canceling synthetic reverberation with greater than 20 dB of feedback suppression. We have also shown that the system can be calibrated and integrated into an existing speaker array, using currently available room microphones to pick up live sounds, and function in real-time by running with off-the-shelf software. Importantly, our system allows flexible and dynamic experiences for performers, audiences, and other users. In theatrical, musical, or other live performance situations, this system does not require performers to wear microphones, transmitters, or battery packs in order to be processed through artificial reverberation, thus expanding performance possibilities. Similarly, in emergent virtual, augmented, or mixed reality settings, such as those one might find in industrial simulations, home entertainment systems, and artistic installations, our system does not require use of headphones to facilitate immersive auralizations.

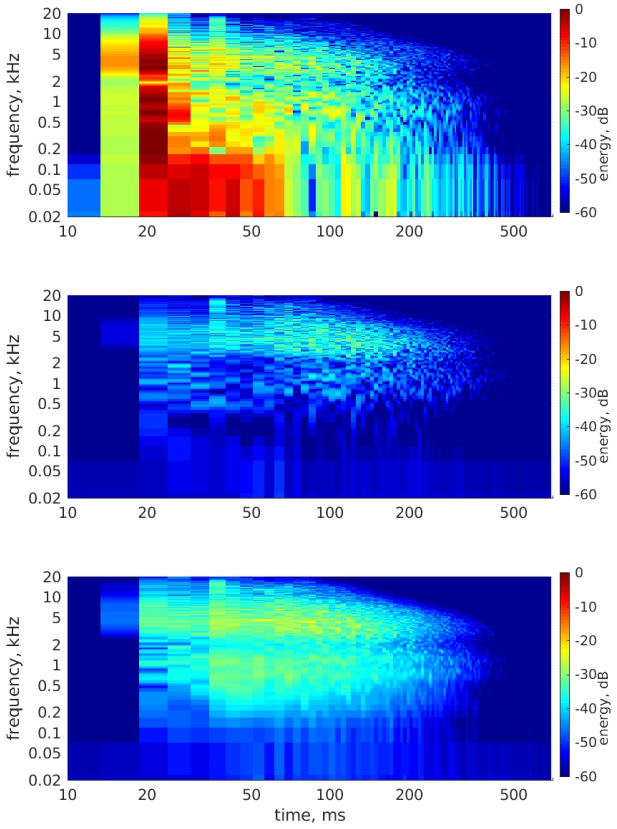


Figure 11: Canceler Performance Example, Residual Energy. The loudspeaker-microphone impulse response spectrogram (top) is shown along with the root-mean-square canceler residual for the unoccupied CCRMA Stage (middle) and occupied CCRMA Stage (bottom). Note that a substantial amount of the loudspeaker energy has been canceled, particularly at the impulse response beginning and for frequencies below about 2 kHz.

We have tested our system in several rooms at CCRMA, Stanford University. Additionally, the system has been used in a series of network-audio concerts between Stanford University and Stockholm, Sweden [21]. We are planning to continue to develop our system for further electroacoustic music works, for virtual reality and virtual acoustic research, music and theatrical rehearsals and performances, art installations, and for other academic and industrial research projects at Stanford University. In particular, we are installing a larger system (4 microphones and 8–16 loudspeakers) for a study of performance practice using vocal repertoire written for Rome's *Chiesa di Sant'Aniceto* using impulse responses recorded in Rome during 2017–18, [22, 23].

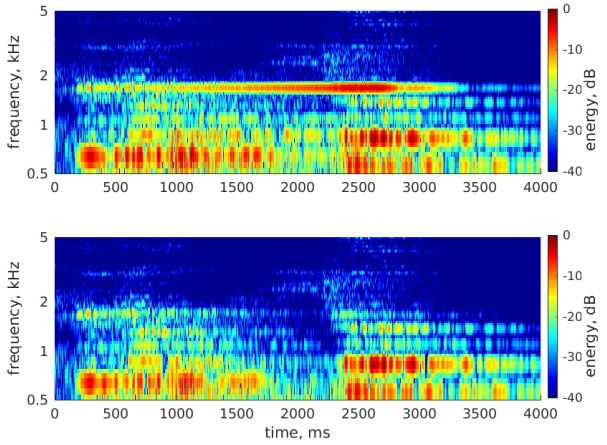


Figure 12: Feedback Example. A spectrogram of a recording of the canceling auralizer simulating the 5-second-long reverberation of Memorial Church at Stanford University is shown (bottom), along with the spectrogram of the same segment, but with the canceler component of the system switched off near 500 ms, and switched back on a little after 3000 ms (top). Note the rapid build up and subsequent suppression of feedback near 1800 Hz with the temporary removal of the cancellation processing.

5. REFERENCES

[1] Michael Vorländer, *Auralization: Fundamentals of Acoustics, Modelling, Simulation, Algorithms and Acoustic Virtual Reality*, Springer, 2007.

[2] Mendel Kleiner, Bengt-Inge DalenBack, and Peter Svensson, "Auralization—an overview," *Journal of the Acoustical Society of America*, vol. 41, no. 11, pp. 861–75, November 1993.

[3] Meyer Sound, "Constellation acoustic system," `https://meyersound.com/product/constellation/`, 2006.

[4] David Griesinger, "Improving room acoustics through time-variant synthetic reverberation," in *Proceedings of the 90th Audio Engineering Society Convention*, 1991.

[5] Wieslaw Woszcyk, Tom Beghin, Martha de Francisco, and Doyuen Ko, "Development of virtual acoustic environments for music performance and recording," in *Proceedings 25th Tonmeistertagung VDT International Convention*, 2008.

[6] Mark A. Poletti, "Active acoustic systems for the control of room acoustics," in *Proceedings of International Symposium on Room Acoustics*, 2010.

[7] Shang Li, Nobuaki Takahashi, and Tsuyoshi Takebe, "Fast stabilized adaptive algorithm for IIR bandpass/notch filters for a single sinusoid detection," *Electronics and Communications in Japan (Part III: Fundamental Electronic Science)*, vol. 76, no. 8, pp. 12–24, 1993.

[8] Manfred R. Schroeder, "Improvement of acoustic-feedback stability by frequency shifting," *The Journal of the Acoustical Society of America*, vol. 36, no. 9, pp. 1718–24, 1964.

[9] Mark A. Poletti and Roger Schwenke, "Prediction and verification of powered loudspeaker requirements for an assisted reverberation system," in *Proceedings of the 121st Audio Engineering Society Convention*, 2006.

[10] Jonathan S. Abel, Nicholas J. Bryan, Patty P. Huang, Miriam Kolar, and Bissera V. Pentcheva, "Estimating room impulse responses from recorded balloon pops," in *Proceedings of the 129th Audio Engineering Society Convention*, 2010.

[11] Bernard Widrow, J. R. Glover, J. M. McCool, J. Kaunitz, C. S. Williams, R. H. Hearn, J. R. Zeidler, J. Eugene Dong, and R. C. Goodlin, "Adaptive noise cancelling: Principles and applications," *Proceedings of the IEEE*, vol. 63, no. 12, pp. 1692–716, Dec 1975.

[12] Emanuël Habets, "Fifty years of reverberation reduction: From analog signal processing to machine learning," AES 60th Conference on DREAMS, 2016.

[13] Patrick A Naylor and Nikolay D Gaubitch, Eds., *Speech Dereverberation*, Springer, 2010.

[14] Francis Rumsey, "Reverberation... and how to remove it," *Journal of the Acoustical Society of America*, vol. 64, no. 4, pp. 262–6, April 2016.

[15] Paul DeMarinis, "Personal communication," 2014.

[16] Jonathan S. Abel and Elliot K. Canfield-Dafilou, "Recording in a virtual acoustic environment," in *Proceedings of the 143rd Audio Engineering Society Convention*, 2017.

[17] William G. Gardner, "Efficient convolution without latency," *Journal of the Audio Engineering Society*, vol. 43, 1993.

[18] Guillermo Garcia, "Optimal filter partition for efficient convolution with short input/output delay," in *Proceedings of the 113th Audio Engineering Society Convention*, 2002.

[19] Frank Wefers and Michael Vorländer, "Optimal filter partitions for real-time FIR filtering using uniformly-partitioned FFT-based convolution in the frequency-domain," in *Proceedings of the 14th International Conference on Digital Audio Effects*, 2011, pp. 155–61.

[20] Alexander Harker and Pierre Alexandre Tremblay, "The HISSTools impulse response toolbox: Convolution for the masses," in *Proceedings of International Computer Music Conference*, 2012.

[21] Leif Handberg, Ludvig Elblaus, Chris Chafe, and Elliot Kermit Canfield-Dafilou, "Op 1254: Music for neutrons, networks and solenoids using a restored organ in a nuclear reactor," in *Proceedings of the Twelfth International Conference on Tangible, Embedded and Embodied Interactions*, 2018.

[22] Jonathan Berger, "Reanimating the music of *La Chiesa di Sant'Aniceto a Palazzo Altemos*, in Rome," in *QMUL School of Electronic Engineering and Computer Science Distinguished Lecturer Series*, 2017.

[23] Jonathan Berger, Jonathan S. Abel, Talya Berger, and Elliot K. Canfield Dafilou, "Timbre, texture, space and musical style: Rome's *Chiesa di Sant'Aniceto* and its music," in *Timbre is a Many-Splendored Thing*, 2018.

EFFICIENT SIGNAL EXTRAPOLATION BY GRANULATION AND CONVOLUTION WITH VELVET NOISE

Stefano D'Angelo

Independent researcher
Agropoli, Italy
s@dangelo.audio

Leonardo Gabrielli

Università Politecnica delle Marche,
Ancona, Italy
l.gabrielli@univpm.it

ABSTRACT

Several methods are available nowadays to artificially extend the duration of a signal for audio restoration or creative music production purposes. The most common approaches include overlap-and-add (OLA) techniques, FFT-based methods, and linear predictive coding (LPC). In this work we describe a novel OLA algorithm based on convolution with velvet noise, in order to exploit its sparsity and spectrum flatness. The proposed method suppresses spectral coloration and achieves remarkable computational efficiency. Its issues are addressed and some design choices are explored. Experimental results are proposed and compared to a well-known FFT-based method.

1. INTRODUCTION

Several techniques have been devised since the advent of digital signal processing for the creative generation of *textures* and signal *freezing* effects. Some of these methods, or variations thereof, are also employed for audio restoration (see e.g. [1]), as they allow to mimic a given signal and extend its time duration. Several techniques have been proposed [2], among which some of the most used ones are:

- Overlap-and-add (OLA) techniques [3, 4];
- FFT-based methods based on spectral analysis and resynthesis [5];
- Linear Predictive Coding (LPC) schemes ([6, 7]).

OLA techniques constitute the foundation of granular synthesis, which essentially consists in summing together several time-shifted copies of a small number of short and usually windowed signals (grains) to form the output signal. Generally, however, granulation is meant as a creative tool, thus grains are often processed, e.g. with constant or time-varying pitch-shifting. Despite its conceptual simplicity, this synthesis method finds use in a wide variety of applications. For extrapolation and freezing, it is sufficient to employ a single input grain and have a sufficient density of overlapping repetitions. The relative computational efficiency of such algorithms is anyway normally counterbalanced by spectral coloration, modulation effects, and phase-related artifacts, unless countermeasures are taken [4].

Vocoding [8] is a well-known FFT-based method for signal analysis and resynthesis and it has been used for the purpose of freezing or texturing of a signal. Being block-based, it results in nonuniform execution time and/or high implementation complexity, and significant difficulties arise in handling parameter changes.

Finally, LPC methods generally achieve best output performance in terms of timbre quality, and extensions of these methods can also work in the time-frequency domain, thus allowing for accurate modeling of transients [9]. The good output quality normally obtained by LPC methods is however traded for high computational cost due to the adaptive filtering techniques [10] they are based on.

In this work we describe an OLA method for signal extrapolation which, unlike previous approaches [3, 4], is targeted not only for efficiency, but also for maximal spectral flatness, leading to results that are on par with FFT-based techniques.

The outline of the paper follows. In Section 2 we introduce OLA techniques and justify our proposition from a theoretical perspective. Section 3 reports implementation details, experimental and comparative data. Finally, Section 4 concludes the paper and discusses the outcomes of this research.

2. PROPOSED METHOD

Overlap-and-Add methods are widely used in digital signal processing to evaluate the convolution between two signals, one of which has finite length, e.g. a filter kernel, and another that can theoretically be infinitely long. If $s[n]$ is the latter signal, we can decompose it in non-overlapping blocks of length L, i.e.

$$s[n] = \sum_{r=0}^{+\infty} s_r[n - rL]. \tag{1}$$

Thus, the result of the convolution between such running signal and a finite impulse response $h[n]$ can be defined as

$$c[n] = \sum_{r=0}^{+\infty} s_r[n - rL] * h[n] = \sum_{r=0}^{+\infty} c_r[n - rL]. \tag{2}$$

If $h[n]$ has length P, then $c_r[n - rL]$, has length $L + P - 1$. Therefore, each two contiguous blocks c_r and c_{r+1} need to be overlapped and added (hence the name) to obtain the corresponding portion of $c[n]$.

Many signal extrapolation methods work by summing time-shifted copies of the windowed input signal $x_\mathrm{w}[n]$. This is conceptually equivalent to applying the OLA method to compute the convolution between $x_\mathrm{w}[n]$, impersonating the fixed-length signal, and an impulse train $v[n]$ as the running signal. If the impulses in $v[n]$ are equally spaced, as is often done, the operation will inevitably produce spectral coloration. This can be intuitively understood by considering that such a process corresponds to feeding $x_\mathrm{w}[n]$ into a feedback comb filter with unitary gain, thus resulting in significant cancellation of spectral components that cannot be compensated by post-equalization.

For our purposes, we need $v[n]$ not only to have infinite temporal duration, but also a sufficiently flat spectrum. Two well-known signals that have these properties are white noise and dense

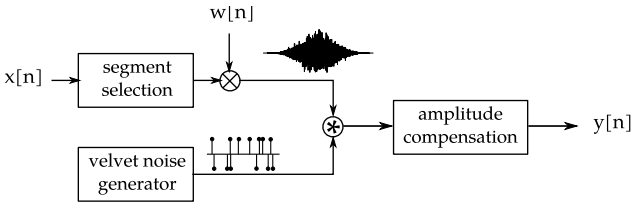


Figure 1: Overview of the proposed system.

velvet noise [11]. Velvet noise, in particular, consists of randomly-spaced unitary bipolar impulses and it has been shown to approximate white noise from a psychoacoustical standpoint when its pulse density rises above a certain threshold [12]. Due to its sparsity and the constant amplitude of impulses, convolution with velvet noise can be efficiently implemented in the time domain by simply summing together multiple randomly time-shifted copies of $x_{\mathrm{w}}[n]$ with random sign. The random nature of $v[n]$ implies random fluctuations of the local energy, requiring, thus, an amplitude compensation mechanism to reduce this undesired phenomenon, later discussed. The overall architecture is shown in Figure 1.

2.1. Issues and Implementation

The proposed method exposes three degrees of freedom in its design and operation: grain length, choice of window function, and velvet noise density.

In granular synthesis, the user can often directly choose which window function is applied as this has noticeable effect on the sound, and especially when using short grains. In our case, we definitely need windowing to eliminate potential discontinuities at the extremes of the input grain, and it would be preferable to pick a function that has high dynamic range and that is easy to compute. However, since grains need to be relatively long to retain low frequency components in the output, we can pragmatically choose the window function based on computational cost alone. The Welch window seems to be a valid choice because it is twice differentiable, except at the extremes, and has an exceptionally low computational cost. In Section 3 a few low cost windows, namely the triangular, half sine, and Welch windows, are compared.

While many musical signal processing devices nowadays are able to perform real-time convolution between a running signal and a long impulse response, the complexity and computational cost of our system can be largely reduced by leveraging the concept of *voices*, as in other forms of synthesis. Each voice is a sample playback engine, triggered randomly and with random sign, thus reducing the convolution operation to a limited number of random memory accesses, sums, and sign changes per output sample. The only potential drawback of this approach is that a finite number of voices needs to be defined beforehand, thus limiting the number of possible simultaneous grain playbacks, which theoretically corresponds to imposing a maximum "instantaneous density" to the velvet noise signal.

Given the suggested implementation approach, we believe it makes most sense to parameterize in terms of simultaneous grains on average, which corresponds to the product of the average velvet noise density (spikes over time) and the grain length. It is probably impossible to determine an optimal density for a given input signal, and especially when the input grain is somehow not sonically uniform (e.g., it contains transients), however we have empirically verified that satisfactory results can be in most cases obtained by employing relatively few voices, usually less than 30. Furthermore, preallocating twice the number of average voices reduces the likelihood of running out of available voices at any instant to at most a few percentage points.

A last issue that needs to be addressed derives from the local energy fluctuations of $v[n]$ that are inherent to its random nature. Those are also found in the output signal and need be compensated for. Significant variations of the amplitude are indeed usually noticeable in our experiments. To attenuate these, at least in a psychoacoustic sense, we propose applying a time-varying gain which depends on the signal volume. We propose employing a simple VU meter-inspired envelope detector for volume estimation, which performs full-wave rectification and conversion to the dB scale (with a lower limit of -120 dB), then applying a one-pole lowpass filter with a rise/fall time of 300 ms for 99% excursion (i.e., $\tau \approx 65.144$ ms). In order to match input and output levels, the same volume estimator can be also applied to the input signal to establish a target level. In any case, the gain factor needs to be limited to avoid the occurrence of loud peaks.

A schematic overview of the implemented algorithm is shown in Figure 2 where the amplitude compensation strategy described in Section 2 is detailed.

3. EXPERIMENTAL RESULTS

The algorithm has been implemented as a GNU Octave script to determine the quality of the audio output. The script and sound samples are available at `http://www.dangelo.audio/dafx2018-freeze.html`. A C++ implementation has also been developed for execution on regular desktop computers and on an embedded system running ELK by Mind Music Labs[1]. It was tested on two laptops, an Acer Extensa 5220 (Intel Celeron M 530 1.73 GHz single-core CPU, 1 GB DDR2 RAM) and an Acer Aspire E1-522 (AMD A4-5000 1.5 GHz quad-core CPU, 4 GB DDR3 RAM), both running 64-bit Arch Linux and using an external Focusrite Scarlett 2i4 sound card. In all cases (laptops and embedded system), the CPU load never exceeded 9% for a grain density of 32 simultaneous grains on average, at a sample rate of 44.1 kHz and with different buffering configurations.

3.1. Qualitative Results

Informal listening tests have been conducted with several audio source materials. An example of such experiments is reported in Figure 4, where a small excerpt of a male voice singing an */a/* phoneme tuned to A2 is taken as source. Its spectrogram is shown in Figure 4(a) and its DFT is shown in Figure 4(d). This signal has been extrapolated according to the proposed algorithm yielding a signal of length 5 s. Its spectrogram is shown in Figure 4(b) and its time-domain representation is shown in Figure 4(c). Random fluctuation of the overall amplitude is visible, however within a range of 3 dB maximum. The DFT from the original and the extrapolated signals are depicted in Figure 4(d-e) and show high resemblance, as expected, due to the spectral flatness of velvet noise. Similar experiments have been done with less stationary audio material, such as a guitar chord, polyphonic music and percussive instruments, with similar outcomes, see Figures 3, 8, 7.

[1] `https://www.mindmusiclabs.com/`

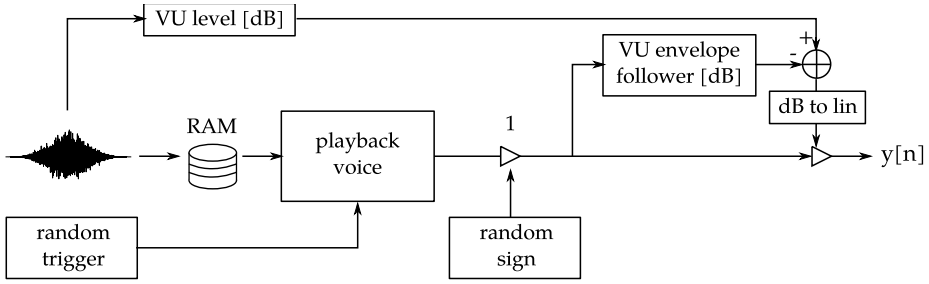


Figure 2: Schematic overview of the implemented algorithm, including the amplitude compensation strategy and exploiting multiple sample playback voices to reduce the computational cost of the convolution.

Figure 5 shows the DFT from the impulse train method. This signal has been synthesized by convolution with an impulse train having the same density as in the previous experiments, i.e. 32 pulses per second. In this case, it is quite evident that the signal has a comb-like pattern, with peaks at multiples of 32 Hz. As discussed previously, since the convolution with an impulse train has the same effect of a comb filter with unitary gain, the peaks are very pronounced, losing the timbre of the original signal.

We have also verified that the output sound quality has little dependency on the choice of the window function when the input grain is sufficiently long. The DFT from signals extrapolated using three window types, triangular, half sine, and Welch, are shown in Figure 6(a),(b) and (c), respectively. The results are almost identical, as expected. Please note that this is also true for any pulse density.

3.2. Comparison to FFT analysis-resynthesis

In this section we compare the proposed method with a well-known method based on FFT analysis-resynthesis, dubbed *timbre stamping* in [5]. In general, the quality of such an FFT-based method is rather high if the number of DFT bins is sufficiently large. In Figures 7 and 8 we compare the proposed method and the FFT-based method with a polyphonic music excerpt (trumpet playing a scale and accompanying jazz combo in the background) and a percussive jazz excerpt (containing a double bass note and cymbals) respectively. Both methods retain features of the original spectra. For instance, the polyphonic excerpt overlaps the notes of the scale are contained in the window. The time envelope of the FFT-based method is perceived as smoother for long grain size (1 s or more), however with shorter windows, such as those used in the figures (32 windows per second and window size of 300ms) the FFT method shows periodic repetitions in the output that are easily perceived especially in the presence of transients in the windowed signal. This is even clearer with percussive audio material. In the FFT-based method, transients may result smeared and are repeated periodically. The proposed method shows to have a smoother temporal domain envelope with respect to the FFT-based method, resulting in a less mechanical behavior and a denser output.

4. CONCLUSIONS

This paper described a novel method for signal extrapolation that has a low computational cost and is, thus, easily implemented in real-time applications. The method is mathematically formulated as a convolution problem with spectral flatness as a constraint. Owing from overlap-and-add methods we derived a formulation that ensures maximal spectral flatness. The low computational cost of this method is an additional benefit that allows for real-time implementations with a very low effort, as it processes the signal directly in the time domain and requires no filter adaptation, as in LPC methods. The real-time implementation can take advantage of the sparsity of the velvet noise reducing the convolution to the playback of randomly triggered *voices*. The method requires an additional step of automatic gain control to reduce random fluctuations of the output signal energy. In the current work we describe a mechanism that is widely used in the literature, however, this may be improved upon taking in consideration both the velvet noise density and the fluctuations inherent to the input signal as well. Experimental results are provided showing the preservation of the original spectrum and the minimal effect of the window type, which can be, thus, selected depending on computational constraints. As a future work, subjective listening tests could be performed to compare it to other well-known methods. The quality of these effects is very subjective, thus, some audio semantic descriptors may be employed as well for the evaluation.

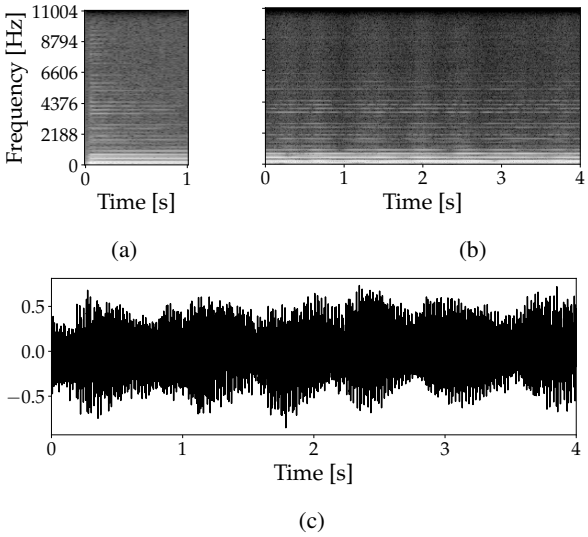


Figure 3: Extrapolation of a guitar chord: spectrogram of the original excerpt (a), spectrogram of the extrapolated audio (b) and waveform (c).

5. REFERENCES

[1] Ismo Kauppinen and Kari Roth, "Audio signal extrapolation–theory and applications," in *the Proc. of the DAFx Conference*, 2002, pp. 105–110.

[2] Diemo Schwarz, "State of the art in sound texture synthesis," in *the Proc. of the DAFx Conference*, 2011.

[3] Jim R. Parker and Brad Behm, "Creating audio textures by example: tiling and stitching," in *the Proc. of the DAFx Conference*, 2004.

[4] Martin Frojd and Andrew Horner, "Fast sound texture synthesis using overlap-add," in *the Proc. of the 2007 International Computer Music Conference, Copenhagen, Denmark*, 2007.

[5] Miller Puckette, *The Theory and Technique of Electronic Music*, World Scientific Press, 2007.

[6] Florian Keiler, Daniel Arfib, and Udo Zölzer, "Efficient linear prediction for digital audio effects," in *the Proc. of the DAFx Conference*, 2000.

[7] Xinglei Zhu and Lonce Wyse, "Sound texture modeling and time-frequency lpc," in *the Proc. of the DAFx Conference*, 2004, vol. 4.

[8] Udo Zölzer, *DAFX-Digital Audio Effects*, John Wiley and Sons, 2011.

[9] Marios Athineos and Daniel PW Ellis, "Sound texture modelling with linear prediction in both time and frequency domains," in *the Proc. of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP'03)*. IEEE, 2003, vol. 5, pp. V–648.

[10] Simon Haykin, *Adaptive Filter Theory*, Prentice Hall, Englewood Cliffs, NJ, USA, second edition, 1991.

[11] Matti Karjalainen and Hanna Järveläinen, "Reverberation modeling using velvet noise," in *the Proc. of the 30th International Conference on Intelligent Audio Environments*, 2007.

[12] Vesa Välimäki, Heidi-Maria Lehtonen, and Marko Takanen, "A perceptual study on velvet noise and its variants at different pulse densities," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 21, no. 7, pp. 1481–1488, 2013.

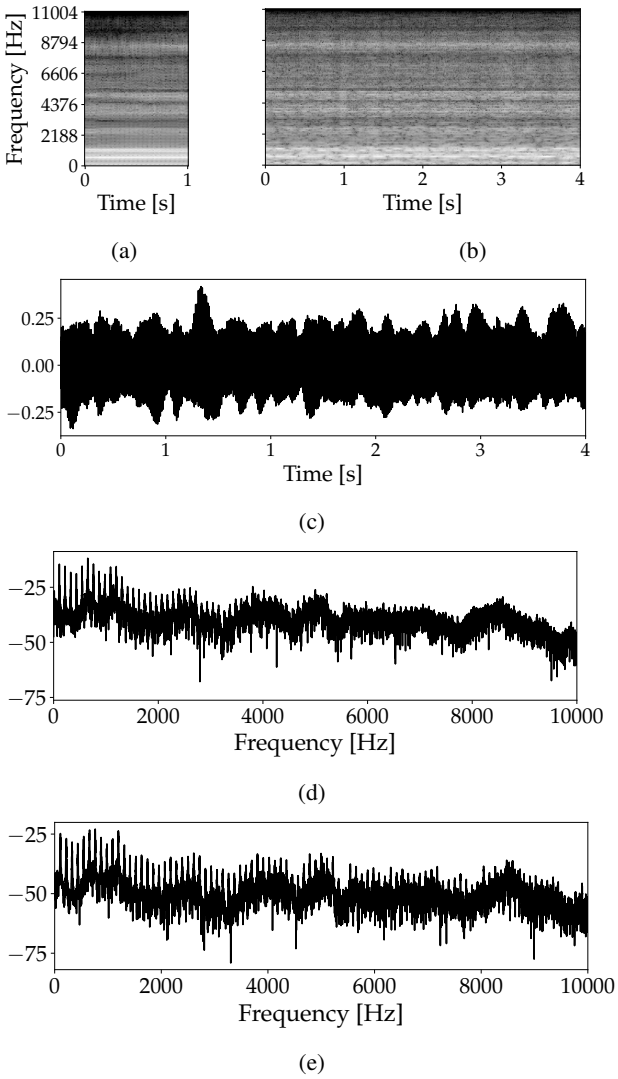


Figure 4: Experiments with voice extrapolation. The input signal is an excerpt of an /a/ phoneme by a male singer tuned to A2. Its spectrogram is shown in (a) and the resulting extrapolated signal is shown in (b), where the impulse density is set to 32 pulses per second. The original phoneme length was 1 s, while the extrapolated signal lasts 5 s. The time domain plot of the extrapolated signal is shown in (c), while the DFTs for the original and the extrapolated signals are respectively shown in (d-e). All signals are sampled at 44100 Hz.

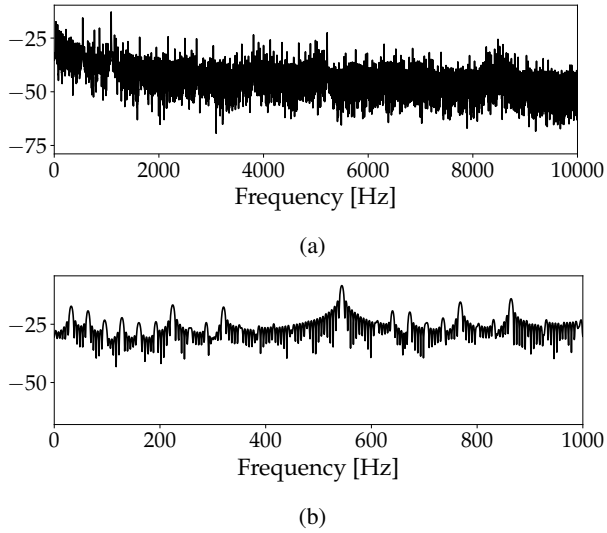


Figure 5: Repeating the experiment of Figure 4 with an impulse train instead of velvet noise with impulse density 32. The DFT is shown in (a). A detailed view shows that the periodicity can be clearly seen by the peaks emerging at multiples of 32 Hz, reducing the effect to a comb filter with unitary gain. The sampling rate is 4100 Hz.

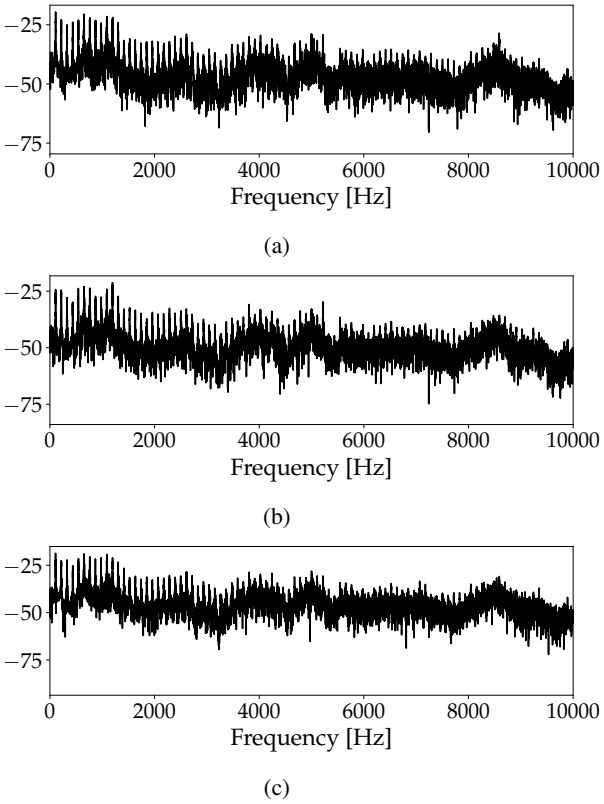


Figure 6: Comparison between signals extrapolated from the vocal signal in Figure 4(a) with window duration of 0.3 s and different window types: triangular (a), half sine (b) and Welch (c). All signals were generated using a grain density of 32 simultaneous grains on average. All signals are sampled at 44100 Hz.

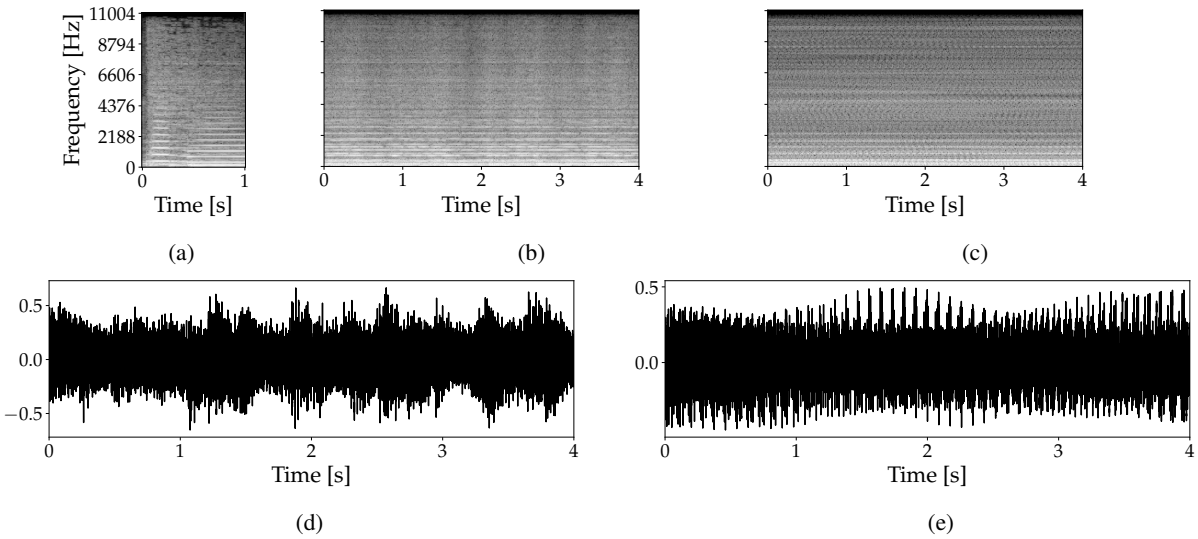


Figure 7: Extrapolation of jazz polyphonic music: spectrogram of the original excerpt (a), spectrogram of the extrapolated audio using the proposed method (b) and a FFT-based method (c). The time waveform are the one from the proposed method (d) and from the FFT-based method (e). The FFT-based method and the proposed extrapolation method use same window size. All signals are sampled at 44100 Hz.

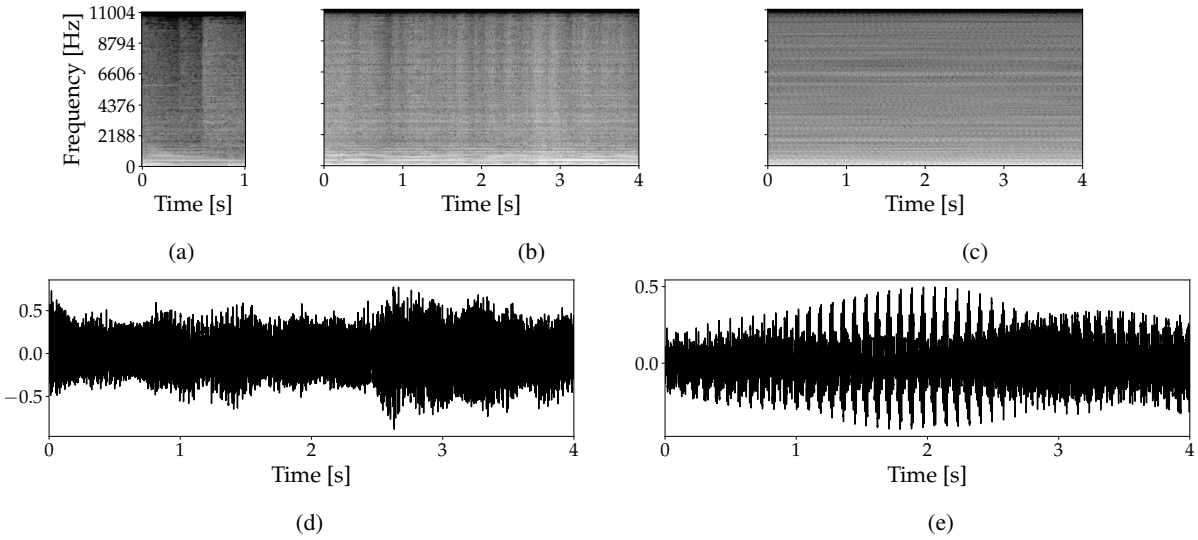


Figure 8: Extrapolation of a jazz excerpt with double bass and cymbals: spectrogram of the original excerpt (a), spectrogram of the extrapolated audio using the proposed method (b) and a FFT-based method (c). The time waveform are the one from the proposed method (d) and from the FFT-based method (e). The FFT-based method and the proposed extrapolation method use same window size. All signals are sampled at 44100 Hz.

IMPROVING INTELLIGIBILITY PREDICTION UNDER INFORMATIONAL MASKING USING AN AUDITORY SALIENCY MODEL

Yan Tang

Acoustics Research Centre,
University of Salford
Salford, UK
Y.Tang@salford.ac.uk

Trevor J. Cox

Acoustics Research Centre,
University of Salford
Salford, UK
T.J.Cox@salford.ac.uk

ABSTRACT

The reduction of speech intelligibility in noise is usually dominated by energetic masking (EM) and informational masking (IM). Most state-of-the-art objective intelligibility measures (OIM) estimate intelligibility by quantifying EM. Few measures model the effect of IM in detail. In this study, an auditory saliency model, which intends to measure the probability of the sources obtaining auditory attention in a bottom-up process, was integrated into an OIM for improving the performance of intelligibility prediction under IM. While EM is accounted for by the original OIM, IM is assumed to arise from the listener's attention switching between the target and competing sounds existing in the auditory scene. The performance of the proposed method was evaluated along with three reference OIMs by comparing the model predictions to the listener word recognition rates, for different noise maskers, some of which introduce IM. The results shows that the predictive accuracy of the proposed method is as good as the best reported in the literature. The proposed method, however, provides a physiologically-plausible possibility for both IM and EM modelling.

1. INTRODUCTION

Speech communication often takes place in non-ideal listening environments. Speech intelligibility is often negatively affected by background noise, leading to the potential failure of information transmission. In order to efficiently quantify the extent to which the background noise harms intelligibility, a great number of objective intelligibility measures (OIM) have been proposed in the last decades. They have been used as a perceptual guide in activities such as development of modification algorithms for highly-intelligible speech [1], speech enhancement [2], production of TV or radio broadcast [3] and research in hearing impairment [4]. OIMs have an important role in developing speech and noise processing algorithms for an inclusion.

Standard measures, such as the Speech Intelligibility Index (SII, [5]) and the Speech Transmission Index [6], and early OIMs (e.g. [4, 7]) make intelligibility predictions based on long-term masked audibility (e.g. SII) or modulation reduction (e.g. STI) of the target speech signal. More recent methods [8, 9, 10, 11] operate on short windows (10-300 ms), in order to improve the predictive accuracy in temporally-fluctuating noise maskers. In addition, some of the measures [10, 11] were developed on the basis of sophisticated auditory models, and have demonstrated more robust predictive power in a wide range of conditions [12]. In the light of the fact that listener do not need all time-frequency (T-F) information to successfully decode the speech [13], Cooke proposed a glimpsing model of speech perception in noise [14]. In [14], the percentage of the T-F regions of speech with a local speech-to-noise ratio (SNR) meeting a given criteria was calculated as the intelligibility proxy, known as the glimpse proportion (GP). It can be thought of the overall contribution from the local audibility of all the T-F regions to intelligibility in noise. Tang and Cooke further extended the GP to a complete intelligibility measure – the extend GP (ext. GP) – which performs detail modelling of the masking effect taking place in the auditory peripheral from the outer-middle ear, through the cochlea to the inner-hair cells. The predictions by ext. GP are well correlated with listener word recognition performance in various noise maskers, with correlation coefficients greater than 0.85 [11].

Energetic masking (EM) and informational masking (IM) introduced by the noise masker mainly account for the reduced intelligibility. EM is the consequence of interactions of physical signals acting in the auditory peripheral. IM is different as it obstructs auditory identification and discrimination at the late stage of auditory pathway, when a sound is perceived in the presence of other similar sounds [15, 16, 17]. However, this is overlooked by the aforementioned OIMs, which can only quantify the impact of EM on intelligibility from the physical attributes of the speech and noise signals. When comparing the GP in speech-shaped noise (SSN) and competing speech (CS), Tang and Cooke found that to achieve the same intelligibility much fewer glimpses are required in SSN than in CS, which introduces large IM [11]. They postulated that IM made some glimpses ineffective.

With a classification of the glimpsed T-F regions based on energy, it was further found that the regions with energy above the average are more robust in noise. Those with energy under the average are more susceptible to both EM and IM [11]. Importantly, the amount of high-energy glimpses is broadly consistent for the same speech signal in SSN and CS under SNRs leading to the similar listener performance. Although this method, know as high-energy glimpse proportion (HEGP), is a crude approach for making consistent predictions when the masker is in presence or absence of IM, it confirmed an early hypothesis that IM may affect the effectiveness of the glimpsed T-F regions available for speech.

One possible explanation is that the listener switches attention between the target and the competing sources, leading to some of the target components that have triggered activities at the auditory peripheral not being further processed by the brain. The perceptual and cognitive resources that a human's nervous system can use to process the input sensory stimuli received in a short time window is limited. Consequently, the brain temporarily and selectively stores only a subset of available sensory information in short-term working memory for further processing [18, 19, 20]. This selection is a combination of rapid bottom-up signal-driven

(task-independent) attention, as well as slower top-down cognitive (task-dependent) attention. First, the bottom-up processing occurs and attracts attention towards conspicuous or salient locations of the scene in an unconscious manner. Then, the top-down processing shifts the attention voluntarily towards locations of cognitive interest. Only the information selectively attended to is allowed to progress through the cortical hierarchy for high-level processing and detailed analysis [20, 21]. Therefore, saliency detection is considered to be a key attentional mechanism used to economically allocate and efficiently use the brain's limited processing capacity [22, 23].

The saliency of an object is the state or quality by which it stands out relative to its neighbours or background. In a complex auditory scene, a salient sound object may stand a bigger chance relative to other competing sources to gain a listener's attention. Saliency-based approaches were initially proposed as a major component in modelling bottom-up visual attention [24, 25, 26, 18]. The way in which the auditory cortex responds to sound stimuli is similar in terms of feature analyses on spectral or temporal modulation for instance [27, 28, 29, 30]. Many studies (e.g. [31, 32, 33, 34, 35]) on auditory saliency adopt the same analytical feature extraction mechanism to model auditory attention. The features used mainly include intensity, temporal contrast, spectral contrast and orientation which simulates the dynamics of the auditory neuron responses to moving ripples [36, 37]. In general, the modelling of auditory attention closely resembles that of visual attention, in which features essentially approximate the receptive field sensitivity profile of orientation-selective neurons in the primary visual cortex [38].

The output of the saliency analysis is usually a spectro-temporal representation called a saliency map. Kayser et al. generated the saliency map from intensity, temporal and spectral contrasts using a standard Fourier analysis [31]. By comparing the model prediction to the results from behavioural studies on human listeners and macaque monkeys, it was confirmed that different primate sensory systems rely on common principles for extracting relevant sensory events. In more recent studies [32, 33, 34, 35], the features used for composing the saliency map were extracted from the output of auditory peripheral analysis instead of via Fourier. This in principle provided a more physiologically-valid representation for the saliency analysis. Besides the same features used in [31], Kalinli and Narayanan included the orientation information in the saliency map [32]. A saliency score, which was a function of time, was further computed by collapsing the saliency map across frequencies followed by normalisation. This was used to predict the 'prominent' syllables and words in sentences drawn from a speech corpus. Their model achieved a better accuracy than when orientation information was excluded. However, further adding pitch information did not improve the model accuracy. Some other features were also used for generating a saliency map. Kaya and Elhilali added temporal envelope, rate and bandwidth as features to further emphasise the impact of the spectro-temporal modulations [35].

As both contemporary auditory saliency and glimpse analyses are performed on T-F representations, it is therefore possible to use a common representation at the early stage of the models for the purposes. This study aims to integrate saliency analysis into the ext. GP measure, in an attempt of quantifying the IM effect in a physiologically-plausible approach. The performance of the proposed method were evaluated along with another three reference OIMs, by comparing the model predictions to measured subjective intelligibility in noise maskers, some of which introduce IM.

2. PROPOSED METHOD

The proposed method consists of two main parts, as illustrated in Fig. 1. The first part is the ext. GP [11], which models the energetic masking taking place at the auditory peripheral. The second part (shaded and on the left of Fig. 1) performs saliency analysis on the given auditory scene as a whole, quantifying the probability of the T-F regions on the scene gaining processing in the later stage of the auditory pathway in a bottom-up process. The output of this part, the saliency map SM, is subsequently combined with glimpse representation G' from ext. GP, in order to adjust the contribution of the glimpses to the final intelligibility.

2.1. Quantifying energetic masking

Energetic masking is modelled using ext. GP [11]. To generate the auditory representations – the spectro-temporal excitation patterns (STEP) – for the signals, the clean speech signal s and noise signal n are passed through 64-gammatone filterbanks[1]. The centre frequencies of the 64 filters are evenly distributed on the equivalent rectangle bandwidth (ERB) scale, ranging from 100 to 7500 Hz, with a spectral resolution of 0.51 ERB. An outer-middle ear transfer function [39] is applied to the filter outputs, in order to account for the auditory sensitivity (i.e. hearing threshold) to the level of the signal at different frequencies. The Hilbert envelopes of each frequency band, $E(f)$, is then extracted, smoothed by a leaky integrator with an 8 ms time constant and downsampled to 100 Hz. A log-compression is imposed on the final output.

The glimpses are determined by comparing the STEP of the speech signal $STEP_s$ against that of the noise signal $STEP_n$. A glimpsed T-F region must possess a local SNR above a given threshold (Δ=3 dB), and be above the hearing level (HL, set to 25 dB),

$$G(t,f) = STEP_s(t,f) > max(STEP_n(t,f) + \Delta, HL) \quad (1)$$

To account for forward masking, the raw glimpses G are further validated using an inner-hair cell model (IHC, [40]), which also takes the envelopes of speech-plus-noise mixture E_m as the input. The glimpsed T-F regions surviving from simultaneous masking are considered valid only when their corresponding IHC outputs are not masked during the IHC depleting and replenishing process. Hence, the IHC-validated glimpse G' is defined as,

$$G'(t,f) = G(t,f) \wedge \neg g(t,f) \quad (2)$$

where $\wedge$ indicates logic '*and*', and g denotes the masked glimpses due to forward masking. For the rules for IHC validation, see [11] for details.

The plots in the second row of Fig. 2 exemplify the valid glimpsed T-F regions on a speech signal in SSN and CS at 1 and -7 dB SNR, respectively. The chosen SNRs led to a similar intelligibility in the two maskers [41].

2.2. Generating saliency map

A saliency map is also a T-F representation produced from STEP. Generating a saliency map often involves feature extraction, nor-

[1] Saliency analysis requires a greater number of filters to maintain the T-F resolution of its output than previously used for ext. GP. Instead of 34-channel described in [11], 64-channel STEPs were used here for ext. GP, in order to keep the representations consistent. Tests have shown that filter numbers above 34 have little impact to the performance of ext. GP.

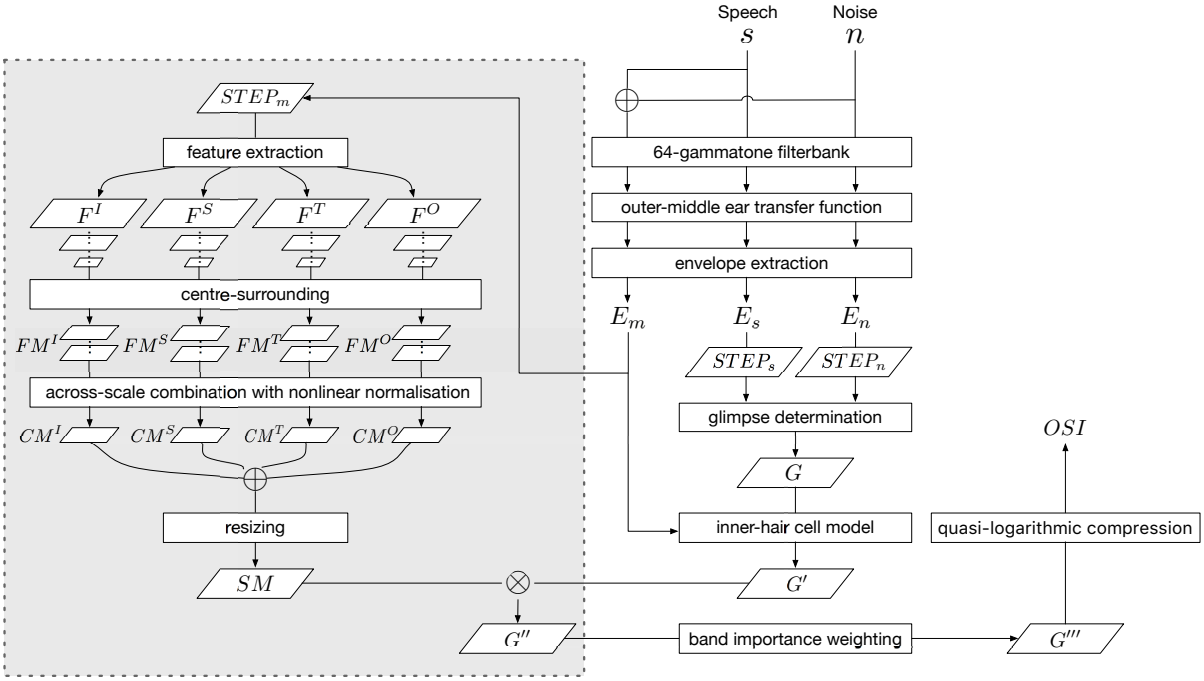


Figure 1: *Diagram of the proposed system. The shaded part on the left performs saliency analysis, partly accounting for the effect of informational masking. The unshaded part on the right describes the mechanism of the extended GP [11].*

malisation, combination and resizing. After [42, 32, 33], the features $F(\sigma, \theta, \alpha)$ including intensity F^I, spectral contrast F^S, temporal contrast F^T and orientation F^O are extracted from the STEP of the speech-plus-noise mixture $STEP_m$. This is performed in a multi-scale manner [18]: eight scales $\sigma = \{1, ..., 8\}$ are used, and the input $STEP_m$ is filtered and decimated by a factor of two iteratively for seven times; the output of the last iteration is the input of the next. This results in size reduction factors ranging from 1:1 to 1:128. The resized STEPs are then convolved by the Gabor filters (which are the product of a cosine grating and a 2D Gaussian envelope) with different θ, which represents one of the four target features, as listed in Table 1:

Table 1: *Parameters of the Gabor filters for each feature*

Feature	θ	α
Intensity	$\pi/2$	0
Spectral contrast	0	1
Temporal contrast	$\pi/2$	1
Orientation	$\{\pi/4, 3\pi/4\}$	1

In order to mimic the properties of local cortical inhibition, the 'centre-surrounding' differences are calculated after extracting features at multiple scales, yielding a set of feature maps $FM(c, s)$. This is done by across-scale subtraction between a centre finer scale $c \in \{2, 3, 4\}$ and a surrounding coarser scale s:

$$FM(c, s) = |FM(c) - FM(s)| \tag{3}$$

where $s = c + \delta, \delta \in \{3, 4\}$. As the size of the feature representation varies across scales, it (from scale 1 to 8) needs to be be normalised prior to the across-scale subtraction. Here the representations for each feature are resized to that of scale 4. The centre-surrounding step finally results in 6×5 feature maps[2], 6 of which represent each of the features in different scales.

Across-scale combination aims to generate a so-called 'conspicuity map', CM, for each feature from the feature maps at different scales, using across scale addition. Due to different dynamic ranges resulting from the extraction process for each feature, the feature maps must be first handled by a nonlinear normalisation procedure in order to bring them into a comparable scale. Another purpose of the normalisation is to simulate competition between neighbouring salient locations [43]. This nonlinear normalisation consists of certain number of iterations (three times is used here), each of which consists of self-excitation and inhibition induced by neighbours. To implement, a 2-D difference of Gaussians (DoG) filter is convolved with each feature map, followed by clamping the negative values to zero. A feature map FM is then transformed in each iteration as follow:

$$CM \longleftarrow |FM + FM * DoG - 0.02| \geq 0 \tag{4}$$

After normalisation, the normalised feature maps of different scales can then be summed up to a single conspicuity map. This is repeated for all the four features, resulting in four maps. The final saliency map, SM, is a linear combination of all the four normalised maps. A further resizing is required to recovery the map size (currently at scale 4) back to the original size (at scale 1, same to the $STEP_m$).

[2]Due to orientation having two sub-conditions (i.e., $\theta \in \{pi/4, 3\pi/4\}$), there are therefore 12 feature maps for orientation in total.

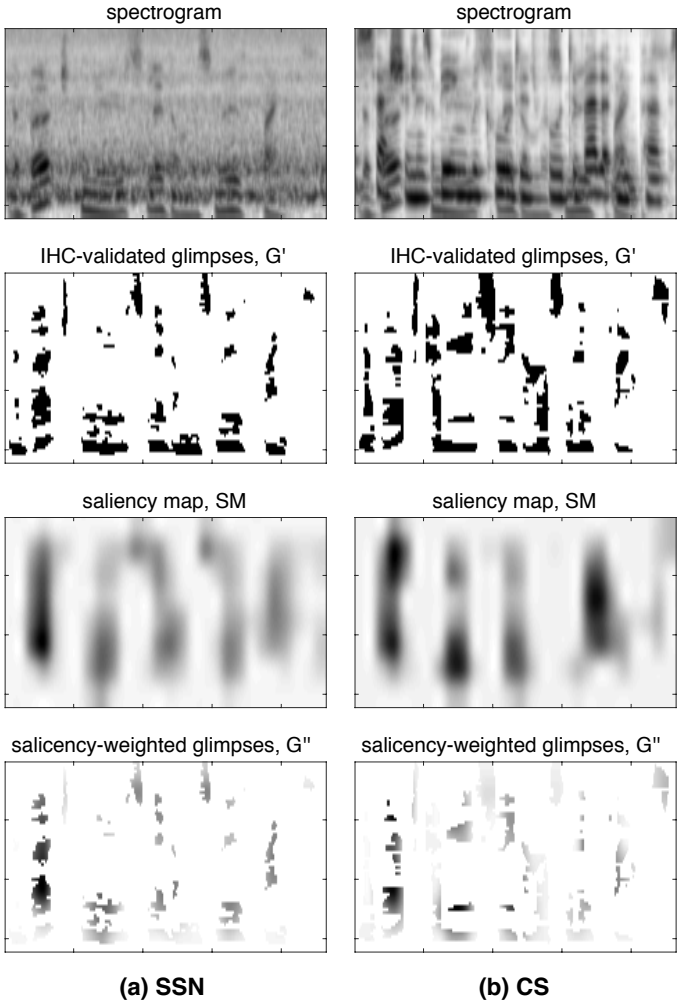


Figure 2: *Spectrograms, IHC-validated glimpses (G'), saliency maps (SM) and saliency-weighted glimpses (G'') of the sentence 'the birch canoe slid on the smooth planks' in SSN (left column) and CS (right column) at 1 and -7 dB SNR, respectively.*

The plots in the third row of Fig. 2 show saliency maps of the same speech signal corrupted by SSN and CS. While the T-F regions where the glimpses occur are mostly salient in SSN, this is not always the case in CS. For CS, the glimpsed fricative components of speech that have energy concentrated at mid-high frequencies are scarcely salient in the example. These glimpses might have very limited contribution to intelligibility due to IM.

2.3. Intelligibility prediction

The final saliency-adjusted glimpses G'' is the product of the IHC-validated glimpse G' and the saliency map SM. The effect of this operation on the glimpses is visualised in the plots at the bottom of Fig. 2. The remaining procedure follows the calculation of ext. GP: G'' is subsequently weighted by the band importance function K [5] , followed by a quasi-logarithmic compression in a form of

$$v(x) = \log(1 + x/0.01)/\log(1 + 1/0.01),$$

$$OSI = v\left[\frac{1}{T}\sum_{f=1}^{F}\left(K(f)\sum_{t=1}^{T}G'(t,f)\cdot SM(t,f)\right)\right] \quad (5)$$

where F=64 and T are the number of frequency bands and time frames, respectively. The final predictive index falls between 0 and 1, with the greater number indicating the better intelligibility.

3. EVALUATION

For the reference performance, ext. GP, HEGP and SII were evaluated along with the proposed method.

3.1. Subjective data

The subjective data was drawn from [41, 44]. In the two studies, the listener intelligibility was measured as the sentence-level word recognition rate in SSN and CS at three SNR levels for each masker, i.e. -9, -4 and 1 dB for SSN and -21, -14 and -7 dB for CS. The chosen SNRs led to the intelligibility of approximately 25%, 50% and 75% in each masker. While the target sentences were uttered by a male native English speakers, the CS was produced by a female speaker. In contrast to SSN, CS is able to cause strong IM [45]. In total, this corpus offers 180 conditions, covering the intelligibility range from 5% to 95%. As this corpus consists of 30 types of speech including those algorithmically-modified for better intelligibility and synthetic speech, it is rather challenging for OIMs to predict from. Tang et al. evaluated up to seven state-of-the-art OIMs using this corpus, the average overall performance – the correlation between the listener performance and the model predictions – across all the OIMs was merely 0.67, with 0.83 being the best [12]. Nevertheless, the use of SSN and CS maskers in the corpus provided this study with an ideal experiment protocol (i.e. inclusion of maskers which do or do not introduce IM) for evaluation the proposed method.

3.2. Procedure

The raw model output, O, was transformed to the estimated listener performance using a two-parameter sigmoid function (Eqn. 6), in order to make a direct comparison with the subjective data.

$$W = \frac{1}{1 + \exp(-(a + b \cdot O))} \quad (6)$$

where a and b are the two open parameters, the values of which are chosen to give a best fit to the subjective data for each OIM; values are presented in Table 2.

Table 2: *Values of parameters a and b used in the sigmoid transformation for the OIMs*

	proposed	ext. GP	HEGP	SII
a	-2.201	-2.864	-4.007	-1.009
b	8.284	5.837	8.024	5.339

The main performance of the OIM was evaluated as the Pearson correlation coefficient ρ between the measured and estimated intelligibility, as well as the root-mean-square error $RMSE$.

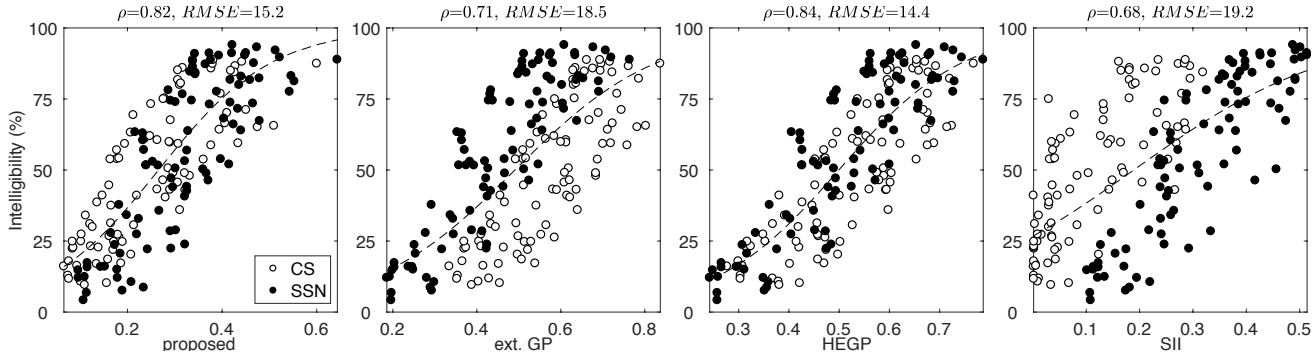


Figure 3: *Listener intelligibility versus model predictions in all 180 conditions. The dashed line in each plot is the sigmoid fitting for the OIM.*

Table 3: *Subjective-model Pearson correlation correlations ρ and RMSEs (in parentheses) as the model performance in each sub-conditions. Figure in squared brackets indicates the number of data points from which ρ and $RMSE$ were calculated.*

	proposed	ext. GP	HEGP	SII
SSN [90]	0.89 (13.0)	0.88 (13.0)	0.88 (13.1)	0.87 (13.5)
CS [90]	0.82 (14.0)	0.81 (14.4)	0.83 (13.7)	0.77 (15.7)
natural [132]	0.86 (13.5)	0.73 (17.8)	0.87 (13.0)	0.70 (18.7)
synthetic [48]	0.92 (9.0)	0.79 (13.5)	0.93 (8.0)	0.74 (14.8)
overall [180]	0.82 (15.2)	0.71 (18.5)	0.84 (14.4)	0.68 (19.2)

3.3. Results

Fig. 3 compares the model predictions against the measured intelligibility in the 180 conditions. Overall, ext GP and SII exhibited visually poorer performance than the other two due to the discrepancy between the predictions in SSN (solid circles) and in CS (open circles). While ext GP overestimated in CS or underestimated in SSN, SII displays opposite behaviour. This will be discussed later. By accounting for the effect of IM using the saliency map to further weight the contribution of glimpses, the proposed method decreases the discrepancy observed for ext. GP in Fig. 3. This led to a significant improvement in accuracy for the proposed method ($\rho = 0.82$) over ext. GP ($\rho = 0.71$) [$Z = 3.216$, $p < 0.01$]. SII performance was similar to ext. GP ($\rho = 0.48$) [$Z = 0.781$, $p = 0.535$]. With such overall listener-model correlation, the proposed method performed as almost the best as reported in [12] ($\rho = 0.83$). The proposed is also comparable to HEGP [$Z = 0.970$, $p = 0.332$], despite the latter leading to the highest correlation ($\rho = 0.84$).

The performances of the OIMs were also examined in a series of sub-conditions, as displayed in Table 3. For individual maskers, all the OIMs achieved similar performance [all $\chi^2(3) \leq 3.923$, $p \geq 0.270$]. When making predictions separately for natural and synthetic speech, the proposed was equivalent to the HEGP [all $Z \leq 0.797$, $p \geq 0.426$], however was clearly more robust than the other two OIMs [all $Z \geq 2.927$, $p < 0.01$], especially for synthetic speech.

4. DISCUSSION

The current study aimed to improve the predictive power of the ext. GP metric [11] under informational masking by incorporating an auditory saliency model into the OIM. Having observed that the speech-dominant T-F regions contribute to intelligibility differently in the face of different maskers [11], a weighting based on the likelihood of a region being selected for further auditory processing in a bottom-up procedure, can help account for the IM effect. Hence, improved performance over the original ext. GP metric is seen, especially when performing across maskers which do or do not introduce IM.

The overall performance of both ext. GP and SII suffers from the separation of their outputs in the two maskers, as seen in Fig. 3. Since speech is more tolerant of EM in CS (i.e. fluctuating masker) than in SSN (stationary masker) at the same SNR level, speech in CS must be presented at a lower global SNR to obtain the same intelligibility level as in SSN. However, due to its large envelope modulations, CS provides more opportunities for glimpsing T-F regions on the target signal than in SSN. Even so, the additional glimpses in CS are not translated to intelligibility gain. The overestimation of ext. GP in CS is thus attributed to the IM effect not being accounted for. On the other hand, Tang et al. explained that SII scores lower in CS than in SSN is due to its long-term spectral SNR-based calculation being sensitive to any change in global SNR [12], which is a more dominant factor to speech intelligibility in noise than IM [45].

The proposed method achieved the same performance as the HEGP metric, which assumes that the amount of the high-energy T-F regions on the speech signal is determinant for intelligibility prediction in noise. In terms of EM, more energy offers bigger chance of surviving from the masking to this group of T-F regions, it is therefore more likely for them to be glimpsed by the listener. In the meantime, relatively high intensity in these regions may cause large spectral and temporal contrasts across both the time and frequency at the boundaries when intensity dramatically increases or decreases, e.g. at the transition between a consonant and a vowel. Consequently, these T-F regions are likely to be more salient than others, and hence more probable to win the completion of the auditory attention during the bottom-up processing. Despite the similar fundamental mechanism and predictive performance, the proposed method presents a finer and more transparent modelling of speech intelligibility in noise than HEGP. As it quantifies the EM and IM effects in different components, modelling of each

effect could be further improved and extended separately. There is some evidence suggesting that in English the glimpses taking place on vowels are more important to the intelligibility than those on consonants [46, 47], implying that the contribution of the glimpsed T-F regions could be further re-weighted for voicing and invoicing segments. In addition, a top-down auditory spotlight searching [48] could be also considered in the metric for better modelling of IM.

5. CONCLUSIONS

An auditory saliency model was used in conjunction with a state-of-the-art OIM to improve the accuracy for intelligibility prediction under IM. The evaluation confirmed the validity of this approach, whose performance for the given dataset was comparable to the best reported in the literature. This study presents a detailed and yet physiologically-plausible approach for modelling both EM and IM to speech intelligibility. The proposed method could be thus used as a perceptual guide in audio production and reproduction, where speech intelligibility is a concern. However, the complexity of IM occurring at the later stage of the auditory pathway warrants investigations in future.

6. ACKNOWLEDGMENTS

This work was supported by the EPSRC Programme Grant S3A: Future Spatial Audio for an Immersive Listener Experience at Home (EP/L000539/1) and the BBC as part of the BBC Audio Research Partnership.

7. REFERENCES

[1] Y. Tang and M. Cooke, "Learning static spectral weightings for speech intelligibility enhancement in noise," *Computer Speech and Language*, vol. 49, pp. 1–16, 2018.

[2] Q. Liu, W. Wang, P. J. B. Jackson, and Y. Tang, "A Perceptually-Weighted Deep Neural Network for Monaural Speech Enhancement in Adverse Background noise," in *2017 European Signal Processing Conference*, Kos island, Greece, 2017.

[3] Y. Tang, B. M. Fazenda, and T. J. Cox, "Automatic speech-to-background ratio selection for maintaining speech intelligibility in broadcasts using an objective intelligibility metric," *Appl. Sci.*, vol. 8, no. 1, pp. 59, 2018.

[4] Inga Holube and Birger Kollmeier, "Speech intelligibility prediction in hearing-impaired listeners based on a psychoacoustically motivated perception model," *J. Acoust. Soc. Am.*, vol. 100, pp. 1703–1716, 1996.

[5] ANSI S3.5, "ANSI S3.5-1997 Methods for the calculation of the Speech Intelligibility Index," 1997.

[6] IEC, ""Part 16: Objective rating of speech intelligibility by speech transmission index (4th edition)," in IEC 60268 Sound System Equipment (Int. Electrotech. Commiss., Geneva, Switzerland)," 2011.

[7] J. M. Kates and K. H. Arehart, "Coherence and the speech intelligibility index," *J. Acoust. Soc. Am.*, vol. 117, no. 4, pp. 2224–2237, 2005.

[8] K. S. Rhebergen, N. J. Versfeld, and W. A. Dreschler, "Extended speech intelligibility index for the prediction of the speech reception threshold in fluctuating noise," *J. Acoust. Soc. Am.*, vol. 120, no. 6, pp. 3988–3997, 2006.

[9] C. H. Taal, R. C. Hendriks, R. Heusdens, and J. Jensen, "A short time objective intelligibility measure for time-frequency weighted noisy speech," in *Proc. ICASSP*, 2010, pp. 4214–4217.

[10] C. Christiansen, M. S. Pedersen, and T. Dau, "Prediction of speech intelligibility based on an auditory preprocessing model," *Speech Comm.*, vol. 52, no. 7-8, pp. 678–692, 2010.

[11] Y. Tang and M. Cooke, "Glimpse-Based Metrics for Predicting Speech Intelligibility in Additive Noise Conditions," in *Proc. Interspeech*, San Francisco, US, 2016, pp. 2488–2492.

[12] Y. Tang, M. Cooke, and C. Valentini-Botinhao, "Evaluating the predictions of objective intelligibility metrics for modified and synthetic speech," *Computer Speech and Language*, vol. 35, pp. 73–92, 2016.

[13] R. V. Shannon, F. G. Zeng, V. Kamath, J. Wygonski, and M. Ekelid, "Speech recognition with primarily temporal cues," *Science*, vol. 270, no. 5, pp. 303–304, 1995.

[14] M. Cooke, "A glimpsing model of speech perception in noise," *J. Acoust. Soc. Am.*, vol. 119, no. 3, pp. 1562–1573, 2006.

[15] G. Miller, "The masking of speech," *Psychol. Bull.*, vol. 44, pp. 105–129, 1947.

[16] I. Pollack, "Auditory informational masking," *J. Acoust. Soc. Am.*, vol. 57, no. S1, pp. S5–S5, 1975.

[17] G. Kidd Jr and H. S. Colburn, "Informational masking in speech recognition," in *The Auditory System at the Cocktail Party*, pp. 75–109. Springer, 2017.

[18] L. Itti, C. Koch, and E. Niebur, "A model of saliency-based visual attention for rapid scene analysis," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 20, no. 11, pp. 1254–1259, 1998.

[19] A. S. Bregman, *Auditory Scene Analysis: The Perceptual Organization of Sounds.*, The MIT Press, 1990.

[20] C. Alain and S. R. Arnott, "Selectively attending to auditory objects," *Front. Biosci.*, vol. 55, pp. 202–212, 2000.

[21] S. Harding, M. Cooke, and P. Koenig, "Auditory gist perception: An alternative to attentional selection of auditory streams," in *WAPCV2007*, 2007.

[22] W. Schneider and R. M. Shiffrin, "Controlled and automatic human information processing: I. detection, search, and attention," *Psychological Review*, vol. 84, no. 1, pp. 1–66, 1977.

[23] R. M. Shiffrin and W. Schneider, "Controlled and automatic human information processing: II perceptual learning, automatic attending and a general theory," *Psychological Review*, vol. 84, no. 2, pp. 127–190, 1977.

[24] C. Koch and S. Ullman, "Shifts in Selective Visual Attention: Towards the Underlying Neural Circuitry," *Human Neurobiology*, vol. 4, pp. 219–227, 1985.

[25] R. Milanese, S. Gil, and T. Pun, "Attentive Mechanisms for Dynamic and Static Scene Analysis," *Optical Eng.*, vol. 34, no. 8, pp. 2428–2434, 1995.

[26] S. Baluja and D. A. Pomerleau, "Expectation-Based Selective Attention for Visual Monitoring and Control of a Robot Vehicle," *Robotics and Autonomous Systems*, vol. 22, no. 3–4, pp. 329–344, 1997.

[27] J. P. Rauschecker, B. Tian, and M. Hauser, "Processing of complex sounds in the macaque nonprimary auditory cortex," *Science*, vol. 268, pp. 111–114, 1995.

[28] C. E. Schreiner, H. L. Read, and M. L. Sutter, "Modular organization of frequency integration in primary auditory cortex," *Rev. Neurosci.*, vol. 23, pp. 501–529, 2000.

[29] L. M. Miller, M. A. Escabi, H. L. Read, and C. E. Schreiner, "Spectrotemporal receptive fields in the lemniscal auditory thalamus and cortex," *J. Neurophysiol.*, vol. 87, pp. 516–527, 2002.

[30] S. Kaur, R. Lazar, and R. Metherate, "Intracortical pathways determine breadth of subthreshold frequency receptive fields in primary auditory cortex," *J. Neurophysiol.*, vol. 91, pp. 2551–2567, 2004.

[31] C. Kayser, C. I Petkov, M. Lippert, and N. K. Logothetis, "Mechanisms for Allocating Auditory Attention: An Auditory Saliency Map," *Current Biology*, vol. 15, no. 21, pp. 1943–1947, 2005.

[32] O. Kalinli and S. S. Narayanan, "A saliency-based auditory attention model with applications to unsupervised prominent syllable detection in speech.," in *Proc. Interspeech*, 2007, pp. 194–1944.

[33] B. De Coensel and D. Botteldooren, "A model of saliency-based auditory attention to environmental sound," in *Proc. 20th International Congress on Acoustics (ICA 2010)*, 2010, pp. 1–8.

[34] T. Tsuchida and G. W. Cottrell, "Auditory saliency using natural statistics," in *Proc. Annual Meeting of the Cognitive Science (CogSci)*, 2012, pp. 1048–1053.

[35] E. M. Kaya and M. Elhilali, "A temporal saliency map for modeling auditory attention," in *46th Annual Conference on Information Sciences and Systems (CISS*, 2012.

[36] R. C. deCharms, D. T. Blake, and M. M. Merzenich, "Optimizing sound features for cortical neurons," *Science*, vol. 280, pp. 1439–1443, 1998.

[37] S. Shamma, "On the role of space and time in auditory processing," *Trends in cognitive sciences*, vol. 5, no. 8, pp. 340–348, 2001.

[38] A. G. Leventhal, *The Neural Basis of Visual Function: Vision and Visual Dysfunction, vol. 4.*, Boca Raton, Fla.: CRC Press, 1991.

[39] ISO 389-7:2006, "Acoustics – Reference Zero For The Calibration Of Audiometric Equipment – Part 7: Reference Threshold Of Hearing Under Free-field And Diffuse-field Listening Conditions," 2006.

[40] M. Cooke, *Modelling Auditory Processing and Organisation*, Cambridge University Press, 1993.

[41] M. Cooke, C. Mayo, C. Valentini-Botinhao, Y. Stylianou, B. Sauert, and Y. Tang, "Evaluating the intelligibility benefit of speech modifications in known noise conditions," *Speech Comm.*, vol. 55, pp. 572–585, 2013.

[42] D. Walther and C. Koch, "Modeling attention to salient proto-objects," *Neural Networks*, vol. 19, no. 9, pp. 1395–1407, 2006.

[43] L. Itti and C. Koch, "Feature combination strategies for saliency-based visual attention systems," *J. Electron. Imaging*, vol. 10, no. 161–169, pp. 1102–1116, 2001.

[44] M. Cooke, C. Mayo, and C. Valentini-Botinhao, "Intelligibility-enhancing speech modifications: the Hurricane Challenge," in *Proc. Interspeech*, 2013, pp. 3552–3556.

[45] D. S. Brungart, "Informational and energetic masking effects in the perception of two simultaneous talkers," *J. Acoust. Soc. Am.*, vol. 109, no. 3, pp. 1101–1109, 2001.

[46] R. Cole, Y. Yan, B. Mak, and M. Fanty, "The contribution of consonants versus vowels to word recognition in fluent speech," *J. Acoust. Soc. Am.*, vol. 100, pp. 2689, 1996.

[47] D. Fogerty and D. Kewley-Port, "Perceptual contributions of the consonant-vowel boundary to sentence intelligibility," *J. Acoust. Soc. Am.*, vol. 126, no. 2, pp. 847–857, 2009.

[48] S. Treue, "Directing the auditory spotlight," *Nature Neuroscience*, vol. 100, pp. 2689, 2006.

ACOUSTIC ASSESSMENT OF A CLASSROOM AND REHABILITATION GUIDED BY SIMULATION

Raquel Ribeiro *

Faculdade de Engenharia da Universidade do Porto - DEEC
ee12169@fe.up.pt

Diamantino Freitas

Faculdade de Engenharia da Universidade do Porto - DEEC
dfreitas@fe.up.pt

ABSTRACT

The acoustics of spaces whose purpose is the acoustic communication through speech, namely classrooms, is a subject that has not been given the due importance in architectural projects, with consequences in the existence of adverse acoustic conditions, which affect on a daily basis the learning of the students and the well-being of teachers.

One of the lecture rooms of the Faculty of Engineering of the University of Porto (FEUP) was chosen, with a criterion of generality, in which the acoustic conditions were evaluated and compared with those that are known to be necessary for the intended acoustic communication effect. Several measurements were made in the space to investigate the acoustic parameters situation relatively to the appropriate range.

An acoustic model of the amphitheater under study was developed in the EASE software, with which it was possible to obtain simulated results for comparison with the previously measured parameters and to introduce changes in the model to perceive their impact in the real space. In this phase it was possible to use the auralization resources of the software to create perception of how the sound is heard in the built model. This was useful for the phase of rehabilitation of the space because it was possible to judge subjectively the improvement of the sound intelligibility in that space.

Finally, possible solutions are presented in the acoustic domain and using electroacoustic sound reinforcement aiming to provide a better acoustic comfort and communicational effectiveness for the people who use it.

1. INTRODUCTION

In today's society there is still no great concern with the acoustic problems of the daily frequented places, however, if an analysis is done on this subject, we quickly see how harmed we sometimes are due to the poor acoustics of a space, either by the excessive effort to the understanding of speech or by the vocal effort caused on the speaker. Among the most critical cases of this situation are the classroom spaces. Often they do not present favorable acoustic conditions for a good understanding of the words, so impairing student learning [1].

This problem arises from the lack of awareness in the project stage of the space about the acoustic specifications necessary for its purpose. It is known in acoustic science that, if the space is used for communication by means of the word, then the intelligibility of the transmission will only be assured by deliberate attention and should therefore be a factor to be taken into account [2].

* Acknowledgments: Support from MIEEC programme of work developed in the scope of the MIEEC master dissertation; Support of Acoustics Laboratory of FEUP - António Costa (M.Eng.).

Through direct contact with the problem and the perception of its impact it became necessary to intervene.

This problem was identified in some lecture spaces of FEUP from internal reports and studies and above all, through the common experience of students and teachers [3, 4]. In the course of previous studies in some of those spaces, a clear diagnosis of the problem was achieved, through objective and subjective tests, and some pilot interventions were carried out, however, the changes made in the chosen rooms were quite profound and expensive, making it difficult to generalize to the whole school.

A new study was carried out in the Amphitheater B013 of FEUP, one of the more abundant types of lecture rooms that may be encountered at FEUP, in terms of quantity times capacity ranking, to evaluate how far is it from the necessary conditions for the purpose and to present solutions.

Is it possible to evaluate the acoustic conditions of a space not only through experimentation but also by modeling the space using appropriate software. This is a great advantage when it comes to evaluate and simulate changes to the space in an economic way. To do this there are several softwares available, such as, EASE [5], ODEON [6], Olive Tree Lab-Room[7] and CATT [8], among others. In these softwares, auralization might also be available allowing a subjective evaluation of the space through its digital model where it is possible to actually ear a simulated sound as if the person was inside, on a specific spot of the model. This also brings the advantage of having a perception of how sound will be heard after simulating an intervention, saving all the costs of implementing the solution experimentally in the place, which is very important in the case of a preliminary study.

2. METHODOLOGY OF STUDY, ASSESSMENT AND ACOUSTIC DESIGN OF CLASSROOMS

Since the problem under analysis is centered on the evaluation of an existing lecture room, a methodology is proposed for its acoustic enhancement, that, in this case, a corrective one, since the space in study is already constructed. The work addressed in this paper can be separated in two phases, a preliminary phase and an implementation phase. The present paper describes only the first phase which was already accomplished. Thus, the workflow proposed for the preliminary phase consists of the steps presented in figure 1.

For the following implementation phase, where the design will be applied to the space, what is recommended is to intervene with alteration of the acoustic architecture by means of some acoustic materials for passive correction as well as with introduction of a simple speech reinforcement system, composed of microphones, amplifier and one loudspeaker. Finally, the project should finish by taking measurements for verification of the effectiveness of the intervention.

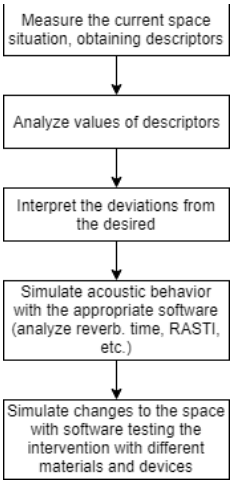


Figure 1: Workflow proposed for the preliminary phase of the acoustic assessment and rehabilitation.

3. CASE STUDY: AMPHITHEATER B013 (FEUP)

3.1. Acoustic evaluation

The acoustic evaluation consisted of an *in loco* data collection for calculation of a set of descriptive acoustic parameters of the space. For the amphitheater B013, composed of 98 seats, of which two pictures are presented in figures 2 and 3, the set of parameters, which were considered important, taking into account that the space has the purpose of speech communication, was composed by: reverberation time, RASTI, definition, clarity, and also background noise [9] [10]. Two types of sound sources besides a RASTI emitter were used to excite the room space, and a couple of measurement microphones, a sound-level meter and a RASTI receiver were employed to record sound and measure, respectively. Some pictures of the equipment used are presented in figure 4. Posteriorly, a specially developed Matlab program was employed to process the recorded signals and obtain not only the values for the definition and clarity but also reverberation times for additional sub-bands not given by the sound-level meter. The impulse responses of the space were also obtained for two distinct locations, in rows 2 and 6, using the same software with additional averaging.

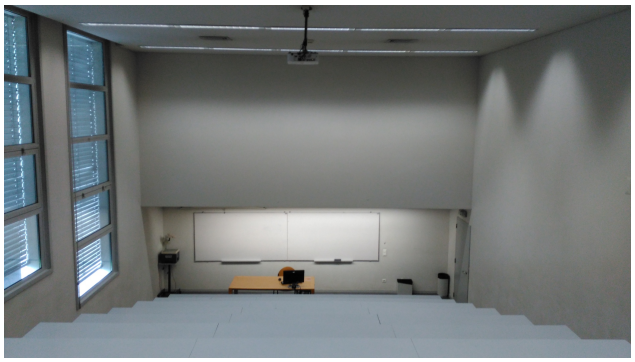


Figure 2: Front of amphitheater B013.

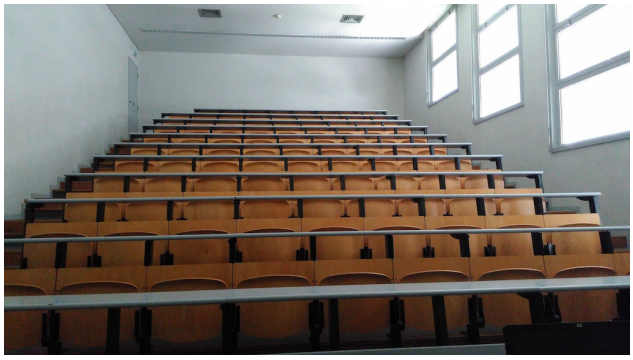


Figure 3: Rear of amphitheater B013.

Figure 4: Material used for the measurements. From left to right in the top row are two sound sources and a sound-level meter. In the bottom row are a RASTI emitter and a RASTI receiver.

For the wideband global reverberation time (averaging the results for the octave bands of 500 Hz, 1 kHz and 2 kHz) a value of 2.35 s was obtained. A graphic with the RT frequency distribution can be seen in figure 5 .The other mean obtained values were: 0.45 for RASTI, for D_{50} and C_{50} in row 2, 0.45 and -0.79 dB, respectively and 38.4 dBA for background noise. Table I presents the obtained values in comparison with the ones that would be adequate for this room[2] [11] [12]. Through a reflectometry analysis of the room impulse responses, we observed a large number of important late reflections that contribute to impair the speech intelligibility.

With those values so distanciated from the desired levels, a clear need for intervention in the space was proved.

3.2. Acoustic simulation

Exploring the possibility of evaluating the space not only through experimentation, but also using simulation software allows to take profit of the advantage that, after the required model is completed, changes may be inserted simulating space interventions virtually, in a rather quick and economic way. The studied space was sim-

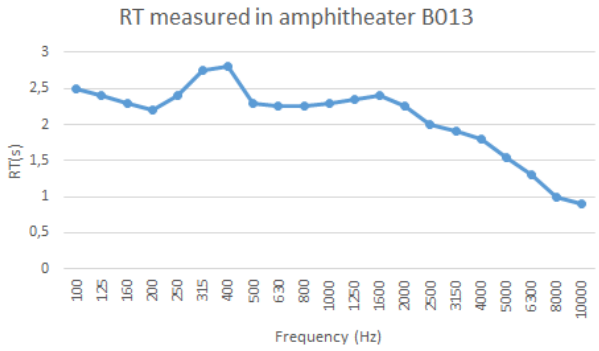


Figure 5: Measured reverberation time on amphitheater B013.

Table 1: *Presentation of the values obtained for the selected acoustic parameters in the amphitheater B013 in relation to their appropriate values.*

Parameter	Appropriate range	Obtained value
RT [500-1kHz] (s)	0,7-0,8	2,42
RASTI	$\geq$ 0,6	0,39-0,54
Definition	$>$ 0,5	$\leq$ 0,45
Clarity (dB)	$>$ 0	$\leq$ -0,79
Background noise (dBA)	$<$ 40	38,4

ulated in EASE and the descriptive acoustic parameters were obtained from the model. The model construction required some iterations and fine-tuning. Finally, simulation results were close to the experimental ones, allowing to conclude that the model was a good approximation of the reality, as can be verified by comparing figures 5 and 8 (no intervention) RT plots.

When constructing the model the first things that were needed to be taken into account were the dimensional aspects of the room, which include the dimensions of the space and its elements such as stairs, doors and windows. Consultation of the building construction blueprints still left some dimensions to be measured on site due to small alterations that were not clear in the drawings. After this part was accomplished it was necessary to carefully close the model in geometrical terms, otherwise the simulated room geometry would have leaks and the software would not allow a good simulation.

After modeling the space geometrically, several essential aspects were considered such as discovering which materials grades were used in the amphitheater in order to reproduce them in the model. It was taken into account the absorptions coefficients variations by frequency, thicknesses, mounting of false ceilings, etc. All these aspects have an impact on the acoustics of the space and for this reason they should be considered in the model.

Having reached this phase it became possible to check the accuracy of the model, not only in an objective, but also in a subjective way, using auralization. This is the process of producing the sound field created by a source in the space in a virtual way, in order to simulate a listener's binaural sensation in a defined position of the modeled space. In the developed work, auralization allowed to compare the audio recordings previously done in the space to the ones obtained with the model and verify that the sounds obtained were similar, giving the model a subjective validation as a good representation of reality.

4. SIMULATED ACOUSTIC REHABILITATION

4.1. Simulated rehabilitation of the amphitheater B013

With the calibrated model, and noting the need for intervention in the space, some solutions were considered and studied for the required improvement of its characteristics. Thus, a set of three valences were analyzed, namely, spreaded change of absortion on the enclosure surface, the use of spot absortion devices and the use of sound reinforcement as a complement.

For this amphitheater after several computational simulations, it was found that the use of the absorbent material K13 applied at the rear of the space, on the back wall and the rear part of the ceiling (2.5m length along the width of the ceiling), presented the best relation between obtained results and cost. This conclusion was obtained through the reflectometry study with which it was possible to find where the most adverse reflections come from and to guide the placement of absorbent material to attenuate them. Thus, the use of this material is proposed mainly to improve the intelligibility of the space. However, when examining the direct sound pressure level at the audience it was also noted that there was a decrease and this created a need to introduce sound reinforcement in the space to allow the direct sound to reach the listeners on the rear rows with sufficient intensity [13] [14]. Thus, a study was made on the best minimal approach for sound reinforcement taking into account the localization of the speaker and which loudspeaker type and location to use. The choice of the loudspeaker type is crucial since its characteristics models the sound radiation and how directly it reaches the listeners. For this phase the simulation is a specially important tool which allows to study the position and comparison of several loudspeakers as well as their driving parameters, time delay and power, and to select the one which produces the desired results.

In this way, the complete intervention proposal presented in this work for the studied amphitheater consists of a combination of the placement of absorbent material (K13) in the back of the amphitheater and the introduction of sound reinforcement as a complement. A representation of the model geometry of this intervention is depicted in figure 6 and the proposed electroacoustic chain to use in figure 7. The view point of the representation in figure 6 is below the floor and two walls are removed. The white board backside is visible in green, the professor's desk in brown, the loudspeaker in light blue and the new absorbent material in dark blue. If needed, color pictures are available by request to one of the authors or by downloading from the following link: `https://www.dropbox.com/s/emc74rtyv7bfhac/Model.PNG?dl=0`.

Simulating in the EASE software the changes proposed above, there was a decrease in the reverberation time in function of frequency between 0.31 s and 1.51 s, as can be seen in figure 8. It was also possible to increase the mean RASTI value to 0.6, thus reaching a subjective rating evaluated as good. In figures 9 and 10 the simulated distribution of RASTI before and after the intervention is shown. The simulated mean value of C50 after intervention also changed from -4.28 dB to 0.36 dB.

By using the proposed chain presented in figure 7, the direct sound pressure level received by the listeners is increased from 52dB before rehabilitation to near 61dB. In figure 11 the distribution of the direct sound pressure level before rehabilitation sim-

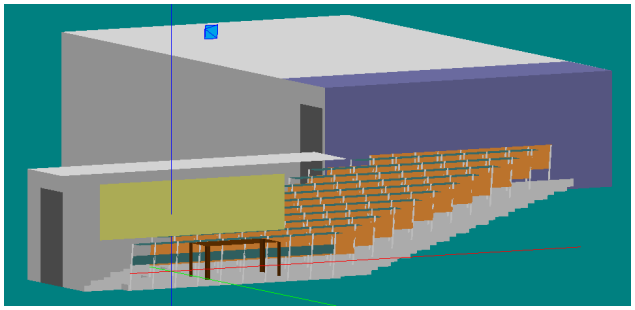


Figure 6: Model geometry of the amphitheater after intervention on EASE.

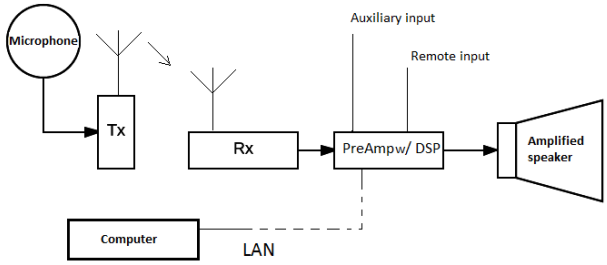


Figure 7: Proposed audio chain for sound reinforcement in the amphitheater B013.

ulated on EASE is shown and in figures 12 and 13, respectively, the direct and total sound pressure levels with acoustic reinforcement simulated on EASE, are represented. It may be concluded that this simple borderline intervention can guarantee a significant improvement of the values of the descriptive acoustic parameters of the space.

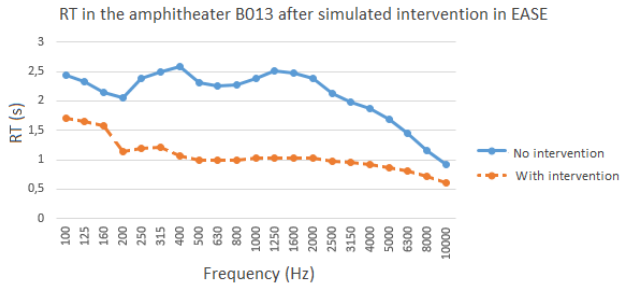


Figure 8: Simulated reverberation time before and after the intervention on EASE.

4.2. Use of auralization for subjective apreciation

As previously mentioned, auralization is a powerful tool for the subjective evaluation of a simulated model. In this work it had an important role in the validation of the model and in the simulation phase of rehabilitation since it allowed to verify the sound quality improvement introduced in the space with the application of the proposed solution. In order to do this, a speech sound segment was recorded in an anechoic chamber and the same sound recorded in the space under study, in rows 2, 6 and 10. The anechoic recording was submitted to auralization sound treatment with the EASE software where the result of this process is a binaural recording demonstrating how the sound would be perceived in the room. The result of this whole process before the rehabilitation allowed not only the quantitative but also the qualitative validation of the model, by the authors and a small group of test listeners, and to verify the improvement of the same after rehabilitation.

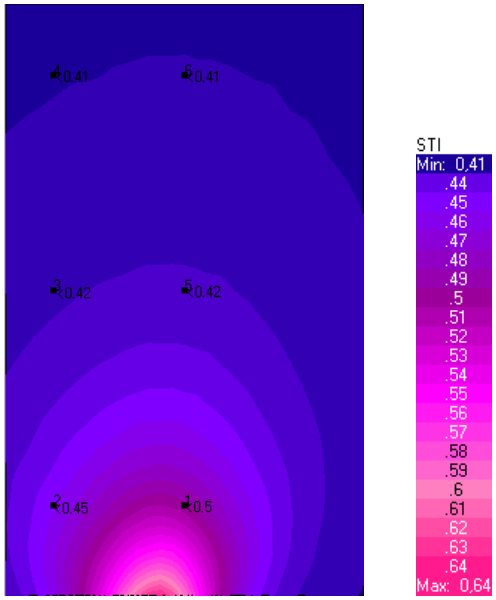


Figure 9: Distribution of simulated RASTI before intervention on EASE.

5. CONCLUSIONS

The objective of this work was fully achieved, having reached the enhancement of the acoustic design of a space adapted to speech communication with minimized implementation costs, through a proposal of intervention using a minimal amount of absorbent material and complementary introduction of a simple sound reinforcement system. Thus, a workflow is proposed for the acoustic assessment and rehabilitation design, which can be applied to several spaces, and also a way to combine acoustics and electroacoustics while reaching all main quality specifications is presented with the purpose of minimizing application prices. These are the main contributions of this work which can serve as guidelines. In this way, to implement this proposal, is a solution to provide greater acoustic comfort to the people who attend the treated space.

It is also concluded with this work that the power of the acoustic design simulation tools was demonstrated, allowing that in a simple and effective way, time and resources be saved since after the model is built it allows to simulate several changes to the space to arrive at the desired result. It is still important to emphasize the importance of auralization in the process of acoustic design since it allows to subjectively predict how the sound will be perceived

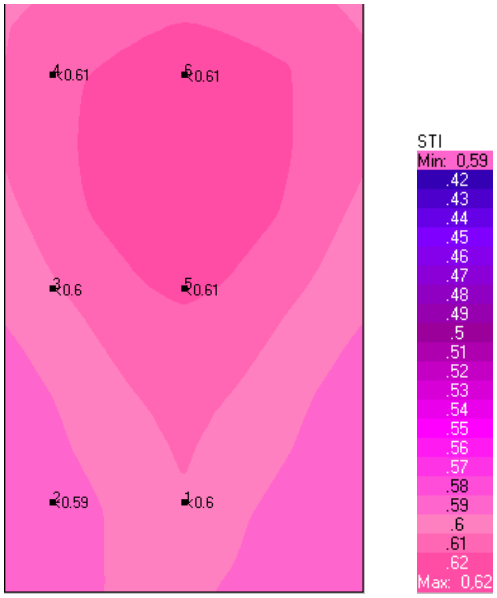


Figure 10: Distribution of simulated RASTI after intervention on EASE.

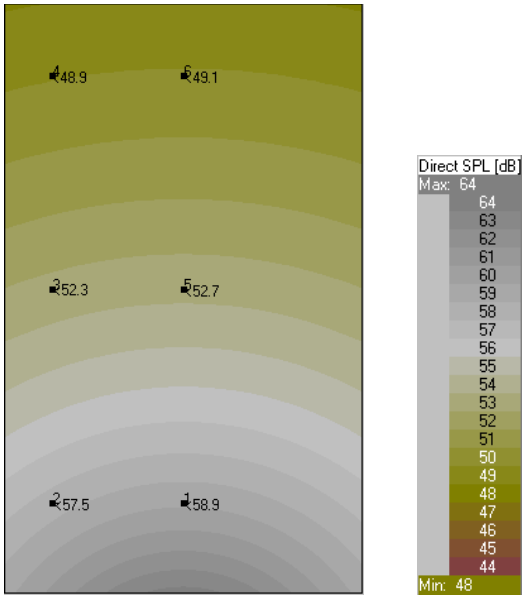


Figure 11: Distribution of direct sound pressure before rehabilitation simulated on EASE.

even before the space is built or to undergo intervention.

For future work the characterization of the average of people generated noise, the integration of other sources of background noise into the model and the other acoustic effects of the audience, mainly its per capita sound absorption, should be done to replace average empirical coefficients that are generally used and therefore, increase model accuracy. Also, in order to make the evaluation of the space more complete, it would be appropriate to introduce in this study the performance evaluation by means of subjective panel tests.

6. REFERENCES

[1] C. M. Silva, "O tempo de reverberação e a inteligibilidade da palavra," July 2013, Masters dissertation, FEUP, (in Portuguese), Universidade do Porto.

[2] A.P.Oliveira de Carvalho, *Acústica Ambiental e de Edifícios (in Portuguese)*, 2016, Edition 8.12.

[3] N. Rodrigues N. R. e Castro P. de Sousa R. Pinto J. Gonçalves, N. Diogo, "Inteligibilidade nas salas de aula da feup," 2005, internal report (in Portuguese) available by request, FEUP.

[4] L. Afonso M. Chivarria R. Nelson C. Lopes, J. Carvalho, "Inteligibilidade nas salas da feup," 2005, internal report (in Portuguese) available by request, FEUP.

[5] "Ease website," `http://ease.afmg.eu/`.

[6] "Odeon website," `https://odeon.dk/`.

[7] "Olive tree lab-room website," `https://www.mediterraneanacoustics.com/`.

[8] "Catt-acoustic website," `https://www.catt.se/`.

[9] F. A. Everest, *Master Handbook of Acoustics*, McGraw-Hill, Fourth edition, 2001.

[10] L. E.Kinsler, *Fundamentals of Acoustics*, Wiley, 1982.

[11] J. J. Smaldino C. Flexer, C. C. Crandell, *Sound Field Amplification*, Thomson, Second edition, 2005.

[12] G. M. Ballou, *Handbook for Sound Engineers*, Focal Press, Third edition, 2002.

[13] B. P. Ortega M. R. Romero, *Electroacústica: Altavoces y Microfónos*, Pearson Educación, 2003, (in Spanish).

[14] J. Eargle, *JBL professional's sound system design reference manual*, 1999.

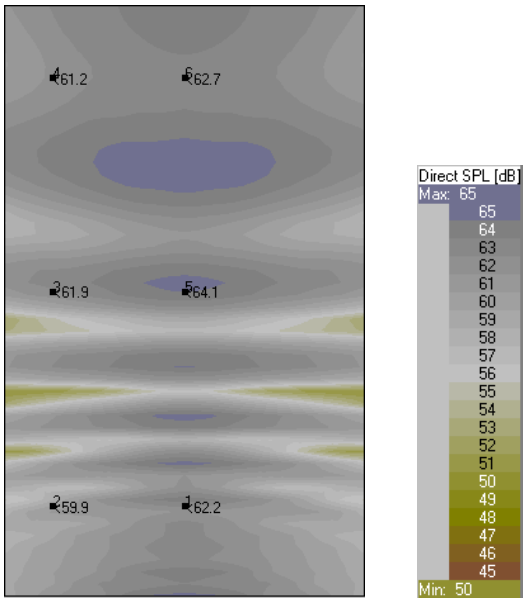


Figure 12: Distribution of direct sound pressure after rehabilitation simulated on EASE.

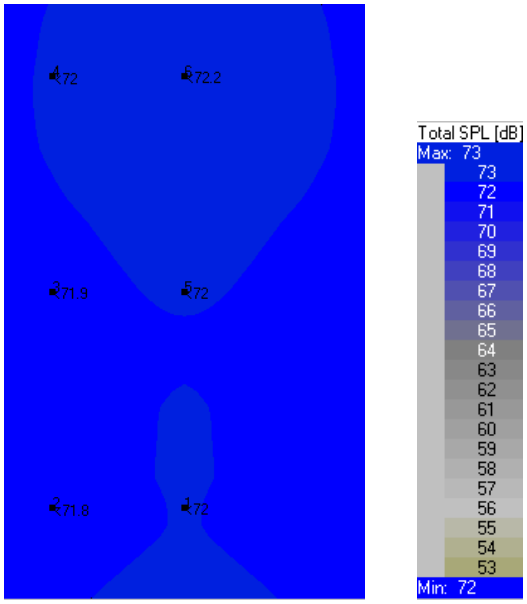


Figure 13: Distribution of total sound pressure after rehabilitation simulated on EASE.

USING SEMANTIC DIFFERENTIAL SCALES TO ASSESS THE SUBJECTIVE PERCEPTION OF AUDITORY WARNING SIGNALS

Joana Vieira
Ergonomics Lab
Faculty of Human Kinetics
CIAUD, Lisbon School of Architecture, Universidade de Lisboa
ALGORITMI, University of Minho, Braga
CCG - Centre for Computer Graphics
Guimarães, Portugal
joana.vieira@ccg.pt

Jorge Almeida Santos
School of Psychology
University of Minho
Braga, Portugal
ALGORITMI, University of Minho, Braga
CCG -Centre for Computer Graphics
Guimarães, Portugal
jorge.a.santos@psi.uminho.pt

Paulo Noriega
Ergonomics Lab
Faculty of Human Kinetics
CIAUD - Lisbon School of Architecture
Universidade de Lisboa,
Portugal
pnoriega@campus.ul.pt

ABSTRACT

The relationship between physical acoustic parameters and the subjective responses they evoke is important to assess in audio alarm design. While the perception of urgency has been thoroughly investigated, the perception of other variables such as pleasantness, negativeness and irritability has not. To characterize the psychological correlates of variables such as frequency, speed, rhythm and onset, twenty-six participants evaluated fifty-four audio warning signals according to six different semantic differential scales. Regression analysis showed that speed predicted mostly the perception of urgency, preoccupation and negativity; frequency predicted the perception of pleasantness and irritability; and rhythm affected the perception of urgency. No correlation was found with onset and offset times. These findings are important to human-centred design recommendations for auditory warning signals.

1. INTRODUCTION

The study of the psychological correlates of physical parameters motivated early psychophysical research. This was considered a tool to better study and understand the mind [1]. Early studies focused on sensory thresholds of humans, associating the human response to the systematic variation of a physical stimulus. Nowadays, this interest in human response is broader. Could we know more than sensory responses? Could similar methods be used to comprehend the relation between physical parameters and affective responses?

Several experimental methodologies attempt to understand the association of physical parameters with subjective perceptions and evaluations by humans. Mostly derived from these early works, it is common to have controlled laboratorial set-ups to understand how certain emotional states can be triggered. This happens because there is consensus and robustness in what a culturally similar group of participants finds *pleasant*, *attractive*, or *annoying*.

For instance, semantic profiling stemmed from the wine tasting industry and is currently being applied in other areas such as acoustics [2], [3]. Here, the evaluators can taste and compare several samples of wines and then verbally create a vocabulary describing the perceptual differences between the wines. Later, consensus is achieved among all gathered vocabularies. Another technique, Kansei Engineering [4], originated in the automotive industry in Japan intending to quantitatively connect affective responses of the customers to physical design specifications. The evaluation method pairs representative samples of the product under evaluation with representative words usually presented in a semantic differential scale (a scale between two polar adjectives).

In the auditory modality, the semantic differential scale method is used to understand which variations in which acoustic parameters should be implemented in order to trigger the appropriate affective, attentional or motor response. While the method is commonly applied in alarm design (e.g.: trendsons [5]), disciplines such as sound design for products [6] or music theory [7] are also interested in knowing exactly which acoustic structure originates which affective response.

In the auditory alarm design context, early work by Roy D. Patterson [8], [9], Judy Edworthy and Elizabeth Hellier [10]–[12] has set the fundamental work grounds to understand the perception of urgency. However, not all auditory warning signals are associated with urgent events, and thus the same work needs to be made to comprehend which acoustic parameters might trigger, for instance, irritability, preoccupation, unpleasantness or others - depending on their context and adequate response. This knowledge will allow designing more appropriate audio alarms or warning signals for environments heavily populated with alarming sounds, such as control rooms, intensive care units or operating theatres.

The purpose of this study is to use a semantic differential scale methodology to understand which acoustic parameters of simple computer-generated sounds have an effect on perceived urgency, pleasantness, irritability, preoccupation, speed, and positiveness. Its specific aim is to create a predictive model that indicates which acoustic parameters (spectral or temporal) activates the subjective perceptions mentioned above. In the future, these findings will be used for the design of auditory warning signals from medical devices.

2. METHOD

The selected methodology was based on previous studies of Kansei Engineering [13] and semantic differential scales applied to psychoacoustic studies [10], [11], [14].

2.1. Selection of representative pairs of words

When using semantic differential scales, it is of extreme importance to select pairs of words that can adequately describe the object under evaluation [4]. For this, in a pilot study, people were asked to suggest words they associated with artificial sounds, in all possible contexts. Any words, adjectives or not, were accepted. Examples of sounds were referred, such as sounds from household devices, electronics, sounds from inside the vehicle, or alarms

from queuing services. In total, 183 words were suggested that described sensations, emotions and perceptions evoked by sounds. The most frequent words were *shrieking, loud, alert, irritating, deafening, confusing, noisy, pleasant, short,* and *sweet*. Other words related with a) physical properties (*low, short, long, fast, vibrant, synchronous, slow, repetitive, harmonious*); b) positive feelings (*relief, calm, curiosity, fresh, gentle, positive, relaxing, pleasant, melodic, peaceful, soft*); c) negative feelings (*boring, anxious, unpleasant, strident, fiddly, nervous, stressful, irritating, intrusive, angry, frustrating, penetrating*); d) other words (*critical, strong, important, order, respect, safety, attention, artificial*).
All were grouped considering similitude of meaning and frequency. This resulted in 11 words and corresponding negation. Then, five human factors and acoustics researchers selected the most fitted pairs to describe artificial sounds/alarms, resulting in 6 pairs. The final six pairs of words are in Table 1.

Table 1: *Pairs of Words used for evaluation.*

1	Not very - very Urgent
2	Unpleasant-Pleasant
3	Not very - Very Irritating
4	Not very - Very Preoccupying
5	Slow - Fast
6	Negative - Positive

All pairs of words were presented in an analog visual scale, ranging from 0 to 100 mm without numbers (*Figure 2)*

2.2. Selection of acoustic parameters

This phase consisted in selecting the acoustic parameters to be manipulated, so in the evaluation phase they could be paired with the chosen pairs of words. Two types of parameters were selected: spectral and temporal characteristics of sound. Four acoustic parameters were analysed in the present study:

1) *Frequency*: referred to by Hertz (Hz) where 1 Hz is one cycle per second.
2) *Amplitude Envelope*: the shape of a waveform's intensity throughout time. Rise (onset) and fall (offset) times were edited in milliseconds (ms).
3) *Speed*: determined by the inter-pulse interval with faster bursts possessing shorter inter-pulse intervals.
4) *Rhythm*: regular occurrence of an auditory event in time. This occurrence can have a given pattern that can be cyclic, thus having periodicity.

A total of three levels were defined for Frequency, Speed and Onset. Rhythm had two levels. The objective was to have three different levels of priority, similarly to an emergency signal (level 1 in table 2), a warning signal (level 2) and an information notification (level 3). Table 2 depicts all levels for each parameter.
Values and directionality of the variations were established after literature and international standards on the design of audio warning signals, detailed in the following sections.

Table 2*: Levels of variation in each acoustic parameter*

	1	**2**	**3**
F0 Frequency	2500 Hz	1500 Hz	500 Hz
Speed	x4	x2	x1
Rhythm	Regular	Regular	Regular
	Syncopated 0	Syncopated 5	Syncopated 10
Onset	Regular	Slow onset	Slow offset

2.2.1. Frequency

The fundamental frequency of a signal should depend on the purpose and context of the signal. Whether it is an emergency or an information signal, or whether it is to be used in a public or private space. For instance, Begault and Godfroy [15] proposes a range between 300 Hz – 1000 Hz for NASA's crew exploration vehicles, while ISO 7731 [16] for danger signals proposes frequency components in the 500 Hz to 2 500 Hz range. Specifically for medical devices, IEC 60601- 1-8:2012 [17] proposes a frequency range between 500 Hz and 3 000 Hz. Because the aim of this study is to help in the design of medical devices' audio alerts, three levels of the range suggested by [17] standard were chosen as a fundamental frequency.
All agree the auditory signals should have several harmonics. Begault and Godfroy [15] state that "*there should be four or more harmonically related spectral contents*"; IEC 60601- 1-8:2012 [17] and ISO 7731 [16] also propose four or more harmonics to improve spatial localization and signal audibility.
For this study, three levels of frequency (F0) were chosen: 2500 Hz, 1500 Hz, and 500 Hz. All had four harmonics.

2.2.2. Speed

ISO 7731 [16] recommends the temporal distribution of the signal to be pulsating rather than continuous in time; Patterson, Edworthy, and Lower [9] mention speed as the main variable for the perception of priority. ANSI/ASA S3.41 [18] recommends a temporal pattern of three 1-s pulses with 1.5s silence; ISO 9703-1:1994 [19] (this standard has been withdrawn) proposed multiple pulses with an interval between of 0.15- 0.5 s, depending on the priority of the alarm. Similarly, IEC 60601- 1-8:2012 [17] proposes three different pulse duration patterns according to high, medium or low priority of the alarm, respectively 75 ms to 200 ms (high) and 125 ms to 250 ms (medium and low), but only mentions the interpulse interval should be "*speeding up > regular/slowing*".
For this experiment, the strategy adopted by Edworthy, Loxley, and Dennis [10] was applied by creating three levels of speed with a systematic relationship: the faster speed was twice the speed of the moderate one, which was twice the speed of the slower one. The temporal distribution of the "pulse + silence" was repeated three times when the speed was x1 (slow) and x2 (moderate), and it was repeated five times when speed was in x4 (fast). However, the silence duration differed according to speed. In x1 it had 1 s, in x2 it had 0.5 s and in x4 it had 0.25 s.

2.2.3. Rhythm

The standard IEC 60601- 1-8:2012 [17] suggests syncopated or "off-beat" rhythms for higher priority alarms and regular rhythms

for medium and low priority alarms. Edworthy, Loxley, and Dennis [10] have found the inverse relation with syncopated rhythms being perceived as less urgent than a regular one.
For this study, the stimuli rhythm was based on the syncopation index of Fitch and Rosenfeld [20] (index 0, 5 and 10), and all stimuli were tested both with regular and syncopated rhythm.

2.2.4. Onset – Offset

The rise and fall time of an auditory warning is defined in IEC 60601- 1-8:2012 [17] as "*the interval over which the pulse increases from 10 % to 90 % of its maximum amplitude*". While initially this standard proposed a rise time of 10 to 20% of the stimuli's total duration, a 2012 amendment changed this to allow for rise times of up to 40% of the total duration. Due to hardware constraints, this rise time should not be less than 10-ms long. The manipulation of rise times provide, according to the standard, more psychoacoustic cues of greater urgency, where rapid rise times are perceived as more urgent than slow rise times. Edworthy, Loxley, and Dennis [10] found that a regular 20 ms onset was considered more urgent than a pulse with a slower onset.
In this study, stimuli had either a slow onset (180 ms; offset of 20 ms), a regular onset and offset (20 ms) or a slow offset (180 ms; onset of 20 ms).

2.3. Auditory Stimuli

In order to test all variables, a combination of all parameters was performed, generating 54 stimuli (3 Frequency x 3 Speed x 2 Rhythm x 3 Onset/Offset). All audio stimuli were generated in R Studio using Seewave [21] and TuneR [22] packages. A modular approach as first proposed by Patterson [8] and used in Edworthy, Loxley, and Dennis [10] was applied, where pulses were firstly created and then grouped into longer bursts of sound, which were then intercalated with periods of silence to form the full warning. All pulses were 200-ms long. Figure 1depicts two warning signals.

a)

b)

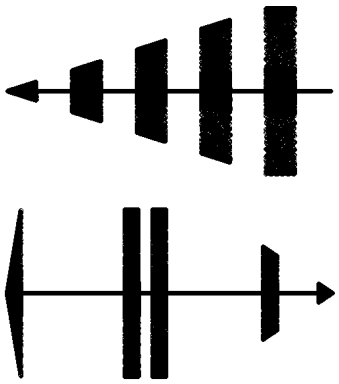


Figure 1: *Depictions of a) Stimuli with 180-ms onset, regular rhythm, speed level 4 (1500 Hz); b) Stimuli with 20-ms onset, 180-ms offset, syncopated rhythm, speed level 1 (500 Hz))*

2.4. Participants

Twenty-six participants took part in the study (17 female, 9 male), from 20 to 50 years (M=33, SD=10), all with normal hearing and most (22) without formal musical education. Data collections were carried out in two geographical locations in order to gather a higher number of participants, using the same equipment.

2.5. Apparatus

The study took place in a quiet room, where the participant was seated in front of a display and made the sound evaluation using a computer mouse by clicking on the visual analog scale. Participants used AKG Pro Audio K271 MKII headphones and all stimuli were presented using PsychoPy [23] software running on a Lenovo G500s with a 3rd generation Intel® Core™ i7-3612 processor and a Conexant Audio HD. Audio stimuli were presented in 77 dB SPL.

2.6. Procedure

Participants were welcomed and explained the main objective of the study, which consisted in evaluating several sounds according to a set of properties. They sat in front of a screen and placed the headphones. There was one participant per experimental session. After signing an informed consent and answering demographic questions, the instructions were given by the experimenter. These referred that after presenting a sound, an adjective was going to be presented, and participants should evaluate that sound according to that adjective. There were a total of six adjectives, and participants were told they should evaluate how much the sound was pleasant or unpleasant, irritating or not, preoccupying or not, slow or fast, urgent or not urgent and, finally, negative or positive (Table 3).
Participants were told they could only make the evaluation after hearing the entire sound once, which could last between 2 to 5 seconds. Participants could navigate with the mouse on the line of the scale, but after clicking with the mouse, it could not be changed. The scale was a continuous 100-mm scale. Before starting the experiment, all participants went through a training phase with the same scales and four different sounds (from [17]).

Table 3: *Descriptors per pair of words. A sheet with this information was always near the participant*

Pair of words	Description
Unpleasant Pleasant	I dislike the sound and it bothers me// I like the sound and it does not bother me
Not Irritating Very Irritating	The sound does not make me feel irritated and impatient// The sound makes me feel irritated and impatient
Not Preoccupying Very Preoccupying	The sound does not make me feel worried and alarmed// The sound makes me feel worried and alarmed
Slow Fast	The sound has a slow pace// The sound has a fast pace
Not Urgent Very Urgent	The sound communicates a need that may not be immediate// The sound communicates an immediate need
Negative Positive	The sound communicates a negative information// The sound communicates a positive information

After, the experimental session began, the screen displayed one pair of words at the time (Figure 2). The presentation of sound files was randomized, as well as the presentation of the pairs of words.

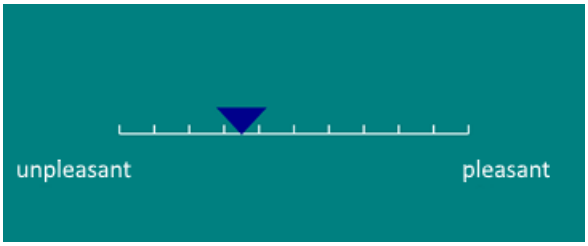


Figure 2: *Image of the evaluation interface with the semantic differential scale under evaluation*

Due to the great number of stimuli to be evaluated, there were two intervals, which had the length the participant preferred. The total procedure lasted between 30 to 40 minutes. Each participant evaluated each stimuli once for each pair of words. In total, each participant made 324 evaluations (54 stimuli x 6 pairs of words).

3. RESULTS

Because participants did not repeat the evaluation, it was important to assess the degree of agreement between participants as raters of a given stimuli. Outliers were removed from the sample using Tukey's method due to its independency from data distribution. This method ignores the mean and standard deviation, which are influenced by the outliers, by using an inter-quartile range approach (above and below the 1.5*IQR).

3.1. Inter-participant concordance

Kendall's W (also known as Kendall's coefficient of concordance) is a non-parametric statistic and can be used for assessing agreement among raters. Kendall's W ranges from 0 (no agreement) to 1 (complete agreement). The value of Kendall's W was calculated per pair of words to verify if the stimuli were rated in more or less the same order per participant. The results are in *Table 4*. All tests revealed a significant value of Kendall's W. As expected, because it was the most objective adjective, the pair "Slow-Fast" obtained the highest value of concordance, followed by "Not Urgent – Urgent".

Table 4: Values of Kendall's W

	Kendall's W
Slow – Fast	0.70 ***
Not Urgent - Very Urgent	0.61 ***
Not Irritating - Very Irritating	0.45 ***
Not Preoccupying -Very Preoccupying	0.42 ***
Unpleasant – Pleasant	0.36 ***
Negative – Positive	0.12 ***

*** *Significant (p < .001)* ** *(p <.01)* * *(p < .05)*

This analysis only shows consistency, and does not reveal the nature of the classification made by the participants. For this purpose, correlational (Table 5) and linear regression analysis were performed after all data was pooled and averaged.

Table 5: *Correlations between the four acoustic parameters and the six pairs of words*

	Frequency	**Speed**	**Rhythm**	**Onset**
Irritating	0.69 ***	0.13	0.27 *	0.04
Positive	-0.27 *	-0.70 ***	-0.18	0.02
Pleasant	-0.88 ***	-0.26	-0.14	0.04
Preoccupying	0.15	0.80 ***	0.31 *	-0.03
Urgent	0.11	0.83 ***	0.37 *	0.00
Fast	0.14	0.78 ***	0.41 ***	-0.01

	Irritating	**Positive**	**Pleasant**	**Preoccupying**	**Urgent**	**Fast**
Irritating	---					
Positive	-0.41 ***	---				
Pleasant	-0.82 ***	0.53 ***	---			
Preoccupying	0.40 ***	-0.88 ***	-0.48 ***	---		
Urgent	0.38 *	-0.86 ***	-0.45 ***	0.96 ***	---	
Fast	0.40 *	-0.85 ***	-0.47 ***	0.97 ***	0.98 ***	---

*** *Significant (p < .001)* ** *(p <.01)* * *(p < .05)*

The significant correlations found with *Frequency* were with Irritating ($r(52) = .69$, $p < .001$), Positive ($r(52) = -.27$, $p < .05$) and Pleasant ($r(52) = -.88$, $p < .001$); with *Speed*, the stronger correlations were with Positive ($r(52) = -.70$, $p < .001$), Preoccupying ($r(52) = .80$, $p < .001$), Urgent ($r(52) = .83$, $p < .001$), and Fast ($r(52) = .78$, $p < .001$); with *Rhythm* were Irritating ($r(52) = .27$, $p < .05$), Preoccupying ($r(52) = .31$, $p < .05$), Urgent ($r(52) = .37$, $p < .05$), and with Fast ($r(52) = .41$, $p < .001$). No correlations were found with the acoustic parameter Onset-Offset. For this reason, this variable will not be used in further analysis.
Additionally, it can be seen that the pair of words Irritating correlated significantly with all other words, negatively with Positive and Pleasant. The pair of words Positive and Pleasant were negatively correlated with Preoccupying, Urgent and Fast. And Preoccupying was correlated with Urgent, and Fast.
Following this, all relations between acoustic parameters and pairs of words were explored using linear regression models (Table 6-9).

3.1.1. Frequency

The *Frequency* (Table 6) variable had three levels, and each level increased the perception of unpleasantness of our participants, with 500 Hz ($B= 53.96$, $F = 100.2$, $R^2 = .80$, $p < .001$) 1500 Hz ($B= -15.55$, $p < .001$) and 2500 Hz ($B= -23.77$, $p < .001$). A similar pattern was found regarding the perception of irritableness, with 500 Hz ($B= 39.11$, $F = 29.43$, $R^2 = .54$, $p <$

.001) 1500 Hz (B= 19.82, p < .001) and 2500 Hz (β= 24.75, p < .001). No significant relations with Frequency were observed among the other pairs of words.

Table 6: *Results of linear regression by levels of Frequency (500 Hz, 1500 Hz and 2500 Hz). N = 54. 95% Confidence Interval (only* $R^2 > 0.5$ *are depicted)*

FREQUENCY	Unpleasant – Pleasant		Not Irritating - Very Irritating	
	B	**CI**	**B**	**CI**
Intercept (500)	53.96 ***	51.54 – 56.38	39.11 ***	34.26 – 43.96
1500	-15.55 ***	-18.97 – -12.12	19.82 ***	12.96 – 26.68
2500	-23.77 ***	-27.19 – -20.34	24.75 ***	12.96 – 26.68
F	100.2		29.43	
R^2	.797		.536	

**** Significant (p < .001) ** (p <.01) * (p < .05)*

These two regression models are plotted in Figure 3, with the y-axis depicting the 20-80 mm fraction of a 100-mm visual analog scale.

According to these results, the higher the sound's frequency, the more irritant and the less pleasant the sound is evaluated.

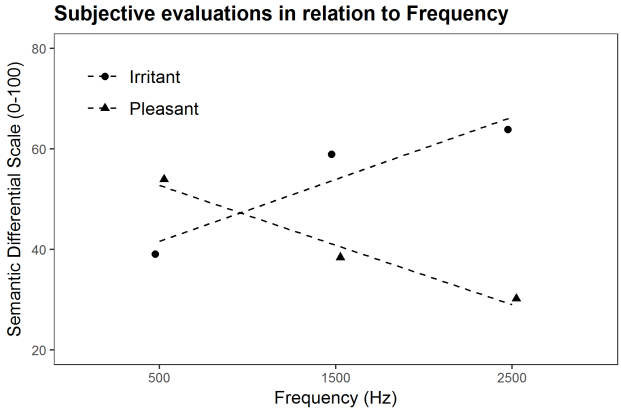


Figure 3*: Significant regressions for Frequency as predictor. Relationship between participant's evaluation of a sound as Irritable (R^2 = .54) or Pleasant (R^2 = .80) and three levels of increasing frequency.*

3.1.2. Speed

The *Speed (*Table 7*)* variable also had three levels and it was the variable which better explained the variance of four pairs of words. As the speed increased, so did the perception of Urgency (Speed x1 B= 31.70, p < .001, Speed x2 B= 23.82, p < .001 and Speed x4 B= 39.68, p < .001, F = 76.11, R^2 = .75): Preoccupation (Speed x1 B= 34.31, p < .001, Speed x2 B= 20.37, p < .001 and Speed x4 B= 30.78, p < .001, , F = 69.47, R^2 = .73) and Speed (Fast) (Speed x1 B= 32.95, p < .001, Speed x2 B= 27.33, p < .001 and Speed x4 B= 40.85, p < .001, , F = 63.9, R^2 = .72). The inverse pattern was observed in the Negative-Positive pair of words, with the perception of positiveness decreasing as the speed increased (Speed x1 B= 50.60, p < .001, Speed x2 B= -8.75, p < .001 and Speed x4 B= -11.95, p < .001, F = 37.9, R^2 = .60). No significant relations with Speed were observed among the other pairs of words.

The four regression models are plotted in Figure 4, with the y-axis depicting the 20-80 mm fraction of a 100-mm visual analog scale.

According to these results, the higher the sound's speed, the more urgent, fast and preoccupying and the less positive it is evaluated.

Table 7: *Results of linear regression by levels of Speed (x1, x2, x4). N = 54, 95% Confidence Interval (only* $R^2 > 0.5$ *are depicted)*

SPEED	Not Preoccupying- Very Preoccupying		Slow – Fast		Not Urgent – Very Urgent		Negative – Positive	
	B	**CI**	**B**	**CI**	**B**	**CI**	**B**	**CI**
Interc. (x1)	34.3 ***	30.5 – 38.1	33.0 ***	27.7 – 38.2	31.7 ***	27.1 – 36.3	50.6 ***	48.6 – 52.6
x2	20.3 ***	15.0 – 25.7	27.3 ***	19.9 – 34.7	23.8 ***	17.3 – 30.3	-8.8 ***	-11.6 – -5.9
x4	30.7 ***	25.4 – 36.1	40.9 ***	33.5 – 48.2	39.7 ***	33.2 – 46.2	-12.0 ***	-14.8 – -9.1
F	69.47		63.9		76.11		37.9	
R^2	.731		.715		.749		.598	

**** Significant (p < .001) ** (p <.01) * (p < .05)*

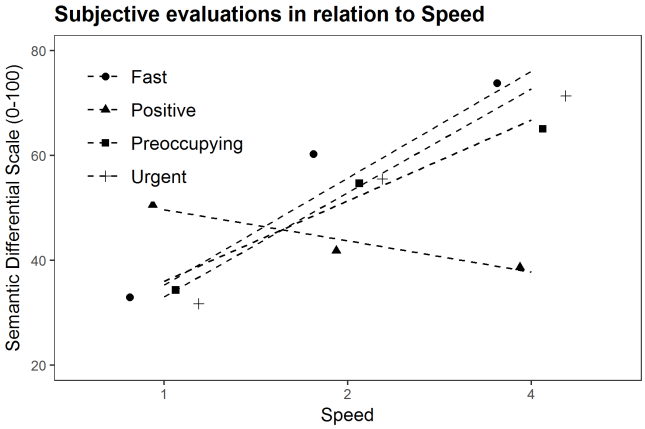


Figure 4: *Significant regressions for Speed as predictor. Relationship between participant's evaluation of a sound as Fast (R^2=.71), Positive (R^2=.60), Preoccupying (R^2=.73), or Urgent (R^2=.75) and three levels of increasing speed*

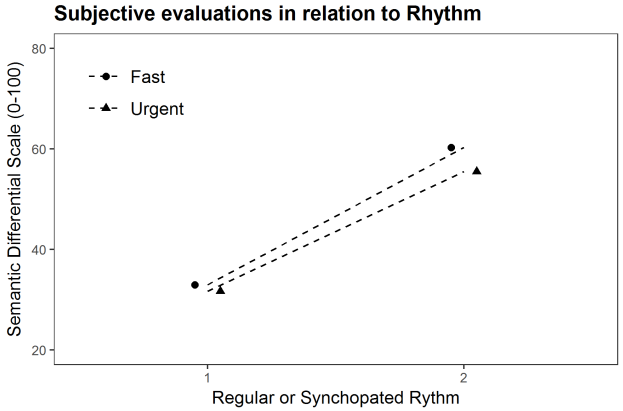


Figure 5: *Significant regressions for Rhythm as predictor. Relationship between participant's evaluation of a sound as Fast (R^2=.17) and Urgent (R^2=.14) and two levels of Rhythm (regular or syncopated).*

3.1.1. Rhythm

The *Rhythm (Table 8)* variable had two levels and the perception of speed (word fast) (Regular B= 47.47, p < .001, Syncopated B= 16.41, p < .001, F=10.4, R^2 = .17) and urgency (Regular B= 45.89, p < .001, Syncopated B= 13.95, p < .001, F=8.26, R^2 = .14) increased when the rhythm was syncopated.

Table 8: Results of linear regression by levels of Rhythm (Regular, Syncopated). N = 54, 95% Confidence Interval (only R^2 > 0.5 are depicted)

RHYTHM	Slow – Fast		Not Urgent – Very Urgent	
	B	CI	B	CI
Intercept (Regular)	47.47 ***	40.25 – 54.69	45.89 ***	39.00 – 52.78
Syncopated	16.41 **	6.20 – 26.62	13.95 **	4.21 – 23.69
F	10.4		8.26	
R^2	.167		.137	

**** Significant (p < .001) ** (p <.01) * (p < .05)*

Having significant regression coefficients means the *Rhythm* is correlated with both subjective perceptions, nevertheless, the model does not account for the variability found among the data.

4. DISCUSSION

The obtained results demonstrate how the semantic differential scale methodology is robust and useful for the analysis of relations between subjective perceptions and physical acoustic parameters. Firstly, although some pairs of words were extremely subjective, like *unpleasant – pleasant*, and vague, like *negative-positive*, there was consistency among participants, revealed in the significant values of Kendall's *W* in all pairs of words.

This was an important result, as it allows to somewhat balance an obvious limitation of this study, which was the lack of repetitions of the evaluation sessions. At first, one could expect large inter personal variability regarding such subjective perceptions, but these observations serve as an addition to the strengths of this simple method. It is important to add that during the data collection phase, some participants had informally mentioned they had trouble in classifying a sound as negative or positive, even though they had the definition sheet nearby. It is then somewhat surprising to understand that, although difficult, the classification was congruent among raters, later relating significantly to the manipulation of the *Speed* variable.

Regarding the associations between subjective and physical variables, with *Frequency*, it was observed that the subjective perceptions in which it had more effect were *Pleasantness* and *Irritability*. Again, although apparently a very personal evaluation, most participants found high-frequency audio signals as unpleasant and irritant. This is an important result that confirms that an alarm, to essentially fit its purpose of communicating an urgent event, does not need to increase its frequency. In fact, it should not, as it only affects the negative affective perception of the signal

Also importantly, and in agreement with Patterson's suggestions and standard norms, *Speed* is the variable which mostly affects an alarm's perception of urgency, communication of preoccupation or that "something" negative is happening. In applied settings, it is important to bear in mind that an increase in these subjective perceptions should be made via inter-pulse interval.

Rhythm obtained results also aligned with the [17] standard, with participants evaluating as significantly more urgent those auditory signals with syncopation than those with regular rhythm. However, the association found was not robust, and no more elations can be made. One explanation can be that the irregularity of the rhythm might have been affected by the slow onsets and offsets, not allowing to hear the full structure of the auditory signal.

Contrary to the literature and standards, the onsets and offsets of the auditory signals had no effect on the perception of any pair of words. In the future, the variations of this parameter should be more numerous, and evaluations should consider this manipulation only. This would allow clarifying the effect this parameter has without interacting with other manipulations.

With this study, it was possible to understand which acoustic features trigger what affective state when designing for auditory

warning signals. For instance, that a signal to be understood as urgent should have shorter and irregular inter-pulse intervals, preferably with lower frequencies. However, these sound design recommendations must co-exist with other requirements such as the ability to localize audio warning signals in an open space, and the ability to recognize it among other devices with similar spectral and temporal patterns.

5. CONCLUSION

A study was performed to better understand the psychological correlates of acoustic parameters. Fifty-four stimuli were created manipulating frequency, speed, rhythm and onset and offset times. Twenty six participants listened to each stimuli six times, each time considering a different pair of words presented in a visual analog scale. These words were selected among more than a 100 sound-related words. The applied methodology consisted in using semantic differential scales. The findings allowed to consolidate this method as a good evaluator of subjective perceptions. Results have demonstrated that the acoustic features which most contribute to the perception of these states in audio stimuli are frequency (pleasantness and irritability) and speed (urgency, preoccupation and negativity). Rhythm also affected the perception of urgency, although to a lesser extent, with irregular rhythms obtaining higher ratings for the perception of urgency.
This was the first study intending to use a human-centred approach to the design of auditory warning signals. After these fundamental associations between acoustic parameters and subjective perception have been established, the next step will be to apply them in the design of better auditory warning signals for medical devices.

6. ACKNOWLEDGMENTS

This work was supported by grant no. POCI-01-0145-FEDER-031943, co-financed by COMPETE2020 under the PT2020 programme, and supported by FEDER

7. REFERENCES

[1] G. A. Gescheider, *Psychophysics: The Fundamentals*, Third Edit. London: Lawrence Erlbaum Associates, 1997.

[2] T. Lokki, H. Vertanen, A. Kuusinen, J. Pätynen, and S. Tervo, "Auditorium acoustics assessment with sensory evaluation methods," *Proc. Int. Symp. Room Acoust. Melbourne, Aust.*, no. August, pp. 1–10, 2010.

[3] T. Lokki, "Tasting music like wine: Sensory evaluation of concert halls," *Phys. Today*, vol. 67, no. 1, pp. 27–32, 2014.

[4] S. Ishihara, "Psychological Methods of Kansei Engineering," in *Kansei/Affective Engineering*, M. Nagamachi, Ed. Boca Raton: CRC Press, 2011, pp. 31–38.

[5] J. Edworthy, E. Hellier, K. Aldrich, and S. Loxley, "Designing Trend-Monitoring Sounds for Helicopters: Methodological Issues and an Application," *J. Exp. Psychol. Appl.*, vol. 10, no. 4, pp. 203–218, 2004.

[6] E. Özcan-Vieira, *Product Sounds - Fundamentals & Applications*. 2008.

[7] H. Von Helmholtz, *On the Sensations of Tone as a Physiological Basis for the Theory of Music.* Longmans, Green, 1912.

[8] R. D. Patterson, "Guidelines for Auditory Warning Systems on Civil Aircraft," Eindhoven, The Netherlands, 1982.

[9] R. Patterson, J. Edworthy, and M. Lower, "Alarm sounds for medical equipment in intensive care areas and operating theatres," London, 1986.

[10] J. Edworthy, S. Loxley, and I. Dennis, "Improving auditory warning design: relationship between warning sound parameters and perceived urgency.," *Hum. Factors*, vol. 33, no. 2, pp. 205–231, 1991.

[11] E. Hellier, J. Edworthy, and I. Dennis, "A comparison of different techniques for scaling perceived urgency," *Ergonomics*, vol. 38, no. 4, pp. 659–670, 1995.

[12] J. Edworthy, E. Hellier, and R. Hards, "The semantic associations of acoustic parameters commonly used in the design of auditory information and warning signals.," *Ergonomics*, vol. 38, no. 11, pp. 2341–2361, 1995.

[13] J. Vieira, J. M. A. Osório, S. Mouta, P. Delgado, A. Portinha, J. F. Meireles, and J. A. Santos, "Kansei engineering as a tool for the design of in-vehicle rubber keypads," *Appl. Ergon.*, vol. 61, 2017.

[14] E. Hellier and J. Edworthy, "On using psychophysical techniques to achieve urgency mapping in auditory warnings," *Appl. Ergon.*, vol. 30, no. 2, pp. 167–171, 1999.

[15] D. Begault and M. Godfroy, "Auditory Alarm Design for NASA CEV Applications," in *13th International Conference on Auditory Display*, 2007, pp. 131–138.

[16] ISO, *ISO 7731 - Ergonomics - Danger signals for public and work areas — Auditory danger signals*, vol. 2003. 2003.

[17] AAMI, *ANSI/AAMI/ IEC 60601- 1-8:2006 & A1:2012 MEDICAL ELECTRICAL EQUIPMENT – Part 1-8: General requirements for basic safety and essential performance – Collateral Standard: General requirements, tests and guidance for alarm systems in medical electrical equip*. 2013.

[18] A. S. of America, *ANSI/ASA S3.41 Audible Emergency Evacuation Signal*. 2015.

[19] I. O. for Standardization, *ISO 9703-1:1994 - Anaesthesia and respiratory care alarm signals*. 1994.

[20] W. T. Fitch and A. J. Rosenfeld, "Perception and Production of Syncopated Rhythms," *Music Percept. An Interdiscip. J.*, vol. 25, no. 1, pp. 43–58, Sep. 2007.

[21] J. Sueur, T. Aubin, and C. Simonis, "SEEWAVE, a free modular tool for sound analysis and synthesis," *Bioacoustics*, vol. 18, no. 2, pp. 213–226, Jan. 2008.

[22] U. Ligges, S. Krey, O. Mersmann, and S. Schnackenberg, "tuneR: Analysis of music," 2016.

[23] J. W. Peirce, "PsychoPy—Psychophysics software in Python," *J. Neurosci. Methods*, vol. 162, no. 1–2, pp. 8–13, May 2007.

SOUNDSCAPE AURALISATION AND VISUALISATION: A CROSS-MODAL APPROACH TO SOUNDSCAPE EVALUATION

Francis Stevens
Audio Lab
Department of Electronic Engineering
University of York
York, United Kingdom
frank.stevens@york.ac.uk

Damian T Murphy
Audio Lab
Department of Electronic Engineering
University of York
York, United Kingdom
damian.murphy@york.ac.uk

Stephen L Smith
Intelligent Systems Group
Department of Electronic Engineering
University of York
York, United Kingdom
stephen.smith@york.ac.uk

ABSTRACT

Soundscape research is concerned with the study and understanding of our relationship with our surrounding acoustic environments and the sonic elements that they are comprised of. Whilst much of this research has focussed on sound alone, any practical application of soundscape methodologies should consider the interaction between aural and visual environmental features: an interaction known as cross-modal perception. This presents an avenue for soundscape research exploring how an environment's visual features can affect an individual's experience of the soundscape of that same environment. This paper presents the results of two listening tests[1]: one a preliminary test making use of static stereo UHJ renderings of first-order-ambisonic (FOA) soundscape recordings and static panoramic images; the other using YouTube as a platform to present dynamic binaural renderings of the same FOA recordings alongside full motion spherical video. The stimuli for these tests were recorded at several locations around the north of England including rural, urban, and suburban environments exhibiting soundscapes comprised of many natural, human, and mechanical sounds. The purpose of these tests was to investigate how the presence of visual stimuli can alter soundscape perception and categorisation. This was done by presenting test subjects with each soundscape alone and then with visual accompaniment, and then comparing collected subjective evaluation data. Results indicate that the presence of certain visual features can alter the emotional state evoked by exposure to a soundscape, for example, where the presence of 'green infrastructure' (parks, trees, and foliage) results in a less agitating experience of a soundscape containing high levels of environmental noise. This research represents an important initial step toward the integration of virtual reality technologies into soundscape research, and the use of suitable tools to perform subjective evaluation of audiovisual stimuli. Future research will consider how these methodologies can be implemented in real-world applications.

1. INTRODUCTION

To provide a context for the methods used in the two listening test presented in this paper, this section includes a summary of the various research areas informing this study. This includes soundscape theory and evaluation, cross-modal perception, and green infrastructure.

[1]This work is part of an EPSRC supported doctoral training studentship: reference number 1509136.

1.1. Soundscape Theory

In his seminal text 'The Soundscape: Our Sonic Environment and the tuning of the World', R. Murray Schafer defines a soundscape as [1]:

> *'The sonic environment. Technically, any portion of the sonic environment regarded as a field for study. The term may refer to actual environments, or to abstract constructions such as musical compositions and tape montages, particularly when considered as an environment.'*

Soundscape analysis looks at the holistic experience of all sound in a given location, and aims to explore an individual's perception of, and interaction with, that environment [2]. In this way, soundscape analysis describes both the physical and perceptual properties of an environment [3]. This explains soundscape research's position as a convergence of multiple disciplines, including acoustic ecology, musicology, sociology, psychology, architecture, and acoustics [4, 5].

1.2. Cross-modal Perception

Cross-modal perception is where the stimulation of one sensing modality (for example vision) can influence the experience of another (e.g. hearing). A famous example of this phenomenon is the McGurk effect [6] where a change in the appearance of mouth movement can alter the phoneme heard in recorded speech.

In a soundscape context, cross-modal perception has been considered as a way of understanding how the visual setting of an environment can change the perception of that environment's soundscape. For example, Lercher and Schulte-Fortkamp showed living on a 'pretty' street could reduce noise annoyance [7] and Viollon et al. found that exposure to still images of natural environments incorporating natural features reduced the perceived 'noisiness' of a soundscape [8]. Research into this area is of great importance to human health and well-being, in terms of reduced stress due to lower levels of noise annoyance and other health effects (for example, a patient's recovery following an operation has been shown to be faster if the patient has access to a window with a pleasant view [9]).

1.3. Green Infrastructure

Broadly speaking, when considering noisy soundscapes, the kind of visual features that may be present to improve one's experience of noise can be collected under the term Green Infrastructure. A definition of Green Infrastructure is given in [10]:

> *'It can be considered to comprise of all natural, semi-natural and artificial networks of multifunctional ecological systems within, around and between urban areas, at all spatial scales.'*

Whilst the acoustic impact (noise level reduction, acoustic absorption to reduce reverberation times etc.) of green infrastructure may be minimal, the impact on perception of sound may be much more pronounced [11]. An underlying motivation for this research is to investigate to what extent the presence of green infrastructure and other natural, pleasant, visual features can reduce the negative effects of acoustic noise in a soundscape. This aligns with the Biophilia thesis, originating from the field of environmental psychology, which posits that human beings have an innate appreciation for, and affinity with, natural environmental features: particularly water and vegetation [12].

The motivation for the work presented here is to make use of visualisation and soundscape methodologies to understand how the presence of certain visual features can change the emotional response evoked by a soundscape. This includes a preliminary test making use of still panoramic images and ambisonic UHJ renderings of soundscape stimuli, and a main test making use of panoramic videos and dynamic binaural rendering of FOA soundscape recordings.

2. METHODS

This section will consider the research methods and approaches applied to this study, including the soundscape evaluation methodologies used, and the data collection process.

2.1. Subjective Evaluation

2.1.1. The Self-Assessment Manikin

A previous study [13] made a direct comparison between semantic differential (SD) pairs and the Self-Assessment Manikin (SAM) as methods for measuring a test participant's experience of a soundscape.

The use of SD pairs is a method originally developed by Osgood to indirectly measure a person's interpretation of the meaning of certain words [14]. The method involves the use of a set of bipolar descriptor scales (for example 'calming-annoying' or 'pleasant-unpleasant') allowing the user to rate a given stimulus. SD pairs are a well established aspect of listening test methodology in soundscape research [15–17]. Whilst useful in certain scenarios, they can be time-consuming and unintuitive [13]. An alternative subjective assessment tool to use is the SAM.

The SAM is a method for measuring emotional responses developed by Bradley and Lang in 1994 [18]. It was developed from factor analysis of a set of SD pairs rating both aural [19] and visual stimuli [20] (using, respectively, the International Affective Digital Sounds database, or IADS, and the International Affective Picture System, or IAPS). The three factors developed for rating emotional response to a given stimuli are:

- **Valence:** How positive or negative the emotion is, ranging from unpleasant feelings to pleasant feelings of happiness.
- **Arousal:** How excited or apathetic the emotion is, ranging from sleepiness or boredom to frantic excitement.
- **Dominance:** The extent to which the emotion makes the subject feel they are in control of the situation, ranging from not at all in control to totally in control.

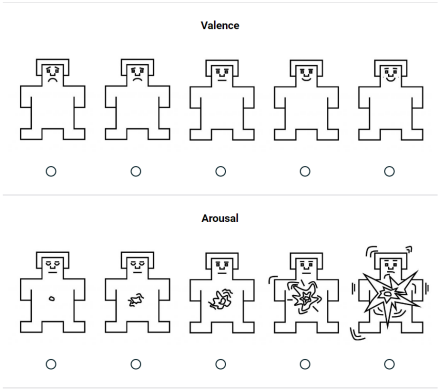


Figure 1: The Self-Assessment Manikin (SAM) as used in this study, after [18].

These results were then used by Bradley and Lang to create the SAM itself as a set of pictorial representations of the three identified factors. The version of the SAM used in this experiment (as shown in Fig. 1) contained only the Valence and Arousal dimensions following results from a previous study [13].

2.1.2. Soundscape Categorisation

The soundscape recordings used in this test were selected in order to cover as wide a range of sound sources as possible. In order to determine what such a set of soundscape recordings would contain, a review of soundscape research indicated that in a significant quantity of the literature [21–24] three main groups of sounds are identified:

- **Natural:** These include animal sounds (such as bird song), and other environmental sounds such as wind, rustling leaves, and flowing water.
- **Human:** Any sounds that are representative of human presence/activity that do not also represent mechanical activity. Such sounds include footsteps, speech, coughing, and laughter.
- **Mechanical:** Sounds such as traffic noise, industrial and construction sounds, and aeroplane noise.

Following results from a previous test [25] it was decided to include ratings scales for the test participants to evaluate the soundscape in terms of the three above categories. Fig. 2 shows the category ratings question as presented to the test participants. The purpose of including this question, in both the preliminary and main listening tests, was to see how the presence of visual features can alter the perceived category of an environment, and how this relates to evoked emotional state.

2.2. Data Collection

The data used in this study were collected from various locations around the North of the United Kingdom, including: Dalby forest, a natural environment; Pickering, a suburban/rural environment; and Leeds city centre, a highly developed urban environment. All of the soundscape recordings were made in FOA using a Soundfield STM 450 microphone [26]. Concurrent A-weighted noise level measurement were taken to allow for calibration of later auralisation.

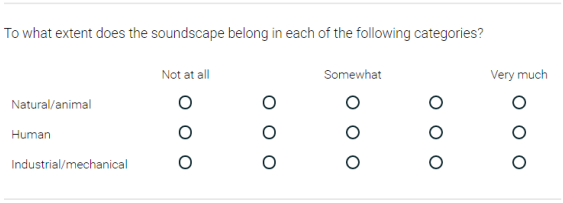


Figure 2: The category ratings question as presented to test participants.

Table 1 gives details of the sound sources present in each of the 16 clips used in the listening test. These clips were 30 seconds long and extracted from the 10 minutes of soundscape recording made at each location. These clips have been used in previous stages of this research [13, 27].

The visual data was collected at each recording location using six GoPro cameras mounted as the faces of a cube in a Freedom360 rig [28]. At each location a still image was taken immediately before recording began, and then full motion video recordings were made alongside the FOA sound recordings.

3. PRELIMINARY LISTENING TEST

This section covers the content creation and test procedure for the preliminary listening test, as well as its results. This includes the conversion of the FOA soundscape recordings to stereo UHJ format, and the stitching of the still GoPro photographs to create panoramic images of the recording location.

3.1. Stereo UHJ Conversion

In order to present the recorded soundscape material over headphones without head-tracking, the FOA signals had to be converted to a suitable two-channel format. It was decided to make use of Ambisonic UHJ stereo format, where the $\mathbf{W}$, $\mathbf{X}$, and $\mathbf{Y}$ channels of an FOA recording are used to translate the horizontal plane of the soundfield into two-channels [29]. The resultant signal can the be shared online and reproduced over headphones, allowing the FOA recordings to be used with the spatial content of the $\mathbf{W}$, $\mathbf{X}$, and $\mathbf{Y}$ channels preserved in reproduction. The use of this format has been established as ecologically valid in a prior stage of this research [30], where it was shown that emotional states evoked by exposure to the stereo UHJ format soundscape recordings were significantly similar to those evoked by full FOA renderings in a 16-loudspeaker listening rig.

The following equations are used to convert from the $\mathbf{W}$, $\mathbf{X}$, and $\mathbf{Y}$ channels of the FOA signal to two stereo channels:

$$S = 0.9397\mathbf{W} + 0.1856\mathbf{X} \quad (1)$$

$$D = j(-0.342\mathbf{W} + 0.5099\mathbf{X}) + 0.6555\mathbf{Y} \quad (2)$$

$$L = 0.5(S + D) \quad (3)$$

$$R = 0.5(S - D) \quad (4)$$

where j is a $+90^\circ$ phase shift and L and R are the left and right channels respectively of the resultant stereo UHJ signal [31]. Note that the Cartesian reference for FOA signals is given by ISO standard 2631 [32], and the $\mathbf{Z}$ channel of the FOA recording is not used.

3.2. Preliminary Test Procedure

The listening test was presented using Qualtrics [33] to administer the questions to the test participants, and using MATLAB to play the stereo UHJ audio and present the panoramic images using FSPViewer [34] (a freely downloadable viewer for spherical panoramic images). Presenting the images in this way allowed participants to click-and-drag the panoramic image to 'look' around the environment (which they were encouraged to do). These images were created using Kolor Autopano [35] to stitch together the still images from the GoPro cameras into single equirectangular spherical panoramic images. An example image can be accessed online [36].

All 16 soundscape clips were presented to the test participants in both the aural and audiovisual stages. These were presented in a random order each time and were preceded by two orienting stimuli. The audio-only test was completed by 31 test participants, and the audiovisual test was completed by 11 participants. Of the 31 audio-only test participants, 20 were male, and 16 were aged under 26. No demographic data were collected for the audiovisual test, as analysis of previous results did not indicate any significant effect on test results due to demographic factors. The next section includes an evaluation and discussion of the test results.

3.3. Preliminary Test Results

A Shapiro-Wilk's test was applied to all of the rating scales for each test stimuli as a test for normality [37]. Only a handful were identified as normally distributed. As such, in order to make comparisons between the results for the different stimuli, the Mann-Whitney test was used [38]. This test is suitable for comparing the values of two variables that are not normally distributed [39]. It is also suitable for comparing variables with small, arbitrary, sample sizes, including where the sample sizes of the two variables are different.

The purpose of applying the Mann-Whitney test was to indicate where the test results were significantly different for each of the five rating scales (Valence, Arousal, Natural, Human, and Mechanical) when comparing the results for the audiovisual stimuli with the audio alone. Fig. 3 shows the Mann-Whitney test results for the preliminary listening test data, indicating these significant differences. The next section will discuss theses results.

3.3.1. Significant Differences

The three clips showing a significant difference in arousal values are 6A, 6B, and 7B. For all three of these clips the arousal rating value was significantly larger when the clip was presented with the visual stimuli. Both of these recording locations were in Leeds city centre: one next to a main road (location 7); one on a pedestrianised street (location 6). This increase in arousal is therefore possibly due to the presences of cars and people in the images of the scenes that are not so pronounced in the soundscape recordings.

The 6 clips showing a significant difference in valence values are 1A-2A, 3A-3B, and 8A. As with the arousal results, for all of these clips the presence of visual stimulus results in an increase in valence. For clips 1A and 1B this is unsurprising: the soundscape clips contain some birdsong and insect noise, but despite their hi-fidelity (where the sound sources present are clearly defined with little background noise [1]) there is little information given to indicate the features of the recording location. As such it is to be anticipated the presence of the visual features with the soundscape results in an increased valence rating.

Location	Site	Clip A Sound Sources	Clip B Sound Sources
Dalby Forest (Rural/Natural)	1. Low Dalby Path	Birdsong, Owl Hoots, Wind	Birdsong and honking, Insects, Aeroplane flyby
	2. Staindale Lake	Birdsong, Wind, Insects, Single car	Insects, Birdsong, Water
North York Moors (Rural/Suburban)	3. Hole of Horcum	Birdsong, Traffic, Bleating	Birdsong, Traffic, Conversation
	4. Fox & Rabbit Inn	Traffic, Car door closing, Car starting	Traffic, Footsteps, Car starting
	5. Smiddy Hill, Pickering	Traffic, Car door starting, Conversation	Birdsong, Distant traffic
Leeds City Centre (Urban)	6. Albion Street	Busking, Footsteps, Conversation, Distant traffic	Workmen, Footsteps, Conversation, Distant traffic
	7. Park Row	Traffic, Buses, Wind, Busking	Busking, Footsteps, Conversation, Distant traffic
	8. Park Square	Birdsong, Traffic, Conversation, Shouting	Workmen, Traffic, Conversation, Birdsong

Table 1: Details of the sound sources present in the two 30 second long clips (labelled A and B) recorded at each of the eight locations.

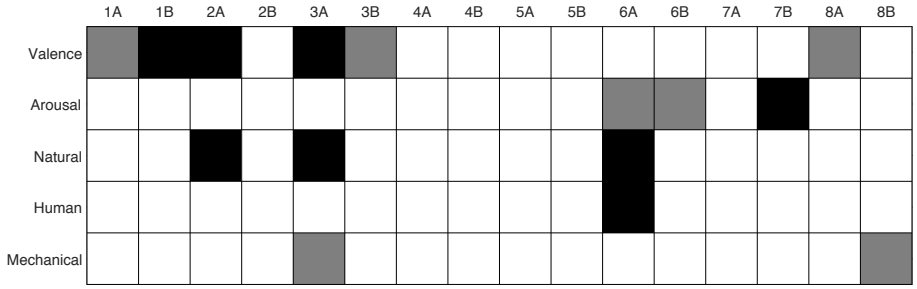


Figure 3: Mann-Whitney test results for the preliminary listening test, comparing results for each of the five rating scales for each of the 16 test stimuli when presented as the soundscape alone and with accompanying still panoramic images. Dark squares indicate a significant difference at 95% confidence ($p < 0.05$), and Light marked squares at 90% confidence ($p < 0.1$). White squares indicate no significant difference at either confidence level.

For clip 2A a similar effect can be observed, due to the presence of single car driving past. These results suggest that the visual setting (greenery and trees, peaceful lake, big sky) results in a significantly increased valence rating.

The significant increases in valence value for the audiovisual presentation of clips 3A and 3B also show the same effect: the aural information in these clips contains some natural sounds and traffic noise that indicate little about of the surrounding countryside of the North York Moors national park.

Likewise the soundscape of clip 8A contains some birdsong alongside quiet traffic noise (and some sounds of human activity), but the visuals recorded at that location show an inner city park with foliage, flowers, and some trees. This green infrastructure is clear when viewing the scene, but not evident in any explicit way in the audio-only presentation, and is likely responsible for evoking an alternative emotional state where reported valence levels (i.e. how pleasant the scene is) are higher.

The significant differences in the natural rating scale support this argument in part: clips 2A and 3A show a significant increase in the natural rating with the presence of visual stimuli, which includes a forest and countryside respectively. Clip 6A (recorded on a pedestrianised shopping centre street) also shows a significant increase in the natural rating with the presence of visual information. This environment contains some very minor elements of green infrastructure in the form of a couple of trees in some small pots. Whilst this cannot directly be correlated with a change in the valence rating for the environment, it does indicate how even a very slight presence of green infrastructure can change an individual's experience and perception of a location. This location also sees a significant decrease in the human category rating for the audiovisual presentation of the clip relative to the soundscape alone. This is possibly due to the difference between reality and expectation of the visual setting: the dominant sound sources in this clip are human sounds (including very loud conversation, footsteps, and some shouting) with only some distant traffic noise. However the visual setting is dominated by concrete in the form a pavement, shop-fronts and some larger inner city buildings reducing the impact of the human activity.

The two soundscapes showing a significant difference in the mechanical category rating are 3A and 8B, both of which saw a decrease in mechanical rating with the introduction of visual stimuli. In a way these two clips can be considered as the corollary of one another: clip 3A shows a natural environment 'interrupted' by the presence of a busy road; and clip 8B shows a green-infrastructure (a park) in the context of a large city. As such both of these soundscape clips indicate little about the features of the visual settings, resulting in a decreased mechanical rating for the audiovisual presentation.

3.3.2. Perceptual Noise Impact Rating

In order to further investigate the effect of certain visual features on the emotional state evoked by a soundscape, the valence and arousal rating scales can be combined to form a single measure of the emotional state evoked by a noisy soundscape. This new measure is called the Perceptual Noise Impact Rating (PNIR) and was introduced as part of this body of research in [40]. It is formulated by:

$$\mathrm{PNIR} = 1 - 0.5(1 - \mathrm{A} + \mathrm{V}) \tag{5}$$

where A and V represent the Arousal and Valence scores respectively (where the scores are normalised between 0 and 1).

Fig. 4 shows a summary of PNIR results from the preliminary listening test. Indicated in this plot are the mean PNIR values across all participants for each of the 16 stimuli for both the audio-only and audiovisual listening conditions. These results show a trend in the data towards three groups of PNIR values:

1. Clips 1A-2B: These soundscapes were recorded at two locations in Dalby forest, and are comprised of many natural

sounds (birdsong, insects, wind) and visual features (trees, a lake, open sky).

2. Clips 4A-7B: These soundscapes were recorded in highly developed environments, including various locations in the centre of the city of Leeds, and next to a road in the town of Pickering. The most commonly identified sound sources in these clips were traffic noise, other mechanical noise, and human sounds (footsteps and conversation).
3. Clips 3A-3B and 8A-8B: These soundscapes were recorded in environments that can be considered as being on the interface between the recording locations of the two above categories. Location 3 was next to a country road overlooking a wide expanse of countryside, and location 8 was in a park in Leeds city centre. Both of these environments contained a mixture of mechanical and natural sounds (i.e. relatively quiet traffic noise and birdsong) and visual features (i.e. flowers, trees and other greenery alongside the roads and buildings).

These three emotional groups were used alongside the Mann-Whitney test results to identify which of the soundscape clips to use in the main listening test.

Clips 1B and 2A were chosen to represent group 1: clip 1B was recorded in Dalby forest and contains natural sounds and visual elements; clip 2A was recorded at a nearby lake and again presents many natural sounds and visual elements, as well as a single car drive by.

Clips 6A and 7B were chosen to represent group 2: clip 6A was recorded on a pedestrianised street lined with shops; clip 7B was recorded next to a busy road in Leeds city centre. Both of these clips contain mainly human and mechanical sounds, with little in the way of natural sounds or visual elements.

Clips 3A and 8A were chosen to represent group 3: clip 3A was recorded next to a road in the North York Moors national park; clip 8A was recorded in a small park in the centre of Leeds. As stated above, these locations both represent something of an interface between natural and developed habitats and contain both human and natural sounds and visual elements, including the presence of green infrastructure.

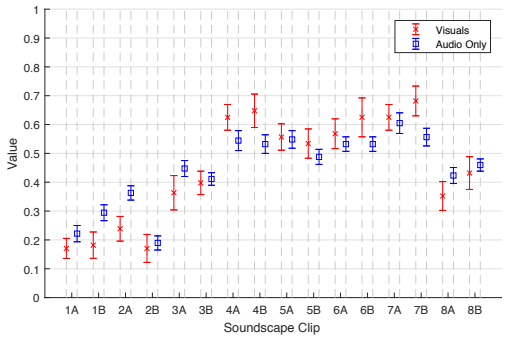


Figure 4: A summary of PNIR ratings from the preliminary listening test results.

4. MAIN LISTENING TEST

This section covers the creation of VR content and the test procedure methodologies used in the main listening test.

4.1. Virtual Reality Content Creation

Fig. 5 depicts a flow diagram for the creation of full motion spherical audiovisual content ready for playback on YouTube, either via a VR headset or on a standard computer monitor. Firstly Kolor Autopano is used to stitch together the six feeds of GoPro footage into a single equirectangular panoramic video [35]. FFMPEG [41], a free software project designed for handling multimedia data, is then used to add the FOA audio (with its channels in ACN, rather than Furse-Malham, order) to the panoramic footage [42]. In order for this file to then be uploadable to YouTube [43] the Spatial Media Metadata Injector [44] is used to indicate that the file contains a panoramic video. For the 'audio-only' stimuli a still image of equirectangular perspective lines was used as the visual component, in order to give the test participants some sense of orientation [45]. The resultant content can be viewed in the following two YouTube playlists: the audio-only playlist [46]; and the full audiovisual playlist [47].

4.2. Main Test Procedure

For the main listening test there were 20 participants, split into two groups of 10. Each group was exposed to the six chosen soundscape recordings: one group experienced the audio-only soundscapes first, and then experienced them with accompanying video footage; the other group of participants experienced the stimuli with the order reversed. Within each listening condition the presentation order was randomised. As with the audiovisual stage of the preliminary listening test no demographic data were collected here. In each viewing condition participants were encourage to pan and 'look around' the environment, with YouTube updating the binaural rendering of the FOA audio according to the visual perspective.

The soundscapes were presented as YouTube content embedded in Qualtrics. The presentation order within each set of stimuli was randomised. As with the preliminary test, each stimulus was rated in terms of valence and arousal, and in terms of the three established soundscape categories. Test participants were also asked to list the sound sources and visual elements in the scene.

4.3. Main Test Results

This section presents an evaluation and analysis of the results of the main listening test. As with the preliminary listening test, a Shapiro-Wilks test for normality was used. Similarly only a very small number of variables were shown to demonstrate a non-normal distribution. The main listening test results were therefore suitable to be compared using the Mann-Whitney U-test.

Initially the results for all test participants are all compared with no consideration of the order in which the two sets of stimuli were presented. Further analysis is then presented in order to investigate how the order in which test participants were exposed to the aural and audiovisual stimuli has affected their experience of the soundscape.

4.3.1. Overall Comparison

Fig. 6a shows the results from Mann-Whitney U-test applied to the main listening test results, comparing the results for the audio-only soundscape presentations with the audiovisual ones.

As this figure indicates, there are relatively few significant differences in any of the rating scales when comparing the two

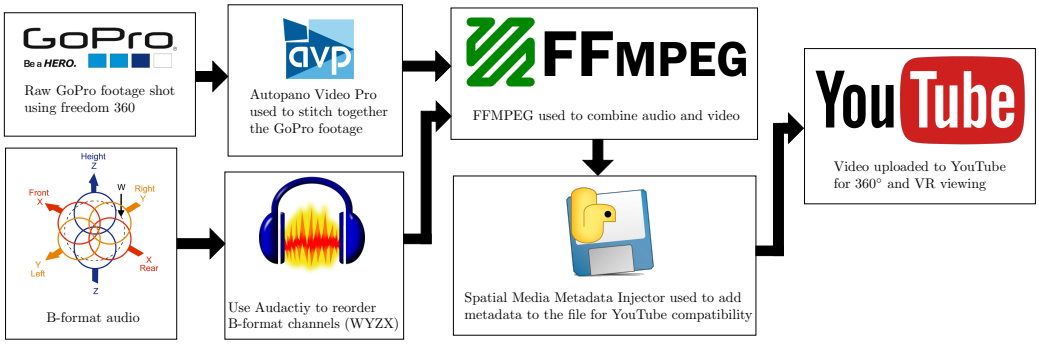


Figure 5: A flow diagram showing the method used in this study for VR content creation.

listening conditions. The clip that shows the most significant differences are for clip 7B, which was recorded next to a busy road in Leeds city centre. Compared to the audio only presentation of this soundscape clip, the ratings for the audiovisual presentation show significantly increased valence and human ratings, and a significantly reduced PNIR rating.

There are two aspects of the visual setting of this clip that have likely contributed to these differences: firstly, it is hard from listening to the soundscape alone to get a sense of how close to the road the listener is, as the traffic sounds are very loud, whilst the visual setting makes it clear that recording position is safely away from the road; secondly, the square that this recording was made at is lined with some trees which were clearly identified by test participants as a major visual feature of the scene.

The only other significant difference shown in Fig. 6a is for clip 3A, where the presence of visuals alongside the soundscape results in a significantly higher natural rating (as expected from the preliminary test results).

4.3.2. Order Dependence

Having now considered all of the results for both listening conditions for both groups of test participants, a breakdown of results by presentation order will now be considered.

Fig. 6b shows the results of applying the Mann-Whitney U-test to just the first listening condition experienced by each group: i.e. the audio-only results for the group that experienced those clips first compared with the audiovisual results from the other group.

Firstly it is interesting to note that the significant differences shown in this figure are not the same as those shown in Fig. 6a. These results show that for clip 1B, recorded at Dalby forest, the version of the clip presented with the accompanying visuals received a significantly greater valence rating, and a significantly lower mechanical rating. As with the preliminary test results, the change in valence rating is most likely due to the pleasantness of the trees and open sky in the visual setting. The mechanical rating is also lower with the presence of visuals for this clip. The soundscape contains some ambiguous noise that may be distant traffic, wind, or aircraft flying overhead. When presented with visual features this ambiguity is resolved and the natural visual elements take precedence.

A significant difference in mechanical rating can also be seen for clip 7B; this is most likely due to the human elements (people walking past) and minor elements of green infrastructure (some trees lining the square) that reduce the impact of the mechanical noise on the audiovisual experience of the soundscape.

Also shown in Fig. 6b are two significant differences in the ratings for clip 6A: the audiovisual presentation of this clip received significantly lower valence and human ratings than the audio-only version. This is most likely due to, again, elements of the visual environment that are not manifest in the soundscape itself: in this case the inner city shopping district buildings. In the audio-only presentation the dominant features are conversation and footsteps, whilst in the visual presentation the large buildings are the dominant feature. The presence of these buildings and paved streets also possibly gives some orientation for the background noise in the clip, grounding its otherwise ambiguous nature and indicating to participants that there is some distant traffic noise present.

Fig. 6c shows the Mann-Whitney U-test results comparing the two listening conditions for the group who experienced the audio-only soundscapes first, followed by audiovisual presentation. For clip 3A, recorded next to the Hole of Horcum in the North York Moors national park, there is a significant increase in the natural rating for the audiovisual presentation of the clip relative to the audio-only version due to the rolling countryside (something not obviously present in the soundscape itself).

The category ratings for all other soundscapes show no significant differences between listening conditions, but for clips 7B and 8A there are some differences in the emotion ratings. For clip 7B this means a significantly higher valence rating, and a significantly lower PNIR, once again showing how the presence of a relatively small amount of green infrastructure can improve the experience of a location.

Also of note in Fig. 6c is that for clip 8A, recorded at an inner city park in Leeds, there is indicate a significant decrease in the PNIR for the clip presented with visuals relative to the audio alone. This is interesting as neither the valence nor arousal ratings on their own show significant differences, but when these ratings are combined a significant difference can be demonstrated.

4.4. Discussion

When taken together the above results can be summarised as three main findings. Firstly, many of the significant differences in emotional or categorical ratings for the different soundscape clips are (perhaps unsurprisingly) due to the visual features that are not manifest in the soundscape clips. This makes clear the need for a cross-modal approach to soundscape evaluation as any real-life soundscape evaluation procedure will have to consider the visual context of that soundscape.

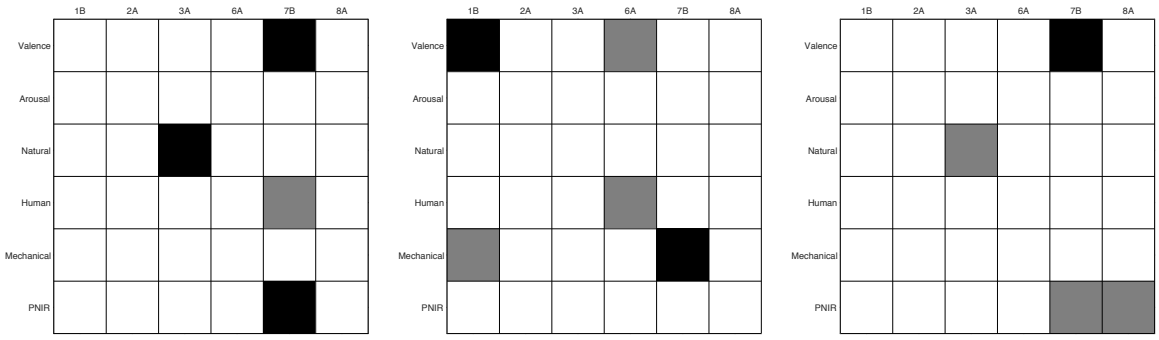


Figure 6: Mann-Whitney test results indicating significant differences between the two listening conditions. Plot (a) compares all of the results from both groups for each clip (b) compares the results for the first listening condition experienced by each group, and (c) compares only the results from the participants that experienced the soundscapes as audio-only first and then audiovisually. Dark marked squares indicate a difference at 95% confidence ($p < 0.05$), and light marked squares indicate a difference at 90% confidence ($p < 0.1$).

Secondly, for many of the differences in perception of the soundscape clips, the presence of elements of green infrastructure can be identified. This lends credence to the idea that green infrastructure, whilst not necessarily resulting in a significant change to an environment's acoustic properties, can improve the experience of that location.

Thirdly, the SAM, which has been examined thoroughly throughout this research in terms of its usefulness for soundscape evaluation, has been shown to be very useful in examining differences between the emotional states evoked by different soundscape. The PNIR, a combination of the valence and arousal dimensions of the SAM into a single perceptual rating, has also been shown to be useful in this study for discerning significant differences between emotional states evoked by soundscapes.

5. CONCLUSION

This paper has presented the results of two listening tests, each making use of soundscape recordings and images of the recording locations to investigate how a cross-modal approach to soundscape evaluation can be use to measure the impact of green infrastructure. The SAM and category ratings were used to conduct this evaluation: first in a preliminary test making use of stereo-UHJ renderings of the soundscape clips and still images; and then in a main listening test presenting the soundscapes in dynamically rendered binaural audio accompanied by full motion panoramic video footage.

Whilst the results presented in this paper show some significant differences in emotion and category rating between the audio only and audiovisual clip presentation, further work should be conducted comparing ratings for audiovisual soundscape presentation where the visual setting is altered, for example through the addition of trees or other aspects of green infrastructure. Such research would build on the results presented here, which validate the methodology in terms of the rating scales used, and the VR content creation and presentation methods.

6. REFERENCES

[1] R. Schafer, *The Soundscape: Our Sonic Environment and the Tuning of the World*. Inner Traditions/Bear, 1993. [Online]. Available: http://books.google.co.uk/books?id=ltBrAwAAQBAJ

[2] S. Payne, W. Davies, and M. Adams, "Research into the practical policy applications of soundscapes concepts and techniques in urban areas. DEFRA report NANR200, june 2009," 2009.

[3] E. Thompson, *The Soundscape of Modernity: Architectural Acoustics and the Culture of Listening in America, 1900-1933*. MIT Press, 2004. [Online]. Available: http://books.google.co.uk/books?id=7jvtvGbatv4C

[4] G. Keizer, *The Unwanted Sound of Everything We Want: A Book About Noise*. PublicAffairs, 2010. [Online]. Available: https://books.google.co.uk/books?id=yZ44DgAAQBAJ

[5] B. Truax, *Acoustic Communication*. Greenwood Publishing Group, 1984.

[6] J. Macdonald and H. McGurk, "Visual influences on speech perception processes," *Perception & Psychophysics*, vol. 24, no. 3, pp. 253–257, 1978. [Online]. Available: http://dx.doi.org/10.3758/BF03206096

[7] P. Lercher and B. Schulte-Fortkamp, "The relevance of soundscape research to the assessment of noise annoyance at the community level," in *Proceedings of the Eighth International Congress on Noise as a Public Health Problem*, 2003, pp. 225–231.

[8] S. Viollon, L. C., and C. Drake, "Influence of visual setting on sound ratings in an urban environment," *Applied Acoustics*, vol. 63, no. 5, pp. 493 – 511, 2002. [Online]. Available: http://www.sciencedirect.com/science/article/pii/\S0003682X01000536

[9] R. Ulrich, "View through a window may influence recovery," *Science*, vol. 224, no. 4647, pp. 224–225, 1984.

[10] K. Tzoulas, K. Korpela, S. Venn, V. Yli-Pelkonen, A. Kaźmierczak, J. Niemela, and P. James, "Promoting ecosystem and human health in urban areas using green infrastructure: A literature review," *Landscape and urban planning*, vol. 81, no. 3, pp. 167–178, 2007.

[11] D. T. Murphy, A. Southern, and F. Stevens, "Sounding our smart cities: Soundscape design, auralisation and evaluation for our urban environment," in *Sound + Environment 2017*, Hull, UK, 2017.

[12] L. Steg, A. van den Berg, and J. de Groot, *Environmental Psychology: An Introduction*, ser. BPS textbooks in psychology. Wiley, 2012. [Online]. Available: http://books.google.co.uk/books?id=RFHmw57kiNwC

[13] F. Stevens, D. T. Murphy, and S. L. Smith, "Emotion and soundscape preference rating: using semantic differential pairs and the self-assessment manikin," in *Sound and Music Computing conference, Hamburg, 2016*, Hamburg, Germany, 2016.

[14] C. Osgood, "The nature and measurement of meaning." *Psychological bulletin*, vol. 49, no. 3, p. 197, 1952.

[15] J. Kang and M. Zhang, "Semantic differential analysis of the soundscape in urban open public spaces," *Building and environment*, vol. 45, no. 1, pp. 150–157, 2010.

[16] W. Davies, N. Bruce, and J. Murphy, "Soundscape reproduction and synthesis," *Acta Acustica United with Acustica*, vol. 100, no. 2, pp. 285–292, 2014.

[17] S. Viollon and C. Lavandier, "Multidimensional assessment of the acoustic quality of urban environments," in *Conf. proceedings "Internoise", Nice, France, 27-30 Aug*, vol. 4, 2000, pp. 2279–2284.

[18] M. Bradley and P. Lang, "Measuring emotion: the self-assessment manikin and the semantic differential," *Journal of behavior therapy and experimental psychiatry*, vol. 25, no. 1, pp. 49–59, 1994.

[19] M. Bradley and P. J. Lang, *The International affective digitized sounds (IADS)[: stimuli, instruction manual and affective ratings*. NIMH Center for the Study of Emotion and Attention, 1999.

[20] M. Bradley, B. Cuthbert, and P. Lang, "Picture media and emotion: Effects of a sustained affective context," *Psychophysiology*, vol. 33, no. 6, pp. 662–670, 1996.

[21] A. Léobon, "La qualification des ambiances sonores urbaines," *Natures-Sciences-Sociétés*, vol. 3, no. 1, pp. 26–41, 1995.

[22] W. Yang and J. Kang, "Acoustic comfort and psychological adaptation as a guide for soundscape design in urban open public spaces," in *Proceedings of the 17th International Congress on Acoustics (ICA)*, 2001.

[23] L. Anderson, B. Mulligan, L. Goodman, and H. Regen, "Effects of sounds on preferences for outdoor settings," *Environment and Bevior*, vol. 15, no. 5, pp. 539–566, 1983.

[24] G. Watts and R. Pheasant, "Tranquillity in the scottish highlands and dartmoor national park–the importance of soundscapes and emotional factors," *Applied Acoustics*, vol. 89, pp. 297–305, 2015.

[25] F. Stevens, D. T. Murphy, and S. L. Smith, "Soundscape categorisation and the self-assessment manikin," in *Proceedings of the 20th International Conference on Digital Audio Effects (DAFx-17)*, Edinburgh, UK, 2017.

[26] E. Benjamin and T. Chen, "The native b-format microphone," in *Audio Engineering Society Convention 119*, 10 2005.

[27] F. Stevens, D. T. Murphy, and S. L. Smith, "Soundscape auralisation and perception for environmental sound modelling," in *Sound + Environment 2017*, Hull, UK, 2017.

[28] "Freedom 360 mount," 2015. [Online]. Available: http://freedom360.us/shop/freedom360/

[29] R. Elen, "Ambisonics: The surround alternative," in *Proceedings of the 3rd Annual Surround Conference and Technology Showcase*, 2001, pp. 1–4.

[30] F. Stevens, D. T. Murphy, and S. L. Smith, "Ecological validity of stereo uhj soundscape reproduction," in *In Proceedings of the 142nd Audio Engineering Society (AES) Convention*, Berlin, Germany, 2017.

[31] M. A. Gerzon, "Ambisonics in multichannel broadcasting and video," *Journal of the Audio Engineering Society*, vol. 33, no. 11, pp. 859–871, 1985.

[32] ISO, *Mechanical Vibration and Shock: Evaluation of Human Exposure to Whole-body Vibration. Part 1, General Requirements: International Standard ISO 2631-1: 1997 (E)*. ISO, 1997.

[33] J. Snow and M. Mann, "Qualtrics survey software: handbook for research professionals," 2013.

[34] "FSPViewer," 2017. [Online]. Available: http://www.fsoft.it/FSPViewer/

[35] "Kolor autopano," 2015. [Online]. Available: http://www.kolor.com/autopano-video/#start

[36] F. Stevens, "Dalby forest panoramic image," 2017. [Online]. Available: http://www.dermandar.com/p/bIrnfi

[37] S. Shapiro and M. Wilk, "An analysis of variance test for normality (complete samples)," *Biometrika*, pp. 591–611, 1965.

[38] H. Mann and D. Whitney, "On a test of whether one of two random variables is stochastically larger than the other," *The annals of mathematical statistics*, pp. 50–60, 1947.

[39] S. Harriet, "Application of auralisation and soundscape methodologies to environmental noise," Ph.D. dissertation, University of York, 2013.

[40] A. Southern, F. Stevens, and D. T. Murphy, "Sounding out smart cities: Auralization and soundscape monitoring for environmental sound design," *The Journal of the Acoustical Society of America*, vol. 141, no. 5, pp. 3880–3880, 2017.

[41] "FFMPEG." [Online]. Available: https://www.ffmpeg.org/

[42] B. Wiggins, "Youtube, ambisonics and vr," 2016. [Online]. Available: https://www.brucewiggins.co.uk/?p=666

[43] "Google support: Upload 360-degree videos." [Online]. Available: https://support.google.com/youtube/answer/6178631\?hl=en-GB

[44] "Spatial media metadata injector," 2016. [Online]. Available: https://github.com/google/spatial-media/releases

[45] D. Swart, "Equirectangular perspective lines," 2016. [Online]. Available: https://www.flickr.com/photos/dmswart/26363697850

[46] F. Stevens, "Final test audio stimuli YouTube playlist," 2017. [Online]. Available: https://www.youtube.com/playlist?list=PL-3kCuZ4n30QM5zUhzqfn9vkiwZPl2QzD

[47] ——, "Final test visual stimuli YouTube playlist," 2017. [Online]. Available: https://www.youtube.com/playlist?list=PL-3kCuZ4n30TIn40XSXdz5Y-sPr5brNet

REAL-TIME WAVE DIGITAL SIMULATION OF CASCADED VACUUM TUBE AMPLIFIERS USING MODIFIED BLOCKWISE METHOD

Jingjie Zhang, Julius O. Smith III

Center for Computer Research in Music and Acoustics (CCRMA), Stanford University
660 Lomita Drive, Stanford, CA 94305, USA
[jingjiez|jos]@ccrma.stanford.edu

ABSTRACT

Vacuum tube amplifiers, known for their acclaimed distortion characteristics, are still widely used in hi-fi audio devices. However, bulky, fragile and power-consuming vacuum tube devices have also motivated much research on digital emulation of vacuum tube amplifier behaviors. Recent studies on Wave Digital Filters (WDF) have made possible the modeling of multi-stage vacuum tube amplifiers within single WDF SPQR trees. Our research combines the latest progress on WDF with the modified blockwise method to reduce the overall computational complexity of modeling cascaded vacuum tube amplifiers by decomposing the whole circuit into several small stages containing only two adjacent triodes. Certain performance optimization methods are discussed and applied in the eventual real-time implementation.

1. INTRODUCTION

Having been displaced by semiconductor technologies in almost all areas of electronics, vacuum tube circuits are still widely used in hi-fi audio amplifiers and high-end guitar amplifiers due to the unique harmonic distortion characteristics produced by overdriven tubes that are preferred by human ears. On the other hand, driven by certain shortcomings of vacuum tube devices, such as large size and weight, poor durability and high power consumption, digital simulation of the behaviors of vacuum tube amplifiers, especially tube guitar amplifiers, has been an emerging research topic since the mid-1990s. In [1], Pakarinen and Yeh reviewed several digital techniques that emulate vacuum tube guitar amplifier behaviors.

Introduced by Fettweis, Wave Digital Filters (WDF) [2] are a class of digital filters that mimic classical filter structures, preferably lattice or ladder structures, by utilizing a wave-variable representation. Because of their superior numerical properties and stability under finite-arithmetic conditions, WDF have been successfully applied in digital modeling of lumped electronic or physical systems over the past few decades, as these systems can be typically represented by a set of blocks connected with each other through electrical or physical ports. It is thus reasonable that in recent years WDF have become widely applied in the field of nonlinear audio system modeling as a solid approach.

Digital modeling using classical WDF is able to handle series and parallel circuits containing a single-port delay-free nonlinearity. Reflection-free ports [3] are introduced to resolve the delay-free loop created by port connections, and a single-port nonlinearity [4] can be accommodated in a binary connection tree [5, 6]. However, the mathematical model of a vacuum tube triode is usually a dual-port or triple-port delay-free module. Moreover, circuits containing vacuum tube triodes usually cannot be simply decomposed into series and parallel topologies due to the feedback around the ports. Previous practices [7, 8, 9, 10] broke the multiple delay-free loops within WDF triode and JFET models by means of ad hoc unit delays, at the cost of accuracy and even stability.

Recently, Werner *et al.* [11, 12, 13, 14] extended the classical WDF adaptors to include the $\mathcal{R}$(Rigid)-type adaptor, a wave-domain scattering matrix [15] formed by a general approach based on Modified Nodal Analysis (MNA) [16] that resolves both arbitrary complex topologies [17] and multiple/multiport nonlinearities. The K-method [18] was used to resolve the multiple delay-free loops within these nonlinearities and thus make them tractable through tabulation. However, multidimensional tabulation often results in high memory-space consumption. Hence, while 1D nonlinearities are easily handled by linearly interpolated lookup tables [19], piecewise polynomial interpolation [20] or canonical piecewise-linear representation [4, 21], multidimensional iterative techniques [22] are typically used as an alternative to the tabulation approach when there are several interacting nonlinearities. Extending the previous binary connection tree containing three-port WDF adaptors, the $\mathcal{R}$-type adaptor has led to a new tree structure—an SPQR tree [17], which is able to absorb multiple nonlinearities into one single tree. Such a strategy was taken in [23] to simulate a multi-stage tube guitar amplifier, whereas it still remains to be improved for the sake of real-time capability, due to the dramatically increasing complexity of solving multidimensional nonlinear equations as the dimension increases. Although recent works on system identification and gray-box modeling techniques by Eichas *et al.* [24] are capable of modeling nonlinear guitar amplifiers with relatively low computational load, they are not based on the knowledge of the circuits and hence, cannot exactly model the behavior of the knobs in amplifiers.

In this paper, a certain type of vacuum tube amplifier circuit—cascaded vacuum tube amplifiers—are of concern, since a large proportion of vacuum tube amplifiers, especially vacuum tube guitar preamplifiers, appear to have a cascading structure. Previous studies done by Mačák [25, 26, 27, 28] introduced a modified blockwise method, which decomposes cascaded vacuum tube amplifiers properly into separate small stages to keep low the dimension of the local nonlinear system to be numerically solved each time and therefore, reduce the overall computational complexity of the whole simulation without affecting the mutual interactions between adjacent amplifier circuits. Similar strategy was also devised in [29] where two stages of the TR-808 bass drum were carefully separated. As a demonstration of combining WDF modeling techniques with the modified blockwise method, a case study of a vacuum tube guitar preamplifier in cascading structure is presented. Performance optimization methods that contribute to the real-time behavior of the eventual implementation are discussed, as well as simulation results.

This work was supported by Stanford Art Institute 2017-18 Fellowship

The remainder of this paper is structured as follows: Section 2 reviews the previous research on resolving multiple/multiport WDF nonlinearities within a single SPQR tree. Section 3 illuminates the details of the modified blockwise method and its significance to the modeling of cascaded vacuum tube amplifiers. The case study is given in Section 4. Section 5 summarizes the results and discusses future research directions.

2. PREVIOUS WORK

In this Section, recent developments in WDF modeling of nonlinear circuits are reviewed. In particular, two approaches that resolve the multiple delay-free loops within multiple/multiport nonlinearities within $\mathcal{R}$-type adaptors are discussed.

Previous studies [14, 17] have developed a general approach that can decompose any given circuit into **S**eries, **P**arallel, and **R**igidly connected WDF elements, and thus form a WDF SPQR tree, which uses an $\mathcal{R}$-type adaptor to absorb any complex (neither series nor parallel) topologies. All nonlinearities are placed at the "roots" of the SPQR tree, while the remaining subtrees containing series, parallel, or even other R-type connected linear elements can be modeled using conventional WDF theory. Thévenin port equivalents and Modified Nodal Analysis (MNA) [16] are utilized to compute the wave scattering matrix $\mathbf{S}$ for each $\mathcal{R}$-type adaptor:

$$\begin{bmatrix}\mathbf{b}_I\\ \mathbf{b}_E\end{bmatrix} = \mathbf{S}\begin{bmatrix}\mathbf{a}_I\\ \mathbf{a}_E\end{bmatrix} = \begin{bmatrix}\mathbf{S}_{11} & \mathbf{S}_{12}\\ \mathbf{S}_{21} & \mathbf{S}_{22}\end{bmatrix}\begin{bmatrix}\mathbf{a}_I\\ \mathbf{a}_E\end{bmatrix}, \tag{1}$$

where $\mathbf{a}_I$ and $\mathbf{b}_I$ represent the vectors of internal incident and reflected waves from the nonlinearities, while $\mathbf{a}_E$ and $\mathbf{b}_E$ represent the external incident and reflected waves from the subtrees.

Whereas the wave domain nonlinear relationship between $\mathbf{a}_I$ and $\mathbf{b}_I$ can be represented by $\mathbf{a}_I = F_w(\mathbf{b}_I)$, it is much easier to obtain the Kirchhoff domain nonlinear relationship:

$$\mathbf{i}_C = F_k(\mathbf{v}_C), \tag{2}$$

since the behaviors of most nonlinear electronic devices are usually defined in the Kirchhoff domain, while only some specific nonlinearities can be modeled in wave domain using the Lambert $\mathcal{W}$ function [30, 31, 32].

Therefore, the internal wave vectors $\mathbf{a}_I$ and $\mathbf{b}_I$ are converted to the corresponding Kirchhoff vectors $\mathbf{i}_C$ and $\mathbf{v}_C$ using a w–K converter matrix $\mathbf{C}$:

$$\begin{bmatrix}\mathbf{v}_C\\ \mathbf{a}_I\end{bmatrix} = \begin{bmatrix}\mathbf{C}_{11} & \mathbf{C}_{12}\\ \mathbf{C}_{21} & \mathbf{C}_{22}\end{bmatrix}\begin{bmatrix}\mathbf{i}_C\\ \mathbf{b}_I\end{bmatrix} = \begin{bmatrix}-\mathbf{R}_I & \mathbf{I}\\ -2\mathbf{R}_I & \mathbf{I}\end{bmatrix}\begin{bmatrix}\mathbf{i}_C\\ \mathbf{b}_I\end{bmatrix}, \tag{3}$$

where $\mathbf{R}_I$ is a diagonal matrix of internal port resistances. Combining (1) and (3) yields a new scattering relationship:

$$\begin{bmatrix}\mathbf{v}_C\\ \mathbf{b}_E\end{bmatrix} = \begin{bmatrix}\mathbf{F} & \mathbf{E}\\ \mathbf{N} & \mathbf{M}\end{bmatrix}\begin{bmatrix}\mathbf{i}_C\\ \mathbf{a}_E\end{bmatrix}, \tag{4}$$

where

$$\begin{cases}\mathbf{E} = \mathbf{C}_{12}(\mathbf{I} + \mathbf{S}_{11}\mathbf{H}\mathbf{C}_{22})\mathbf{S}_{12}\\ \mathbf{F} = \mathbf{C}_{12}\mathbf{S}_{11}\mathbf{H}\mathbf{C}_{21} + \mathbf{C}_{11}\\ \mathbf{M} = \mathbf{S}_{21}\mathbf{H}\mathbf{C}_{22}\mathbf{S}_{12} + \mathbf{S}_{22}\\ \mathbf{N} = \mathbf{S}_{21}\mathbf{H}\mathbf{C}_{21},\end{cases} \tag{5}$$

with $\mathbf{H} = (\mathbf{I} - \mathbf{C}_{22}\mathbf{S}_{11})^{-1}$.

Plugging (2) into (4) yields the delay-free loops within the $\mathcal{R}$-type adaptor:

$$\mathbf{v}_C = \mathbf{E}\mathbf{a}_E + \mathbf{F}F_k(\mathbf{v}_C). \tag{6}$$

The delay-free loops in (6) can be resolved using either K-method or iterative techniques. In terms of high speed data access and memory consumption, using multidimensional tables transformed by K-method [12, 18] in real-time simulation becomes more expensive as the dimension increases. In addition, the neighbor searching and scattered interpolation of multidimensional table data further aggravates the computational load. As a more general approach, multidimensional iterative techniques solve instantaneous loops by finding numerical solutions to the given nonlinear systems, and hence are applied in [22] to offer an alternative to K-method. To solve for $\mathbf{v}_C$ in (6), the following multidimensional nonlinear equation can be constructed:

$$H(\mathbf{v}_C) = \mathbf{E}\mathbf{a}_E + \mathbf{F}F_k(\mathbf{v}_C) - \mathbf{v}_C = 0. \tag{7}$$

Several iterative approaches are available to obtain the numerical solution to this equation. The simplest and typically most effective way is multidimensional Newton's method. For the multidimensional function $H(\mathbf{v}_C)$, given an initial guess $\mathbf{v}_C^0$ in a sufficiently close neighborhood of one of its zeros, a numerical approximation of the solution can be obtained iteratively by

$$\mathbf{v}_C^{k+1} = \mathbf{v}_C^k - J_H(\mathbf{v}_C^k)^{-1}H(\mathbf{v}_C^k), \tag{8}$$

where J_H is the Jacobian matrix of H. The choice of the initial guess $\mathbf{v}_C^0$ is detailed in [22]. Although several advanced iterative algorithms based on Newton's method can be devised to achieve a higher convergence rate, the overall computational complexity within each iteration expands dramatically as the dimension of the inverse Jacobian matrix J_H^{-1} increases, especially in [23] where four WDF triode models were involved in one single SPQR tree.

3. MODIFIED BLOCKWISE METHOD

As discussed in the previous section, performance degradation in high-dimensional cases is dramatic in the multi-nonlinearity WDF systems resolved by either K-method or iterative methods, while such circumstances are inevitable when cascaded vacuum tube amplifiers are modeled as a whole. On the other hand, as a common approach to deal with complex cascaded systems, simply decomposing cascaded tube amplifiers into minimal separate stages (*i.e.*, one tube per stage) is also not applicable due to the strong mutual interactions that comes from the loading effect between two adjacent tube amplifiers, although it minimizes the dimension of the local nonlinear equations to be solved each time.

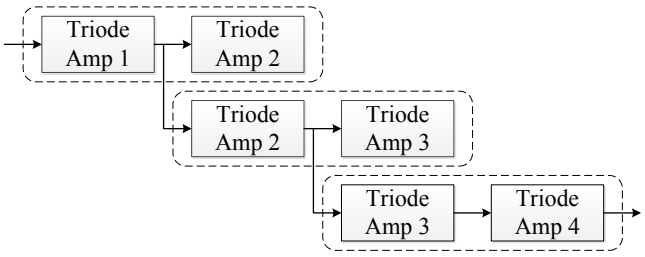


Figure 1: *Decomposing cascaded vacuum tube amplifiers using the modified blockwise method.*

In [28], the loading effects between three cascaded typical common-cathode triode amplifiers have been measured and compared. It has been proved that there is very small interaction between the first and the third amplifier and hence, it is sufficient to consider only the second amplifier as the nonlinear load for the first one, which lays the foundation of the modified blockwise method. Applied in several previous practices [25, 26, 27] on equation-based simulation of cascaded vacuum tube amplifiers, the modified blockwise method decomposes the cascaded amplifiers into several coupled triode amplifier stages that are modeled separately as illustrated in Fig. 1. It is noteworthy that the extra computational load introduced by the redundant triode amplifiers involved in the simulation is far outweighed by the reduced overall computational complexity of the whole system.

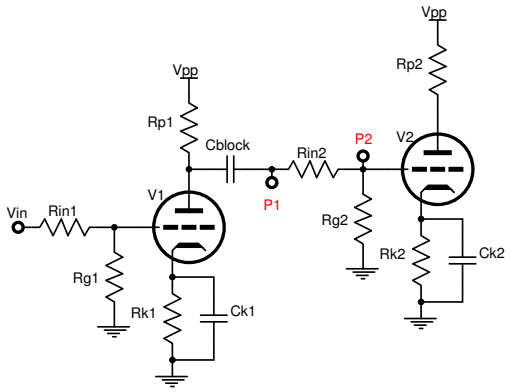


Figure 2: *Extracting the proper output signal of the first amplifier in a coupled common-cathode triode amplifier stage.*

The modified blockwise structure ensures that in each stage, the nonlinear current flowing into the grid of the second triode is taken into account and therefore, the output signal of the first triode amplifier is correct and ready to be fed into the next stage. Fig. 2 points out the circuit node P_1 where the output signal of the first amplifier is usually extracted in a coupled common-cathode triode amplifier stage. On the other hand, extracting the signal at P_2 is usually not applicable, although it cuts down the number of redundant components in the next stage. Such a conclusion is drawn on the basis of the triodes' grid limiting behavior [33] illustrated in Fig. 3. As the input voltage V_{in} to the triode amplifier is made larger, the grid current I_g increases, causing an increased voltage drop across the grid resistor R_{in}. This tends to make the grid voltage V_g increase much less than the input. As a result, the grid resistor R_{in} is of great significance and hence cannot be simply separated from the simulation unless the triode's operating point is not located in the grid limiting region under any circumstances.

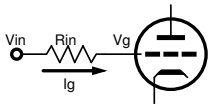


Figure 3: *Grid limiting behavior of a vacuum tube triode.*

4. CASE STUDY

As an example of combining multi-nonlinearity WDF modeling techniques with the modified blockwise method, we study the preamplifier stage of the MESA/Boogie® Mark II-B™ guitar amplifier, which consists of five cascaded vacuum tube triode amplifiers. The prototype of this circuit was patented by Smith [34] in 1980 as the world's first high gain dual mode channel switching amplifier.

4.1. System Decomposition

As shown in Fig. 5a and 5b, both "Clean" and "Lead" branches are cascaded vacuum tube triode amplifiers that can be decomposed into small coupled triode amplifier stages using the modified blockwise method. The circuit nodes where signals should be extracted are marked in these schematics. In Fig. 5a, the output signal of *Stage* 3 is extracted at P_2 rather than the node between capacitor C_8 and resistor R_4 simply because it is much easier to get the plate voltage of triode V1B in the corresponding WDF SPQR tree as presented in Fig. 6; whereas extracting signal elsewhere requires extra subtraction. It is also worth mentioning that the circuit node P_3 in both Fig. 5a and 5b is carefully chosen so that the "Lead" and "Clean" branches can share the same coupled triode output stage without extra computational load. Such a strategy is based on careful measurements of the input and output signals of *Stage* 5, a cathode follower which has no gain but a constant grid-to-cathode voltage.

On the basis of the marked circuit nodes in Fig. 5, the reference circuit is decomposed into five coupled triode amplifier stages organized in the modified blockwise structure and then modeled using WDF techniques. Fig. 6 shows the resulting SPQR trees of each stage and a diagram of the system structure at the top right corner that illustrates the relationship between the dual triode stages in the modified blockwise structure and the original cascaded circuit stages, where the original stages containing triodes are marked with a darker color. All the linear elements in the circuits are modeled using voltage wave variables. The active sections of a potentiometer are treated separately and identified by suffix numbers (*e.g.*, the potentiometer "Volume" in *Stage* 3 is separated into "Vol1" and "Vol2").

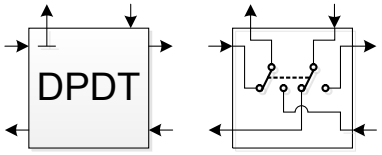
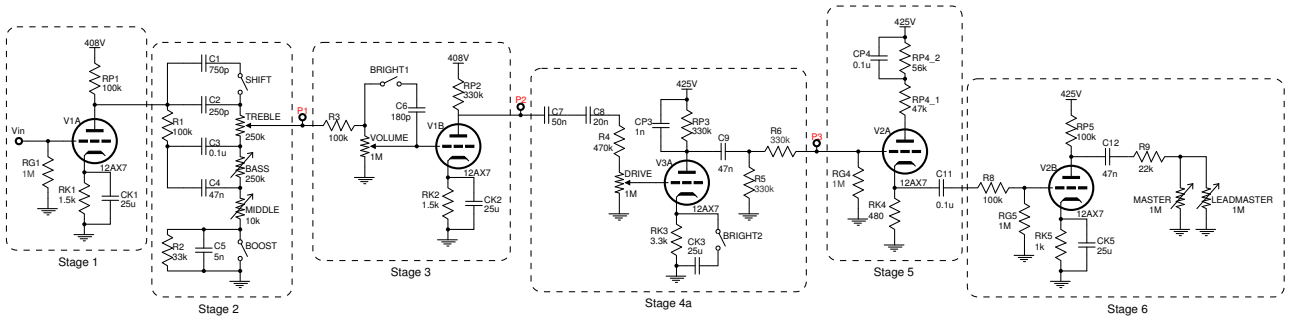
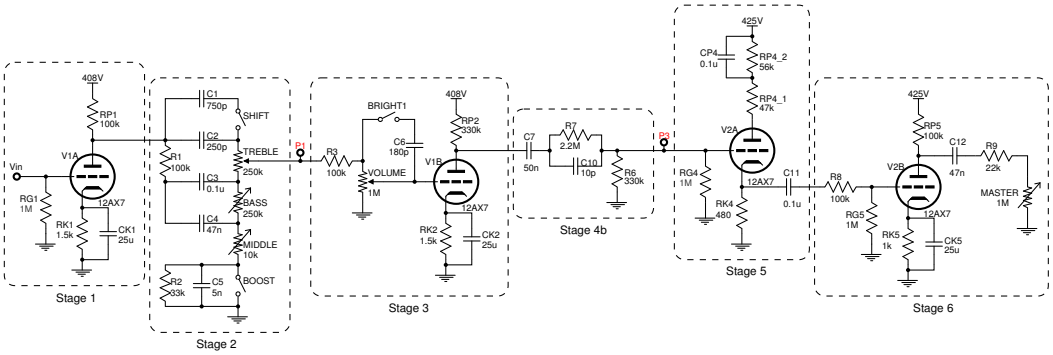


Figure 4: *Internal structure of a wave-domain double-point, double-throw (DPDT) switch.*

In previous WDF modeling practices [29, 35], the single-pole, single-throw (SPST) switch are usually modeled as non-adaptable elements at the root of the trees. In this work, a wave-domain double-point, double-throw (DPDT) switch is devised to model the equivalent behavior of some circuits containing SPST switches. The common port of a DPDT switch can only be connected with one of the two sub-ports, as illustrated in Fig. 4. The two different states of an SPST switch in a circuit will result in two different local topologies. If the differences are within one subtree of an $\mathcal{R}$-type adaptor, then this wave-domain DPDT switch can be utilized to adapt the two local subtrees derived from the two topolo-



(a) *"Lead" mode*



(b) *"Clean" mode*

Figure 5: *Schematic of MESA/Boogie® Mark II-B™ guitar preamplifier in both "Lead" and "Clean" modes.*

gies. In the process, some elements will inevitably appear twice in the whole SPQR tree, therefore, the two identical elements are distinguished from each other by an extra apostrophe in the labels (*e.g.*, the potentiometer "Middle" in *Stage* 2 has two identical WDF models "M" and "M′" in two local subtrees).

4.2. Resolving Coupled Triodes

Although the modified blockwise decomposition of the reference circuit reduces the dimension of the local nonlinear system from 8D/10D to 4D (coupled triode amplifiers), the size of a K-method-transformed multidimensional lookup table is still too large for real-time simulations. Thus, in this case study, multidimensional Newton's method is used to resolve the dual triodes within one SPQR tree, which involves utilizing the Jacobian matrix J as a direction to find the root of the nonlinear equations (7) formed by the mathematical models of these triodes. However, previous triode models [36, 37, 38, 39] are all piece-wise nonlinear functions that result in poor performance near the points of discontinuity.

Preferred by recent research [23, 28, 40, 41], the physically-motivated Dempwolf triode model [42] smooths the discontinuity by combinations of exponential and logarithmic functions:

$$\begin{cases} I_{gk} = f(V_{gk}) = G_g \cdot (\frac{1}{C_g}\log(1+e^{C_g \cdot V_{gk}}))^{\xi} + I_{g0} \\ I_k = g(V_{gk}, V_{pk}) = G \cdot (\frac{1}{C}\log(1+e^{C\cdot(\frac{V_{pk}}{\mu}+V_{gk})}))^{\gamma} \\ I_{pk} = I_k - I_{gk} = g(V_{gk}, V_{pk}) - f(V_{gk}), \end{cases} \quad (9)$$

Table 1: *Dempwolf triode model parameters of a 12AX7 tube.*

G_g	C_g	ξ	I_{g0}
6.177E-4	9.901	1.314	8.025E-8
G	C	γ	μ
2.242E-3	3.4	1.26	103.2

with grid-to-cathode voltage and current V_{gk}, I_{gk}, plate-to-cathode voltage and current V_{pk}, I_{pk}, cathode current I_k, and constant model parameters such as perveances G_g, G, adaption factors C_g, C and exponents ξ, γ. For a typical **12AX7** tube, the model parameters are given in Table 1.

For dual triode amplifier stages, plugging (9) into (2) yields

$$\mathbf{i}_C = \begin{bmatrix} I_{gk1} \\ I_{pk1} \\ I_{gk2} \\ I_{pk2} \end{bmatrix} = F_k(\mathbf{v}_C) = F_k(\begin{bmatrix} V_{gk1} \\ V_{pk1} \\ V_{gk2} \\ V_{pk2} \end{bmatrix}) = \begin{bmatrix} f_1 \\ g_1 - f_1 \\ f_2 \\ g_2 - f_2 \end{bmatrix}, \quad (10)$$

where f_i denotes $f(V_{gki})$ and g_i denotes $g(V_{gki}, V_{pki}) - f(V_{gki})$.

Hence, the multidimensional nonlinear equation (7) of each dual triode amplifier stage can be expressed as

$$H(\begin{bmatrix} V_{gk1} \\ V_{pk1} \\ V_{gk2} \\ V_{pk2} \end{bmatrix}) = \mathbf{E}\mathbf{a}_E + \mathbf{F}\begin{bmatrix} f_1 \\ g_1 - f_1 \\ f_2 \\ g_2 - f_2 \end{bmatrix} - \begin{bmatrix} V_{gk1} \\ V_{pk1} \\ V_{gk2} \\ V_{pk2} \end{bmatrix} = 0. \quad (11)$$

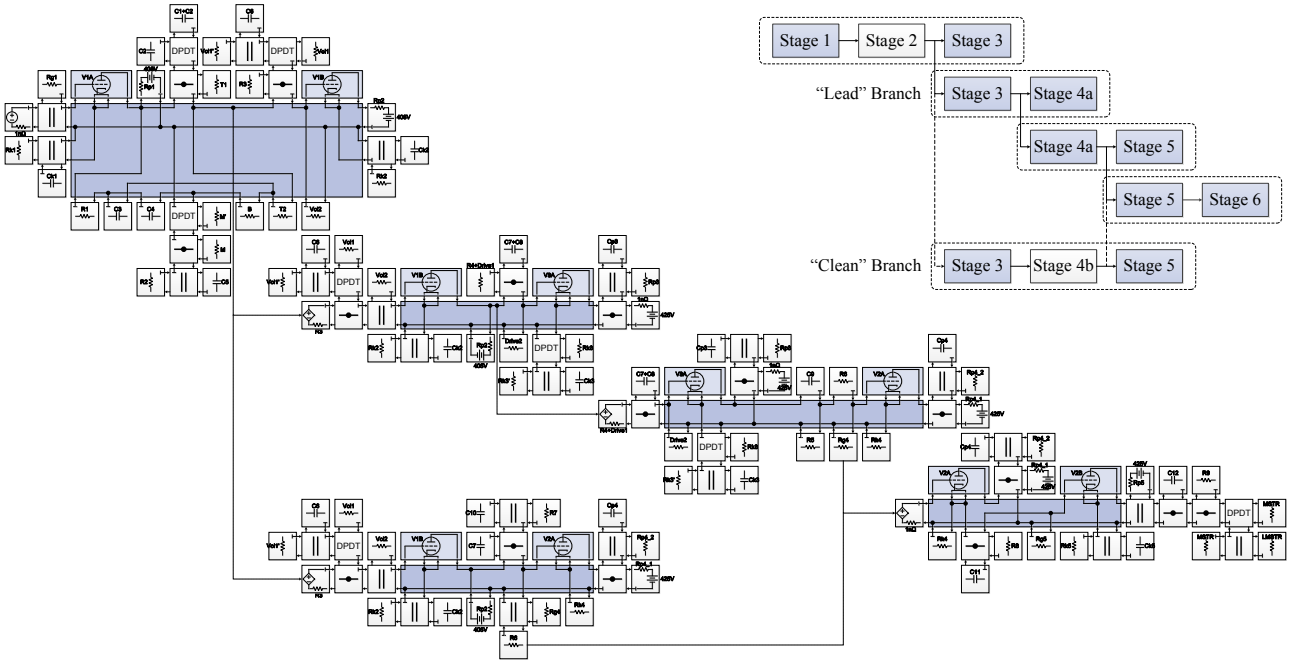


Figure 6: *Modified blockwise WDF model of MESA/Boogie® Mark II-B™ guitar preamplifier.*

The corresponding iteration expression (8) thus becomes

$$\begin{bmatrix} V_{gk1} \\ V_{pk1} \\ V_{gk2} \\ V_{pk2} \end{bmatrix}^{k+1} = \begin{bmatrix} V_{gk1} \\ V_{pk1} \\ V_{gk2} \\ V_{pk2} \end{bmatrix}^{k} - J_H(\begin{bmatrix} V_{gk1} \\ V_{pk1} \\ V_{gk2} \\ V_{pk2} \end{bmatrix}^{k})^{-1} H(\begin{bmatrix} V_{gk1} \\ V_{pk1} \\ V_{gk2} \\ V_{pk2} \end{bmatrix}^{k}), \quad (12)$$

where the four-dimensional Jacobian matrix J_H is given by

$$J_H = \mathbf{F} J_{F_k} - \mathbf{I}, \quad (13)$$

with

$$J_{F_k} = \begin{bmatrix} \frac{\partial f_1}{\partial V_{gk1}} & 0 & 0 & 0 \\ \frac{\partial g_1}{\partial V_{gk1}} - \frac{\partial f_1}{\partial V_{gk1}} & \frac{\partial g_1}{\partial V_{pk1}} & 0 & 0 \\ 0 & 0 & \frac{\partial f_2}{\partial V_{gk2}} & 0 \\ 0 & 0 & \frac{\partial g_2}{\partial V_{gk2}} - \frac{\partial f_2}{\partial V_{gk2}} & \frac{\partial g_2}{\partial V_{pk2}} \end{bmatrix}. \quad (14)$$

4.3. Performance Optimization Methods

Various methods can be used to optimize the performance of the WDF simulation system to make it run in real time. A widely applied one is to tabulate the nonlinearity with a proper uniform interval and introduce linear interpolation into the table lookup process [19]. In this case study, the two nonlinear functions $I_{gk} = f(V_{gk})$ and $I_k = g(V_{gk}, V_{pk})$ in the Dempwolf triode model (9) are tabulated into two one-dimensional lookup tables corresponding to a pair of indices V_1, V_2 transformed through

$$\begin{bmatrix} V_1 \\ V_2 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 1 & \frac{1}{\mu} \end{bmatrix} \begin{bmatrix} V_{gk} \\ V_{pk} \end{bmatrix}. \quad (15)$$

Combining (9) and (14) also yields

$$\frac{\partial g(V_{gk} + \frac{V_{pk}}{\mu})}{\partial V_{pk}} = \frac{1}{\mu} \frac{\partial g(V_{gk} + \frac{V_{pk}}{\mu})}{\partial V_{gk}}. \quad (16)$$

Thus, all the elements in the Jacobian matrix J_{F_k} (14) can also be covered by two 1D nonlinear tables $\frac{\partial f}{\partial V_{gk}}$ and $\frac{\partial g}{\partial V_{gk}}$ corresponding to the same pair of indices V_1, V_2 given in (15).

Linear interpolation is applied when utilizing the four 1D tables $f(V_1)$, $\frac{\partial f}{\partial V_{gk}}(V_1)$, $g(V_2)$, $\frac{\partial g}{\partial V_{gk}}(V_2)$. For a 1D table $y[n]$, given an accurate value x between two adjacent integer indices n and $n+1$, the linear interpolation result $y[x]$ is defined by

$$y[x] = y[n] + (x - n)(y[n+1] - y[n]), \quad (17)$$

which can be further optimized by introducing a different table $\Delta y[n] = y[n+1] - y[n]$ that replaces the extra subtraction:

$$y[x] = y[n] + (x - n)\Delta y[n] \quad (18)$$

Another approach to speed up the simulation process is developed from the perspective of matrix-operation performance tuning. In this study, the open source C++ linear algebra library Armadillo [43] is used to cover the basic matrix operations such as addition, multiplication and inversion. However, when solving the 4D local nonlinear system (11) within each coupled triode amplifier stage, the most time-consuming process in each iteration (12) is the inversion of the 4D matrix J_H, although the inversions of small matrices up to 4D are carried out explicitly inside Armadillo. This is due to the large overheads introduced by Armadillo wrappers to take different measures according to different matrix dimensions. The same situation occurs when Armadillo is calling the general matrix-vector multiplication (GEMV) in Basic Linear

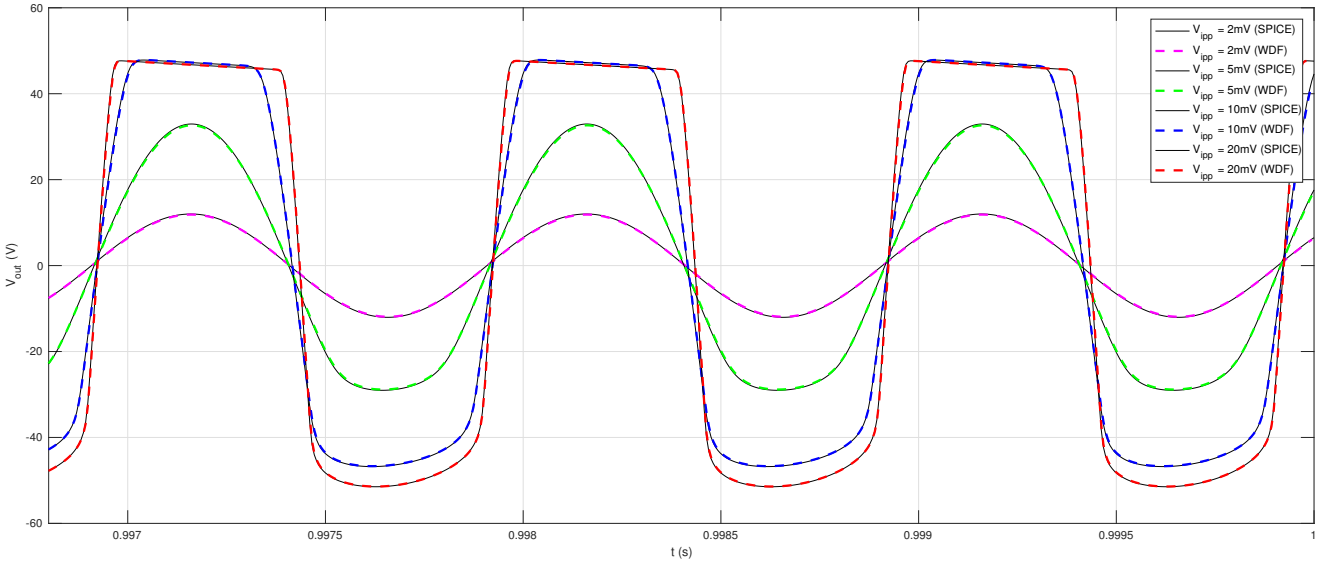


Figure 7: *Comparison of WDF and SPICE's time domain responses to different input signal levels.*

Algebra Subprograms (BLAS), in which case more overheads is introduced since BLAS is row-oriented so that an extra transposition is required when called by the column-oriented Armadillo. Therefore, explicit 4D matrix inversion and 4D matrix-vector multiplication are implemented to avoid extra overheads.

Finally, to further increase the simulation speed, a certain level of accuracy can be carefully sacrificed by increasing the error tolerance threshold TOL of the iterative root-finding process, which serves as a termination criterion for iteration:

$$||H(\mathbf{v}_C^k)|| \leq TOL \tag{19}$$

4.4. Simulation Results

The modified blockwise WDF system was implemented in C++ language and tested on a MacBook Pro with 2.3 GHz Intel Core i5 and 8GB RAM at 4x oversampling of a typical audio sampling rate of 44.1 kHz (176.4 kHz). After several successful initial offline behavioral tests, the system was tested in real-time with a buffer size of 256 samples and a 15.1% maximal CPU load.

To test the system's time domain response, 1kHz sinusoids with a small peak-to-peak voltage of 2mV, 5mV, 10mV and 20mV were used as input signal to observe in particular the transition from linear amplification to soft clipping in "Lead" mode. As presented in Fig. 7, the output waveforms of the WDF system show excellent agreement with LTspice simulation results of the same circuit. As the input amplitude increases gradually, the grid limiting first starts to appear in the negative cycle of the output waveform, and when the input level is even higher, the positive cycle is cut off. The error of a fully clipped output signal corresponding to a 1kHz, 250mV peak-to-peak input sinusoid is shown in Fig. 8. Most errors occur during zero-crossing with a maximum of 4.5V, which might be introduced by the alignment deviation after resampling SPICE results onto the time grid of the WDF simulation.

Fig. 9 shows the comparison of WDF and SPICE's frequency responses to a 1kHz, 250mV peak-to-peak input sinusoid. The SPICE results have been offset by 50Hz to create clarity. The first twenty harmonic peak frequencies of the WDF simulation

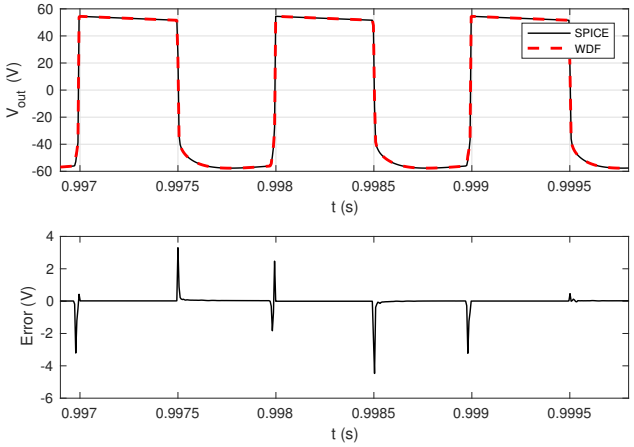


Figure 8: *Comparison of WDF and SPICE's time domain responses (top) and error (bottom) for a* 1*kHz,* 250*mV peak-to-peak input sinusoid.*

agree well with the SPICE results. To verify the system's behavior across the audible range, exponential sine sweeps [44] between 20Hz and 20kHz is used as input signal, the response spectrograms of "Clean" and "Lead" mode are presented in Fig. 10 and 11 respectively. It is confirmed that the harder clipping in "Lead" mode results in higher harmonics in the corresponding output signal.

Preserving a reasonably high accuracy of simulation, the modified blockwise WDF system shows superior performance advantages when compared with SPICE and single WDF SPQR tree system. For the "Lead" mode circuit containing five cascaded triode amplifiers, given a 1-second input sinusoid, the modified blockwise WDF system only spends around 370ms to finish the whole simulation, while a single WDF SPQR tree model of the same circuit requires more than 20s, and the corresponding simulation time of SPICE even exceeds 80s.

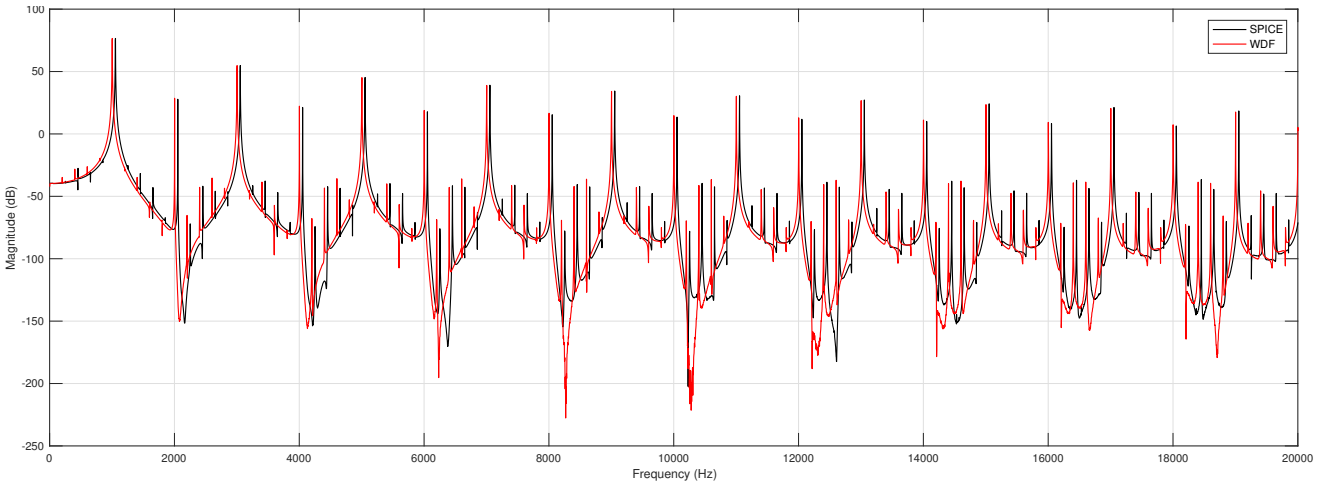


Figure 9: *Comparison of WDF and SPICE's frequency responses, a* 50*Hz offset is applied to SPICE result for intelligibility.*

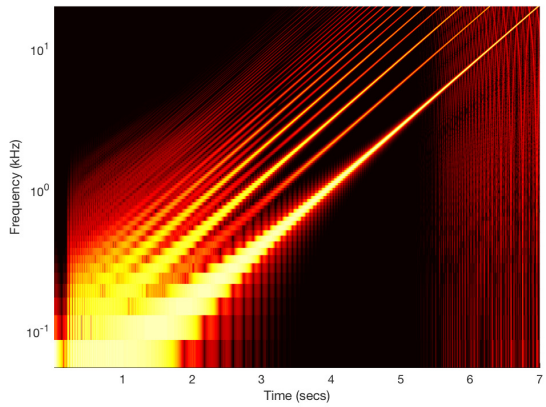


Figure 10: *Exponential sine sweep response spectrogram for* 20-20*kHz* 250*mV peak-to-peak input signals ("Clean" mode).*

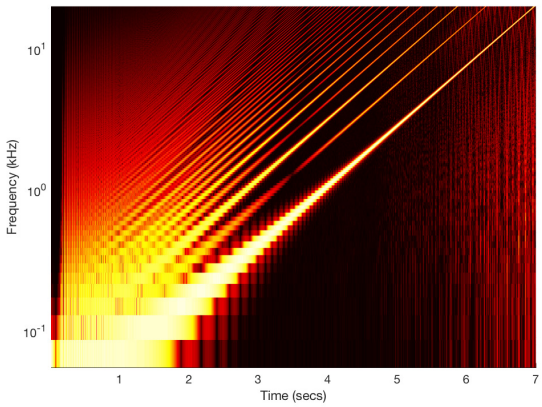


Figure 11: *Exponential sine sweep response spectrogram for* 20-20*kHz* 250*mV peak-to-peak input signals ("Lead" mode).*

5. CONCLUSION AND FUTURE WORK

In this paper, the modified blockwise method is applied to the WDF modeling of cascaded vacuum tube amplifiers to reduce the overall computational complexity of solving high-dimensional nonlinear systems. The cascaded tube amplifier was decomposed into several small stages containing two adjacent triodes. With the help of several performance optimization methods, such as lookup tables with linear interpolation and explicitly implemented matrix operations, the resulting modified blockwise WDF simulation system preserves a reasonably high precision in both time and frequency domains while exhibiting extremely high simulation speed on a standard laptop and hence, is capable of running in real-time.

As stated in the previous sections, the K-method transformed nonlinear multidimensional lookup tables are currently not competitive due to their high memory consumption and slow data-access speed. However, given enough memory, a table-lookup will be much faster than the iteration process. Sparse memory techniques [45] can be pursued, but they must ultimately be less expensive than Newton iterations. We therefore believe that future research could focus on 4D nonuniform tabulation, and high-speed 4D nearest neighbor searching algorithms. Unlike most vacuum tube preamplifiers in cascading structures, most vacuum tube power amplifiers are push-pull tube circuits that cannot be simulated by simply applying the strategies mentioned in this paper. Hence, further research can be done in this direction as well.

6. ACKNOWLEDGMENTS

Thanks to Brad Nelson and Steven R. Brill from the Stanford Computational Consulting (C^2) Service for helpful discussions on optimizing matrix-operation performance.

7. REFERENCES

[1] J. Pakarinen and D. T. Yeh, "A review of digital techniques for modeling vacuum-tube guitar amplifiers," *Comput. Music J.*, vol. 33, no. 2, pp. 85–100, 2009.

[2] A. Fettweis, "Wave digital filters: Theory and practice," *Proc. IEEE*, vol. 74, no. 2, pp. 270–327, Feb 1986.

[3] A. Fettweis and K. Meerkötter, "On adaptors for wave digital filters," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. 23, no. 6, pp. 516–525, Dec 1975.

[4] K. Meerkötter and R. Scholtz, "Digital simulation of nonlinear circuits by wave digital filter principles," in *Proc. IEEE Intl. Symp. Circ. & Sys.*, Portland, OR, May 8–11, 1989, vol. 1, pp. 720–723.

[5] A. Sarti and G. De Sanctis, "Systematic methods for the implementation of nonlinear wave-digital structures," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 56, no. 2, pp. 460–472, Feb 2009.

[6] G. De Sanctis and A. Sarti, "Virtual analog modeling in the wave-digital domain," *IEEE Trans. Audio, Speech, Language Process.*, vol. 18, no. 4, pp. 715–727, May 2010.

[7] M. Karjalainen and J. Pakarinen, "Wave digital simulation of a vacuum-tube amplifier," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, Toulouse, France, May 15–19, 2006.

[8] J. Pakarinen, M. Tik, and M. Karjalainen, "Wave digital modeling of the output chain of a vacuum-tube amplifier," in *Proc. 12th Int. Conf. on Digital Audio Effects (DAFx-09)*, Como, Italy, Sep 1–4 2009.

[9] J. Pakarinen and M. Karjalainen, "Enhanced wave digital triode model for real-time tube amplifier emulation," *IEEE Trans. Audio, Speech, Language Process.*, vol. 18, no. 4, pp. 738–746, May 2010.

[10] D. Hernandez and J. Huang, "Emulation of junction field-effect transistors for real-time audio applications," *IEICE Electron. Express*, vol. 13, no. 12, pp. 1–11, 2016.

[11] K. J. Werner, J. O. Smith III, and J. S. Abel, "Wave digital filter adaptors for arbitrary topologies and multiport linear elements," in *Proc. 18th Int. Conf. on Digital Audio Effects (DAFx-15)*, Trondheim, Norway, Nov 30 – Dec 3, 2015.

[12] K. J. Werner, J. O. Smith III, and J. S. Abel, "Resolving wave digital filters with multiple/multiport nonlinearities," in *Proc. 18th Int. Conf. on Digital Audio Effects (DAFx-15)*, Trondheim, Norway, Nov 30 – Dec 3, 2015.

[13] K. J. Werner, V. Nangia, J. O. Smith III, and J. S. Abel, "A general and explicit formulation for wave digital filters with multiple/multiport nonlinearities and complicated topologies," in *Proc. IEEE Workshop Appl. Signal Process. Audio Acoust. (WASPAA)*, New Paltz, NY, Oct 18–21, 2015.

[14] K. J. Werner, A. Bernardini, J. O. Smith III, and A. Sarti, "Modeling circuits with arbitrary topologies and active linear multiports using wave digital filters," *IEEE Trans. Circuits Syst. I, Reg. Papers*, Jun 2018, In Press, DOI: https://doi.org/10.1109/TCSI.2018.2837912.

[15] V. Belevitch, *Classical network theory*, San Francisco, CA: Holden-Day, 1968.

[16] C.-W. Ho, A. Ruehli, and P. Brennan, "The modified nodal approach to network analysis," *IEEE Trans. Circuits and Syst.*, vol. 22, no. 6, pp. 504–509, Jun 1975.

[17] D. Fränken, J. Ochs, and K. Ochs, "Generation of wave digital structures for networks containing multiport elements," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 52, no. 3, pp. 586–596, Mar 2005.

[18] G. Borin, G. De Poli, and D. Rocchesso, "Elimination of delay-free loops in discrete-time models of nonlinear acoustic systems," *IEEE Trans. Speech Audio Process.*, vol. 8, pp. 597 – 605, Oct 2000.

[19] J. O. Smith III, "Efficient simulation of the reed-bore and bow-string mechanisms," in *Proc. Int. Comput. Music Conf.*, The Hague, The Netherlands, 1986, pp. 275–280.

[20] P. Cook and G. Scavone, *The synthesis toolkit in C++ (STK), version 4*, 2010. [Online]. Available: http://ccrma.stanford.edu/software/stk.

[21] A. Bernardini and A. Sarti, "Canonical piecewise-linear representation of curves in the wave digital domain," in *Proc. 25th Eur. Signal Process. Conf. (EUSIPCO)*, Kos, Greece, Aug 28–Sep 2, 2017, pp. 1125–1129.

[22] M. J. Olsen, K. J. Werner, and J. O. Smith III, "Resolving grouped nonlinearities in wave digital filters using iterative techniques," in *Proc. 19th Int. Conf. on Digital Audio Effects (DAFx-16)*, Brno, Czech Republic, Sep 5–9, 2016.

[23] W. R. Dunkel, M. Rest, K. J. Werner, M. J. Olsen, and J. O. Smith III, "The Fender Bassman 5F6-A family of preamplifier circuits—a wave digital filter case study," in *Proc. 19th Int. Conf. on Digital Audio Effects (DAFx-16)*, Brno, Czech Republic, Sep 5–9, 2016.

[24] F. Eichas, S. Möller, and U. Zölzer, "Block-oriented gray box modeling of guitar amplifiers," in *Proc. 20th Int. Conf. on Digital Audio Effects (DAFx-17)*, Edinburgh, UK, Sep 5–9, 2017, pp. 184–191.

[25] J. Mačák, "Modified blockwise method for simulation of guitar tube amplifiers," in *Proc. 33rd Int. Conf. on Telecom. and Signal Process. (TSP-10)*, Baden, Austria, Aug 17–20, 2010, pp. 1–4.

[26] J. Mačák and J. Schimmel, "Real-time guitar tube amplifier simulation using an approximation of differential equations," in *Proc. 13th Int. Conf. on Digital Audio Effects (DAFx-10)*, Graz, Austria, Sep 6–10, 2010.

[27] J. Mačák and J. Schimmel, "Real-time guitar preamp simulation using modified blockwise method and approximations," *EURASIP J. Adv. Signal Process.*, 2011, Article #629309.

[28] J. Mačák, *Real-time digital simulation of guitar amplifiers as audio effects*, Ph.D. thesis, Brno University of Technology, Brno, 2012.

[29] K. J. Werner, *Virtual analog modeling of audio circuitry using wave digital filters*, Ph.D. thesis, Stanford University, Stanford, CA, pp. 165–170, 2016.

[30] R. C. D. Paiva, S. D'Angelo, J. Pakarinen, and V. Valimaki, "Emulation of operational amplifiers and diodes in audio distortion circuits," *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 59, no. 10, pp. 688–692, Oct 2012.

[31] K. J. Werner, V. Nangia, A. Bernardini, J. O. Smith III, and A. Sarti, "An improved and generalized diode clipper model for wave digital filters," in *Proc. 139th Int. Audio Eng. Soc. (AES)*, New York, NY, Oct 29– Nov 11, 2015.

[32] A. Bernardini, K. J. Werner, A. Sarti, and J. O. Smith III, "Modeling nonlinear wave digital elements using the Lambert function," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 63, no. 8, Aug 2016.

[33] T. E. Rutt, "Vacuum tube triode nonlinearity as part of the electric guitar sound," in *Proc. 76th Int. Audio Eng. Soc. (AES)*, New York, NY, Oct 8–11, 1984.

[34] R. C. Smith, "Dual mode music instrument amplifier," U.S. Patent 4,211,893, issued Jul 8, 1980.

[35] K. J. Werner, W. R. Dunkel, and F. G. Germain, "A computational model of the Hammond organ vibrato/chorus using wave digital filters," in *Proc. 19th Int. Conf. on Digital Audio Effects (DAFx-16)*, Brno, Czech Republic, Sep 5–9, 2016.

[36] W. M. Leach Jr, "SPICE models for vacuum-tube amplifiers," *J. Audio Eng. Soc.*, vol. 43, no. 3, pp. 117–126, 1995.

[37] N. Koren, "Improved vacuum tube models for spice simulations," *Glass Audio*, vol. 8, no. 5, pp. 18–27, 1996.

[38] G. C. Cardarilli, M. Re, and L. Di Carlo, "Improved large-signal model for vacuum triodes," in *IEEE Int. Symp. Circuits Syst. (ICSCAS)*, Taipei, Taiwan, May 24–27, 2009.

[39] S. D'Angelo, J. Pakarinen, and V. Valimaki, "New family of wave-digital triode models," *IEEE Trans. Audio, Speech, Language Process.*, vol. 21, no. 2, pp. 313–321, Feb 2013.

[40] J. Mačák, J. Schimmel, and M. Holters, "Simulation of fender type guitar preamp using approximation and state space model," in *Proc. 15th Int. Conf. on Digital Audio Effects (DAFx-12)*, York, UK, Sep 17–21, 2012.

[41] P. Raffensperger, "Toward a wave digital filter model of the Fairchild 670 limiter," in *Proc. 15th Int. Conf. on Digital Audio Effects (DAFx-12)*, York, UK, Sep 17–21, 2012.

[42] K. Dempwolf and U. Zölzer, "A physically-motivated triode model for circuit simulations," in *Proc. 14th Int. Conf. on Digital Audio Effects (DAFx-11)*, Paris, France, Sep 19–23, 2011.

[43] C. Sanderson and R. Curtin, "Armadillo: a template-based C++ library for linear algebra," *J. Open Source Softw.*, vol. 1, pp. 26, 2016.

[44] S. Müller and P. Massarani, "Transfer-function measurement with sweeps," *J. Audio Eng. Soc.*, vol. 49, no. 6, pp. 443–471, 2001.

[45] M. Holters and U. Zölzer, "A k-d tree based solution cache for the non-linear equation of circuit simulations," in *Proc. 24th Eur. Signal Process. Conf. (EUSIPCO)*, Budapest, Hungary, Aug 29–Sep 2, 2016, pp. 1028–1032.

TIME WARPING IN DIGITAL AUDIO EFFECTS

Gianpaolo Evangelista

Institute for Composition, Electroacoustics and Sound Engineering Education
MDW University of Music and Performing Arts, Vienna, Austria
evangelista@mdw.ac.at

ABSTRACT

Time warping is an important paradigm in sound processing, which consists of composing the signal with another function of time called the warping map. This paradigm leads to different points of view in signal processing, fostering the development of new effects or the conception of new implementations of existing ones. While the introduction of time warping in continuous-time signals is in principle not problematic, time warping of discrete-time signals is not self-evident. On one hand, if the signal samples were obtained by sampling a bandlimited signal, the warped signal is not necessarily bandlimited: it has a sampling theorem of its own, based on irregular sampling, unless the map is linear. On the other hand, most signals are regularly sampled so that the samples at non-integer multiples of the sampling interval are not known. While the use of interpolation can partly solve the problem it usually introduces artifacts. Moreover, in many sound applications, the computation already involves a phase vocoder. In this paper we introduce new methods and algorithms for time-warping based on warped time-frequency representations. These lead to alternative algorithms for warping for use in sound processing tools and digital audio effects and shed new light in the interaction of time warping with phase vocoders. We also outline the applications of time warping in digital audio effects.

1. INTRODUCTION

Time warping is, in principle, a simple operation consisting in the composition of the time signal with another function of time called the warping map. As a result, the signal is deformed and its plot vs. time appears as if the original time axis had been warped.

Together with its dual operation defining frequency warping, time warping allows for the introduction of new or known effects, which are often described adopting different points of view, and allows for the mapping of signal representations into other representations. For example, the introduction of vibrato in a signal can be either seen as a result of passing the signal through a time-varying delay line or as a result of time warping the signal. Adding the original signal to a collection of time warped versions of the same signal, one can achieve different realizations of flanging and chorus effects. Even frequency or phase modulation as in FM synthesis could be considered as a version of time warping, where the warping map is one-to-one only for small values of the modulation index.

Used in conjunction with time-frequency or time-scale representations, invertible time and frequency warping help allocating analysis time intervals and / or frequency bands which differ from the ones provided by the original representative elements [1, 2, 3, 4, 5]. This makes it possible to obtain, e.g., non-uniform resolution from the uniform resolution of the original representation, or more sophisticated allocations than the ones sketched by rigid and simplified mathematical rules. In recent times, non-stationary Gabor frames were introduced [6, 7] and linked to time and frequency warping operators subject to redressing methods [8, 9].

In phase vocoder based schemes, time-warping of the windows can be considered as a building block for time stretching sound signals; another building block being the adjustments of the phases to provide alignment of the sinusoidal components across the overlapping stretched windows. In the most conventional implementations, a constant stretching as defined by a linear time-warping map is applied. However, the use of a piecewise linear or curvilinear map generally achieves better results in which, e.g., only the stationary part of the signal is stretched or compressed while the transients, especially at the attack of sounds, are left unaltered.

This paper is organized as follows. In Section 2 we recall the definition of warping as an operator and of its unitary version. We explore useful maps for audio processing and effects in Section 2.1 and evaluate the warped sampling expansion as an algorithm for time warping in Sections 2.2, 2.3 and 2.4. In Section 3 we introduce original methods for time-warping and consider the interaction of time-warping with Gabor frames or phase vocoder in Section 3.1. Two original algorithms for warping are shown in Section 3.2 and 3.3, respectively, where the results of experimentation shown in Section 3.4 provide an assessment of the SNR with test signals together with an analysis of the computational costs found in Section 3.5. In Section 4 we give a brief outline of the use of time warping for time stretching and pitch shifting audio signals. Finally, in Section 5 we draw our conclusions.

Examples and experimental code will be made available at the author's web page:

http://members.chello.at/~evangelista/

under the Sound Examples tab - Time Warping.

2. TIME WARPING OPERATORS

Given a function of time γ, which will play the role of the warping map, a time warping operator $\mathbf{W}_\gamma$ is identical to a composition-by-γ operator $\mathbf{C}_\gamma$ acting in the time domain. Thus, we have:

$$s_{tw} = \mathbf{W}_\gamma s = \mathbf{C}_\gamma s = s \circ \gamma, \tag{1}$$

where s_{tw} is the time-warped version of the signal s, γ is the time warping map and $\circ$ denotes function composition. Thus, for any signal $s(t)$ we have

$$s_{tw}(t) = s(\gamma(t)). \tag{2}$$

Conditions that guarantee the boundedness and invertibility of the warping operators can be found in [8] and references therein.

The conditions for the definition of unitary warping / composition operators are generally less strict than those for the boundedness and invertibility of the non-unitary warping operators (see [8] and references therein). If the warping map γ is almost everywhere strictly increasing, one-to-one and differentiable then one can define a unitary time-warping operator $\mathbf{U}_\gamma$ simply by multiplying the non-unitary operator $\mathbf{W}_\gamma$ in (1) by the square root of the magnitude derivative of the map, in which case:

$$s_{tw}(t) = [\mathbf{U}_\gamma s](t) = \sqrt{\left|\frac{d\gamma}{dt}\right|}\, s(\gamma(t)). \tag{3}$$

For simplicity, we assume that the warping maps γ of interest are almost everywhere increasing, so that they are invertible [8], and that both the first derivatives of γ and γ^{-1} are essentially bounded from below. Since the maps are increasing, their derivatives are positive so that the magnitude sign under the square root in (3) can be dropped.

2.1. Some Maps of Interest for Audio Effects

Time warping with arbitrary maps can be used per se as an audio effect, introducing simultaneous pitch modulation and local or global stretching or compression of the signal. The warped signal can also be mixed with the original signals and / or with other differently warped versions of the signal. As it will be shown in Sections 3 and 4, time warping can also be employed in conjunction with time-frequency representations in order to obtain alternative algorithms for warping and to build modified phase vocoders for time stretching or pitch shifting of audio. In this section we illustrate some time-warping maps that are of interest for the construction of new or known audio effects.

Usually we are only interested in the shape of the map for non-negative values of time. If needed, in order to define a map over the entire real axis we may extend the generic map by enforcing odd parity: $\gamma(-t) = -\gamma(t)$.

The simplest time-warping map is linear, also known as affine transformation:

$$\gamma_{lin}(t) = \alpha t + c, \tag{4}$$

with inverse

$$\gamma_{lin}^{-1}(t) = \frac{t-c}{\alpha}, \tag{5}$$

where $\alpha \neq 0$ and c are constants.

In the linear map (4), the parameter α is usually chosen as positive in order to maintain the direction of time, while a negative value produces time reversal effects useful, for example, in granular synthesis. With the generic linear map, each sinusoidal component of the signal at frequency f_0 is brought to frequency $f_1 = \alpha f_0$. Time-wise, the signal is dilated by a factor $1/\alpha$, which means it is stretched if $\alpha < 1$ and compressed if $\alpha > 1$.

Usually, one selects $c = 0$ in order to map the time origin into itself. However, it is always possible to let c be a negative number in order to introduce a time delay, which might be useful, for example, for online computation of time warping. Indeed, as shown in Fig.1, the identity line $g(t) = t$, which represents the present (the loci of time instants that map into themselves), divides the past (the loci of time instants that map into previous times) from the future (the loci of time instants that map into subsequent times). In the same figure, shown is the map $\gamma(t) = t - d$, where d is a positive number, which completely lies in the past and introduces a uniform delay d. For a nonlinear warping map $\gamma(t)$, such that $\gamma(t) - t$ is bounded, it is possible to introduce a delay through composition with a linear delay map so that the whole map does not have points belonging to the future, which makes causal computation possible. That is, given the map $\gamma(t)$, such that $\gamma(t) > t$ in some region, we form the causal map $\tilde{\gamma}(t) = \gamma(t) - d$ where $d \geq \sup_t(\gamma(t) - t)$, which has no points in the future.

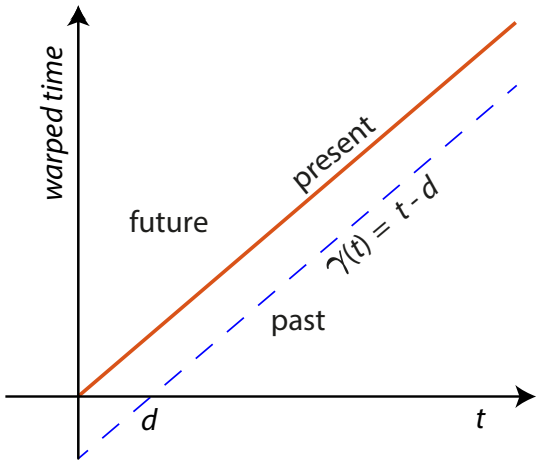


Figure 1: *Subdivision of the time-warped time plane into past, present and future. Also shown is a linear map introducing a delay d.*

Piecewise linear maps can also be exploited in order to produce dynamic warping effects in a simple way, where several versions of (4) and (5) are used on contiguous finite disjoint intervals and their mapped intervals, respectively. In that case, the constant c of each map is chosen to guarantee continuity, where the initial value of the map matches the final value of the previous adjacent map.

Another class of maps of interest is given by the chirps, examples of which are

$$\gamma_l(t) = t + \beta_l t^2, \tag{6}$$

which, when applied to a sinusoidal signal component produces a linear chirp and

$$\gamma_q(t) = t + \beta_q t^3, \tag{7}$$

which gives a quadratic chirp.

When applied to audio signals, the chirp maps produce glissandos. The linear chirp map brings any sinusoidal component of the signal of frequency f_0 to a signal having instantaneous frequency $f_0(1 + 2\beta_l t)$. The parameter β_l can be selected to achieve a target frequency f_1 after a time interval of duration τ, in that case we set $\beta_l = (f_1 - f_0)/2f_0\tau$. It is convenient to express β_l in a form that does not depend on f_0 as follows:

$$\beta_l = \frac{\rho - 1}{2\tau}, \tag{8}$$

where ρ it the ratio of frequency change in the lapse τ. Only values of $\rho > 1$, corresponding to $\beta_l > 0$, guarantee the invertibility of the map at all times. This is the case of upward chirps where the frequency increases. Downward chirps with $\beta_l < 0$ can still

be used on finite length intervals provided that one checks, for invertibility, that the derivative of the map does not change sign.

For $\beta_l > 0$ the inverse of the linear chirp map is

$$\gamma_l^{-1}(t) = \frac{-1 + \sqrt{4\beta_l t + 1}}{2\beta_l}. \tag{9}$$

This map can also be used, by exchanging the roles of the direct and inverse maps, to produce downward chirps, which are, however, not linear.

The quadratic chirp warping map (7) dynamically maps the frequency f_0 to the instantaneous frequency $f_0(1 + 3\beta_q t^2)$. Here again, for complete invertibility we require $\beta_q > 0$ and we can express this parameter in terms of the frequency change ratio ρ in the lapse τ as follows:

$$\beta_q = \frac{\rho - 1}{3\tau^2}, \tag{10}$$

with $\rho > 1$ for an upward chirp. For $\beta_q > 0$, the inverse of the quadratic chirp map is given as follows:

$$\gamma_q^{-1}(t) = \frac{\sqrt[3]{\frac{2}{3\beta_q}}}{Q_{\beta_q}(t)} - \frac{Q_{\beta_q}(t)}{\sqrt[3]{18\beta_q^2}} \tag{11}$$

which gives a downward chirps, where

$$Q_{\beta_q}(t) = \sqrt[3]{\sqrt{81\beta_q^2 t^2 + 12\beta_q} - 9\beta_q t}. \tag{12}$$

To conclude our exploration of relevant warping maps, we consider the phase modulation map

$$\gamma_{pm}(t) = t + I_m sin(2 * \pi f_m t), \tag{13}$$

where f_m is the modulating frequency and I_m is the modulation index expressed in multiples of the carrier frequency, i.e., of the frequency f_c of the sinusoidal component of the signal to which the map is applied. As it is easy to check from its derivative, this warping map is invertible only if $I_m < 2\pi f_m$. It is useful for producing vibratos, for small values of the modulating frequency and for phase modulating audio signals as a special FM effect where the carrier is not a necessarily single sinusoid. The inverse map of (13) cannot be expressed in closed form. When an inverse is desired, one can resort to linear interpolation from the direct map by exchanging the abscissae with the ordinates. Alternatively, one can numerically find for each time point t of interest the zero of the function $\gamma(x) - t$.

Yet another possibility is given by the approximation of the phase modulation map by means of the invertible map

$$\gamma_{apm}(t) = t + \frac{1}{\pi f_m} \tan^{-1}\left(\frac{b_m \sin(2\pi f_m t)}{1 - b_m \cos(2\pi f_m t)}\right), \tag{14}$$

which is inspired from the phase response of a first order real all-pass filter. The inverse map of (14) can be readily found by changing the sign of b_m. This parameter controls the modulation index and can be optimized for the map to approximate (13). By matching the points of maximum deflection from the linear component t of (13) with the values of (14) at the same points, i.e. at the points $2\pi f_m t = \pi/2 + 2k\pi$, $k \in \mathbb{Z}$, one can see that a good estimate for b_m is $\hat{b}_m = \tan(\pi f_m I_m)$. The detail of the approximation of the phase modulation map with the map is shown in Fig.2. This way we obtain an invertible map in closed form that is more practical for phase modulation effects.

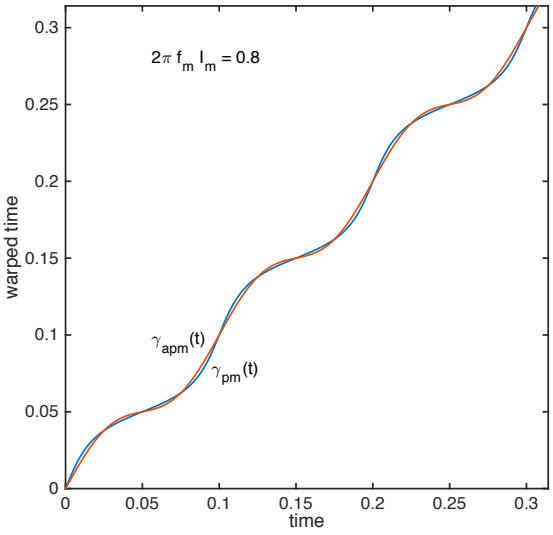


Figure 2: *Detail of the approximation of the phase modulation map with the map γ_{apm} in (14).*

2.2. Sampling Theorem for Time-Warped Bandlimited Signals

Assume that $s_{tw}(t) = s(\gamma(t))$ as in (2), where the function γ is invertible. If the original signal $s(t)$ is bandlimited to $\left]-\frac{f_s}{2}, +\frac{f_s}{2}\right[$, where $f_s/2$ is the Nyquist frequency, then it admits a sampling reconstruction formula

$$s(t) = \sum_n s(nT)\text{sinc}\left(\frac{t}{T} - n\right), \tag{15}$$

where $\text{sinc}\,(t) = \frac{\sin \pi t}{\pi t}$ and $T = 1/f_s$ is the sampling interval. With the simple observation that $s_{tw}(\gamma^{-1}(t)) = s(t)$ one can conclude that $s(nT) = s_{tw}(\gamma^{-1}(nT))$. Thus, by time-warping both sides of (15) one obtains the following sampling reconstruction for the warped signal:

$$s_{tw}(t) = \sum_n s_{tw}(\tau_n)\text{sinc}\left(\frac{\gamma(t)}{T} - n\right), \tag{16}$$

where $\tau_n = \gamma^{-1}(nT)$ are sampling instants that are not regularly spaced unless the map is linear. However, as conjectured in [10], if $\gamma(t)$ is an invertible function, the time-warped signal $s(\gamma(t))$ is not guaranteed to be bandlimited, unless $\gamma(t)$ is an affine map (4). This conjecture is shown to be true for a wide class of maps, including the ones arbitrarily close to a linear map and the piecewise linear ones [11, 10, 12].

The warped sampling expansion (16) constitutes an important algorithm for time-warping discrete-time signals. In fact, knowing that $s(nT) = s_{tw}(\tau_n)$ and disregarding possible aliasing, one can compute the discrete-time time-warped signal by evaluating (16) at uniformly spaced sampling instants $t_r = rT$, for any $r \in \mathbb{Z}$:

$$s_{tw}(rT) = \sum_n s(nT)\text{sinc}\left(\frac{\gamma(rT)}{T} - n\right). \tag{17}$$

Since in computations the sinc function is impractical as it extends over the entire time axis, one can approximate it by a windowed sinc interpolating kernel like the Lanczos kernel

$$\varphi_L(t) = \begin{cases} \text{sinc}\left(\frac{t}{L}\right)\text{sinc}(t) & t \in [-L, +L[\\ 0 & \text{otherwise} \end{cases} \quad (18)$$

or the von Hann windowed sinc [13]:

$$\varphi_L(t) = \begin{cases} \cos^2\left(\frac{t}{2L}\right)\text{sinc}(t) & t \in [-L, +L[\\ 0 & \text{otherwise} \end{cases} \quad (19)$$

Here L is an integer parameter which controls the extension of the approximation interval. In both cases, and in many other similar choices of window function, we have $\lim_{L\to\infty} \varphi_L(t) = \text{sinc}(t)$. Thus,

$$s_{tw}(rT) \approx \sum_n s(nT)\varphi_L\left(\frac{\gamma(rT)}{T} - n\right) \quad (20)$$

is a discrete-time approximation of the time warped signal, which is increasingly better as L grows.

The time and frequency domain characteristics of both the Lanczos interpolating kernel and the von Hann windowed sinc are shown in Fig.3. While the two interpolating kernels are very similar in the time domain, the magnitude Fourier transform of Lanczos' kernel shows a slightly steeper frequency roll-off of the main lobe.

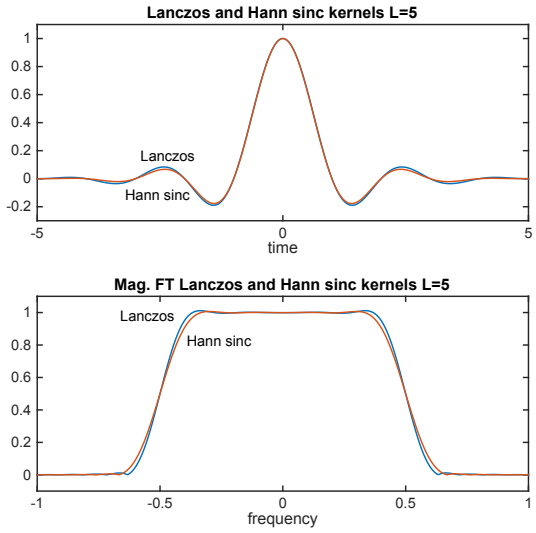


Figure 3: *Lanczos and von Hann windowed sinc interpolating kernels: time domain and magnitude Fourier transforms.*

2.3. Experimental Results

In order to assess the quality of the approximation (20) we performed tests with artificial signals which have an analytic closed form and we measured the SNR. The error is estimated as the difference of the signal warped as in (20) and the one obtained by applying the warping map function directly to the artificial signal. Clearly, the results depend on the map and on the signal. In Fig.4 we plot the SNR as a function of the width parameter L for both the Lanczos and the von Hann windowed sinc kernels, choosing as warping map the linear function $\gamma(t) = \alpha t$.

The test signals consisted of 1 KHz sinusoids with smooth envelopes and we chose a sampling rate of 44.1 KHz for the sampling expansion. Varying the parameter α of the warping map, we concluded from the observations that for $\alpha \geq 1$ a 255 dB SNR (not shown in the figure) was achieved in all cases, which means no error above machine precision. The worst cases are for $\alpha \ll 1$; the values of the SNR shown in the figure are computed with $\alpha = 1/16$, which warps the 1 KHz sinusoid down to 67 Hz. The intuition about the lower performance at lower map derivatives lies in the fact that the sample instants τ_n of s_{tw} in the RHS of (16), of which (20) is an approximation, are linked to the inverse map $\gamma^{-1}(t) = t/\alpha$. Thus, since $\tau_n = nT/\alpha$, their density is lower for smaller values of α and so is the quality of the approximation. However, increasing the sampling rate did not show great benefits. This might be due to the fact that we are dealing with sinusoids still within average Nyquist rate, while increasing the sampling rate proportionally increases the number of the terms in the sampling expansion estimate (20), which brings a proportionally higher approximation error.

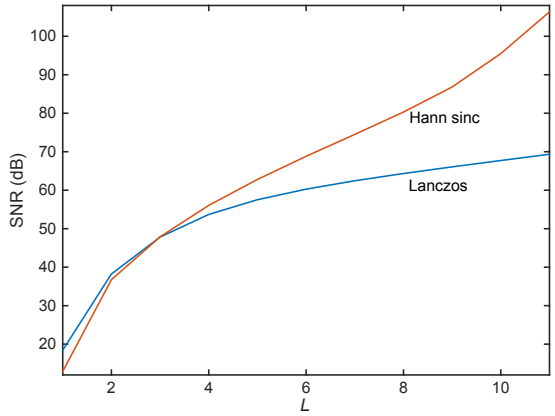


Figure 4: *SNR characteristics of Lanczos and von Hann windowed sinc kernels as a function of the support width parameter L.*

From Fig.4 we remark that for values of $L > 3$ the von Hann windowed sinc outperforms the Lanczos interpolating kernel. A choice of $L = 5$ or larger leads to good approximations where the artifacts are inaudible at 56 dB SNR or better, reaching 106 dB for $L = 11$.

The SNR results in Fig.4 were confirmed up to the first decimal digit in other tests we performed using the linear and quadratic chirp maps described in Section 2.1. We also tried their inverses, which give downwards chirps, achieving similar SNR and acoustic results.

2.4. Computational Complexity

The computational complexity of the warped sampling expansion can be easily estimated by observing that in (20), in order to produce the output, any input sample is multiplied by the sinc window kernel and these are added together with the shifted windows of

previous and future samples within the window length. The extension of the warped window depends on the warping map. Given an increasing warping map γ and the window extension factor K, we have that the sampled support of the warped window is the set of $r \in \mathbb{Z}$ such that:

$$\frac{\gamma^{-1}((n-L)T)}{T} \leq r < \frac{\gamma^{-1}((n+L)T)}{T}. \tag{21}$$

For a linear warping map $\gamma(t) = \alpha t$ the width of the support is approximately $2L/\alpha$ samples. For the generic map, in order to compute the average complexity, α can be replaced by the average derivative of the map:

$$\bar{\alpha} = \frac{1}{\Delta}\int_0^{\Delta} \frac{d\gamma}{dt}dt = \frac{\gamma(\Delta)}{\Delta}, \tag{22}$$

over a finite interval of duration Δ, where we have assumed $\gamma(0) = 0$, or by its limit as $\Delta \to \infty$, if it exists, for the infinite interval.

In conclusion, the average complexity of the warped sampling theorem based algorithm for time warping is proportional to $2L/\bar{\alpha}$ per sample.

3. TIME WARPING AND TIME-FREQUENCY REPRESENTATIONS

The warped sampling expansion illustrated in the previous section is not the only algorithm for time warping signals. In fact, the expansion of the signal in any set of complete orthogonal or biorthogonal functions of time in $L^2(\mathbb{R})$ can be used and so is the expansion into any time domain frame in the same space, provided that either the analysis or the synthesis functions can be expressed in closed analytic form.

In this paper we consider a new approach to the time warping of discrete-time signals based on Gabor frame expansions. While in [14] we considered a similar approach based on filter banks for the computation of frequency warping, time warping has different requirements, which deserve a separate discussion, and the important computational advantage that finite-length windows are still finite-length after warping. Moreover, for audio processing purposes, the filter bank approach to time-warping allows one to control the output bandwidth simply by limiting the number of frequency components.

3.1. Gabor Frames

In this paper, we consider the signal expansion over a Gabor frame[1] whose representative elements are obtained by modulating and time-shifting a suitable window function $h_s(t)$:

$$\varphi^{(s)}_{n,m}(t) = \left[\mathbf{T}_{na}\mathbf{M}_{mb}h_s\right](t) = h_s(t-na)e^{j2\pi mb(t-na)}, \tag{23}$$

where $\mathbf{T}_\tau$ is the shift by τ operator and $\mathbf{M}_\nu$ is the modulation by ν operator which consists in multiplication by $e^{j2\pi\nu t}$. The constants a and b respectively represent the time-shift sampling interval (hop size) and the frequency sampling interval (distance of the frequency bins), with $ab \leq 1$ a necessary condition for the set $\left\{\varphi^{(s)}_{n,m}(t)\right\}_{n,m\in\mathbb{Z}}$ to form a frame.

We recall that the sequence of functions $\{\varphi_{n,m}\}_{n,m\in\mathbb{Z}}$ in $L^2(\mathbb{R})$ is called a frame if there exist two positive constants A and B such that

$$A\|s\|^2 \leq \sum_{m,n} |\langle s, \varphi_{n,m}\rangle|^2 \leq B\|s\|^2 \quad \forall s \in L^2(\mathbb{R}), \tag{24}$$

where $\|s\|^2 = \langle s, s\rangle$ is the norm square of the signal.

For the Gabor set (23) one can show that perfect reconstruction (PR) is guaranteed if there exists an analysis window $h_a(t)$ such that

$$\sum_n h_a\left(t + \frac{r}{b} - na\right)h_s(t-na) = b\delta_{r,0}, \tag{25}$$

where $\delta_{r,0}$ is the Kronecker delta and the analysis frame is given by

$$\varphi^{(a)}_{n,m}(t) = \left[\mathbf{T}_{na}\mathbf{M}_{mb}h_s\right](t) = h_a(t-na)e^{j2\pi mb(t-na)}. \tag{26}$$

General necessary and sufficient conditions for compact supported and exponentially decaying windows to generate a Gabor frame are given in [15]. For compact supported analysis and synthesis windows with support smaller than $1/b$, the popular overlap-add condition must be satisfied for PR:

$$\frac{1}{b}\sum_n h_a(t-na)h_s(t-na) = 1. \tag{27}$$

It is always possible, in this case, to modify the windows so that the generated frame is tight ($A = B$) so that the analysis and synthesis windows are identical.

If $\left\{\varphi^{(s)}_{n,m}\right\}_{n,m\in\mathbb{Z}}$ forms a Gabor frame with dual frame $\left\{\varphi^{(a)}_{n,m}\right\}_{n,m\in\mathbb{Z}}$, any signal $s(t)$ in $L^2(\mathbb{R})$ can be represented as follows:

$$s(t) = \sum_{m,n} S(na, mb)h_s(t-na)e^{j2\pi mb(t-na)}, \tag{28}$$

where

$$S(\tau, \nu) = \int_{-\infty}^{+\infty} s(t)h_a(t-\tau)e^{-j2\pi\nu(t-\tau)}dt \tag{29}$$

is the Short-Time Fourier Transform of the signal with analysis window $h_a(t)$, so that

$$S(na, mb) = \left\langle s, \varphi^{(a)}_{n,m}\right\rangle \tag{30}$$

is its sampled version on the uniform grid $\{na, mb\}_{n,m\in\mathbb{Z}}$.

A discrete-time version of (28) can be obtained in the same form by uniformly sampling time $t_k = kT$ and by choosing hop-size $a = NT$ and frequency sample interval $b = 1/MT$, where both N and M are integers with $M \geq N$, where, in typical applications, $M = KN$, with the integer K controlling the overlap factor. Furthermore, the summation over the frequency index m is finite in the discrete case.

[1] Throughout this paper we use an equivalent definition of Gabor frames and STFT which includes time shift (na or τ) in the complex sinusoids, so that windows are synchronous with the phase of the exponentials.

3.2. Time Warping by Means of Phase Vocoder: Form I

By time-warping both sides of (28) one has the following expansion for the continuous time time-warped signal:

$$s(\gamma(t)) = \sum_{m,n} S(na, mb) h_s\left(\gamma(t) - na\right) e^{j2\pi mb(\gamma(t)-na)}. \quad (31)$$

Due to unitary equivalence [2] through the unitary warping operator $\mathbf{U}_\gamma$, the set

$$\mathbf{U}_\gamma \varphi^{(s)}_{n,m}(t) = \sqrt{\dot{\gamma}(t)} h_s\left(\gamma(t) - na\right) e^{j2\pi mb(\gamma(t)-na)}, \quad (32)$$

where $\dot{\gamma}(t)$ is the time derivative of the warping map, is a frame if and only if the set in (23) is a frame. Thus, an alternate scheme for warping a continuous-time signal consists of projecting the signal over a Gabor analysis frame and compute the expansion (31) over the time-warped frame (32).

An algorithm of interest for time-warping discrete-time signals can be obtained by discretizing (31):

$$\tilde{s}(k) = \sum_{m=-\lfloor \frac{M}{2} \rfloor}^{+\lfloor \frac{M}{2} \rfloor} \sum_n S_{n,m} h_s\left(g_k - nNT\right) e^{j\frac{2\pi m}{M}\left(\frac{g_k}{T} - nN\right)}, \quad (33)$$

where $g_k = \gamma(kT)$, while $\tilde{s}(k) = s_{tw}(kT) = s(\gamma(kT))$ is the discrete-time warped signal and

$$S_{n,m} = \sum_k s(kT) h_a\left((k - nN)T\right) e^{-j\frac{2\pi m}{M}(k-nN)}, \quad (34)$$

with $n \in \mathbb{Z}$ and $m = -\left\lfloor \frac{M}{2} \right\rfloor, ..., +\left\lfloor \frac{M}{2} \right\rfloor$, are the Gabor expansion coefficients obtained by projection over the corresponding discrete-time frame. We refer to this algorithm as the Form I computation.

The computation of time warping by means of (34) and (33) only involves the warping of the synthesis window and of the complex exponentials, which are continuous-time functions expressed in closed form so this operation does not pose any problem. Assuming that the warping map is invertible, if the original synthesis window $h_s(t)$ has compact support in $\left[-\frac{KNT}{2}, +\frac{KNT}{2}\right[$, the shifted warped windows $h_s\left(\gamma(t) - nNT\right)$ also have compact support in

$$\left[\gamma^{-1}\left(\left(n - \frac{K}{2}\right)NT\right), \gamma^{-1}\left(\left(n + \frac{K}{2}\right)NT\right)\right[. \quad (35)$$

From this it is easy to find the samples instants kT for which the warped window sequence in (34) is nonzero. Since the numbers $\frac{g_k}{T}$ in (33) are not integer, the computation of the synthesis cannot be performed by means of the IFFT.

3.3. Time Warping by Means of Phase Vocoder: Form II

An alternate algorithm for discrete-time time warping by means of Gabor frames, can be obtained by computing the coefficients of the warped signal by projecting it over a frame in the form (23). Thus, we compute the scalar products

$$\tilde{S}(na, mb) = \left\langle \mathbf{U}_\gamma s, \varphi^{(a)}_{n,m} \right\rangle = \left\langle s, \mathbf{U}_{\gamma^{-1}} \varphi^{(a)}_{n,m} \right\rangle, \quad (36)$$

where the last equality is due to the fact that the warping operator is unitary and its adjoint corresponds to unitary warping with the inverse map γ^{-1}. Thus, (36) is equivalent to projecting the signal over the inversely warped frame, whose elements are:

$$\mathbf{U}_{\gamma^{-1}} \varphi^{(a)}_{n,m}(t) = \sqrt{\dot{\gamma}^{-1}(t)} h_a\left(\gamma^{-1}(t) - na\right) e^{j2\pi mb(\gamma^{-1}(t)-na)}. \quad (37)$$

For the synthesis one uses the dual frame $\left\{\varphi^{(s)}_{n,m}\right\}_{n,m\in\mathbb{Z}}$. Passing to discrete time as in Section 3.2, we have the following algorithm to compute (unitary) warping:

$$\tilde{s}(k) = \sum_{m=-\lfloor \frac{M}{2} \rfloor}^{+\lfloor \frac{M}{2} \rfloor} \sum_n \tilde{S}_{n,m} h_s\left((k - nN)T\right) e^{j\frac{2\pi m}{M}(k-nN)}, \quad (38)$$

where $\tilde{s}(k) = \sqrt{\dot{\gamma}(kT)} s(\gamma(kT))$ is the discrete-time unitarily warped signal and

$$\tilde{S}_{n,m} = \sum_k d_k s(kT) h_a\left(g_k - nNT\right) e^{-j\frac{2\pi m}{M}\left(\frac{g_k}{T} - nN\right)} \quad (39)$$

with $n \in \mathbb{Z}$ and $m = -\left\lfloor \frac{M}{2} \right\rfloor, ..., +\left\lfloor \frac{M}{2} \right\rfloor$, are the expansion coefficients obtained by projection over the discrete-time analogue of (37), where $d_k = \sqrt{\dot{\gamma}^{-1}(kT)}$ and $g_k = \gamma^{-1}(kT)$. We refer to this algorithm as the Form II computation of discrete time warping. Since the numbers $\frac{g_k}{T}$ in (39) are not integer, the computation of the analysis coefficients cannot be performed by means of the FFT.

3.4. Experimental Results

In this section we provide an assessment of the quality of the algorithms proposed in Sections 3.2 and 3.2 for computing discrete-time time warping by means of generalized Gabor expansions, namely, through Form I in (33) and (34) or through Form II in (38) and (39). For comparison, we used the same sets of closed form signals to evaluate the SNR as we did in the evaluation of the sampling expansion based method in Section 2.3.

The average results for Form I are shown in Fig.5. Here again, the SNR results did not show great variability across the warping maps we tested in our experiments, from linear map to linear and quadratic chirps.

The results obtained by varying the overlap factor K from 2 to 8 and for several values of the hop-size factor N from 64 to 512 are a bit erratic but most of the SNRs are above 100 dB, which are sufficient for most audio applications. The quality of the results generally improves with the length of the window KN. We note that at equal window lengths but different overlap factors, e.g., on the $N = 512$ curve with $K = 2$ and the $N = 256$ curve with $K = 4$, we obtain similar SNRs.

In both Form I and Form II algorithm one should choose suitably long windows, as these are time-warped, in the analysis algorithm for Form II and in the synthesis for Form I. Thus, for a given choice of the fixed window length, the warped versions can become too short. The dilation factor of the window locally depends on the time derivative of the map. With this into consideration, with dilation factors smaller than 1 we obtained similar results for Form II to those for Form I at the cost of generally larger window lengths. Typical characteristics of the SNR for the Form II are shown in Fig.6, where we used a linear map (4) with coefficient $\alpha = 0.7$ and $c = 0$. We notice that while the window length is modulated in the analysis, the windows size remains constant in the synthesis, thus involving an extra amount of operations, also depending on the overlap factor.

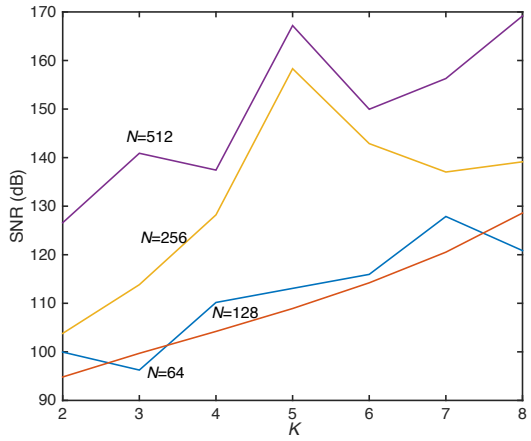


Figure 5: *Average SNR characteristics for the Form I algorithm as a function of the overlap parameter K, for several values of the hop size factor N.*

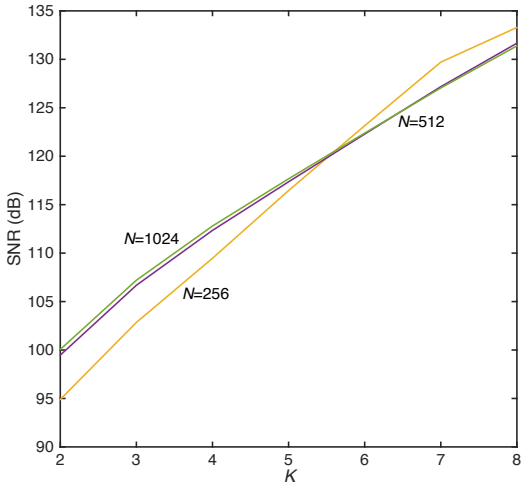


Figure 6: *SNR characteristics for the Form II algorithm for a linear map $\gamma(t) = 0.7 \cdot t$ as a function of the overlap parameter K, for several values of the hop size factor N.*

3.5. Computational Complexity

The computational complexity of the phase vocoder based time-warping algorithms is generally higher than the sampling expansion based algorithm. In fact, in either Form I or Form II algorithms, the analysis and the synthesis cannot be both computed by means of FFT. Thus, a matrix DFT-like form computation is necessary to obtain the warped Fourier transform for each time shift of the window. Due to the variable support of the warped windows, this computation requires on the average an order of $M^2/\bar{\alpha}$ operations in Form I and an order of $\bar{\alpha}M^2$ operations in Form II, where $\bar{\alpha}$ is the average derivative of the map (see discussion in Section 2.4) and M is the number of frequency bins. Each of these computations generates N samples, where N is the hop size.

Thus, the computational complexity of Form I is proportional to $MK/\bar{\alpha}$ operations per sample and that of the Form II to $\bar{\alpha}MK$ operations per samples, where K is the overlap factor such that $M = KN$. Comparing these results with the complexity of the windowed sinc kernel interpolation in Section 2.4, we see that since to achieve similar SNR values, the width factor L of the kernel can be chosen to be smaller than the length M of the window in Form I and Form II, the latter are computationally less efficient. However, in many audio applications a phase vocoder could already be part of the computational structure of the effect. In that case, time warping can be introduced with little extra effort in order to build dynamic effects.

4. TIME WARPING IN TIME STRETCHING AND PITCH SHIFTING

Time warping stretches or compresses signals altering both their pitch and their duration. Often, in sound processing, it is desirable to time stretch the signal without changing the pitch or to pitch shift the signal while preserving its original duration [16]. In the previous parts of this paper we have been considering time warping at the input or output signal level. However, it turns out that for time stretching, time warping in the STFT domain is the best approach.

In order to time stretch a sound signal without altering its pitch, one wishes to scale or time-warp the envelopes of the partials without altering the oscillation time. We are going to perform a similar derivation of the stretching algorithms to the one for uniform STFT albeit in the warped framework. In order to have an idea of the operations involved, first consider the STFT $S(\tau, \nu)$ (29) of a sinusoidal signal of frequency f with a slowly-varying envelope $a(t)$:

$$s(t) = a(t)e^{j2\pi ft}. \tag{40}$$

If, for any shift τ, the envelope is approximately constant over the support of the shifted analysis window $h_a(t-\tau)$, then we have

$$S(\tau, \nu) \approx a(\tau)H_a(\nu - f)e^{j2\pi f\tau}, \tag{41}$$

where $H_a(\nu)$ is the Fourier transform of the analysis window. Thus, for the simple signal (40), the time characteristics of the magnitude STFT only depends on the amplitude envelope and the time characteristics of the phase only depends on the frequency of the sinusoid. If we time warp only the magnitude STFT, i.e., we let $\tilde{S}(\tau, \nu) = \tilde{a}(\tau)H_a(\nu - f)e^{j2\pi f\tau}$ where $\tilde{a}(\tau) = \sqrt{\dot{\gamma}(t)}a(\gamma(\tau))$, and perform reconstruction via the usual synthesis form:

$$\tilde{s}(t) = \int_{-\infty}^{+\infty} d\nu \int_{-\infty}^{+\infty} d\tau \tilde{S}(\tau, \nu)h_s(t-\tau)e^{j2\pi\nu(t-\tau)}, \tag{42}$$

under the assumption that the warped envelope $\tilde{a}(t)$ is still approximately constant over the support of the shifted windows, we obtain

$$\tilde{s}(t) \approx \tilde{a}(t)e^{j2\pi ft}, \tag{43}$$

which is the dynamically stretched sinusoid with the original frequency f.

An alternate equivalent form of (42) consists of pre-unwarping the phase of the STFT with the inverse map γ^{-1}, then warp the result with the map γ and finally perform the synthesis. Clearly, by the unitarity of the warping operator, this is equivalent to performing inverse warping, with respect to time-shift τ, on the synthesis

windows:

$$\tilde{s}(t) = \int_{-\infty}^{+\infty} d\nu \int_{-\infty}^{+\infty} d\tau \tilde{\tilde{S}}(\tau,\nu)\tilde{h}_s(t,\tau)e^{j2\pi\nu(t-\gamma^{-1}(\tau))} \quad (44)$$

where

$$\tilde{h}_s(t,\tau) = \sqrt{\dot{\gamma}^{-1}(\tau)}h_s(t-\gamma^{-1}(\tau))$$

and

$$\tilde{\tilde{S}}(\tau,\nu) = a(\tau)H_a(\nu-f)\sqrt{\dot{\gamma}^{-1}(\tau)}e^{j2\pi f\gamma^{-1}(\tau)}.$$

The latter form (44) can be recognized as a generalization of the most common phase vocoder approach for time stretching. There, for the synthesis we change the hop-size with respect to the original analysis hop-size. This can be seen as a time warping of the hop-size with a linear map.

Once obtained the dynamically time stretched version of the signal from (42) or (44), the dynamically pitch shifted version can be obtained simply by time warping the result with the inverse map γ^{-1}.

By superposition, if the signal partials fall in sufficiently distant frequency bins, which can be adjusted by properly choosing the frequency resolution, our derivation easily extends to signals made out of several enveloped sinusoids. Of course, in reality, things get a bit more complicated than this outline. In fact, just as in the ordinary phase vocoder, multiple sinusoidal partials can interfere within common analysis bins and make the phase of the STFT have a more complicated dependency on the frequencies of the single partials. Unstable pitch due to vibrato or glissando and transients can alter the simple result. Moreover, in practice, one attempts to time stretch signals directly from a sampled version of the STFT given by the phase vocoder.

Sampled counterparts of (42) and (44) can be readily defined. The measures for robustly adapting these methods for use in time stretching and pitch shifting will be the object of forthcoming work.

5. CONCLUSIONS

In this paper we have considered the problem of time-warping discrete-time signals. We compared the algorithm based on the warped sampling expansion with two new methods, Form I and Form II, obtained from the interaction of time warping with a phase vocoder. While the latter require a higher number of operations, all the methods considered achieve high quality in terms of SNR. The phase vocoder based methods have a more flexible design in terms of window length and even transformation scheme. Their use in audio effects could be desirable when a phase vocoder is already present in the computational structure of the effect. Moreover, in conjunction with phase vocoders, time warping is part of the algorithms for time stretching and pitch shifting.

6. REFERENCES

[1] R. G. Baraniuk and D. L. Jones, "Warped wavelet bases: unitary equivalence and signal processing," in *1993 IEEE International Conference on Acoustics, Speech, and Signal Processing*, April 1993, vol. 3, pp. 320–323 vol.3.

[2] R.G. Baraniuk and D.L. Jones, "Unitary equivalence : A new twist on signal processing," *IEEE Transactions on Signal Processing*, vol. 43, no. 10, pp. 2269–2282, Oct. 1995.

[3] A.V. Oppenheim, D.H. Johnson, and K. Steiglitz, "Computation of spectra with unequal resolution using the Fast Fourier Transform," *Proc. of the IEEE*, vol. 59, pp. 299–301, Feb. 1971.

[4] G. Evangelista and S. Cavaliere, "Frequency Warped Filter Banks and Wavelet Transform: A Discrete-Time Approach Via Laguerre Expansions," *IEEE Transactions on Signal Processing*, vol. 46, no. 10, pp. 2638–2650, Oct. 1998.

[5] G. Evangelista and S. Cavaliere, "Discrete Frequency Warped Wavelets: Theory and Applications," *IEEE Transactions on Signal Processing*, vol. 46, no. 4, pp. 874–885, Apr. 1998, special issue on Theory and Applications of Filter Banks and Wavelets.

[6] P. Balazs, M. Dörfler, F. Jaillet, N. Holighaus, and G.A. Velasco, "Theory, implementation and applications of nonstationary Gabor Frames," *Journal of Computational and Applied Mathematics*, vol. 236, no. 6, pp. 1481–1496, 2011.

[7] G.A. Velasco, N. Holighaus, M. Dörfler, and T. Grill, "Constructing an invertible constant-Q transform with nonstationary Gabor frames," in *Proceedings of the Digital Audio Effects Conference (DAFx-11)*, Paris, France, 2011, pp. 93–99.

[8] G. Evangelista, "Redressing Warped Wavelets and Other Similar Warped Time-Something Representations," in *Proceedings of the Digital Audio Effects Conference (DAFx-17)*, Edinburgh, UK, 2017, pp. 260–267.

[9] G. Evangelista, M. Dörfler, and E. Matusiak, "Arbitrary phase vocoders by means of warping," *Music/Technology*, vol. 7, no. 0, 2013.

[10] D. Cochran and J.J. Clark, "On the sampling and reconstruction of time-warped bandlimited signals," in *International Conference on Acoustics, Speech, and Signal Processing*, Apr 1990, pp. 1539–1541 vol.3.

[11] J. Clark, M. Palmer, and P. Lawrence, "A transformation method for the reconstruction of functions from nonuniformly spaced samples," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 33, no. 5, pp. 1151–1165, Oct 1985.

[12] S. Azizi, D. Cochran, and J.N. McDonald, "Reproducing kernel structure and sampling on time-warped spaces with application to warped wavelets," *IEEE Transactions on Information Theory*, vol. 48, pp. 789–790, Mar. 2002.

[13] A.N. Jarrot, C. Ioana, and A. Quinquis, "Toward The Use Of The Time-Warping Principle With Discrete-Time Sequences," *Journal of Computers (JCP)*, vol. 2, no. 6, pp. 49–55, Aug. 2007, NonWOS.

[14] G. Evangelista and S. Cavaliere, "Real-time and efficient algorithms for frequency warping based on local approximations of warping operators," in *Proceedings of the Digital Audio Effects Conference (DAFx-07)*, Bordeaux, France, Sept. 2007, pp. 269–276.

[15] H. Bölcskei and J.E.M. Janssen, "Gabor frames, unimodularity, and window decay," *Journal of Fourier Analysis and Applications*, vol. 6, no. 3, pp. 255–276, May 2000.

[16] J. Driedger and M. Müller, "A review of time-scale modification of music signals," *Applied Sciences*, vol. 6, no. 2, 2016.

JOINT MODELING OF IMPEDANCE AND RADIATION AS A RECURSIVE PARALLEL FILTER STRUCTURE FOR EFFICIENT SYNTHESIS OF WIND INSTRUMENT SOUND

Esteban Maestre
CAML / CIRMMT
McGill University
Montréal, QC, Canada
esteban@music.mcgill.ca

Gary P. Scavone
CAML / CIRMMT
McGill University
Montréal, QC, Canada
gary@music.mcgill.ca

Julius O. Smith
CCRMA
Stanford University
Stanford, CA, USA
jos@ccrma.stanford.edu

ABSTRACT

In the context of efficient synthesis of wind instrument sound, we introduce a technique for joint modeling of input impedance and sound pressure radiation as digital filters in parallel form, with the filter coefficients derived from experimental data. In a series of laboratory measurements taken on an alto saxophone, the input impedance and sound pressure radiation responses were obtained for each fingering. In a first analysis step, we iteratively minimize the error between the frequency response of an input impedance measurement and that of a digital impedance model constructed from a parallel filter structure akin to the discretization of a modal expansion. With the modal coefficients in hand, we propose a digital model for sound pressure radiation which relies on the same parallel structure, thus suitable for coefficient estimation via frequency-domain least-squares. For modeling the transition between fingering positions, we propose a simple model based on linear interpolation of input impedance and sound pressure radiation models. For efficient sound synthesis, the common impedance-radiation model is used to construct a joint reflectance-radiation digital filter realized as a digital waveguide termination that is interfaced to a reed model based on nonlinear scattering.

1. INTRODUCTION

For robust and efficient sound synthesis, many digital waveguide models [1] of wind instruments approximate their air columns as being cylindrical. In a typical digital waveguide model, the air column of an ideal instrument constructed from a cylindrical pipe and a bell can be represented by a pair of delay lines simulating pressure wave propagation inside the pipe, and a termination that includes two digital filters: one that lumps frequency-dependent propagation losses and dispersion, and another one emulating the frequency-dependent bell reflectance. In these efficient schemes, the reed-valve end termination of the pipe is often modeled via a nonlinear scattering element that is interfaced to the air column model through decomposed pressure traveling waves P^+ and P^-, respectively going into and reflected back from the pipe input interface. Approximations with conical elements are possible [2] but often result in inharmonic resonance structures that are difficult to tune for sound synthesis [3].

To account for realistic, non-ideal instrument air column shapes, one could treat the entire air column as a resonant load, observe its linear behavior from frequency-domain experimental data, and propose a modal expansion formulation that characterizes the air column as a series association of second-order ordinary differential equations nonlinearly coupled to a partial differential equation modeling the behavior of the valve [4]. Using a state-space formulation, the valve-resonator coupling used in such framework relies on implicit integration schemes that may cause numerical dispersion and require high computational cost. For sound synthesis purposes, our digital waveguide approach is based on coupling the valve (a nonlinear scattering element) to the resonator via pressure traveling waves. Frequency-domain measurements are used to design an *air column load* input impedance filter model $Z(z)$ (i.e., an input impedance filter) for simulation so that the pressure wave P^- reflected off the air column entrance can be obtained from the incident wave P^+ via

$$P^-(z) = R(z)P^+(z), \tag{1}$$

where $R(z)$ is a digital reflectance model derived from $Z(z)$. The input impedance frequency response

$$Z(\omega) = \frac{P(\omega)}{U(\omega)}, \tag{2}$$

where $P(\omega)$ and $U(\omega)$ respectively correspond to the frequency response of the sound pressure and flow, both at the entrance of the air column. In a previous work [5], a frequency-domain measurement of an air column input impedance is used to construct a discrete-time reflection function $r[n]$ that is suitable for a traveling-wave numerical scheme based on convolution. In that paper, the authors propose a workaround method to evade time-aliasing and other numerical problems that naturally arise from estimating $r[n]$ via inverse Fourier transform of a frequency-domain measurement signal.

This work avoids the aforementioned problems by proposing a methodology for translating an input impedance measurement directly into a recursive digital filter $Z(z)$ of moderately low order, with the added advantage that efficiency is improved with respect to discrete convolution. Moreover, we are interested in using external sound pressure measurements to design a sound pressure radiation filter $E(z)$ able to model how the flow at the entrance of the air column is related to the sound pressure radiated to an external position in the vicinity of the instrument. This paper is an extension of a recent preliminary work [6] were we used the saxophone impedance measurement of a sole fingering position to propose a methodology for designing an impedance parallel filter, and its realization as a reflectance. Here, after taking a full set of measurements including input impedance and sound pressure radiation for all fingering positions, we propose a radiation model in parallel form and revise the reflectance filter formulation to include radiation, leading to a joint reflectance-radiation digital filter formulation with similar properties to those of a recently introduced admittance-radiation model for string instruments [7]. Moreover, we propose a simple model for fingering transitions that is based on linear interpolation of impedance and radiation digital filters. For completeness, this

paper revisits the methodology for designing the impedance filter already introduced in [6].

In a hemi-anechoic space, alto saxophone input impedances were measured using a six-microphone probe calibrated with three non-resonant loads via a least-mean square signal processing technique as described in [8]. Simultaneously, an external measurement microphone was placed near the bell of the instrument to record the radiated sound pressure signal. A sound pressure radiation frequency response $E(\omega)$ was defined in the frequency-domain as

$$E(\omega) = \frac{T(\omega)}{U(\omega)}, \tag{3}$$

where frequency-domain functions $T(\omega)$ and $U(\omega)$ respectively correspond to the radiated sound pressure signal at a point in the external radiation domain (i.e., the signal recorded with the microphone) and the signal of the flow at the entrance of the air column. With this in mind, we aim at constructing a radiation modeling filter $E(z)$ such that the (external) radiated sound pressure $T(z)$ can be obtained from the simulated scalar flow $U(z)$ as $T(z) = E(z)U(z)$.

In Figure 1 we display the magnitude response of some of the measurements, in particular for fingerings E-5 (natural E5), Bb4 (B4-flat), and C#6 (C6-sharp). In the top plots appear the impedance transfer functions, normalized to the characteristic wave impedance of the air column input. As the resonance amplitudes decrease with frequency, the normalized impedance tends to a value of 0 dB, i.e., total transmission. In the bottom plots appear the corresponding radiation transfer functions, where it is possible to observe a shared modal structure with the impedance. This observation motivates the pursuit of a joint formulation for impedance and radiation modeling, and that constitutes the main focus of this work.

The outline is as follows. In Sections 2 and 3 we re-introduce our input impedance model and its optimization-based design technique as it was first described in [6], with slight nomenclature changes that will help in following the rest of the paper. Then, we follow in Section 4 by introducing the sound pressure radiation model. Section 5 provides details on how to jointly realize the input impedance and external radiation models as a common parallel filter in the form of a digital waveguide reflectance. In 6 we present a simple model for emulating the transition between two fingering positions. In Section 7 we briefly describe how to couple the filter to a valve model for efficiently obtaining sound. We conclude in Section 8 by pointing to future experiments and extensions.

2. INPUT IMPEDANCE MODELING

From observation of the resonance structure exhibited by the input impedance and sound radiation measurements, we propose a digital filter formulation akin to the discretization of a modal expansion. Thus, instead of relying on a digital waveguide representation of the air column, we use a different modal structure for each of the F fingering positions analyzed. For each f-th fingering case, we construct an input impedance parallel model $Z|_f(z)$ by creating a basis of $M|_f$ parallel sections each corresponding to a mode, and use the basis over which to project impedance measurements. In the f-th input impedance model, each m-th modal basis parallel section $H|_{f,m}(z)$ is defined as

$$H|_{f,m}(z) = \frac{1 - z^{-1}}{(1 - p|_{f,m} z^{-1})(1 - \bar{p}|_{f,m} z^{-1})}, \tag{4}$$

which corresponds to a one-zero, two-pole resonator with the zero locked at DC. The resonator is defined by a pair of complex conjugate poles $p|_{f,m}$ and $\bar{p}|_{f,m}$, which we relate to the corresponding modal frequency $\nu|_{f,m}$ and bandwidth $\beta|_{f,m}$ (both expressed in Hz) by $2\pi\nu|_{f,m}T_s = \angle p|_{f,m}$ and $\beta|_{f,m} = -\log(|p|_{f,m}|)/\pi$, with T_s being the sampling period. The impedance model $Z|_{f,m}(z)$ is then formulated in parallel as

$$Z|_f(z) = \sum_{m=1}^{M|_f} (b_0|_{f,m} + b_1|_{f,m} z^{-1}) H|_{f,m}(z), \tag{5}$$

where $b_0|_{f,m}$ and $b_1|_{f,m}$ are real-valued coefficients that allow control of both the amplitude and the phase of the the m-th resonator. The main reason behind the choice for our parallel resonator structure is that, while enabling the control of the relative phase between resonators, it imposes a gain of zero at DC irrespective of the coefficients $b_0|_{f,m}$ and $b_1|_{f,m}$. Next we introduce an optimization technique to find the pole positions and numerator coefficients of model (5) given an impedance measurement. For simplicity, in Section 3 we omit the use of the sub-index f for indicating the fingering case, as the methodology presented therein applies to all F fingering cases.

3. INPUT IMPEDANCE FILTER DESIGN

Departing from a target input impedance measurement $\hat{Z}$, the problem of designing the coefficients of the impedance filter model of M digital resonators which approximates the measurement can be stated as the minimization of an error measurement $\varepsilon(Z, \hat{Z})$ between the measurement and the model, with parameters being a vector

$$\boldsymbol{p} = \{p|_1, \cdots, p|_m, \cdots, p|_M\} \tag{6}$$

of complex poles each corresponding to the m-th resonator of the model, and vectors

$$\mathbf{b}_0 = \{b_0|_1, \cdots, b_0|_m, \cdots, b_0|_M\} \tag{7}$$

$$\mathbf{b}_1 = \{b_1|_1, \cdots, b_1|_m, \cdots, b_1|_M\} \tag{8}$$

of respective numerator coefficients. We solve this problem via sequential quadratic programming [9]. At each iteration only pole positions are exposed as the variables to optimize: once they are decided, zeros (i.e., numerator coefficients) are constrained to minimize an auxiliary quadratic cost function, resulting in a simple closed-form solution. The positions of the poles are optimized iteratively: at each step, an error function is successively evaluated by projecting the target frequency response over a basis of frequency responses defined by the pole positions under test. We add a set of linear constraints to guarantee feasibility and to ease convergence. This routine is extended from the filter design technique of [10] as used in [7] to model string instrument input admittances.

3.1. Impedance measurement pre-processing

As it can be observed in the grey curves of Figure 2, the high-frequency region of an impedance measurement typically presents artifacts caused by noise and limitations of the measurement method. It is important to remove those artifacts so that the target normalized impedance effectively tends to 1 as frequency increases. This is needed to help the fitting process in providing an impedance model design for which the normalized impedance also tends to 1 in the

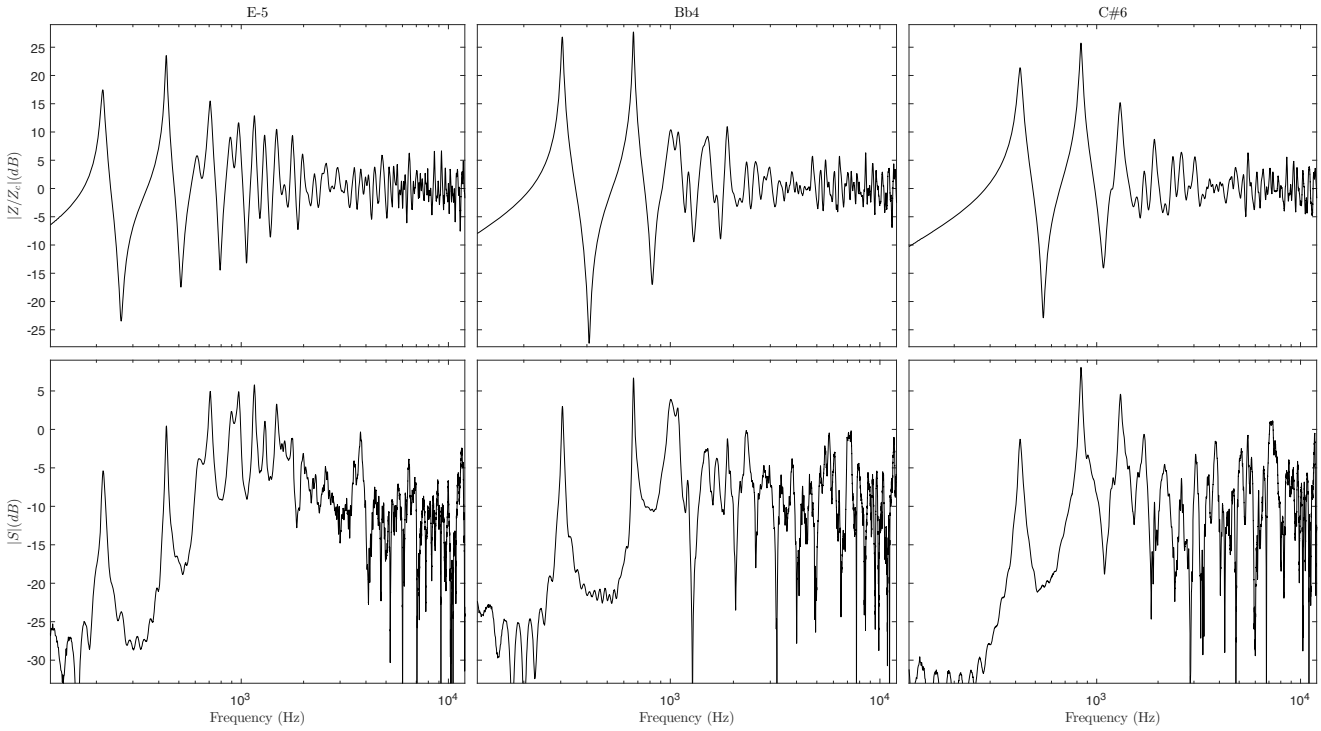


Figure 1: *Magnitude response of impedance (top) and radiation (bottom) measurements for different fingering positions in an alto saxophone.*

high frequency region; otherwise, a derived air column reflectance filter would deliver reflected pressure waves with significant energy around Nyquist, and therefore cause undesired behaviors in the reed-valve nonlinear scattering model. To this end, we perform cross-fading between the normalized impedance measurement and a constant value of one, as illustrated in Figure 2.

3.2. Optimization problem setup

We initialize the model parameters via finding a set of initial pole positions by attending to the magnitude response of the impedance measurement. First, resonance peak selection in the low-frequency region is carried out through an automatic procedure that iteratively rates and sorts spectral peaks by attending to a salience descriptor. For estimating modal frequencies, three magnitude samples (respectively corresponding to the maximum and its adjacent samples) are used to perform parabolic interpolation around selected peaks. For estimating bandwidths, the *half-power* rule [1] is applied using a linear approximation. For the high-frequency region we spread an additional set of poles, uniformly distributed on a logarithmic frequency axis. This leads to a total M modes, each parameterized by a complex pole pair in terms of its angle parameter $w|_m = |\angle p|_m|$ and its radius parameter $s|_m = -\log(1 - |p|_m|)$. This leads to two parameter sets: a set $\boldsymbol{w} = \{w|_1 \cdots w|_m \cdots w|_M\}$ of angle parameter values, and a set $\mathbf{s} = \{s|_1 \cdots s|_m \cdots s|_M\}$ of radius parameter values. With the new parametrization, we state the problem as

$$\begin{aligned} \underset{\boldsymbol{w},\mathbf{s}}{\text{minimize}} \quad & \varepsilon(Z, \hat{Z}) \\ \text{subject to} \quad & \mathbf{C}, \end{aligned} \tag{9}$$

where $\mathbf{C}$ is a set of linear constraints, and numerator coefficients have been left out as they are not exposed as variables in the optimization. A key step before constraint definition is to sort the pole parameter sets so that linear constraints can be defined in a straightforward manner to ensure that the arrangement of poles in the unit disk is preserved during optimization, therefore reducing the number of crossings over local minima. Elements in sets $\mathbf{w}$ and $\mathbf{s}$ are jointly sorted as pairs (each pair corresponding to a complex-conjugate pole) by ascending angle parameter $w|_m$.

From ordered sets $\boldsymbol{w}$ and $\mathbf{s}$, linear constraints $\mathbf{C}$ are defined as follows. First, feasibility is ensured by $0 \leq s|_m$ and $0 \leq w|_m \leq \pi$. Second, to aid convergence we constrain the pole sequence order in set $\boldsymbol{w}$ to be respected. This is expressed by $w|_{m-1} < w|_m < w|_{m+1}$. Moreover, assuming that initialization provides an already trusted first solution, we can bound the search to a region around the initial pole positions. This can be expressed via the additional inequalities $w|_m^- < w|_m < w|_m^+$ and $s|_m^- < s|_m < s|_m^+$, where '$-$' and '$+$' superscripts are used to respectively indicate lower and upper bounds.

3.3. Error estimation

At each i-th step of the optimization, the error $\varepsilon(Z, \hat{Z})$ is estimated as follows. Given K samples of the target impedance frequency response $\hat{Z}(\omega)$ and the set $\mathbf{p}$ of M complex poles defining the modes at the i-th step, numerator coefficient vectors $\mathbf{b}_0$ and $\mathbf{b}_1$ can be obtained via least-squares by

$$\underset{\mathbf{b}}{\text{minimize}} \|\mathbf{H}\mathbf{b} - \hat{\mathbf{z}}\|^2, \tag{10}$$

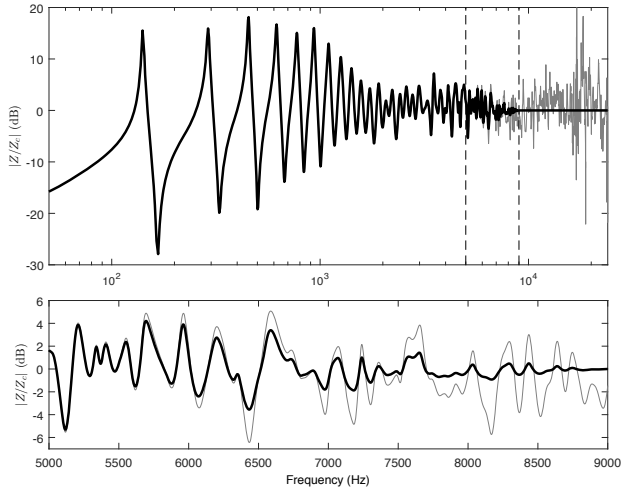


Figure 2: *Magnitude response of an alto saxophone impedance measurement (Bb3 fingering), normalized by the characteristic impedance of the input of the air column. Thin and thick curves are respectively used for raw and pre-processed data. Top: full band, with cross-fading region delimited by vertical lines. Bottom: cross-fading region.*

where $\mathbf{b} = [\,\mathbf{b}_0^T\,\mathbf{b}_1^T\,]^T$ is a real-valued vector; $\hat{\mathbf{z}}$ contains K frequency-domain samples of the impedance measurement $Z(\omega)$ at frequencies $0 \leq \omega_k < \pi$, i.e., $\hat{z}_k = \hat{Z}(\omega_k)$; and $\mathbf{H}$ is a $K \times 2M$ matrix of basis vectors constructed as

$$\mathbf{H} = [\mathbf{h}_0|_1 \cdots \mathbf{h}_0|_m \cdots \mathbf{h}_0|_M\, \mathbf{h}_1|_1 \cdots \mathbf{h}_1|_m \cdots \mathbf{h}_1|_M\,] \quad (11)$$

with column vectors $\mathbf{h}_0|_m$ and $\mathbf{h}_1|_m$ containing the sampled frequency responses of $H|_m(z)$ and $z^{-1}H|_m(z)$ respectively. With numerator coefficients, we evaluate the frequency response of the model and compute the error measure as the l_2-norm of the difference vector, i.e., $\varepsilon(Z, \hat{Z}) = \|\mathbf{H}\mathbf{b} - \hat{\mathbf{z}}\|^2$.

3.4. Final solution

Once poles have been optimized, numerator coefficients of model (5) are found by solving again problem (10). In Figure 3 we display the magnitude and phase responses (top and middle plots) of three example impedance models, respectively obtained from normalized impedance measurements after pre-processing. Although in principle the model (5) is not guaranteed to be positive-real, fitting to measurements of positive-real functions generally provides positive-real designs, as it can be observed from the phase responses. This is important for the stability of the sound synthesis model, as the impedance is going to be realized as a reflectance filter.

4. SOUND PRESSURE RADIATION FILTER

Given the shared modal structure observed in the input impedance and radiation measurements of each fingering, we opt for a radiation model that shares the parallel resonator structure of the impedance model $Z|_f(z)$. We define the sound pressure radiation filter $E|_f(z)$ of the f-th fingering position as

$$E|_f(z) = \sum_{m=1}^{M|_f} (d_0|_{f,m} + d_1|_{f,m} z^{-1}) H|_{f,m}(z), \quad (12)$$

where the $M|_f$ modal basis parallel sections $H|_{f,m}(z)$ are shared with the impedance model (see (4) and (5)), and $d_0|_{f,m}$ and $d_1|_{f,m}$ are real-valued coefficients.

Once the pole positions that define all $H|_{f,m}(z)$ resonators have been found through optimization of the input impedance model $Z|_f(z)$ (see Section 3), numerator coefficients of $E|_f(z)$ are estimated by least-squares. First, in a pre-processing step, all radiation transfer functions are converted to minimum-phase using the real cepstrum [1]. Then, in an analogous manner as for the numerator coefficients of the input impedance model, $d_0|_{f,m}$ and $d_1|_{f,m}$ are arranged into vectors $\mathbf{d}_0|_f$ and $\mathbf{d}_1|_f$ as in (7), (8) and found by solving

$$\underset{\mathbf{d}|_f}{\text{minimize}} \|\mathbf{H}|_f \mathbf{d}|_f - \hat{\mathbf{e}}|_f\|^2, \quad (13)$$

where $\mathbf{d}|_f = [\,\mathbf{d}_0|_f^T\,\mathbf{d}_1|_f^T\,]^T$ is a real-valued column vector; $\hat{\mathbf{e}}|_f$ contains K frequency-domain samples of the radiation measurement $E|_f(\omega)$ at frequencies $0 \leq \omega_k < \pi$, i.e., $\hat{e}_k|_f = \hat{E}|_f(\omega_k)$; and $\mathbf{H}|_f$ is the $K \times 2M$ matrix of basis vectors in (11) that was used for solving the impedance projection problem (13) corresponding to the f-th fingering case. In Figure 3 we display the magnitude responses (bottom plots) of three example radiation models, along with their corresponding impedance models, overlayed on the measurements. A similar quality of approximation was also observed in all other fingering positions.

5. JOINT REALIZATION AS A WAVEGUIDE TERMINATION

From the input impedance model (5), we construct a reflectance that keeps the state of the air column as a resonating element, and allows us to obtain reflected waves from its interface. The formulation that we propose involves the computation of the flow as an intermediate step, therefore allowing us to obtain the external radiated sound pressure as $T(z) = E(z)U(z)$ via model (12). Since both the impedance model and the sound pressure radiation model are constructed so that they share the exact same set of parallel resonators, obtaining the radiated sound comes at a very low additional cost. Thus, via a single set of $M|_f$ resonators corresponding to the f-th fingering position, we are able to model pressure wave reflectance, radiated sound pressure, and (implicitly) energy loss from input transmittance to non-radiating modes and dissipation.

5.1. Reflectance realization

Following the digital waveguide formulation for loaded parallel junctions [1], we can compute the scalar flow $U(z)$ at the input of the air column solely from the input pressure wave $P^+(z)$ as

$$U(z) = \frac{2Y_c P^+(z)}{1 + Y_c Z|_f(z)} \quad (14)$$

where Y_c is the characteristic admittance of the input of the air column, and $Z|_f(z)$ is the input impedance model corresponding to the f-th fingering position. From the flow $U(z)$, it should be

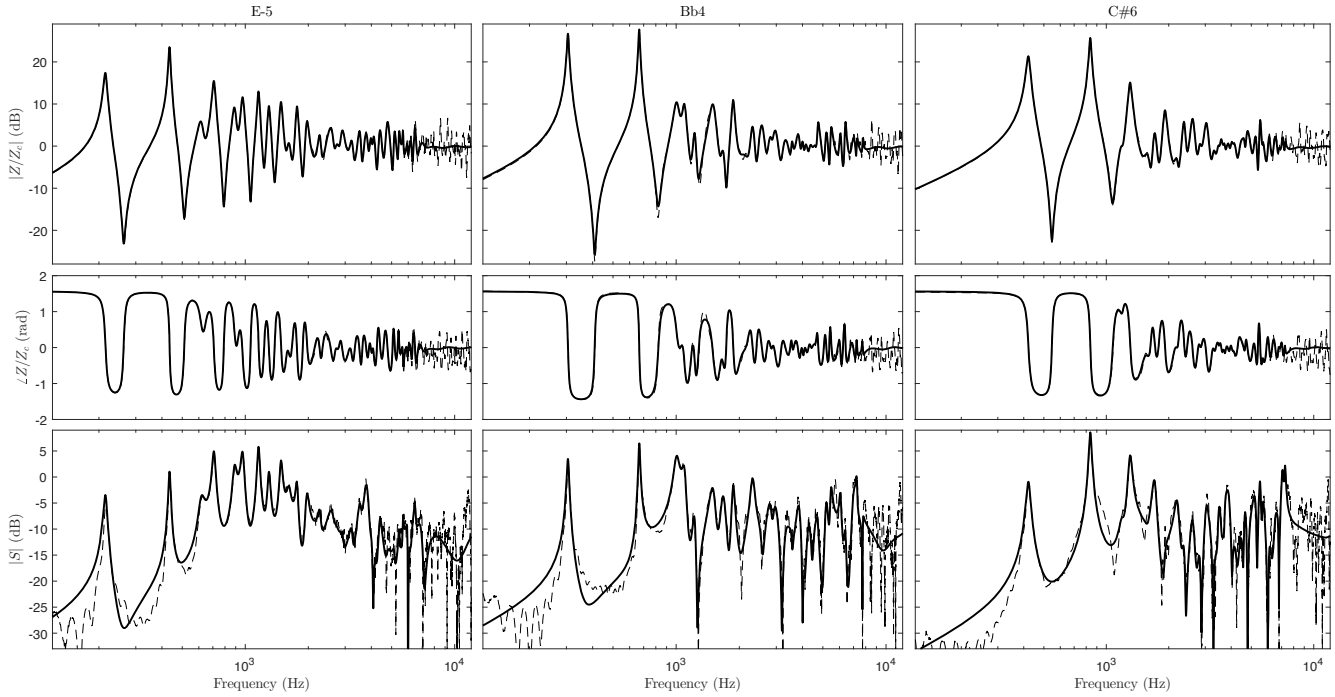


Figure 3: *Example impedance and radiaton models. All three fingerings were modeled with $M = 32$ parallel sections each. From top to bottom: impedance magnitude response, impedance phase response, and radiation magnitude response. In each plot, dashed lines and thick lines are used to depict the measurement and the model respectively.*

straightforward to compute the scalar pressure $P(z)$ at the input of the air column via

$$P(z) = Z|_f(z)U(z). \tag{15}$$

Finally, from the air column pressure $P(z)$ it is possible to obtain the (reflected) outgoing pressure wave $P^-(z)$ by means of

$$P^-(z) = P(z) - P^+(z). \tag{16}$$

Because the formulation of the model (5) presents a parallel structure that we want to maintain, inverting $Z|_f(z)$ as it appears in equation (14) is impractical. To overcome this problem in the realization of the reflectance, we reformulate the impedance in a similar manner as we did for the input admittance of string instruments [7] (inspired by [11]). First, we rewrite each resonator $H|_{f,m}(z)$ of equation (5) as

$$H|_{f,m}(z) = 1 + z^{-1}H_p|_{f,m}(z), \tag{17}$$

with

$$H_p|_{f,m}(z) = \frac{c_0|_{f,m} + c_1|_{f,m}z^{-1}}{1 + a_1|_{f,m}z^{-1} + a_2|_{f,m}z^{-2}}, \tag{18}$$

$c_0|_{f,m} = -1 - a_1|_{f,m}$, and $c_1|_{f,m} = -a_2|_{f,m}$. Note that denominator coefficients are related to pole radius and angle by $a_1|_{f,m} = -2|p|_{f,m}|\cos(\angle p|_{f,m})$ and $a_2|_{f,m} = |p|_{f,m}|^2$. We now can rewrite the impedance model as

$$Z|_f(z) = B_0|_f + z^{-1}B_1|_f + z^{-1}H_0|_f(z) + z^{-2}H_1|_f(z), \tag{19}$$

with

$$B_0|_f = \sum_{m=1}^{M|_f} b_0|_{f,m}, \quad B_1|_f = \sum_{m=1}^{M|_f} b_1|_{f,m}, \tag{20}$$

$$H_0|_f(z) = \sum_{m=1}^{M|_f} b_0|_{f,m}H_p|_{f,m}(z), \tag{21}$$

$$H_1|_f(z) = \sum_{m=1}^{M|_f} b_1|_{f,m}H_p|_{f,m}(z). \tag{22}$$

With this new formulation, we rewrite (14) and (15) as

$$U(z) = \frac{2Y_cP^+(z) - z^{-1}Y_cV|_f(z)U(z)}{1 + Y_cB_0|_f} \tag{23}$$

and

$$P(z) = B_0|_fU(z) + z^{-1}V|_f(z)U(z), \tag{24}$$

where

$$V|_f(z) = B_1|_f + H_0|_f(z) + z^{-1}H_1|_f(z). \tag{25}$$

It is important to notice that now the parallel structure appears in the numerator terms $H_0|_f(z)$ and $H_1|_f(z)$ as part of $V|_f(z)$, making possible its implementation. Moreover, $H_0|_f(z)$ and $H_1|_f(z)$ can be jointly implemented as a sole bank of parallel resonators. Finally, it is worth mentioning that the term $z^{-1}V|_f(z)U(z)$ appears in both equations (23) and (24) but does not need to be implemented twice–once it has been computed to obtain $U(z)$ via equation (23), it can be reused to compute $P(z)$ via equation (24).

5.2. External radiation realization

For the realization of the external radiation model, we take advantage of the fact that the flow $U(z)$ is available as an intermediate step in the computation of the reflected pressure vave $P^-(z)$. Using the decomposition described in (21) for each of the common resonators $H|_{f,m}(z)$, we rewrite the f-th radiation model $E|_f(z)$ in (12) as

$$E|_f(z) = D_0|_f + z^{-1}D_1|_f + z^{-1}L_0|_f(z) + z^{-2}L_1|_f(z), \quad (26)$$

with

$$D_0|_f = \sum_{m=1}^{M|_f} e_0|_{f,m}, \quad D_1|_f = \sum_{m=1}^{M|_f} e_1|_{f,m}, \quad (27)$$

$$L_0|_f(z) = \sum_{m=1}^{M|_f} e_0|_{f,m} H_p|_{f,m}(z), \quad (28)$$

$$L_1|_f(z) = \sum_{m=1}^{M|_f} e_1|_{f,m} H_p|_{f,m}(z). \quad (29)$$

With this, the radiated sound pressure signal $T(z)$ is computed as

$$T(z) = \big(D_0|_f + z^{-1}D_1|_f + z^{-1}L_0|_f(z) + z^{-2}L_1|_f(z)\big)U(z). \quad (30)$$

Please note that all four terms $H_0|_f(z)$, $H_1|_f(z)$, $L_0|_f(z)$, $L_1|_f(z)$ share inputs and parallel structure: each resonator $H_p|_{f,m}(z)$ is present in all four expressions (21), (22), (28), (29) and driven by the flow signal $U(z)$. Therfore, only one bank of $M|_f$ resonators needs to be implemented for the joint realization of the f-th fingering reflectance and external radiation models.

6. MODEL MIXING FOR FINGERING TRANSITIONS

So far, we have treated the impedance and radiation models of each f-th fingering as two parallel structures sharing a bank of resonators. Then we have derived a joint reflectance-radiation filter that simultaneously implements both models and can be interfaced to a reed model as a loaded waveguide termination. Such f-th termination filter replicates the behavior of the air column as observed during the f-th measurement. This means that for each fingering position we have a different termination filter, and in the context of sound synthesis this creates a fundamental problem: how to swap filters when a fingering transition happens? To avoid such an abrupt, non-physical operation we propose to reformulate our air column model as follows.

We define a sole impedance model $Z(z)$ that accounts for all F fingerings simultaneously, via a linear combination of all F single-fingering impedance models. This is expressed as

$$Z(z) = \sum_{f=1}^{F} w|_f Z|_f(z), \quad (31)$$

where $w|_f$ are mixing weights. Assuming that all F models $Z|_f(z)$ are positive-real, we guarantee that the multi-fingering input impedance model $Z(z)$ will be positive-real if all mixing weights are non-negative. With this, $Z(z)$ will lead to a passive termination irrespective of the weights applied in the linear combination. Thus, since fingering transitions are expected to happen at a sufficiently slow speed so that during each simulation step the whole system can be assumed to be quasi-static, a simple time-varying linear mixing of any two impedance models can be used for a smooth, stable transition between fingerings.

For the external radiation model we apply the same idea, leading to a sole radiation model

$$E(z) = \sum_{f=1}^{F} w|_f E|_f(z), \quad (32)$$

where $w|_f$ match those used for impedance mixing. Now it is straightforward to rewrite the expressions for the joint reflectance-radiation realization. First, the impedance model is written as

$$Z(z) = B_0 + z^{-1}B_1 + z^{-1}H_0(z) + z^{-2}H_1(z), \quad (33)$$

where each of the terms is simply a linear combination of each of single-fingering terms in (20) through (22), leading to

$$B_0 = \sum_{f=1}^{F} w|_f B_0|_f, \quad B_1 = \sum_{f=1}^{F} w|_f B_1|_f, \quad (34)$$

$$H_0(z) = \sum_{f=1}^{F} w|_f H_0|_f(z), \quad (35)$$

$$H_1(z) = \sum_{f=1}^{F} w|_f H_1|_f(z). \quad (36)$$

With this, we also rewrite (23) and (24) as

$$U(z) = \frac{2Y_c P^+(z) - z^{-1}Y_c V(z)U(z)}{1 + Y_c B_0} \quad (37)$$

and

$$P(z) = B_0 U(z) + z^{-1}V(z)U(z), \quad (38)$$

where

$$V(z) = B_1 + H_0(z) + z^{-1}H_1(z). \quad (39)$$

An analogous transformation is applied to the radiation part of the model, leading to

$$E(z) = D_0 + z^{-1}D_1 + z^{-1}L_0(z) + z^{-2}L_1(z), \quad (40)$$

where

$$D_0 = \sum_{f=1}^{F} w|_f D_0|_f, \quad D_1 = \sum_{f=1}^{F} w|_f D_1|_f, \quad (41)$$

$$L_0(z) = \sum_{f=1}^{F} w|_f L_0|_f(z), \quad (42)$$

$$L_1(z) = \sum_{f=1}^{F} w|_f L_1|_f(z), \quad (43)$$

and the radiated sound pressure is again computed via

$$T(z) = \big(D_0 + z^{-1}D_1 + z^{-1}L_0(z) + z^{-2}L_1(z)\big)U(z). \quad (44)$$

In Figure 4 we display the input impedance magnitude, input impedance phase, and external radiation magnitude response of the model during a transition from E-5 to Bb4. It is worth noting that, although the complete model will in principle be constructed from all F resonators banks, its run-time operation logic can be implemented as follows: when no transition is happening, one bank of resonators is active; during a transition, two resonator banks are active.

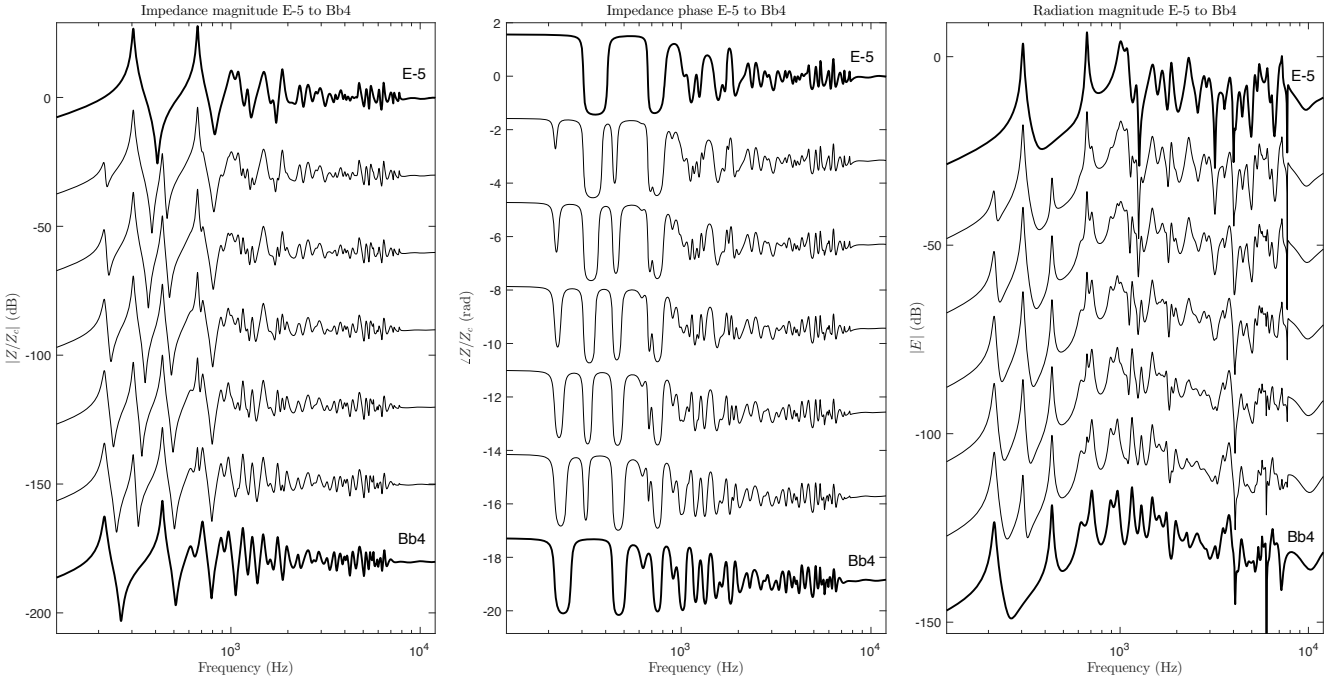


Figure 4: *Impedance and radiation model responses during a fingering transition, from E-5 to Bb4 positions. A linear mixing of 5 steps is performed from the corresponding impedance and radiation models, with $M = 32$ parallel sections each. Thick lines are used to depict the original E-5 (top) and Bb4 (bottom) models, and thin lines are used for the intermediate models. For clarity, impedance magnitude responses, impedance phase responses, and radiation magnitude responses were respectively offset by -30 dB, π radians, and -20 dB per step.*

7. DIGITAL WAVEGUIDE SOUND SYNTHESIS

We construct an efficient sound synthesis scheme by interfacing our joint reflectance-radiation model and a modified version of the digital waveguide reed scattering model used in [12] as follows. At each iteration, two main computations are interleaved: the reed scattering update and the air column reflectance update. During the reed scattering update, the differential pressure driving both the reed motion and the reed channel flow relation (see [12]) is first computed as the difference between the mouth pressure and the value of the scalar air column pressure obtained in the previous reflectance update (see Section 5.1). Then, the pressure wave obtained from the reed scattering is used to feed the next reflectance update. For an average of 32 resonators per fingering, a sampling frequency of 48 kHz, and fingering transitions sparsely happening for about 10% of the simulated time, this model runs at a speed above 30 times faster than real-time in one core of a laptop computer.

In Figure 5 we display the control signals (mouth pressure, fingering weights) and radiated sound of a synthesis example involving two fingering transitions: Bb4 to E-5 and E-5 to A-4, respectively happening at around 1.4 and 2.0 seconds. The first of these transitions involves nominal regimes in both fingerings, while for the second case the high mouth pressure drives the system into its higher octave regime after the transition. With respect to the transition happening at around 0.6 seconds, it does not involve any fingering change but is caused by the system falling from its higher-octave regime to its nominal regime. In Figure 6 we display the reed channel flow (see [12]), the air column input pressure, and the radiated sound during the Bb4 to E-5 transition of the example in Figure 5. The synthetic radiated sound corresponding to this example can be heard online[1].

8. OUTLOOK

Albeit still exploratory and in need of a thorough calibration via automated playability analysis, our results open a promising route for efficient, yet realistic sound synthesis of wind instrument sound with potential applications both in rendering music and in analyzing the timbre and playability of real air column prototypes. Besides the application of this method to modeling other wind instrument air columns, a clear next stage of development involves the use of more sophisticated reed representations, and also the exploration of lip-driven excitation models. Perhaps through subjective tests, it is still necessary to investigate the effects of using more (or less) resonators per fingering, and also different fingering weight profiles. Perhaps through measurements, we could elucidate how well the cross-fading of models during fingering transitions simulates the actual case. Another of the extensions that we are considering involves coupling this model to a vocal tract model also realized as a reflectance that is interfaced to the valve model.

9. REFERENCES

[1] Julius O. Smith, *Physical Audio Signal Processing*, W3K Publishing, 2004, https://-

[1] http://ccrma.stanford.edu/~esteban/wind/dafx2018.wav

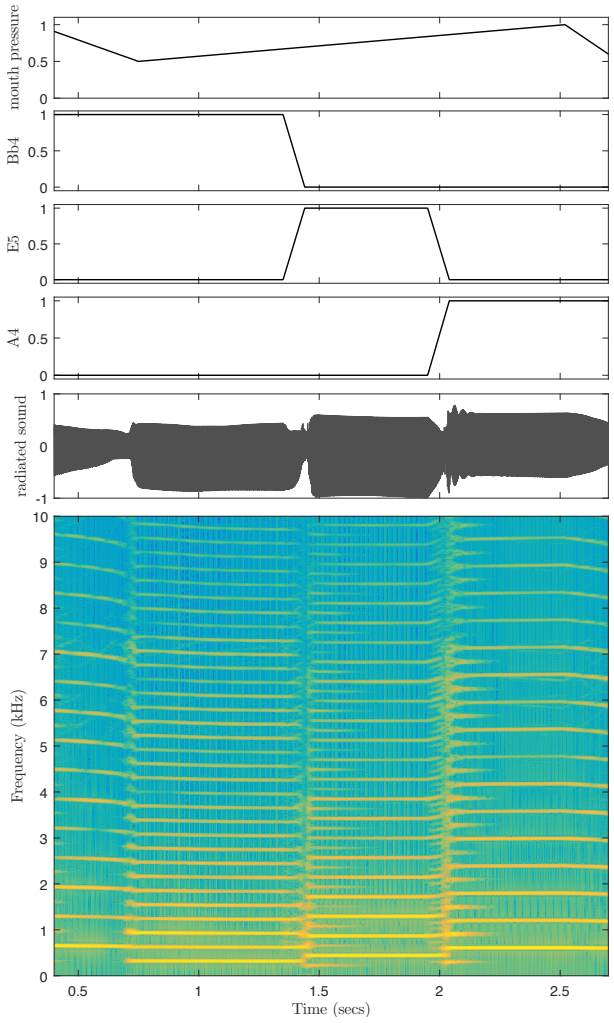


Figure 5: *Sound synthesis example, with $M = 32$ parallel sections per fingering. From top to bottom: mouth pressure (normalized units), weights w_f corresponding to Bb4, E-5, and A-4 fingerings, radiated sound pressure (normalized units), and a spectrogram of the radiated sound pressure.*

ccrma.stanford.edu/~jos/pasp.

[2] David P. Berners, *Acoustics and Signal Processing Techniques for Physical Modeling of Brass Instruments*, Ph.D. thesis, Stanford University, 1999, https://-ccrma.stanford.edu/~dpberner/.

[3] G. Scavone, "Time-domain synthesis of conical bore instrument sounds," in *Proc. of the International Computer Music Conference*, 2002.

[4] F. Silva, C. Vergez, P. Guillemain, J. Kergomard, and V. Debut, "Moreesc: a framework for the simulation and analysis of sound production in reed and brass instruments," *Acta Acustica united with Acustica*, vol. 100(1), pp. 126–138, 2014.

[5] B. Gazengel, J. Gilbert, and N. Amir, "From the measured input impedance to the synthesis signal: where are the traps?," *Acta Acustica*, vol. 3, pp. 445–472, 1995.

[6] E. Maestre, J. O. Smith, and G. P. Scavone, "Analysis-synthesis of saxophone input impedances via recursive parallel filters," in *Proc. of the International Symposium on Musical Acoustics*, 2017.

[7] E. Maestre, G. P. Scavone, and J. O. Smith, "Joint modeling of bridge admittance and body radiativity for efficient synthesis of string instrument sound by digital waveguides," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 25:5, pp. 1128–1139, 2017.

[8] A. Lefevbre and G. P. Scavone, "A comparison of saxophone impedances and their playing behavior," in *Proc. of the Forum Acusticum Conference*, 2011.

[9] J. Nocedal and S. J. Wright, *Numerical Optimization*, Springer, 2006.

[10] E. Maestre, G. P. Scavone, and J. O. Smith, "Design of recursive digital filters in parallel form by linearly constrained pole optimization," *IEEE Signal Processing Letters*, vol. 23:11, pp. 1547–1550, 2016.

[11] M. Karjalainen, "Efficient realization of wave digital components for physical modeling and sound synthesis," *IEEE Transactions on Audio, Speech, and Language Process.*, vol. 16:5, pp. 947–956, 2008.

[12] G. P. Scavone and J. O. Smith, "A stable acoustic impedance model of the clarinet using digital waveguides," in *Proc. of the International Conference on Digital Audio Effects*, 2006.

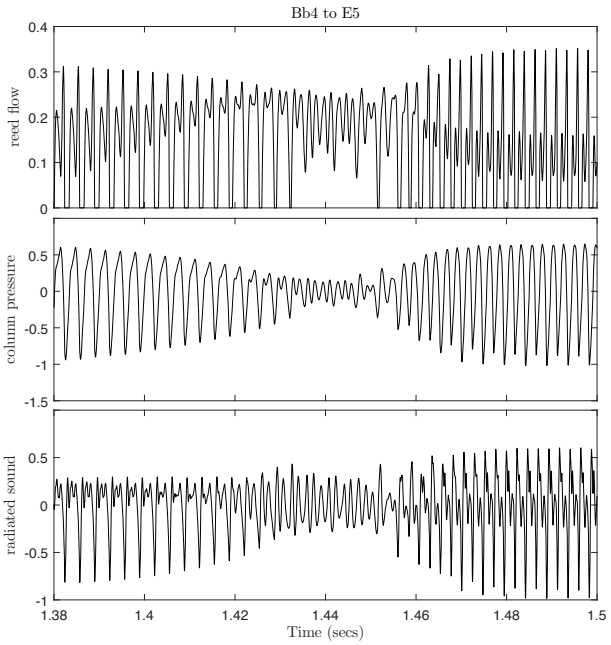


Figure 6: *Detail of the transition beween Bb4 to E-5. From top to bottom, in normalized units: reed flow, air column input pressure, radiated sound pressure.*

INTERPRETATION AND CONTROL IN AM/FM-BASED AUDIO EFFECTS

Antonio José Homsi Goulart

Computer Music Research Group
University of São Paulo
São Paulo, Brazil
ag@ime.usp.br

Marcelo Queiroz

Computer Music Research Group
University of São Paulo
São Paulo, Brazil
mqz@ime.usp.br

Joseph Timoney

Sound and Digital Music Technology Group
Maynooth University
Maynooth, Ireland
joseph.timoney@mu.ie

Victor Lazzarini

Sound and Digital Music Technology Group
Maynooth University
Maynooth, Ireland
victor.lazzarini@mu.ie

ABSTRACT

This paper is a continuation of our first studies on AM/FM digital audio effects, where the AM/FM decomposition equations were reviewed and some exploratory examples of effects were introduced. In the current paper we present more insight on the signals obtained with the AM/FM decomposition, intending to illustrate manipulations in the AM/FM domain that can be applied as interesting audio effects. We provide high-quality AM/FM effects and their implementations, alongside a brief objective evaluation. Audio samples and codes for real-time operation are also supplied.

1. INTRODUCTION

In previous papers [1] [2] we presented our first studies on AM/FM Digital Audio Effects. The AM/FM decomposition enables an analysis-processing-resynthesis approach to work with audio signals. Effects can be implemented by manipulating signals obtained with a decomposition scheme that unravels the original audio time-based representation to an analogous representation based on a pair of new time-based signals with complementary information.

The AM/FM decomposition was firstly adopted in musical processing for synthesis purposes [3], where similarities with the FM synthesis [4] where drawn. More recently, a lot of research was devoted to AM/FM for speech analysis, in the area of works known as Modulation Filtering [5] [6] [7]. Back to the context of music signal processing, in the Modulation Vocoder series of works [8] [9] [10] the decomposition was adopted in order to explore applications like audio codification (compression), control of roughness of audio signals, and pitch transposition. AM/FM decomposition was also used as an extension of sinusoidal modelling for audio analysis/synthesis purposes [11].

In [1] the impact of smoothing in the AM/FM domain was assessed considering different configurations of smoothers and different psychoacoustics metrics. Then, in [2] we investigated effects based on manipulating the AM/FM domain signals with well-known time-domain manipulations of well established effects like octaver, chorus, wah-wah, etc. In the present paper, Section 2 will briefly review the AM/FM Hilbert-based decomposition with an intuitive explanation about the technique, so that manipulations in the AM/FM domain can lead to the design of interesting audio effects. In Section 3 AM/FM effects will be presented alongside audio examples that are available for downloading[1], where the reader will be able to assess the quality of the effects obtained from this study. In Section 4 a brief evaluation of the effects, based on comparisons using audio descriptors, will be presented. Finally, we conclude and point our current and future work. Audio files will be referenced in the paper with the symbol [►filename].

2. AM/FM DECOMPOSITION

2.1. Envelope and instantaneous frequency

The idea behind the AM/FM decomposition is to understand the input signal as a single sinusoidal tone modulated both in amplitude (AM) and frequency (FM). Given an input signal $x(t)$, we want to find a pair of functions $(a(t), f(t))$ such that

$$x(t) = a(t)\cos\left(\int_0^t f(\tau)d\tau\right). \quad (1)$$

The amplitude modulator signal $a(t)$ is estimated with the decomposition as an envelope of the input signal, and the frequency modulator signal $f(t)$ is estimated as the instantaneous frequency (IF) of the input signal. In order to apply audio effects we might process $a(t)$ or $f(t)$ and consider the altered versions $a_{FX}(t)$ and $f_{FX}(t)$ in a resynthesis process

$$x_{FX}(t) = a_{FX}(t)\cos\left(\int_0^t f_{FX}(\tau)d\tau\right). \quad (2)$$

Notice that the argument for the cosine in Eq. 1 is the instantaneous phase, which is the integral of the instantaneous frequency. This is tied to the concept of a phasor (Figure 1), in which the phase (current angle) is given by increments from an initial position (initial angle) in the unit circle.

The increments in the phase are represented by the integral in Eq. 1. If $a(t)$ and $f(t)$ are constant, we will have equal steps around the circle (unit circle if $a(t) = 1, \forall t$), and thus a sinusoid will be obtained with the projection of the phasor onto the x axis (as in Fig. 1). However, if $a(t)$ or $f(t)$ vary, a different kind of signal will be obtained. This leads us to a useful interpretation for the IF, understanding this value as the frequency of a sinusoid that locally (at each time instant t) fits the original signal $x(t)$ [12].

[1] https://www.ime.usp.br/~ag/dl/dafx18.zip

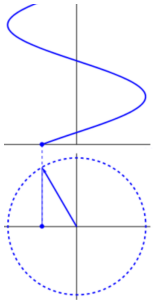


Figure 1: *A regular phasor.*

It is interesting to notice the local aspect of the instantaneous frequency, estimated from an infinitesimal neighborhood of each sample, as opposed to frequencies of sinusoidal components present in the signal spectrum, which have a global scope (the analysis window). It should be noted that the IF might even not be present in the spectrum of a signal, and is sometimes higher than the highest component present in a signal [13]. While additive synthesis [14] would provide a classic example of how to think globally about a signal, there are many situations where a local/instantaneous model of the signal is more appropriate, e.g. in the operation of adaptive devices such as limiters, which measure/change signal values constantly within a feedback loop (lacking knowledge of the whole process) [15]. In order to grasp a good intuition for developing AM/FM effects, this local information is the main concern of this study.

2.2. Ambiguity in the decomposition

Different techniques are available for obtaining an AM/FM decomposition, and as we are unraveling a single signal to a combination of two other signals, an inherent ambiguity permeates this decomposition. Given an input signal $x(t)$ we can find both

$$x(t) = a(t)\cos\left(\phi(t)\right) \tag{3}$$

and

$$x(t) = b(t)\cos\left(\theta(t)\right) \tag{4}$$

in such a way that $b(t) \neq a(t)$ and $\theta(t) \neq \phi(t)$ [13]. Actually we might think of two extreme (and undesirable) cases for the decomposition:

- $a(t) = x(t), \quad \forall t$; and
  $\phi(t) = 0 \rightarrow \cos\left(\phi(t)\right) = 1, \quad \forall t$;
  in this case all the information is coded in the AM portion. The resynthesis would be represented by a pure amplitude modulation, as the cosine value would be constant;
- $a(t) = 1, \quad \forall t$; and
  $\phi(t) = \cos^{-1}\left(x(t)\right), \quad \forall t$;
  in this case the information would go to the FM portion of the decomposition. The resynthesis would be a pure frequency modulation, as the envelope would be constant.

In the development of audio effects we are not really interested in these extreme cases, for in such cases we could work directly on the original time-domain signal. What we usually want is a decomposition that allocates non-trivial information both to the AM and FM portions, so we can develop processing routines that will bring interesting modifications to the dry signal after resynthesis.

2.3. Implementation

In our work we focused on the analytic signal based decomposition, although other techniques, for instance based on energy separation [16] [17] [18] are also available. The analytic signal is a complex signal without any negative frequency components [19]. Given a real signal, its analytic counterpart shows a similar spectrum considering the positive frequencies, but a null contribution from the negative frequencies. For instance, a regular sinusoidal signal $\cos\left(\omega_0 t\right)$ contains two components in its spectrum, localized at $+\omega_0$ and $-\omega_0$ [20]:

$$\cos\left(\omega_0 t\right) = \frac{1}{2}\left(e^{i\omega_0 t} + e^{-i\omega_0 t}\right). \tag{5}$$

Notice that if we consider only the positive component we get the regular phasor represented in Fig. 1 as the analytic signal for $\cos\left(\omega_0 t\right)$.

Now, by eliminating the negative components of any real signal $x(t)$ we get its analytic signal $z(t)$ as

$$z(t) = \frac{1}{2\pi}\int_0^{+\infty} X(\omega)e^{i\omega t}d\omega, \tag{6}$$

where $X(\omega)$ is the Fourier Transform of $x(t)$ [20]. Eq. 6 can be thought as a superposition of an infinite number of phasors, each of them spinning with its own frequency ω and radius $X(\omega)$. Figure 2 represents this view, considering three phasors; the projection onto the x axis gives the original signal $x(t)$.

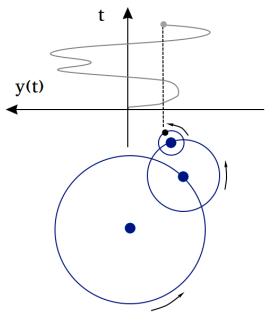


Figure 2: *Superposition of three phasors with different frequencies and radii.*

Considering this negative frequency components elimination perspective, one of the possibilities to obtain the analytic signal is via the Fourier Transform: we can transform the real signal $x(t)$, attribute zero to the negative portion of the spectrum, then apply the inverse transform to obtain $z(t)$ [21]. Another possibility is by applying the Hilbert Transform to $x(t)$, which gives $\hat{x}(t)$, a quadrature version of $x(t)$, where all the components are shifted by 90° [22]. We then can build the analytic signal as

$$z(t) = x(t) + i\hat{x}(t), \tag{7}$$

where $i = \sqrt{-1}$.

The AM/FM decomposition is a matter of finding the envelope and the instantaneous frequency of the analytic signal. Notice that $z(t)$ can be written as

$$z(t) = x(t) + i\hat{x}(t) = a(t)e^{i\phi(t)}, \tag{8}$$

where the envelope is given by

$$a(t) = \sqrt{x^2(t) + \hat{x}^2(t)} = |z(t)|, \tag{9}$$

the instantaneous phase is given by

$$\phi(t) = \arctan\left(\frac{\hat{x}(t)}{x(t)}\right), \tag{10}$$

and by differentiating this quantity we obtain the IF

$$f(t) = \dot{\phi}(t) = \frac{x(t)\dot{\hat{x}}(t) - \dot{x}(t)\hat{x}(t)}{x^2(t) + \hat{x}^2(t)}. \tag{11}$$

Eq. 8 helps us visualize the relation between the concepts of analytic signal, envelope and instantaneous phase and frequency. In the case of a sinusoidal $x(t)$ we will have $z(t)$ as a simple harmonic motion, with constant radius and frequency, so the increments in the angle are always the same. However, for a more general $x(t)$, $z(t)$ will exhibit unequal increments, i.e., in each sample the angle covered around the circle will not be the same; likewise, the radius will not be constant, so we will have a movement that alternates between shrinking and expanding spirals. Notice that the projection of this movement onto the x axis generates $x(t)$.

3. NEW AM/FM DAFX

In order to apply AM/FM effects we must modify $a(t)$ and/or $f(t)$ and then proceed to a resynthesis process. For the modifications we will be dealing with filters, compressors/expanders, and modulators. As these modifications depend on the choice of values for parameters like thresholds, cut-off frequencies, etc., it is important to have an idea about the ranges involved in the original signal. The values will then be chosen according to musical intentions.

In this paper our examples will be focused on a short guitar phrase consisting of a bend and a vibrato[2]. Its varying fundamental frequency represents an important test for the AM/FM decomposition. The waveform in shown in Fig. 3; the envelope and instantaneous frequency signals estimated with the method described in Section 2 are shown in Figures 4 and 5, respectively. Notice that extreme values for the IF are typically associated with the occurrence of very small values in the envelope (the denominator in Eq. 11 is essentially the envelope squared). Figure 6 shows a zoomed view of the IF, up to the value of 1200 Hz, showing that for the length of the signal the IF shows a trend going from (around) 700 Hz to (around) 500 Hz, contaminated with spikes at instants where the envelope values fade out.

The audio file [►resynth-hilb] was created with an AM/FM analysis-resynthesis process, i.e., no manipulations on the envelope and instantaneous frequency were applied. Notice that the analysis is transparent, i.e., the reconstructed signal is identical to the original audio file ([►bend-vibrato]).

3.1. Effects based on filtering

By filtering signals, we are choosing which components will remain unaltered and which will be amplified or attenuated to some extent [23]. For instance, by not allowing high frequencies in the signal we will prevent fast variations to occur, but slow fluctuations remain unaltered.

[2]In the audio files provided, other examples with different musical instruments and phrases are also available.

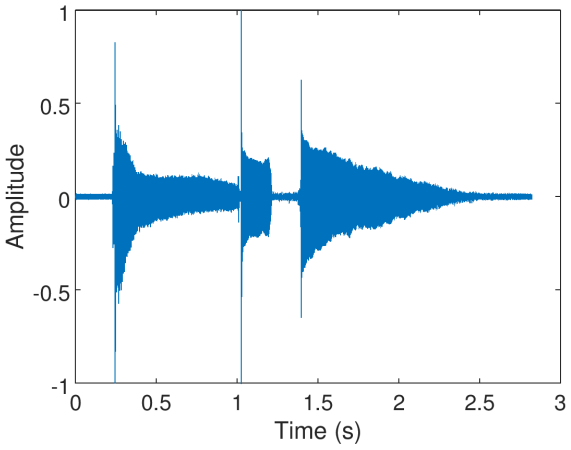


Figure 3: *Waveform of guitar phrase.*

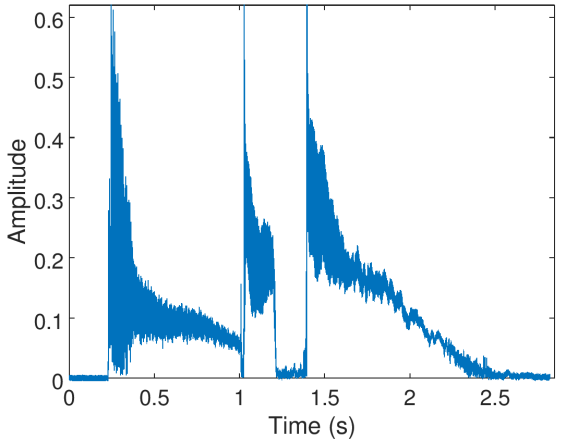


Figure 4: *Envelope of guitar phrase.*

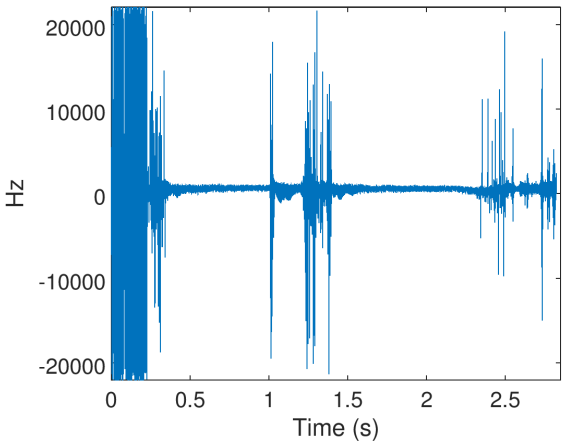


Figure 5: *Instantaneous frequency of guitar phrase.*

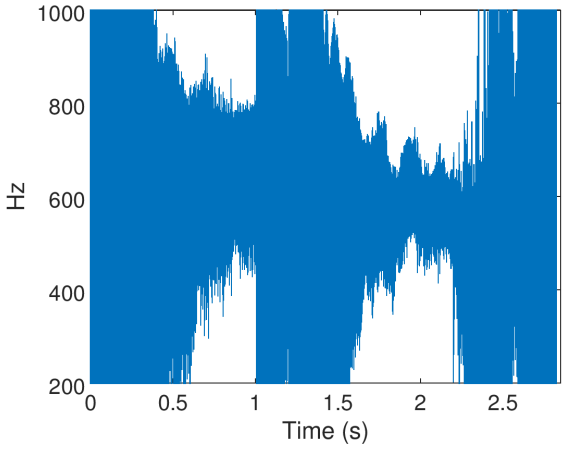


Figure 6: *Instantaneous frequency of guitar phrase (zoom).*

However, modifications in the IF of signals generally have a different meaning. If we, for instance, low-pass filter an IF signal, we will prevent sharp transitions occurring in the IF signal. This will result in a muffled sound, with less articulation, as the IF signal will only keep its slow variations. Thus, in the resynthesis, the phasor will have its increments constrained. We can check this effect by comparing the audio files [►lowp-if-1000] (cut-off frequency set at 1 KHz) and [►lowp-if-500] (cut-off at 500 Hz) with the dry signal [►resynth-hilb]. Fig. 6 shows that the IF values lie around 600 Hz, so the cut-off at 1 KHz will not cause a huge impact, but at 500 Hz it will[3].

Notice that the range of variations in the IF signal is not the matter here, but the frequency with which they occur. Large variations will still exist, but will happen slowly over time. An interesting effect can be achieved by setting the cut-off at extreme values, e.g. 1 Hz. The file [►lowp-if-1] will reveal a chirp, because the sweeping through the IF signal will happen with a limited speed.

By low-pass filtering the envelope component, a different kind of effect is achieved. High frequencies in dynamics are related to percussive sounds, which brings the sensation of the onset of a sound. So, by limiting the envelope only to low frequencies, sounds with a smooth onset, resembling a bowed violin, are obtained, as in the file [►env-lowpf-10] (Butterworth low-pass filter with cut-off frequency at 10 Hz, applied to the envelope).

3.2. Effects based on dynamics processing

Instead of acting on the range of frequencies present in a signal, the manipulation of the dynamics of the envelope and IF signals imparts a selection of the actual values that we allow for these signals. For instance, if we use a limiter to prevent values for the IF higher than an specific threshold, we will prevent, in the resynthesis, angle increments higher than this threshold. In such a way, we can condition the excursion of the signals within a desired range.

Depending on the configurations, similar results can be obtained by using distortion, limiting, or compression [23]. These effects are all used to attenuate large values (higher than a threshold) in the input signal, differing only in the way they operate.

[3]Check also the audio examples considering different instruments (they come in separate folders and are named using the same convention).

Dynamics processing of the envelope or the audio signal itself will have similar results, but working on the IF signal brings interesting musical applications. For instance, as we know (Fig. 6) our IF values lie around 600 Hz, so applying a distortion with a threshold at 400 Hz ([►if-ortion-400]) will result in a sound similar to the one obtained by changing the IF value to a constant equal to 400 Hz ([►fix-if-400]). This will lead, in this example, to a perception of a drone note between a G4 and a G♯4.

3.3. Effects based on modulation

Another family of effects that we will describe is based on altering the value of the IF signal with a LFO (Low Frequency Oscillator) approach. We can both ring modulate the IF, i.e. directly multiply it with a modulator, or apply classic amplitude modulation instead, where the modulation will occur around the IF signal (an offset is added to the modulator signal) [23].

The former case represents the possibility for a very aggressive effect. As we saw in the previous sections, acting on the IF will probably result in pitch modifications. Therefore, a direct multiplication of the IF will result in aggressive transposition in the resynthesised signal.

A slow setting for the modulation frequency will result in a perception of glissandos ([►gliss-if] [4]), while a higher modulation frequency will bring a very unstable kind of pitch variation, since the rapid excursion will sweep a range from the deep lows to the top highs ([►aggress-if] [5]).

In the context of a classic amplitude modulation applied to the instantaneous frequency signal, different types of effects might be achieved depending on the modulator signal configuration. A deep modulation will tend to produce aggressive effects as well, but a more gentle variation might be interesting to create a detuning effect ([►d-if-tune] [6]) or a vibrato ([►v-if-brato] [7]).

4. EVALUATION

In addition to the intuition established with the theory and audition of the audio samples, an objective evaluation based on audio descriptors helps in the development and refinement of the effects.

Audio descriptors are quantities extracted directly from the audio signal, and might be related to models (e.g. psychoacoustic models) or to mathematical manipulations in order to derive some alternative perspective on the signal [24]. We will analyse two descriptors:

- Spectral centroid: indicates the position for the "center of mass" of a signal spectrum, the point that divides the spectrum in two balanced portions [25]. This quantity is strongly related to the brightness of a sound;
- RMS (root mean square): indicates the power, given by the averaged sum of the squared values of the signal [24], being therefore related to the perception of intensity.

The spectral centroid and RMS descriptors were extracted for both the dry signal (the audio with no effect applied) and the wet signals (the resynthesized signals) and compared. The Essentia [26] library was used via its Python API. All the audio samples were generated with Csound [27] [28].

[4]Glissando effect implemented via IF processing.
[5]Aggressive pitch modulation implemented via IF manipulation.
[6]Detuning effect implement via IF processing.
[7]Vibrato effect implemented via IF processing.

Despite the fact that low-pass filtering of the IF would not affect the range of values the IF signal, but only how fast variations can occur, Figure 7 shows that the operation lowered the spectral centroid, as a direct low-pass filtering of the original signal would do. The RMS (Figure 8) is not affected by such an operation; it is actually more influenced by manipulations on the envelope.

Figures 9 and 10 show the influence in the spectral centroid when the IF signal is fixed at a constant value. Knowing that the IF signal in our guitar signal example varies around 600 Hz, a 2 KHz fixed IF results in a higher centroid when compared to the dry signal, and an IF fixed at 200 Hz results in a lower centroid.

The gentle modulations effects do not seem to have much impact on the location of the spectral centroid, because the variations are small and are close to the original IF. Figure 11 shows the *v-if-brato* case. A little less smooth version of the effect is represented in Figure 12. The slow ring modulation commences with null values for the modulator causing the centroid to start at null values, and the centroid progressively reaches the original centroid values as the modulator values become close to 1. Figure 13, however, shows a more extreme effect where the ring modulation is deep and fast, which causes the centroid values to oscillate between zero and the original centroid values for the dry signal.

5. CONCLUSIONS

The incoherent mono-component Hilbert Transform based decomposition has received lots of criticism, specially in the speech analysis literature. We emphasize that the decomposition is transparent, i.e., the signal obtained with an analysis-resynthesis procedure is identical to the original, but the intermediate step of processing in the AM/FM domain can be dangerous, regarding the potential of introducing noise or artifacts into the resynthesised version.

However, we generated many examples where the resynthesis produced a clean signal, without any noise that would invalidate its musical usefulness. In some cases where artifacts do appear after resynthesis, the so-called intelligibility requirement (important in speech analysis) could be loosened in many musical contexts, so the noisy sonorities obtained might also be interestingly explored.

There is a huge potential for the AM/FM approach to be considered as an alternative to other classic modulation effect techniques, like the vocoder [29]. The scheme considered here can be easily applied to a melodic signal, i.e., signals with melodic lines like those created with many wind instruments, guitar solos, bass lines, voice, among many other examples. However, a lot of care should be taken with the envelope and instantaneous frequency signals interpretation, which should not be regarded as simple amplitude (AM, envelope) and frequency/pitch (FM, instantaneous frequency) components, but instead should be acknowledged as signals controlling a single sinusoidal oscillator, which adapts itself to represent all sorts of complex musical signals.

In this paper we focused on low-pass and dynamic range compression examples, but high-pass, band-pass, and expansion of the dynamic range of the envelope or IF work in a similar way. Emphasis was given to processing of the IF signal in the AM/FM representation, but processing the envelope can also lead to interesting effects, especially exploring roughness issues.

The low-pass IF filtering examples were important not only to obtain audio effects per se, but also for showing that the perceptual brightness of a sound is somehow linked to the possibility of the IF signal to vary quickly in an AM/FM representation, in a sense that limiting this speed will produce a perception of a muffled sound. This would not be evident a priori, since the IF can still sweep through all the possible frequencies after being filtered.

Dynamics processing and modulations in the IF will result in pitch modifications on the original signal, a side-effect which should be taken into account when implementing these effects, specially considering the musical scenario where it will be used. Modulation can also modify the pitch of the signal.

The configuration for the parameters' values is challenging, since the very same technique can result in effects from the very subtle to the very aggressive. Evidently, both aesthetic concerns and practicalities of the instruments (dry audio signals sources) will come into play in the design of AM/FM DAFx.

6. FUTURE WORK

We are currently working on effects that explore the separation of the input signal in different bands. After the separation, the AM/FM decomposition can be applied to all bands, and multi-layer effects based on different band-wise configurations of the same effect might lead to interesting results. We are also proceeding to a more thorough evaluation, considering more objective parameters that can elucidate our comprehension between the decomposition-manipulation of the estimated signals and the resulting effect. Subjective evaluations considering instrument players, DJs, and also music appreciators might lead to useful results about the quality and musicality of AM/FM-based effects.

7. ACKNOWLEDGMENTS

This work was partially supported by CAPES (proc. num. 8868-14-0) and was partially realized during Antonio Goulart's internship period at Maynooth University. We are grateful for the insightful comments by the reviewers.

8. REFERENCES

[1] Antonio José Homsi Goulart, Joseph Timoney, Victor Lazzarini, and Marcelo Queiroz, "Psychoacoustic impact assessment of smoothed am/fm resonance signals," in *Proceedings of the Sound and Musical Computing Conference*, Maynooth, Ireland, July 2015.

[2] Antonio José Homsi Goulart, Joseph Timoney, and Victor Lazzarini, "AM/FM DAFx," in *Proceedings of International Conference on Digital Audio Effects (DAFx)*, Trondheim, Norway, December 2015.

[3] James Justice, "Analytic signal processing in music computation," *IEEE Transactions on acoustics, speech and signal processing*, vol. ASSP-27, no. 6, pp. 670–684, 1979.

[4] John Chowning, "The synthesis of complex audio spectra by means of frequency modulation," *Journal of the Audio Engineering Society*, vol. 21, no. 7, pp. 526–534, 1973.

[5] S. Schimmel and L. Atlas, "Coherent envelope detection for modulation filtering of speech," in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing, 2005.*, March 2005, vol. 1, pp. 221–224.

[6] Qin Li and Les Atlas, "Over-modulated am-fm decomposition," in *Proceedings of the SPIE - Advanced Signal Processing Algorithms, Architectures, and Implementations*, Bellingham, WA, 2004, pp. 172–183.

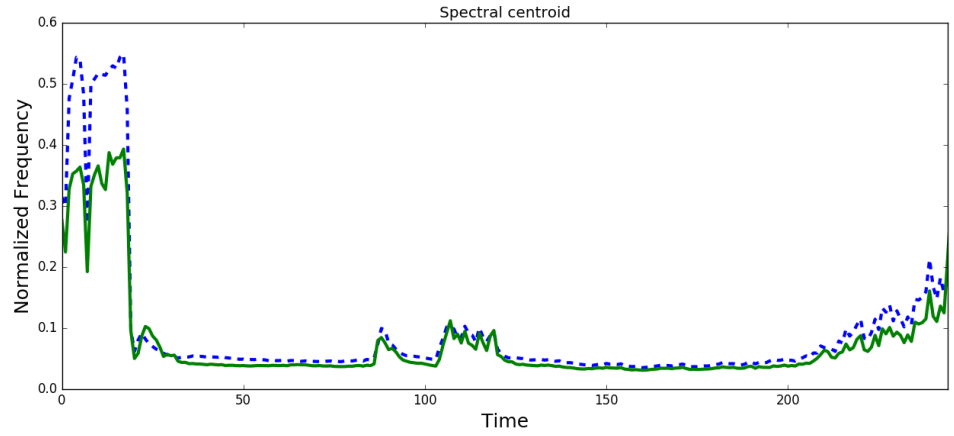


Figure 7: *Spectral centroid after low-pass (Butterworth) filtering the IF with a cut-off frequency at 1 KHz (solid green) versus spectral centroid of the dry signal (dashed blue).*

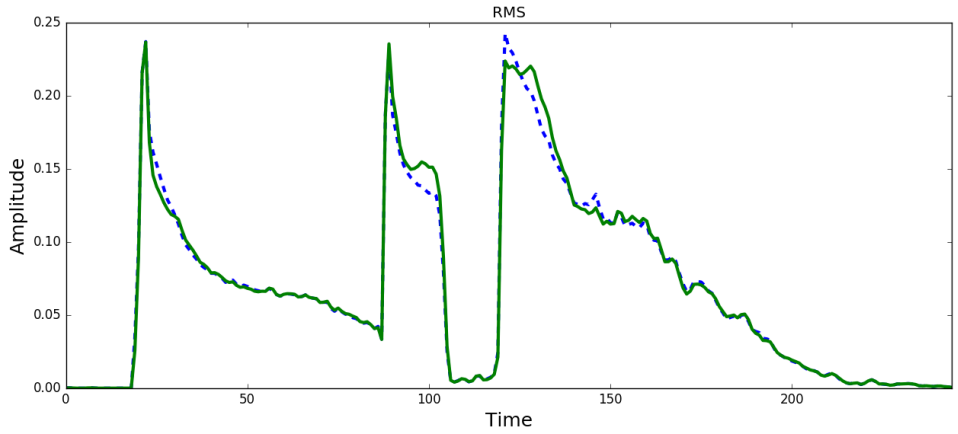


Figure 8: *RMS after low-pass (Butterworth) filtering the IF with a cut-off frequency at 1 KHz (green) versus RMS of the dry signal (blue).*

[7] P. Clark and L. Atlas, "Time-frequency coherent modulation filtering of nonstationary signals," *IEEE Transactions on Signal Processing*, vol. 57, no. 11, Nov 2009.

[8] Sascha Disch and Bernd Edler, "An amplitude and frequency-modulation vocoder for audio signal processing," in *Proceedings of the International Conference on Digital Audio Effects (DAFX-08)*, Espoo, Finland, September 2008.

[9] Sascha Disch and Bernd Edler, "An iterative segmentation algorithm for audio signal spectra depending on estimated local centers of gravity," in *Proceedings of the International Conference on Digital Audio Effects (DAFX-09)*, Como, Italy, September 2009.

[10] Sascha Disch and Bernd Edler, "An enhanced modulation vocoder for selective transposition of pitch," in *Proceedings of the International Conference on Digital Audio Effects (DAFX-10)*, Graz, Austria, September 2010.

[11] M. Caetano, G. P. Kafentzis, A. Mouchtaris, and Y. Stylianou, "Full-band quasi-harmonic analysis and synthesis of musical instrument sounds with adaptive sinusoids," *Applied Sciences*, vol. 6, no. 5, 2016.

[12] B. Boashash, "Estimating and interpreting the instantaneous frequency of a signal-part 1: Fundamentals," *Proceedings of the IEEE*, vol. 80, no. 4, Apr 1992.

[13] Patrick J. Loughlin, "Do bounded signals have bounded amplitudes?," *Multidimensional Systems and Signal Processing*, vol. 9, pp. 419–424, 1998.

[14] Charles Dodge and Thomas Jerse, *Computer Music: Synthesis, composition and performance*, Schirmer Books, New York, NY, USA, 2nd edition, 1997.

[15] D.E. Vakman and L.A. Vainshtein, "Amplitude, phase, frequency - fundamental concepts of oscillation theory," *Sov. Phys. Usp.*, vol. 20, no. 12, pp. 1002–1016, December 1977.

[16] P. Maragos, J. F. Kaiser, and T. F. Quatieri, "On amplitude and frequency demodulation using energy operators," *IEEE Transactions on Signal Processing*, vol. 41, no. 4, pp. 1532–1550, Apr 1993.

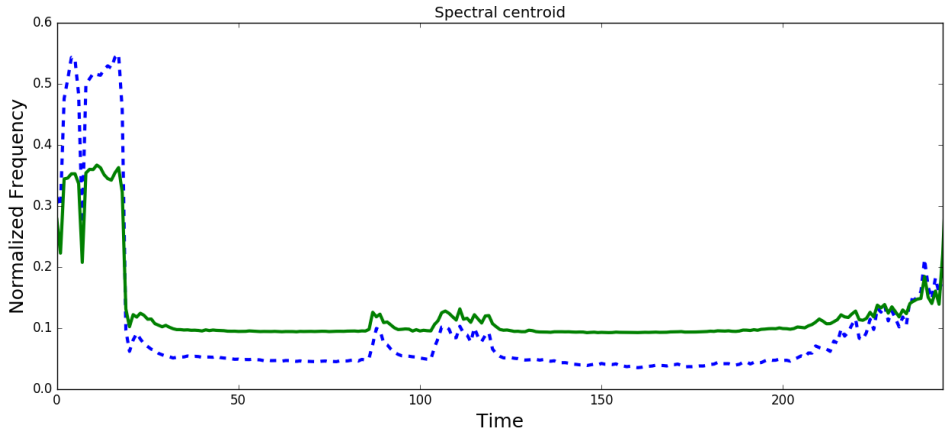


Figure 9: *Spectral centroid after fixing the IF at 2 KHz (green) versus spectral centroid of the dry signal (blue).*

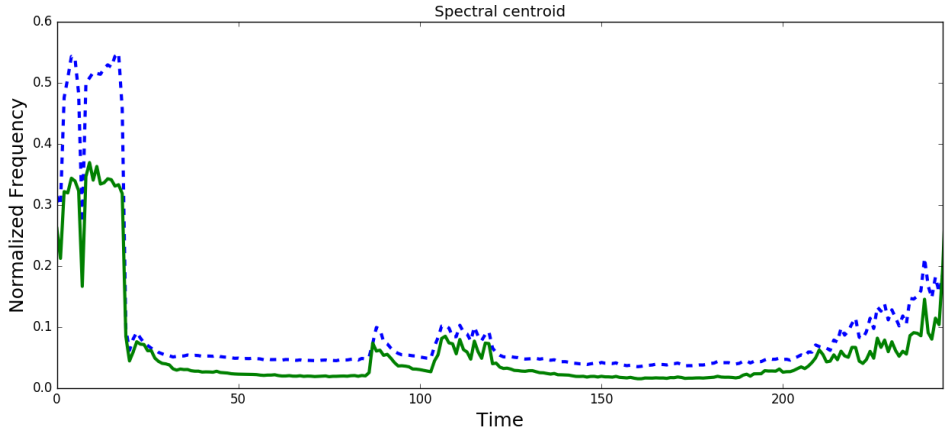


Figure 10: *Spectral centroid after fixing the IF at 200 Hz (green) versus spectral centroid of the dry signal (blue).*

[17] P. Maragos, J. F. Kaiser, and T. F. Quatieri, "Energy separation in signal modulations with application to speech analysis," *IEEE Transactions on Signal Processing*, vol. 41, no. 10, pp. 3024–3051, Oct 1993.

[18] Alexandros Potamianos and Petros Maragos, "A comparison of the energy operator and the hilbert transform approach to signal and speech demodulation," *Signal Processing*, vol. 37, no. 1, pp. 95 – 120, 1994.

[19] Julius Smith, *Mathematics of the Discrete Fourier Transform, with Audio Applications*, W3K Publishing, 2nd edition, 2007.

[20] A.V. Oppenheim and R.W. Schafer, *Digital signal processing*, Prentice Hall, New Jersey, USA, 1975.

[21] B. Picinbono, "On instantaneous amplitude and phase of signals," *IEEE Transactions on Signal Processing*, vol. 45, no. 3, pp. 552–560, Mar 1997.

[22] Stefan Hahn, *Hilbert Transforms in Signal Processing*, Artech House, Norwood, MA, 1996.

[23] Udo Zölzer, Ed., *DAFx: Digital Audio Effects*, Wiley & Sons, 2nd edition, 2011.

[24] Nick Collins, *Introduction to Computer Music*, Wiley and Sons, 2010.

[25] James Beauchamp, "Synthesis by spectral amplitude and 'brightness' matching of analyzed musical instrument tones," *J. Audio Eng. Soc*, vol. 30, no. 6, 1982.

[26] Dmitry Bogdanov, Nicolas Wack, E. Gómez, Sankalp Gulati, Perfecto Herrera, O. Mayor, Gerard Roma, Justin Salamon, J. R. Zapata, and Xavier Serra, "Essentia: an audio analysis library for music information retrieval," in *International Society for Music Information Retrieval Conference (ISMIR)*, Curitiba, Brazil, 04/11/2013 2013, pp. 493–498.

[27] Richard Boulanger, Ed., *The Csound Book: Perspectives in Software Synthesis, Sound Design, Signal Processing, and Programming*, MIT Press, 2000.

[28] Victor Lazzarini, Steven Yi, John ffitch, Joachim Heintz, Øyvind Brandtsegg, and Iain McCurdy, *Csound: A sound and music computing system*, Springer, 2016.

[29] J. L. Flanagan and R. M. Golden, "Phase vocoder," *Bell System Technical Journal*, vol. 45, no. 9, 1966.

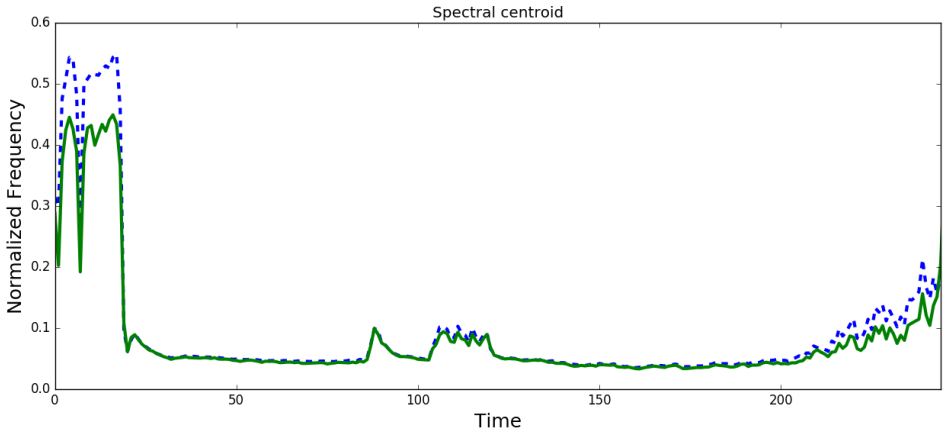


Figure 11: *Spectral centroid after amplitude modulating the IF (14 Hz of modulation depth and modulation frequency at 15 Hz) versus spectral centroid of the dry signal (blue).*

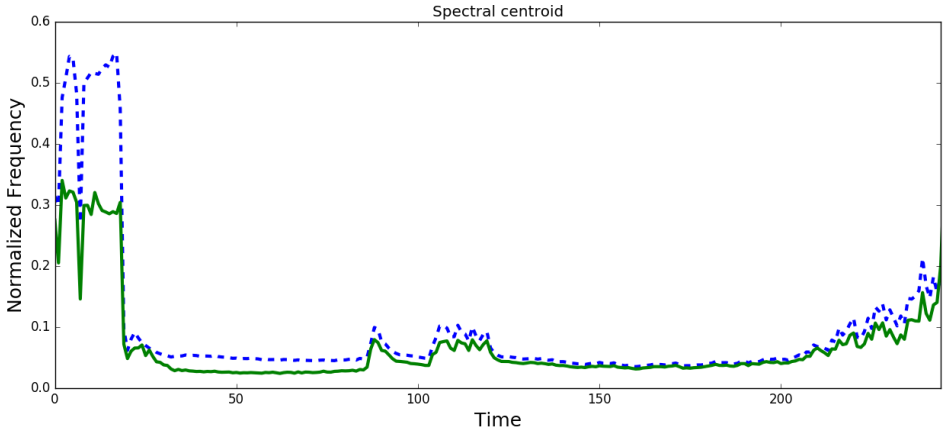


Figure 12: *Spectral centroid after ring modulating the IF (modulation frequency at 0.1 Hz) versus spectral centroid of the dry signal (blue).*

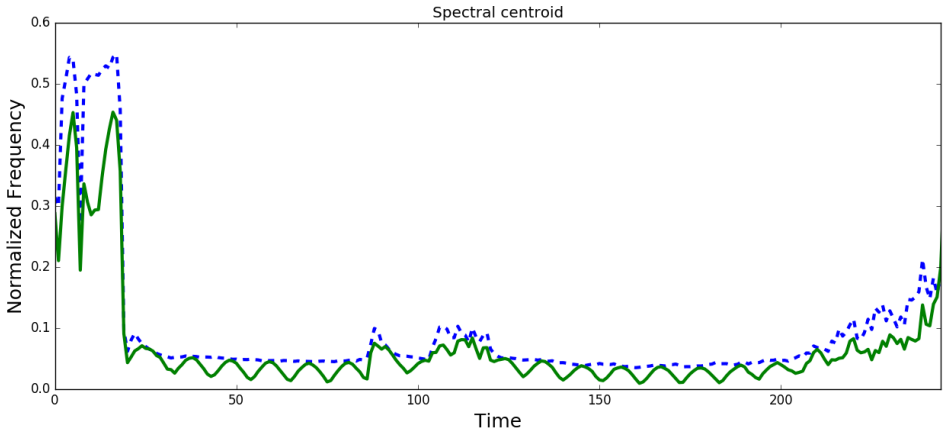


Figure 13: *Spectral centroid after ring modulating the IF (modulation frequency at 4 Hz) versus spectral centroid of the dry signal (blue).*

HIGH FREQUENCY MAGNITUDE SPECTROGRAM RECONSTRUCTION FOR MUSIC MIXTURES USING CONVOLUTIONAL AUTOENCODERS

Marius Miron *

Independent researcher
miron.marius@gmail.com

Matthew E.P. Davies

INESC TEC
Sound and Music Computing Group
Porto, Portugal
mdavies@inesctec.pt

ABSTRACT

We present a new approach for audio bandwidth extension for music signals using convolutional neural networks (CNNs). Inspired by the concept of inpainting from the field of image processing, we seek to reconstruct the high-frequency region (*i.e.*, above a cutoff frequency) of a time-frequency representation given the observation of a band-limited version. We then invert this reconstructed time-frequency representation using the phase information from the band-limited input to provide an enhanced musical output. We contrast the performance of two musically adapted CNN architectures which are trained separately using the STFT and the invertible CQT. Through our evaluation, we demonstrate that the CQT, with its logarithmic frequency spacing, provides better reconstruction performance as measured by the signal to distortion ratio.

1. INTRODUCTION

Audio signals are often low-passed, encoded or compressed before transmitting them through phone lines and Internet streams. This results in the loss of high frequency content and compromises audio quality. Narrow-band audio signals which have information up to a certain frequency cutoff can be perceptually enhanced by reconstructing the higher frequency content. This research task, known as *audio bandwidth extension*, attempts to increase the perceived or real frequency spectrum of audio signals [1, 2, 3, 4, 5].

Audio bandwidth extension methods have been applied to speech signals in an unsupervised and supervised manner. The former are typically statistical approaches which model the relationship between low and high frequency spectral content by relating lower and upper harmonics [1]. For instance, the linear predictive coding (LPC) method in [2] analyzes the lower frequency spectra to synthesize high frequency components. It relies on a codebook: a dictionary of wide-band envelopes, which are matched with the envelope of narrow-band spectral frames. Spectral band replication [6] on the other hand transposes up harmonics from lower and midrange frequencies to higher bands.

Supervised methods learn priors from wide-band signals which are later used to recover the high frequency content of narrow-band signals. Matrix decomposition methods such as non-negative matrix factorization (NMF) [3, 5] treat the magnitude spectrogram as combinations of priors in the form of non-negative bases. At the test stage, these bases are kept fixed and are used to estimate the NMF parameters which best explain the narrow-band signal.

Methods using neural networks learn priors from features derived from time-frequency representations to predict high-band spectral envelopes [7, 8]. Bandwidth extension with deep neural networks has been shown to increase the robustness of speech recognition [8]. In addition, the resolution of raw audio signals, regarded as time series, can be increased using convolutional neural networks (CNNs) [9].

In this paper we seek to estimate high frequency components in time-frequency representations of music signals. Compared to speech, music signals are often complex mixtures, comprising a variety of instruments, both percussive and harmonic, singing voice, and non-linear audio effects. Thus, music signals have broader, richer, and perceptually more relevant high frequency content, which is therefore more difficult to estimate.

While the aim of bandwidth extension for speech is tightly coupled with signal compression and band-limited communication channels, for music signals there are important distinctions both in terms of the constraints of the problem and the potential applications. First and foremost, our aim is to perform bandwidth extension up to CD quality (*i.e.*, 44.1 kHz sampling rate with a Nyquist rate of 22.05 kHz). Given the absence of harmonic information in high frequency musical content (*e.g.*, above 10 kHz), our proposed musical bandwidth extension will be required to reconstruct percussive-type content. Depending on the bandwidth of the narrow-band input signal, it may also be required to reconstruct the upper partials of harmonic content present in the narrow-band signal. In this way, perceptually accurate musical bandwidth extension could be used to replace high-band information typically lost via lossy compression in audio formats such as MP3 and AAC, and thus reduce the bandwidth overhead when streaming music, or allocate a higher bit rate for lower frequency information.

Our specific long term goal is to explore a more creative application of audio bandwidth extension, namely towards the restoration of old music recordings. To this end, we seek to renew old recordings (in particular, jazz from the 1940s and 50s) and thus allow modern-day listeners to experience this music in high audio quality as performed by the original musicians. Towards this ambitious goal, we first investigate the feasibility of full-bandwidth extension for music signals under more controlled conditions, which can be more readily evaluated via access to both the full- and band-limited versions.

Similar to the concept of image inpainting or completion [10, 11], for which CNNs have been shown to be particularly adept, we aim to learn localized features in order to recover the missing higher frequency regions of short-term Fourier transform (STFT) and constant-Q transform (CQT) stereo magnitude spectrograms [12]. However, since the time and frequency axes in STFT and CQT representations do not correlate in the same way

* Marius Miron is currently a post-doctoral researcher at the European Commission Joint Research Center

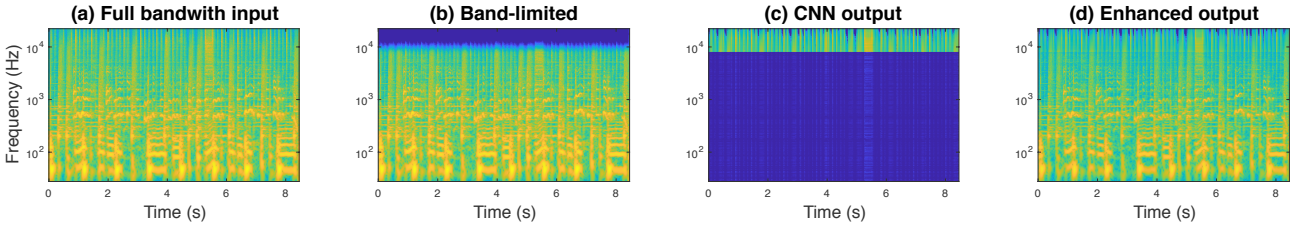


Figure 1: *Illustrative overview of our proposed approach for bandwidth extension. (a) The CQT of a short musical audio input sampled at* 44.1 *kHz. (b) The band-limited version resulting from a low-pass filter with a cutoff frequency of* 7500 *Hz. (c) The high frequency output of the CNN*[1]*. (d) The enhanced output signal obtained by combining the band-limited and CNN reconstruction.*

as the axes of an image, we explore two musically motivated CNN architectures: bottleneck and stride [13, 14] rather than more standard square filters in image processing.

For our musical inpainting problem, we aim to reconstruct or "complete" a strip covering the highest frequency bins of a time-frequency, for which an illustrative example is shown in Figure 1. While this is conceptually related to the idea of filling temporal gaps (*i.e.*, missing vertical strips) [15, 16] these methods exploit temporal redundancy via repetition in the musical input, where as in our approach, the high frequency region is never observed.

A particular novelty of our proposed approach is to leverage implicit knowledge of musical structure by the use of the constant-Q spectrogram. For bandwidth extension, the CQT has a potentially advantageous property over the STFT, which is that, due to the logarithmic spacing of the CQT bins, we can make a richer observation of the narrow-band (*i.e.*, low-frequency) region in order to reconstruct a smaller amount of higher frequency information. Comparing the STFT and CQT in matrix form (where rows correspond to frequency and the columns to time) this means that for an identical cut-off frequency (*e.g.*, of $f_s/4$), and a roughly equal total number of frequency channels, a far smaller amount of data must be reconstructed for the CQT than for the STFT. Until recently, such potential benefits remained theoretical due to the absence of an inverse CQT transform. However, recent work leveraging the non-stationary Gabor transform (NSGT) [17, 18] has demonstrated that perfect reconstruction of the CQT is both possible and executable in reasonable computation time.

For this initial work, our primary focus is towards the reconstruction of magnitude spectrograms, thus we do not attempt any automatic reconstruction of the phase spectrogram. Instead we make use of the original phase from the band-limited version, without any subsequent modification. Our evaluation focuses on the measurement of the signal to distortion ratio (SDR) for the enhanced and band-limited versions. In this way, the extent of the enhancement provided by our approach can be assessed by the increase in SDR over the band-limited versions.

The remainder of this paper is structured as follows. In Section 2 we contrast our approach with existing work in audio bandwidth extension. In Section 3, we detail our proposed method using convolutional neural networks, which we evaluate in Section 4, and provide discussion and conclusions in Section 5.

[1] While the CNN outputs a full wide-band spectrogram, the region below the cut-off has been attenuated for greater visual clarity.

2. RELATION WITH PREVIOUS WORK

With the exception of [5, 9, 19], most previous research in audio bandwidth extension has been applied to speech signals. Regarding the methodology, the deep learning approaches in [8, 9] are the closest to our proposed method. In the same way as [9], which uses a similar approach to image super-resolution [20], we are inspired by recent advancements in image processing using CNNs [10, 11]. Unlike [7] we eliminate all accompanying heuristics and estimate the high-frequency spectra directly with the neural networks.

In contrast to the NMF speaker-specific spectral bases used in [3, 19] or the codebook of the LPC approach [2], we are concerned with the generalization capabilities of our trained model and do not seek to tailor our approach for specific individual pieces of music. Furthermore, we do not tune any method-specific hyper-parameters or weighting coefficients which were previously used in [2] as a part of a chain of signal processing heuristics.

Similar to the convolutional NMF approach in [3], the hidden Markov models (HMM) in [21], and the time-series CNN in [9], we consider cross-frame contextual dependencies. These short-term dependencies are learned by CNNs using horizontal filters for a given time-context, while timbre features are learned using vertical filters [13, 14].

The CNN approaches used in image restoration, completion, or inpainting [22, 10, 11] are exposed to the entire image and not just to the missing patches in order to perform the reconstruction. In a similar fashion, we use the observation of the lower frequencies to better reconstruct the higher frequencies.

3. METHOD

3.1. Overview

An overview of our proposed method, which comprises two stages: training and enhancement, can be seen in Figure 2. For training we require a dataset comprising full-bandwidth music recordings and narrow-band versions which lack high frequency content above a specific cutoff frequency. We obtain narrow-band versions by applying a low-pass filter to the original recordings. Then, we compute the desired time-frequency representation, using the STFT or CQT, and extract the respective magnitude spectrogram for each channel of the stereo recordings. Additionally, we apply the data processing heuristics described in [23] and train the CNNs with the architectures described in Section 3.3 and the training procedure in Section 3.4.

The enhancement stage is detailed in the Section 3.5, where the high-frequency content is obtained by feeding the magnitude

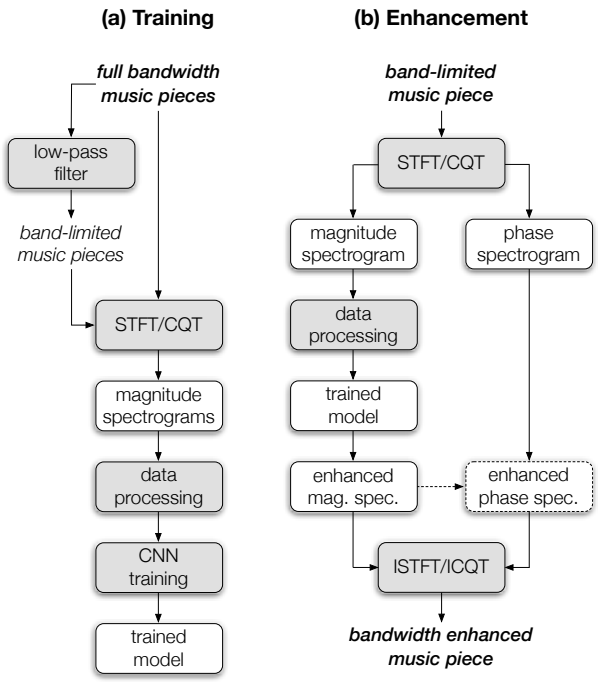


Figure 2: *Overview of our bandwidth extension system. (a) The training stage has access to full-band and band-limited music signals. (b) The enhancement stage only observes the band-limited signals. Boxes shaded in grey indicated processes, where as those in white correspond to data. The term data processing is used to encapsulate the partitioning of the data into overlapping chunks. The dashed arrow and box indicate optional processing which is not undertaken in this work.*

spectrograms forward through the previously trained CNN. The phase spectrogram of the band-limited version is retained to compute the inverse STFT or CQT.

3.2. Feature computation

We calculate the STFT or the CQT [18] of the stereo audio mixture as $\mathbf{X}_i(t, f)$ where $i = 1, 2$ are the stereo channels, t is the time axis and f is the frequency axis. In order to focus on the reconstruction of the magnitude spectrum, we discard the phase when computing the training features for the neural network.

The CNN architectures used in this paper require a fixed input size (T, F), where T is the temporal context in time frames and F is the total number of frequency bins corresponding to the STFT or CQT magnitude spectrograms. To obtain magnitude spectrograms of fixed duration, the variable-size magnitude spectrograms of each music piece are split into overlapping chunks of fixed size T time frames with an overlap of O frames. In addition, splitting the input signal into chunks leads to a smaller network, with fewer parameters to train, and thus a lower computational burden. These data processing heuristics adopted prior to training are described in detail in [23] and were used previously for the task of audio source separation for full length musical recordings [14, 23, 24].

3.3. Convolutional autoencoders

We present two musically motivated CNN autoencoder architectures, the CNN bottleneck in Section 3.3.1 and the CNN stride-2 in Section 3.3.2. Since time and frequency in magnitude spectrograms have different meanings than the horizontal and vertical axes in images, we should not adopt image-processing square filters. Instead, we follow [13, 14] by using vertical filters to model frequency components and horizontal filters to model their temporal evolution. A further distinction is that the magnitude spectrograms of audio signals are sparse [25]. Thus, we use a sparse activation function between the layers, specifically, rectified linear units (ReLU) [26]. In addition, the CNN bottleneck architecture has a dense bottleneck layer with a low number of units to compress, or reduce, the learned features. On a related note, the CNN stride-2 architecture comprises successive convolutions with a stride[2] of two which is the equivalent of learning features by successively downsampling the inputs by a factor of two.

The inputs to both the CNN architectures are multiple magnitude spectrograms of size (T, F), across the channel dimension i. In our case, the learned feature maps are shared between the two input channels [26]. We argue that the CNN can learn more diverse filters from music mixtures with a wide stereo image and therefore we provide magnitude spectrograms for both channels as input. In a further parallel with image processing, this can be considered similar to using the RGB layers of colour images rather a single greyscale image.

The CNN autoencoders comprise an encoding and a decoding stage. The encoding stage contains convolutional and feed-forward layers, while the decoding stage performs the inverse operations of the convolutions in the reverse order such that the output of the CNN has the same dimensions as its input, $(2, T, F)$. Note, we do not use a soft-mask as in music source separation, but instead we directly estimate the magnitude spectrogram with enhanced high-frequency content, $\hat{\mathbf{X}}$. In addition, we assume that the frequency content to be recovered does not have higher energy than the low frequency content. To this end, we limit all the values of $\hat{\mathbf{X}}_i(t, f)$ to the maximum value in channel i at time frame t of the input $\mathbf{X}_i(t, f)$.

3.3.1. CNN bottleneck

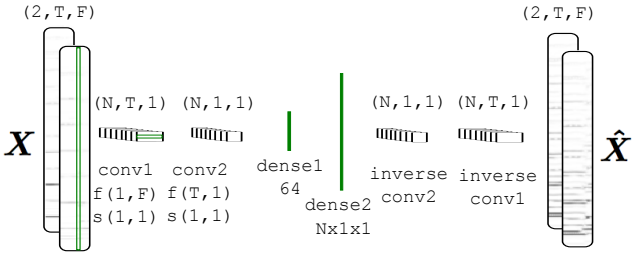


Figure 3: *CNN bottleneck autoencoder architecture [14]. For each layer we give the shape of the filters, strides and feature maps.*

We test a version of the CNN bottleneck successfully used in music source separation [14, 24, 23]. A diagram of the architecture is depicted in Figure 3, and comprises a horizontal convolution, *conv1*, a vertical convolution *conv2*, a bottleneck dense layer

[2]The stride controls how much a filter is shifted on the input.

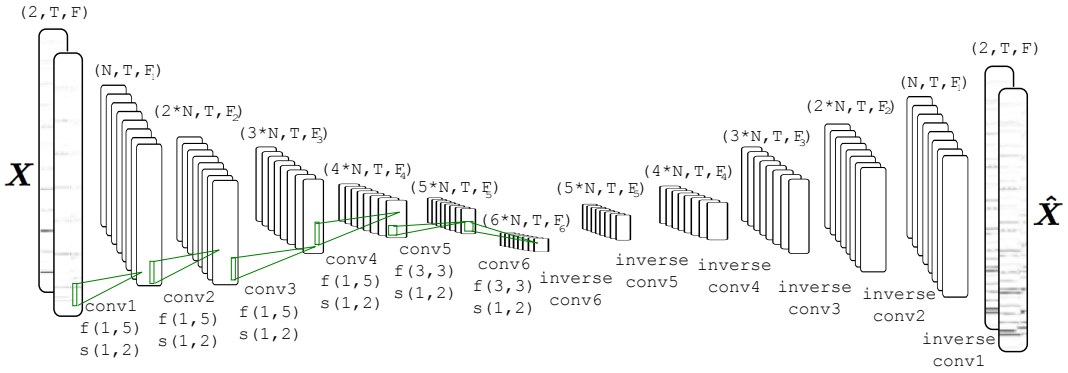


Figure 4: *CNN stride-2 autoencoder architecture. For each layer we give the shape of the filters, strides and feature maps.*

dense1, and another dense layer *dense2* to recover the dimensionality needed to perform the inverse operations of *conv2* and *conv1*. We have N filters for $conv1$ and $conv2$.

3.3.2. CNN stride-2

Small successive convolutional layers with a stride of two have been shown to reduce the number of parameters in a network [27]. Therefore, in contrast to the CNN bottleneck, we target a deep architecture comprising small convolutions. Moreover, time-frequency representations of musical signals often exhibit evenly spaced harmonic components. By modeling frequency content in strides of two we aim to capture high frequency harmonics learned from their low frequency counterparts.

An overview of the stride-2 architecture is shown in Figure 4. For each layer k, the feature maps reduce their frequency size: $F_k = (F_{k-1} - 5)/2 + 1$, as explained in [24]. We have four successive $(1, 5)$ convolutions in frequency, followed by two, two-dimensional $(3, 3)$ convolutions to capture the time-frequency dependencies, each considering the reduction performed by the previous layers.

3.4. Training procedure

Although the output of the CNN, $\hat{\mathbf{X}}$, contains a reconstruction of the magnitude spectrogram across all frequency bins, the parameters of the autoencoder are trained according to a loss function which only considers the reconstruction in higher frequencies. Thus, the loss function L_c depends on the cutoff frequency in bins c and is defined in equation (1) as the mean-squared error (MSE) between the target magnitude spectrogram $\bar{\mathbf{X}}$, and the estimated magnitude spectrograms, $\hat{\mathbf{X}}$:

$$L_c = \sum_{t,f,i} \|u(f-c)(\bar{\mathbf{X}}_i(t,f) - \hat{\mathbf{X}}_i(t,f))\|^2, \qquad (1)$$

where $u(f - c)$ is the unit step function which is 0 for the bins lower than c and 1 for the bins greater than or equal to c.

The parameters of the CNN are updated according to the loss function L_c using mini-batch Stochastic Gradient Descent with the *Adamax* algorithm [28].

3.5. Enhancement

When computing the STFT or CQT for enhancement, we retain the phase and we split the magnitude spectrogram into overlapping chunks of size T time frames with an overlap of O frames as in the training stage. For each chunk $\mathbf{X}$ we obtain an estimation $\hat{\mathbf{X}}$. We then use the estimated chunks to reconstruct the enhanced magnitude spectrogram through the overlap-add procedure as described in [23] and as used in [14, 23, 24].

In contrast to deep learning source separation methods, the estimated spectrogram is not the result of Wiener filtering [29] which ensures that the spectrograms of the sources sum to the input spectrogram. Instead, we need to ensure that the original low-bandwidth content is preserved. To this end, we blend the high-frequency part of the estimations yielded by the network, $\hat{\mathbf{X}}$, with the low-frequency part of the input, $\mathbf{X}$:

$$\tilde{\mathbf{X}}_i(t,f) = (1 - r_c(f))\mathbf{X}_i(t,f) + r_c(f)\hat{\mathbf{X}}_i(t,f) \qquad (2)$$

where $r_c(f) = \max(0, \min(1, f - c))$ is a ramp function depending on the the cutoff frequency in bins c.

As specified in Section 1, we only attempt to reconstruct the magnitude spectrum – without access to phase information when training. However, in order to invert either the reconstructed STFT or CQT we must provide phase information. To this end, we use the phase spectrogram from the band-limited version, as shown in Figure 1(b). Finally, the bandwidth extended audio signals are obtained using with an inverse overlap-add STFT or inverse CQT [18].

4. EVALUATION

The basis of our evaluation is to compare the reconstruction from the STFT and CQT, with the two different CNN autoencoder models: bottleneck and stride-2, and across two cutoff frequencies of 3500 Hz and 7500 Hz. In total, this creates eight reconstruction conditions for comparison.

4.1. Experimental setup

We test our approach on the publicly available Medleydb dataset [30] comprising 121 multi-tracks from which we use the stereo mixes (in uncompressed .wav format sampled at 44.1 kHz and with 16-bit resolution). The dataset covers the following genres: Singer/Songwriter, Classical, Rock, World/Folk, Fusion, Jazz,

Pop, Musical Theatre, Rap. There are 52 instrumental tracks and 70 tracks containing vocals. We randomly split the dataset in training and testing subsets with a ratio of 0.8 (*i.e.*, 80% for training and 20% for testing).

4.1.1. Evaluation metrics

As the basis for the evaluation, we use the BSS_Eval framework [31], a widely used tool to objectively evaluate the quality audio source separation. Within BSS_Eval, the *Source to Distortion Ratio* (SDR) measures the distortion between a target and the estimated multi-channel audio sources. With respect to high-frequency reconstruction, BSS_Eval gives more weight to lower frequency bands and penalizes more frequency content which is not in the target audio, even though this content might be perceptually relevant. In this sense, we recognise that a subjective listening experiment would be a critical important component of future work, but for this initial research, we adopt the SDR as our primary objective measure for this context. It is important to note that we exclude other metrics related to the artifacts, interference, and spatial distortion from BSS_Eval as these are designed particularly for source separation. The SDR is reported for each of the overlapping chunks of 30 seconds with a 15 second overlap.

4.1.2. Time-frequency transform parameterisation

The STFT is computed using a Hann window of length 1024 samples, which at a sampling rate of 44.1 kHz corresponds to 23.2 milliseconds (ms), and a hop size of 512 samples (11.6 ms).

The CQT is computed with the MATLAB toolbox in [18] using the default parameterization, with a minimum frequency of 27.5 Hz, and a frequency resolution of 48 bins per octave. Up to the Nyquist rate of 22.05 kHz this gives 463 logarithmically-spaced frequency bins. Perfect reconstruction via the inverse CQT comes at the expense of high redundancy in time and results in 647 time frames per second, *i.e.*, a temporal resolution of 1.5 ms which is much finer than that of the STFT, while retaining a similar number of frequency bins (463 compared to 513).

Since our goal is to reconstruct the higher frequency end of the magnitude spectrograms, we must contend with the fact that signal energy typically is much lower at higher frequencies than at the lower end. In the context of our convolutional neural network approach this creates a difficulty, since the high frequency magnitude spectrum we seek to predict may have very small values. To partially circumvent this issue, we can apply a logarithmic scaling to both the STFT and CQT magnitude spectrograms prior to training (and subsequently revert back to linear magnitude scaling prior to the eventual output signal reconstruction). However, before applying such a logarithmic scaling we must ensure all magnitude spectrum values (for both the STFT and CQT) are greater than 1, since any values below 1 will be negative after taking the logarithm, and thus ignored by the ReLU. To this end we apply the logarithmic scaling as follows: $\mathbf{X}_{\log} = \log_{10}(\alpha + \beta\mathbf{X})$, where $\mathbf{X}$ refers to either the STFT or CQT. For the CQT we set $\alpha = 1$ and $\beta = 4$, where as for the STFT no scaling is required thus we set $\alpha = 1$ and $\beta = 1$. The final stage of the pre-processing relates to deep learning methods usually requiring data to be normalized to an interval or include a batch-normalization step. Thus, we normalize all the training data to be between 0 and 1 by multiplying with a scale factor, which we set as the maximum of the training data.

To create the band-limited, *i.e.*, low-pass filtered versions of the music pieces for training (and subsequent reconstruction), we use an 8^{th} order Butterworth filter. In order to explore two different conditions, we create one low-pass filtered version with a cutoff of 3500 Hz and another at 7500 Hz (approximately $f_s/12$ and $f_s/6$). For both, we seek to reconstruct the full remaining frequency range of the original recordings up to the Nyquist rate of 22.05 kHz).

We split the STFT or CQT into overlapping chunks of $T = 30$ time frames with an overlap of $O = 10$. Chunks are randomly grouped each epoch into batches of 32. For a fair comparison between bottleneck and stride-2 we use $N = 175$ of filters for bottleneck and $N = 40$ filters for stride-2, such that the number of parameters is equal for both of the architectures (1.8 million). The STFT is trained for 100 epochs. Since CQT has a higher time resolution, we generate more training data and we only train the network for 32 epochs. The initial learning rate is 0.001 for STFT and 0.0001 for CQT.

4.1.3. Implementation details

The code used in this paper is built on top of Pytorch, a framework for neural networks [3]. We ran the experiments on an Ubuntu 16.04 PC with GeForce GTX TITAN X GPU, Intel Core i7-5820K 3.3GHz 6-Core Processor, X99 gaming 5 x99 ATX DDR44 motherboard. Training a condition took 16 hours for the STFT and 44 hours for the CQT; by contrast, the enhancement stage runs faster than real-time on the same hardware. To ensure reproducibility, a fixed seed controls the pseudo-random number generation in Python. This is used when initialize the parameters of the CNN and to randomly split the dataset into training and testing. The results presented in Section 4.2 are for seed 0.

4.2. Results

The results for the bottleneck and stride-2 are shown in terms of SDR in Figure 5a and 5b for the CQT and STFT respectively. In each figure we present the SDR across the cutoff frequencies of 3500 Hz and 7500 Hz and show the difference in performance for examples in the training set versus those withheld for testing. Since we want to measure how much the quality of the reconstruction improves with respect to the low-pass input, we include the SDR for all the low-pass versions of the pieces in the dataset.

On inspection of the figures we can see that the best overall performance for the test set is obtained using the stride-2 architecture for the cutoff of 3500 Hz and the bottleneck architecture for the cutoff of 7500 Hz. In both of these conditions there is a negligible difference between the SDR on those musical recordings used for training, compared to those withheld for testing. In addition to the highest overall mean SDR values, we can additionally observe the greatest relative difference over the mean SDR of the low-pass filtered versions. For both approaches there is a relative increase in SDR of over 4 dB. Since the SDR calculation is made directly on the waveforms, this suggests that relevant high frequency information from the original recordings is being reconstructed based soley on observing the band-limited versions.

When looking across the two architectures for the CQT results, we can observe that the stride-2 approach is less effective for the higher cutoff of 7500 Hz. This may be due to the lower proportion of harmonic content above this cutoff, and hence the reduced impact of the stride's ability to model harmonic relationships.

[3] `http://pytorch.org`

Figure 5: *SDR for (a) CQT and (b) STFT representations. The results compare the difference in SDR for training and testing sets, and the low-pass filtered condition (without enhancement), for the bottleneck and stride-2 CNN architectures and the cutoff frequencies of* 3500 *Hz and* 7500 *Hz. The black vertical lines represent the 95% confidence intervals.*

Looking at the comparison between the CQT and STFT, we can identify two main differences. First, the absolute SDR for the STFT enhanced versions are lower than for the CQT across all conditions, and in turn, the relative improvement over the low-pass filtered versions is also reduced. This behaviour is in line with our original hypothesis concerning the advantage of using the CQT, where, although the frequency range to reconstruct is the same for both time frequency representations, the number of missing rows of the CQT is far smaller than that of the STFT. This is also consistent with results from image completion, in which larger image patches are more difficult to recover than smaller ones [10]. Another important factor may be the difference in temporal resolution for the two time-frequency representations, which is greater by a factor of approximately 8 to 1 for the CQT compared to the STFT; that while both process overlapping chunks of $T = 30$ time frames, the reconstruction of the CQT is much more localised in time than the STFT. We intend to explore this effect in future work by increasing the frame overlap in the STFT to a comparable level to that of the CQT. However, any significant increase in the frequency resolution of the STFT, *e.g.*, by using a larger window size would drastically increase the size of the model to be trained, and thus negate the approximately equal number of frequency channels in the STFT and CQT in our current setup.

To complement these objective results, we provide a set of short sound examples covering the eight reconstruction conditions, together with the original and two low-pass filtered versions. Furthermore, for the two best performing conditions: CQT stride-2 3500 Hz and CQT bottleneck 7500 Hz we provide an informal comparison of different approaches for phase reconstruction. To this end, we include phase reconstruction using: i) the low-pass filtered version (our proposed method); and ii) using low-pass filtered version below the cutoff and random phase above it. All of the sound examples are available at the following website: `http://telecom.inesctec.pt/~mdavies/dafx18/`

5. DISCUSSION AND CONCLUSIONS

We presented a new deep learning method to reconstruct the high frequency content of music recordings. Our evaluation demonstrates that due to to the logarithmic spacing of frequencies, the CQT offers a better time-frequency representation for this problem than STFT in terms of SDR. It is important to stress that these are initial experiments are performed under highly controlled conditions. Due to the high computational cost of training (which took several days using powerful GPUs), we only explored two cutoff frequencies, and used the same type of low-pass filter throughout. On this basis, we do not have sufficient evidence about the generalisation capacities of our trained networks to function under more arbitrary filtering conditions. This is especially important when considering our long term goal of the restoration of old recordings, for which we cannot assume any specific filtering conditions. Furthermore, in this scenario no stereo version of the recording may exist, which would require additional modifications to our approach.

Another important constraint within this study was the treatment of the phase in the reconstruction. While we do not provide unobservable information (*e.g.*, the phase of the original, full-band signal), our approach for using the low-pass filtered version phase could almost certainly be improved via the use of phase reconstruction techniques [32]. Since these are typically applied for an STFT-like representation, we intend to explore the means for doing this directly for in the invertible CQT representation in future work. Furthermore, we recognise the potential of using other time-frequency representations – provided that there is a method to invert them, *e.g.*, using Wavenet as a vocoder [33]. Furthermore, generative adversarial networks have recently became popular in image recovery and super-resolution [10] and can synthesize more realistic time-frequency content, which may yield further improvements to the quality of the signal reconstruction.

With respect to the evaluation, we acknowledge that BSS_Eval

has been primarily designed for audio source separation, and further perceptual experiments are needed to better understand the subjective performance of our proposed method. Furthermore, BSS_Eval metrics do not always correlate with the perceived quality of separation [34]. In contrast to magnitude spectrograms, reconstructed images can be evaluated more directly because the inherent structure in the pixels can be understood in terms of the geometric and textural properties of scenes and objects. However in our approach the images correspond to time-frequency representations which are non-trivial for non-experts to visually interpret, and require an additional transformation stage to be audible. Within our training stage, the loss function relates to the mean squared error between the original magnitude spectrogram and the reconstruction, however our objective evaluation measures the SDR of the reconstructed audio signals, which explicitly includes phase information. Thus, we also intend to explore alternative loss functions (perhaps by using phase information directly) and subsequently investigate their correlation with perceptual ratings of audio quality from trained listeners. As part of this comparison we we intend to incorporate existing approaches for bandwidth extension which have been shown to be effective for music signals sampled at 44.1 kHz.

6. ACKNOWLEDGMENTS

We wish to thank the anonymous reviewers for their expertise and generous feedback which improved the quality of this paper.

Matthew E.P. Davies is supported by Portuguese National Funds through the FCT – Foundation for Science and Technology, I.P., under the project IF/01566/2015. The TITANX used for this research was donated by the NVIDIA Corporation. "TEC4Growth - Pervasive Intelligence, Enhancers and Proofs of Concept with Industrial Impact/NORTE-01-0145-FEDER-00020" is financed by the North Portugal Regional Operational Programme (NORTE 2020), under the PORTUGAL 2020 Partnership Agreement, and through the European Regional Development Fund (ERDF).

7. REFERENCES

[1] H. Yasukawa, "Signal restoration of broad band speech using nonlinear processing," in *European Signal Processing Conference*, 1996, pp. 1–4.

[2] S. Chennoukh, A. Gerrits, G. Miet, and R. Sluijter, "Speech enhancement via frequency bandwidth extension using line spectral frequencies," in *IEEE International Conference on Acoustics, Speech, and Signal Processing*, 2001, vol. 1, pp. 665–668.

[3] D. Bansal, B. Raj, and P. Smaragdis, "Bandwidth expansion of narrowband speech using non-negative matrix factorization," in *Ninth European Conference on Speech Communication and Technology*, 2005, pp. 1505–1508.

[4] E. Larsen and R. M. Aarts, *Audio bandwidth extension: application of psychoacoustics, signal processing and loudspeaker design*, John Wiley & Sons, 2005.

[5] D. L. Sun and R. Mazumder, "Non-negative matrix completion for bandwidth extension: A convex optimization approach," in *IEEE International Workshop on Machine Learning for Signal Processing (MLSP)*, 2013, pp. 1–6.

[6] Martin Dietz, Lars Liljeryd, Kristofer Kjorling, and Oliver Kunz, "Spectral band replication, a novel approach in audio coding," in *Audio Engineering Society Convention 112*. Audio Engineering Society, 2002.

[7] J. Kontio, L. Laaksonen, and P. Alku, "Neural network-based artificial bandwidth expansion of speech," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 15, no. 3, pp. 873–881, 2007.

[8] K. Li, Z. Huang, Y. Xu, and C-H. Lee, "DNN-based speech bandwidth expansion and its application to adding high-frequency missing features for automatic speech recognition of narrowband speech," in *Sixteenth Annual Conference of the International Speech Communication Association*, 2015, pp. 2578–2582.

[9] V. Kuleshov, S. Z. Enam, and S. Ermon, "Audio super resolution using neural networks," *arXiv preprint arXiv:1708.00853*, 2017.

[10] S. Iizuka, E. Simo-Serra, and H. Ishikawa, "Globally and locally consistent image completion," *ACM Transactions on Graphics (TOG)*, vol. 36, no. 4, pp. 107:1–107:14, 2017.

[11] C. Yang, X. Lu, Z. Lin, E. Shechtman, O. Wang, and H. Li, "High-resolution image inpainting using multi-scale neural patch synthesis," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 6721–6729.

[12] J. C. Brown, "Calculation of a constant q spectral transform," *The Journal of the Acoustical Society of America*, vol. 89, no. 1, pp. 425–434, 1991.

[13] J. Pons, T. Lidy, and X. Serra, "Experimenting with musically motivated convolutional neural networks," in *14th International Workshop on Content-Based Multimedia Indexing (CBMI)*, 2016, pp. 1–6.

[14] P. Chandna, M. Miron, J. Janer, and E. Gómez, "Monoaural audio source separation using deep convolutional neural networks," in *International Conference on Latent Variable Analysis and Signal Separation (LVA/ICA)*, 2017, pp. 258–266.

[15] T. Jehan, *Creating music by listening*, Ph.D. thesis, Massachusetts Institute of Technology, School of Architecture and Planning, Program in Media Arts and Sciences, 2005.

[16] N. Perraudin, N. Holighaus, P. Majdak, and P. Balazs, "Inpainting of long audio segments with similarity graphs," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 2018, *(In Press)*.

[17] G. A. Velasco, N. Holighaus, M. Dörfler, and T. Grill, "Constructing an invertible constant-Q transform with non-stationary Gabor frames," *14th International Conference on Digital Audio Effects (DAFx-11)*, pp. 93–99, 2011.

[18] C. Schörkhuber, A. Klapuri, N. Holighaus, and M. Dörfler, "A Matlab toolbox for efficient perfect reconstruction time-frequency transforms with log-frequency resolution," in *Audio Engineering Society Conference: 53rd International Conference: Semantic Audio*. Audio Engineering Society, 2014.

[19] P. Smaragdis and B. Raj, "Example-driven bandwidth expansion," in *IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*, 2007, pp. 135–138.

[20] C. Dong, C. C. Loy, K. He, and X. Tang, "Learning a deep convolutional network for image super-resolution," in *European Conference on Computer Vision*, 2014, pp. 184–199.

[21] P. Jax and P. Vary, "Artificial bandwidth extension of speech signals using MMSE estimation based on a hidden Markov model," in *IEEE International Conference on Acoustics, Speech, and Signal Processing*, 2003, vol. 1, pp. I–680–I–683.

[22] X. Mao, C. Shen, and Y-B. Yang, "Image restoration using very deep convolutional encoder-decoder networks with symmetric skip connections," in *Advances in neural information processing systems*, 2016, pp. 2802–2810.

[23] M. Miron, J. Janer, and E. Gómez, "Generating data to train convolutional neural networks for classical music source separation," in *14th Sound and Music Computing Conference*, 2017, pp. 227–233.

[24] M. Miron, J. Janer, and E. Gómez, "Monaural score-informed source separation for classical music using convolutional neural networks," in *18th International Society for Music Information Retrieval Conference (ISMIR)*, 2017, pp. 55–62.

[25] M. D. Plumbley, T. Blumensath, L. Daudet, R. Gribonval, and M. E. Davies, "Sparse representations in audio and music: from coding to source separation," *Proceedings of the IEEE*, vol. 98, no. 6, pp. 995–1005, 2010.

[26] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, 2012, pp. 1097–1105.

[27] S. Zagoruyko and N. Komodakis, "Wide residual networks," *arXiv preprint arXiv:1605.07146*, 2016.

[28] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[29] A. A. Nugraha, A. Liutkus, and E. Vincent, "Multichannel audio source separation with deep neural networks," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 24, no. 9, pp. 1652–1664, 2016.

[30] R. M. Bittner, J. Salamon, M. Tierney, M. Mauch, C. Cannam, and J. P. Bello, "Medleydb: A multitrack dataset for annotation-intensive MIR research," in *15th International Society for Music Information Retrieval Conference (ISMIR)*, Taipei, Taiwan, 2014, pp. 155–160.

[31] E. Vincent, R. Gribonval, and C. Févotte, "Performance measurement in blind audio source separation," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 14, no. 4, pp. 1462–1469, 2006.

[32] Z. Průša, P. Balazs, and P. L. Søndergaard, "A noniterative method for reconstruction of phase from STFT magnitude," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 25, no. 5, pp. 1154–1164, 2017.

[33] M. Blaauw and J. Bonada, "A neural parametric singing synthesizer," *arXiv preprint arXiv:1704.03809*, 2017.

[34] V. Emiya, E. Vincent, N. Harlander, and V. Hohmann, "Subjective and objective quality assessment of audio source separation," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 19, no. 7, pp. 2046–205, 2011.

[35] Christian R Helmrich, Andreas Niedermeier, Sascha Disch, and Florin Ghido, "Spectral envelope reconstruction via igf for audio transform coding," in *Acoustics, Speech and Signal Processing (ICASSP), 2015 IEEE International Conference on*. IEEE, 2015, pp. 389–393.

A HOLISTIC GLOTTAL PHASE-RELATED FEATURE

Aníbal J. Ferreira *

Department of Electrical and Computers Engineering
Faculty of Engineering - University of Porto
Porto, Portugal
ajf@fe.up.pt

José M. Tribolet

Department of Computer Science and Engineering
Instituto Superior Técnico - University of Lisbon
Lisbon, Portugal
jose.tribolet@inesc.pt

ABSTRACT

This paper addresses a phase-related feature that is time-shift invariant, and that expresses the relative phases of all harmonics with respect to that of the fundamental frequency. We identify the feature as Normalized Relative Delay (NRD) and we show that it is particularly useful to describe the holistic phase properties of voiced sounds produced by a human speaker, notably vowel sounds. We illustrate the NRD feature with real data that is obtained from five sustained vowels uttered by 20 female speakers and 17 male speakers. It is shown that not only NRD coefficients carry idiosyncratic information, but also their estimation is quite stable and robust for all harmonics encompassing, for most vowels, at least the first four formant frequencies. The average NRD model that is estimated using data pertaining to all speakers in our database is compared to that of the idealized Liljencrants-Fant (L-F) and Rosenberg glottal models. We also present results on the phase effects of linear-phase FIR and IIR vocal tract filter models when a plausible source excitation is used that corresponds to the derivative of the L-F glottal flow model. These results suggest that the shape of NRD feature vectors is mainly determined by the glottal pulse and only marginally affected by either the group delay of the vocal tract filter model, or by the acoustic coupling between glottis and vocal tract structures.

1. INTRODUCTION

DFT-based phase processing of speech and musical sounds has been addressed since the birth of signal processing, early in the 60s of the 20th century. As a strong motivation, the theory of Fourier analysis of continuous and discrete signals was already well established, in particular concerning periodic signals, whose spectrum consists of a harmonic structure of sinusoidal components. However, owing to i) the discrete nature of the DFT and its underlying circular properties, ii) the specificity of popular and practical optimization metrics which emphasize quadratic measures, and iii) to a belief that to a considerable extent the 'human ear is insensitive to phase', phase processing in DFT analysis has not received as much attention as magnitude-based processing. A clear evidence of this reality is given by the simple fact that most front-ends for speech recognition and even speaker identification rely on the extraction of acoustic features that are based on spectral magnitude information only. Another reason explaining this reality involves the meaning of phase, especially the meaning in a psychoacoustic sense. Here again, the psychoacoustic meaning that is associated with the spectral magnitude is quite obvious and appealing: for example, it helps to explain pitch (i.e. the fundamental frequency), timbre, dark sounds (low-pass signals) and bright sounds (high-pass signals). On the contrary, phase was never associated with such a clear psychoacoustic interpretation and, in a large number of signal processing applications, such as spectral subtraction in noise reduction, phase is either ignored or simply discarded.

In this paper, we provide an illustrated motivation to the importance of phase as a relevant holistic feature for locally periodic signals, and we focus on its importance to characterize the periodic component of the glottal excitation. Although an in-depth treatment will be addressed in a forthcoming paper, here we use both synthetic and natural voice signals, notably vowel sounds, in order to illustrate holistic phase patterns that reflect idiosyncratic traits due mainly to the periodic glottal source, to illustrate the human diversity in vocal fold operation, and to evaluate how close popular models of the glottal pulse are to practical results.

In this section, we will briefly mention how phase has been looked at and acted upon notably in such areas as speech coding [1, 2] and time-scale modification of speech [3, 4].

Work in speech coding, during the 60e and 70s of the 20th century, especially in the area of frequency-domain coding of speech, has regarded phase as a frequency-domain parameter that could be quantized and coded or replaced by a synthetic phase, on a DFT coefficient basis. With the help of real transforms, such as the Discrete Cosine Transform (DCT), phase was even avoided -at least explicitly- and the focus was rather concentrated on adaptive quantization schemes defining how coarsely or finely the DCT coefficients should be quantized such as to minimize an objective distortion, or such as to minimize the perceptual impact of the quantization and coding noise. Later on, in the 70s, 80s, and 90s, these same principles were applied to wideband speech and high-quality audio coding. In this context, explicit phase-based processing was also avoided by using the Modified DCT [5].

An important class of speech algorithms dealing directly with the DFT phase representation involve time-scale and pitch modification of voiced regions in speech [6, 7]. Although first methods were oriented to phase processing on a DFT coefficient by coefficient basis, the associated subjective quality was considered poor as it was characterized by signal smearing, reverberation and 'phasiness' [8]. Techniques addressing this problem implemented phase modification while preserving certain phase relationships among neighboring DFT channels (or bins) in the region of a local maximum in the magnitude spectrum, a technique known as 'phase locking' [8, 9]. Another category of phase modification involved the harmonics of a periodic waveform. The goal was to preserve the local shape of the waveform even when its duration is artificially modified while preserving the fundamental frequency, or when its fundamental frequency is modified while pre-

* This work was financed by FEDER - Fundo Europeu de Desenvolvimento Regional funds through the COMPETE 2020-Operacional Programme for Competitiveness and Internationalization (POCI), and by Portuguese funds through FCT-Fundação para a Ciência e a Tecnologia in the framework of the project POCI-01-0145-FEDER-029308.

serving the duration. To a significant extent, shape-invariance was implemented in order to avoid the typical poor subjective quality of vocoders and other frequency-domain methods that focused on magnitude modification in the Fourier domain. Phase processing tried as much as possible to preserve the local phase relationships among harmonic frequencies, especially near pitch pulse onset times, because these instants were believed to represent the time 'at which sine waves add coherently' [7, 497], i.e. when they are presumed to be in phase. To our knowledge, this assumption was never really demonstrated and in fact chances are that at pitch pulse onset times the different harmonic frequencies are combined with the same phase relationship, but not necessarily in phase. Furthermore, these methods also depended on robust phase-unwrapping algorithms [10], not only to estimate pitch, but also to create extended phase models allowing to modify the time and frequency scales of a periodic waveform.

With exception of a few works including Di Federico [11] and Saratxaga [12] that we will address in the next section, those phase locking rules, as well as the shape-invariant harmonic phase modification criteria, were not framed as an interpretable holistic phase-related feature, or model, that is amenable to statistical analysis, modification and re-synthesis.

The same remark can also be made regarding the use of phase-related information in speaker recognition. Attempts have been made to include phase directly extracted from a DFT analysis of the speech signal [13], or by first processing it such as to compute a Group Delay Function (GDF) [14]. However, even in this case, phase has been looked at as an additional signal feature conveying information that complements that already provided by classic Mel-Frequency Cepstral Coefficients (MFCCs) [15], and that authors believed to be linked to the glottal source excitation. Yet, a psychophysical meaning was not attached to those phase-related features. In addition, it is quite intriguing that phonetic-oriented segmentation is typically not used to govern phase estimation in this context, which would be particularly meaningful in voiced regions of the speech.

In this paper, we briefly describe and illustrate, with the help of practical examples, a holistic phase-related feature, or model, that is linked to the harmonic phases of a periodic waveform, and that is (time) shift-invariant and independent on the pitch.

The reminder of this paper is organized as follows. In Sec. 2 we explain the nature of NRD and we illustrate it with a simple practical example. In Sec. 3 we illustrate NRD estimation with real vowel sounds. In Sec. 4 we use synthetic and natural signals to characterize the influence of the vocal tract filter on the phase characteristics of the glottal excitation. Section 5 discusses NRD models that may be used to describe the periodic part of the glottal excitation of humans. Finally, Sec. 6 summarizes the main results of this paper and discusses future work.

2. A SHIFT-INVARIANT PHASE-RELATED FEATURE

The holistic phase feature we focus on in this paper emerges directly from the Fourier analysis of the harmonics of a periodic wave. A meaningful way to introduce it is by means of a simple practical example [16]. We use the well known sawtooth waveform which is synthesized using the Fourier series comprising L terms:

$$x(t) = \sum_{\ell=1}^{L} \frac{A}{\pi\ell} \sin \frac{2\pi}{T} \ell t \,, \tag{1}$$

where A represents amplitude and T represents the reciprocal of the pitch. Although the NRD coefficients can be found directly form any periodic wave, for illustration purposes we use the derivative of the sawtooth waveform which can be easily obtained as

$$\frac{d}{dt}x(t) = \sum_{\ell=1}^{L} \frac{2A}{T} \cos \frac{2\pi}{T} \ell t = \sum_{\ell=1}^{L} \frac{2A}{T} \sin \left(\frac{2\pi}{T} \ell t + \frac{\pi}{2} \right) \,. \tag{2}$$

This form is very convenient because it highlights the phase at the *sinusoidal onset* of each harmonic. Let us now split this result in a part consisting of the fundamental frequency, and another part grouping all harmonics:

$$\begin{aligned} \frac{d}{dt}x(t) &= \frac{2A}{T} \sin \left(\frac{2\pi}{T} t + \phi_0 \right) + \sum_{\ell=2}^{L} \frac{2A}{T} \sin \left(\frac{2\pi}{T} \ell t + \phi_\ell \right) \\ &= \frac{2A}{T} \sin \frac{2\pi}{T} \left(t + t_0\right) + \sum_{\ell=2}^{L} \frac{2A}{T} \sin \frac{2\pi}{T} \ell \left(t + \ t_\ell\right) \,, \end{aligned}$$

where $t_0 = T\phi_0/(2\pi)$ and $t_\ell = T\phi_\ell/(2\pi\ell)$ represent the absolute time-shifts of the different terms of the Fourier series. If we concentrate on the second part of this development, we may conveniently introduce a relative time-shift:

$$\begin{aligned} & \sum_{\ell=2}^{L} \frac{2A}{T} \sin \frac{2\pi}{T} \ell \left(t + t_0 + (t_\ell - t_0)\right) \\ = & \sum_{\ell=2}^{L} \frac{2A}{T} \sin \left(\frac{2\pi}{T} \ell (t + t_0) + 2\pi \frac{(t_\ell - t_0)}{T/\ell} \right) \\ = & \sum_{\ell=2}^{L} \frac{2A}{T} \sin \left(\frac{2\pi}{T} \ell (t + t_0) + 2\pi \frac{(\phi_\ell - \ell\phi_0)}{2\pi} \right) \\ = & \sum_{\ell=2}^{L} \frac{2A}{T} \sin \left(\frac{2\pi}{T} \ell (t + t_0) + 2\pi \mathrm{NRD}_\ell \right) \,. \end{aligned} \tag{3}$$

In Eq. (3), NRD_ℓ denotes Normalized Relative Delay (NRD) and expresses a relative delay between harmonic ℓ and the fundamental, which is further normalized by the period of the harmonic [17]. Although the acronym is reminiscent of the way each NRD coefficient is computed in practice, NRDs reflect simply a normalized value in the range $[0.0,\ 1.0[$ which depends on a difference involving the phase of the harmonic and the phase of the fundamental. Thus, the number of NRD coefficients equals the number of harmonics. Other important properties of the NRD coefficients are as follows:

- as a relative phase-related feature, the NRD of the fundamental is zero by definition,
- because NRDs express phase differences, the concepts of phase wrapping and phase unwrapping also apply, in this paper unwrapped NRDs are used since this facilities modeling and understanding,
- NRDs are intrinsically time-shift invariant, and are also independent on the fundamental frequency.

Hence, NRDs express phase relationships that, in addition to the magnitude of the harmonics, explain the shape of a specific periodic waveform, and thus completely define its shape invariance.

The NRD concept has been independently introduced in [17], and has found practical application in singing voice analysis [18],

glottal source modelling [19], speaker identification [16], parametric audio coding [20] and dyspohonic voice reconstruction [21]. It was recently brought to our attention that a similar concept (Relative Phase Delay) had been presented in 1998 by Di Federico [11]. Other smooth phase descriptors for harmonic signals that are similar to NRD were also proposed by Stylianou in 1996 (phase envelope [22, page 44]) and Saratxaga in 2009 (Relative Phase Shift -RPS [12]). Our NRD estimation is closer to the method proposed by Di Federico [11] (that estimates $(t_\ell - t_0)/(T/\ell)$) than that proposed by Saratxaga [12] (that estimates $\phi_\ell - \ell\phi_0$).

To complete the illustration using our example, we use the phase values in Eq. (2) to obtain $\text{NRD}_\ell = \frac{\pi/2 - \ell\pi/2}{2\pi} = \frac{1-\ell}{4}$, $\ell = 2, \ldots, L$. We have synthesized Eq. (2) using $L = 20$ harmonics and 22050 Hz sampling frequency (FS). We obtained the NRD numerical results using the algorithm described in [17] and they are represented in Fig. 1. This algorithm uses phase unwrapping and it can be seen that results are as expected. In particular, for $\ell = 20$, the NRD becomes -4.75. This figure also represents the

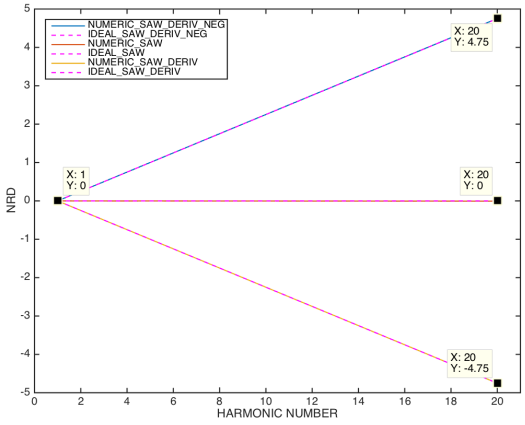


Figure 1: *Unwrapped NRD estimation results for the sawtooth wave, its derivative and its negative derivative. Ideal (analytical) and experimental results are overlapped.*

experimental results regarding the waveform described by Eq. (1), in which case all NRDs are clearly zero. We conclude the illustration of the NRD concept using another synthetic signal alternative. Taking the negative of Eq. (2), we obtain

$$-\frac{d}{dt}x(t) = -\sum_{\ell=1}^{L}\frac{2A}{T}\cos\frac{2\pi}{T}\ell = \sum_{\ell=1}^{L}\frac{2A}{T}\sin\left(\frac{2\pi}{T}\ell t - \frac{\pi}{2}\right), \tag{4}$$

which highlights that the phases at the sinusoidal onset of all harmonics, are all equal to $-\pi/2$. It follows that $\text{NRD}_\ell = \frac{-\pi/2 + \ell\pi/2}{2\pi}$, or $\text{NRD}_\ell = \frac{\ell-1}{4}$, $\ell = 2, \ldots, L$. In particular, for $\ell = 20$, the NRD becomes 4.75. This result is also illustrated in Fig. 1. The experimental results are also shown and it can be seen that the agreement is clear.

Although the NRD concept is a simple one to grasp, the actual computation, or estimation, is less trivial. The major difficulty is that the phases at sinusoidal onsets are not readily available from the DFT or similar transform. What is available is phase information that is referred to a time instant (or sample) corresponding to the delay of the DFT filter bank, and which also depends on the influence of the time analysis window prior to DFT transformation. Thus, this influence must first be compensated for, then phase information (ϕ_ℓ) is converted into time delays (n_ℓ) which are made relative to the time delay of the fundamental (n_0), and further wrapped using the period of each harmonic (P_ℓ). Finally, a normalization by each harmonic period is applied [17]. Fig. 2 illustrates the NRD estimation algorithm. We use the Odd-DFT [23]

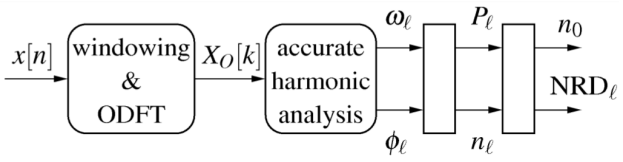


Figure 2: *NRD estimation algorithm [17].*

instead of the plain DFT due to a number of interesting properties which facilitate accurate estimation of the frequencies, phases and magnitudes of the sinusoidal components that exist in a signal. Thus, accurate frequency and phase estimation of each individual sinusoidal component [24] is very important to the reliability, accuracy and robustness of the NRD estimation algorithm.

3. A HOLISTIC PHASE DESCRIBING VOICED SOUNDS

In this section, we present first results for a holistic phase-related feature that consists of unwrapped NRD coefficients. These coefficients are obtained from the accurate frequency analysis, as described in Sec. 2, of the spectrum of voiced vowel signals. The signals correspond to sustained vowel utterances produced by 37 subjects of which 20 are female, and 17 are male. The recordings that are included in the data base were obtained for forensic purposes, focusing on speaker identification, and are described in [25]. Figure 3 represents the magnitude spectrum of an /a/ vowel segment uttered by a female speaker (upper panel), and an overlay of all NRD vectors that are estimated in a sustained vowel region (lower panel) lasting about 1 second. The harmonic structure is signaled in the magnitude spectrum by means of vertical triangles. The dashed line in this figure represents the LPC model (order 22) of the spectral envelope defined by the peaks of all harmonics.

The overlay of NRD vectors suggest a few interesting conclusions. First, a region of consistent and stable NRD coefficients is apparent that involves the first 20 harmonics. These harmonics happen to be the strongest before the spectral valley located at around 4500 Hz. When harmonics have a very small magnitude or are close to the noise floor, then accurate frequency, phase and magnitude estimation is adversely affected in a significant way. Higher order harmonics are also more prone to estimation inaccuracies because their period is quite short, in the order of 3 speech samples or less. Since the period of each harmonic is individually estimated, accounting for some degree of inharmonicity, then shorter periods are more likely to be affected by noise or interferences and, thus, the phase estimation also becomes more unreliable. The impact in terms of unwrapped NRD estimation is a spreading of the NRD values as illustrated in Fig. 3 which may generate visually appealing patterns. However, this spreading is not problematic mainly for two reasons. First, the most important voice formant frequencies are typically accommodated by the NRD region that is stable. Secondly, and this is especially impor-

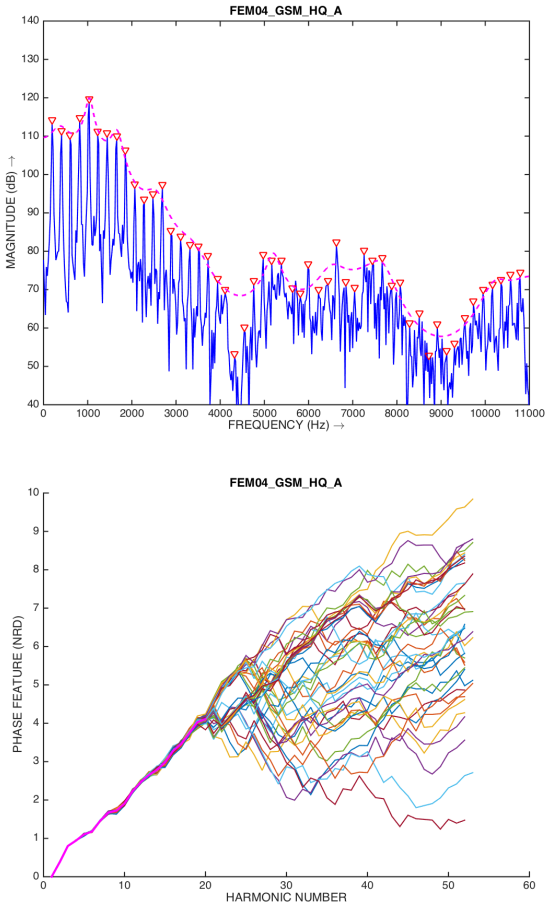


Figure 3: *Magnitude spectrum of a voiced /a/ vowel segment uttered by a female speaker (upper figure). The vertical triangles signal the harmonic structure. The lower figure represents all (unwrapped) NRD vectors found in a sustained /a/ vowel region and that includes the represented magnitude spectrum. The thick magenta line represents the average NRD vector up to harmonic 19.*

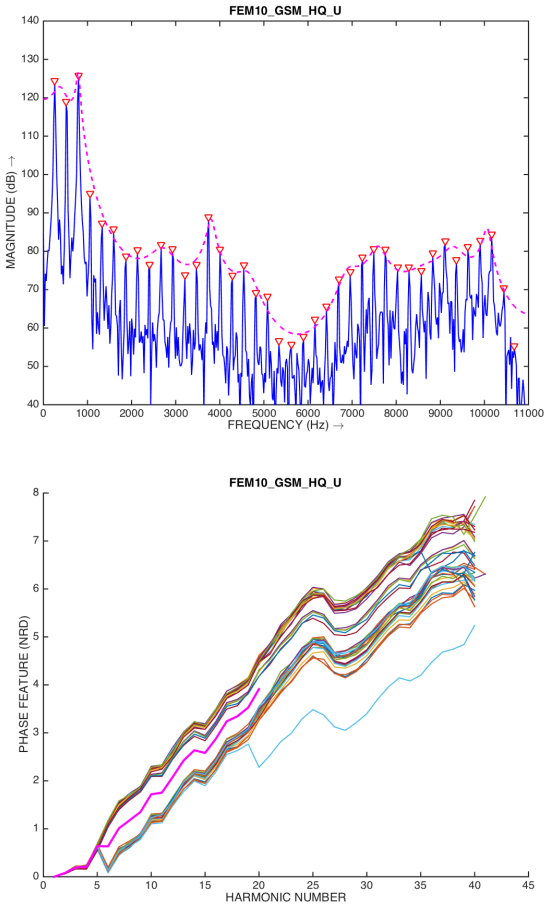


Figure 4: *Magnitude spectrum of a voiced /u/ vowel segment uttered by a female speaker (top). The vertical triangles signal the harmonic structure. The lower figure represents an overlay of all (unwrapped) NRD vectors found in the /u/ vowel region. The thick magenta line represents the average NRD vector up to partial 19.*

tant for synthesis purposes -which is not discussed in this paper-, the NRDs in the 'wild' region, i.e. the region where an exuberant NRD spreading can be observed, can be replaced by the new NRDs that are extrapolated from the stable NRD region.

Figure 4 represents a magnitude spectrum and a peculiar overlay of NRD vectors pertaining to a /u/ vowel uttered by a female. Since this is a back vowel whose two relevant formants have a very low frequency, then the NRD vector is stable only for the first few harmonics, five in this case. Although for other speakers, the stable NRD region may be wider even for this difficult vowel, that has no real relevance as just the first few harmonics define the vowel, both linguistically and in terms of quality.

Figure 5 illustrates the magnitude spectrum and a overlay of NRD vectors pertaining to a /o/ vowel uttered by a male. Since the pitch is about one octave lower than in the case of a female voice, the harmonic density is higher and NRD vectors may have as many as 100 coefficients within the Nyquist range. It can be confirmed in Fig. 5 that the first 4 formant frequencies are represented by the first 42 harmonics, which corresponds to the stable NRD region.

The above results suggest that, in most cases, it is safe to assume that the first 19 coefficients represent stable NRD vectors. Figure 6 illustrates the average NRD vectors for sustained vowel regions pertaining to five different vowels uttered by a male speaker. Results are presented for two repetitions of the same vowel exercise. It can be seen that the profile of the different average NRD vectors are in good agreement, which suggests that there is a trend that is common even for different vowels uttered by the same speaker. Rather than the vocal tract filter, which varies from vowel to vowel realization, what is really common in these situation is the glottal excitation which is mainly characterized by a periodic part due to the vibration of the vocal folds. Thus, the NRDs appear to be mainly determined by the shape of the glottal pulse. It should be noted however that for some speakers, the NRD vectors estimated from /i/ or /u/ vowel regions may deviate from the NRD trend defined by the remaining vowels. As explained above, this may be due to the fact that certain harmonics are very weak, such as in the case of the /i/ vowel which has the largest F1-F2 formant separation, or in the case of the /u/ vowel whose harmonics decay quite strongly just after the F1 and F2 formants.

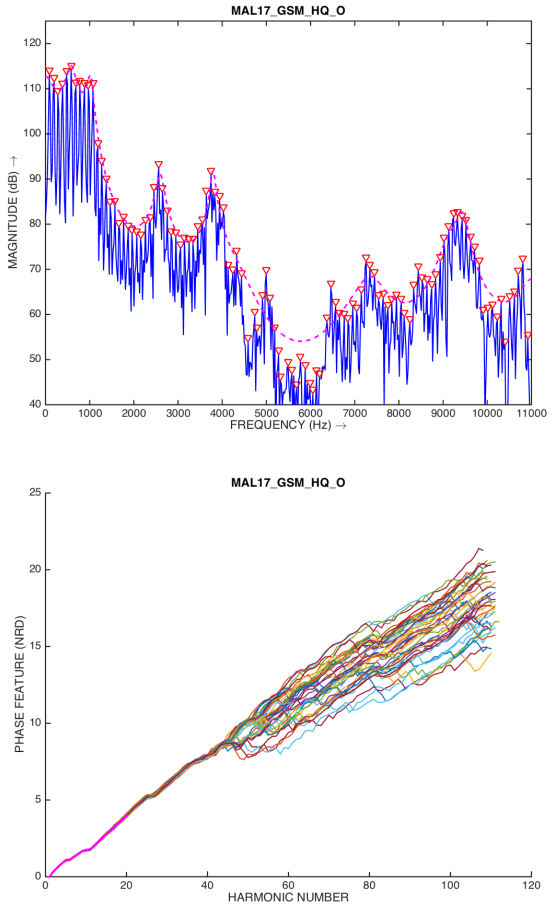


Figure 5: *Magnitude spectrum of a voiced /o/ vowel segment uttered by a male speaker (top). The vertical triangles signal the harmonic structure. The lower figure represents an overlay of all (unwrapped) NRD vectors found in the /o/ vowel region. The thick magenta line represents the average NRD vector up to partial 19.*

4. VOCAL TRACT FILTER PHASE EFFECTS USING SYNTHETIC AND NATURAL VOICED SOUNDS

According to the ideal source-filter model of voice production [26, 27], the signal generated at the glottis is the source signal and includes a stochastic and a periodic part. The supralaryngeal structures, including the oral and nasal cavities, shape the source signal in time and frequency such as to convey a desired linguistic message. This time and frequency shaping, which is mainly influenced to the vocal tract resonant frequencies -also commonly referred to as formants-, is modeled as a filter which may be considered as stationary for sustained sounds, or locally stationary in running speech considering the average syllabic duration, in the order of 10 to 20 ms. Most frequently, the filter is modeled as an all-pole filter; in our experiments, as indicated in Sec. 3, we use a 22nd-order LPC model. The filter may also include the radiation effect due mainly to the lips and nostrils. The radiation effect is usually modeled as a time differentiation operation that converts the air flow into sound pressure.

A very interesting issue that to our knowledge has never been clarified in the literature, deals with the phase contribution due

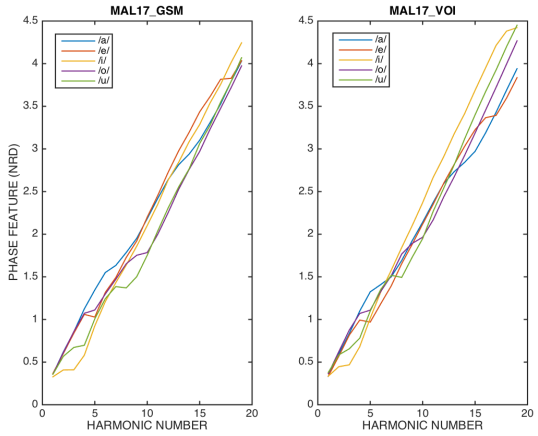


Figure 6: *Average NRD vectors (19-dimensional) obtained from sustained vowels uttered by a male speaker. Results are presented for 5 vowels produced during two different conversations.*

to the source excitation, and that due to the filter. The combined effects are known to be additive in terms of phase or, equivalently, in terms of group delay. However, the clarification of how much the phase contribution -or group delay- due to the filter modifies the phase of the source signal is an open issue.

Using the NRD concept and using the results that were illustrated in the previous section, we may shed some light on the issue. In that regard, we will assume as a plausible periodic glottal source excitation, the derivative of the Liljencrants-Fant model (L-F) of glottal flow [28]. A 210 Hz fundamental frequency glottal excitation using the L-F model has been conveniently generated using the freely available Voicebox Matlab toolbox (FS=22050 Hz).

Figure 7 illustrates a few periods of the L-F glottal flow derivative (upper panel), the corresponding magnitude spectrum with all harmonics signaled by means of vertical triangles (middle panel), and the unwrapped NRD coefficients up to harmonic 50. This fig-

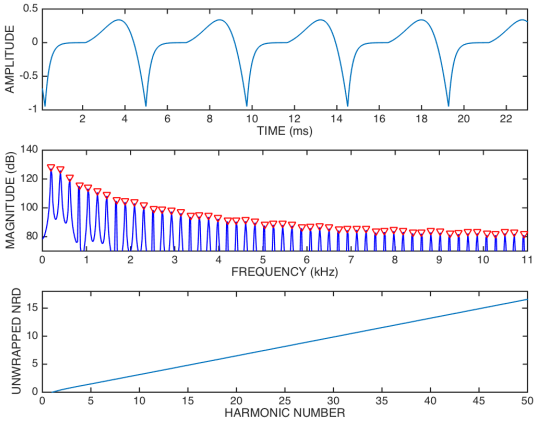


Figure 7: *Analysis of the derivative of the L-F glottal flow model. The top panel represents the time waveform and its magnitude spectrum is represented in the middle panel. The harmonics are signaled by red triangles and the unwrapped NRD coefficients pertaining to the first 50 harmonics are represented in the lower panel.*

ure suggest that the NRD feature vector may be faithfully approximated by means of a simple first order model that is given by

$$\text{NRD}_\ell = -0.207431 + 0.335465\ell,\ \ell = 2, \ldots, L. \qquad (5)$$

As indicated previously, by definition $\text{NRD}_\ell = 0,\ \ell = 1$.

Concerning the filter model, we took advantage of all the LPC models (order 22) that were obtained for all vowels from all speakers. Figures 3, 4 and 5 represent examples of the magnitude frequency responses of the IIR filters corresponding to those models. We took the average of all models separately for vowels /a/, /e/ and /i/. We considered female models only as the formant frequencies characterizing a given vowel, are typically higher in female voices than in male voices (due to anatomical differences between male and female speakers). Then, using the average power spectral density (PSD) of those models, we designed a linear-phase FIR filter (500 taps) and an IIR filter (order 22) having a magnitude frequency response approximating that PSD. The FIR filter has been obtained using a single band Parks-MccClellan optimal equiripple design. The IIR has been obtained using the Levinson-Durbin recursion and after the autocorrelation coefficients are obtained from the PSD using the Wiener-Khintchine theorem. Figure 8 represents the PSD of the average /e/ vowel, as well as the magnitude frequency responses of the FIR and IIR filters. It can be seen that

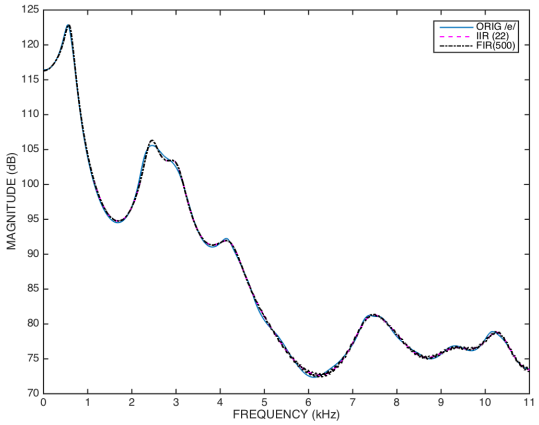


Figure 8: *Average model of the PSD of vowel /e/ uttered by female speakers, and magnitude frequency responses of a linear-phase FIR filter (order 500), and an IIR filter (order 22) approximating that PSD.*

both filters approximate well the PSD. An obvious (and intented) difference lies however in the phase response of both filters. In fact, the linear-phase FIR has a constant group delay response (249.5 samples) while the IIR exhibits a non linear group-delay response that is represented in Fig. 9. Assuming the L-F model as a plausible excitation to the filter, we want to assess how much the NRD coefficients at the output of the filter are affected by the group delay of the filter, according to the two alternatives: linear-phase FIR and IIR filter modeling. In other words, how much are the phase properties of the source excitation affected by the phase properties of the filter ?

To answer this question we filtered the source excitation illustrated in Fig. 7 using the two alternative filters and then, in each case, we extracted the NRD feature vector of the output signals.

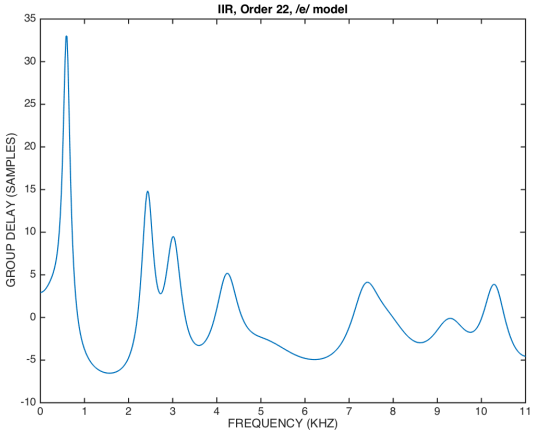


Figure 9: *Group delay of the 22nd-order IIR filter approximating the average PSD of the /e/ vowel uttered by female speakers.*

As indicated above, we repeated the experiment for vowels /a/, /e/ and /i/. In rigour, prior to this operation, we should have compensated the spectral magnitude of the excitation by its spectral tilt such that the signal at the output of the filter exhibits a PSD which corresponds to that of the original vowel PSD. Ignoring this step has however no consequences regarding phase, is just produces an output PSD which has a stronger spectral tilt than the original.

Figure 10 illustrates the NRD feature vector at the output of both filters and taking as a reference the original NRD feature of the excitation. It can be seen that, as expected, in the case of the linear-phase FIR filter, because the group delay is constant, then no modification takes place. However, in the case of the IIR filter, then visible modifications take place, although these do not represent a dramatic modification of the trend defined by the source excitation, exception for vowel /i/. In this case, a plausible explanation is that the group delay of the corresponding filter is such that it modifies significantly the NRD trend of the source excitation. Further research is required to clarify this. Considering however that this vowel represents an exception, it is interesting to compare these results that presume a synthetic excitation signal, and the results displayed in Fig. 6 that were obtained for real natural voices. In both situations, results suggest the vocal/nasal tract filter modifies the phase properties of the glottal excitation although not too strongly as the overall trend in the NRD feature vector of the source excitation is essentially preserved. We believe this is an innovative result that emerges from experimental data with NRDs. It can also be argued that the deviations to the excitation NRD feature vector, after the filter, may be due to the acoustic coupling between the glottis and the vocal/nasal tracts for different configurations of the latter and which modify slightly the shape of the glottal pulse. Clarifying this hypothesis would however imply complex and invasive experiments capturing the signal near the vocal folds.

5. A MODEL OF THE HUMAN GLOTTAL PHASE

In this section, we discuss NRD models that may be used to describe the holistic phase structure of the periodic part of the human glottal excitation.

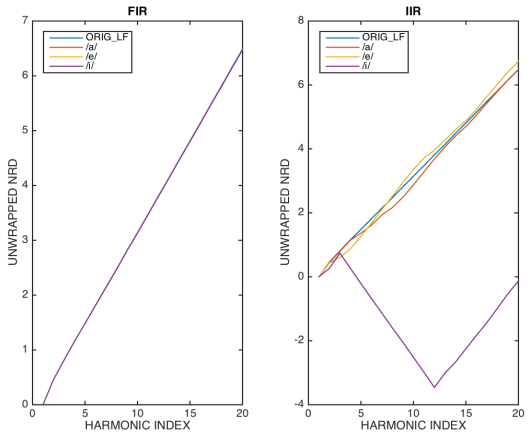


Figure 10: *Illustration of NRD modification of the source excitation due to the phase properties of the filter modeling the PSD of three vowels: /a/, /e/ and /i/. When the filter is a linear-phase FIR filter, no modification exists. When the filter is a 22nd-order IIR filter, its group delay modifies slightly the original NRD feature vector. A strong deviation is observed in the case of vowel /i/.*

Figure 11 represents an overlay of all the average NRD feature vectors that were obtained from the 5 vowel realizations by each speaker. As our data base includes 37 speakers and each speaker has produced two independent realizations for each vowel, Fig. 11 represents 74 true human average NRD data. This figure also

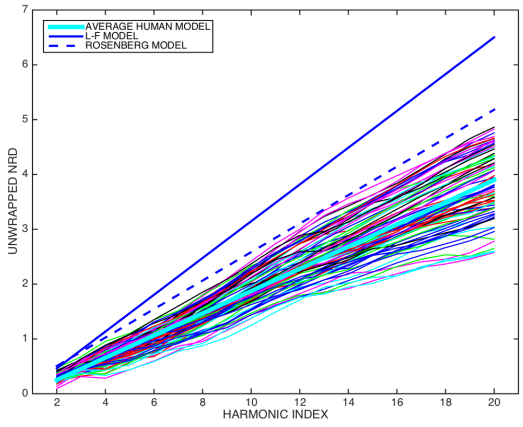


Figure 11: *Overlay of all the average NRD feature vectors for the 5 vowels uttered by each one of the 37 speakers in our data base. The experimental NRD vector of the derivative of both ideal Rosenberg and L-F glottal flow models are also represented.*

represents the NRD feature vectors that have been obtained experimentally from synthetic signals consisting of the derivative of the ideal L-F glottal flow model, and the derivative of the Rosenberg glottal flow model. Both models were generated using the Voicebox toolbox. The L-F NRD model is well approximated by Eq. (5) and has already been illustrated in Figs. 7 and 10. Figure 11 also represents the average NRD model of all human vowel realizations, its first order best approximation is given by

$$\mathrm{NRD}_\ell = -0.1522222 + 0.2025505\ell,\ \ell = 2, \ldots, L\,. \quad (6)$$

For the sake of completeness, the first order best approximation to the Rosenberg NRD model is given by

$$\mathrm{NRD}_\ell = -0.014001 + 0.259785\ell,\ \ell = 2, \ldots, L\,. \quad (7)$$

It can be seen that the L-F NRD model deviates more from the experimental average human NRD model than the Rosenberg model.

To conclude this section, we present a verifiable example of the capability of NRDs in representing the holistic phase properties of any periodic wave. We prepared two `.mat` Matlab files, one of them (`LFmag.mat`) contains the first 20 harmonic magnitudes of the derivative of the L-F glottal flow model, and another one (`LFNRD.mat`) contains the first 20 NRD values pertaining to the corresponding harmonics, including the fundamental. These experimental-based magnitude and NRD values are used to synthesize the derivative of the L-F glottal flow model using

$$\mathrm{dgf}(t) = \sum_{\ell=1}^{L} \mathrm{LFmag}_\ell \cdot \sin\left(\frac{2\pi}{T}\ell t + 2\pi \cdot \mathrm{LFNRD}_\ell\right)\,. \quad (8)$$

In this synthesis we use a fundamental frequency of 210 Hz and FS=22050 Hz. The resulting signal is represented in Fig. 12. We

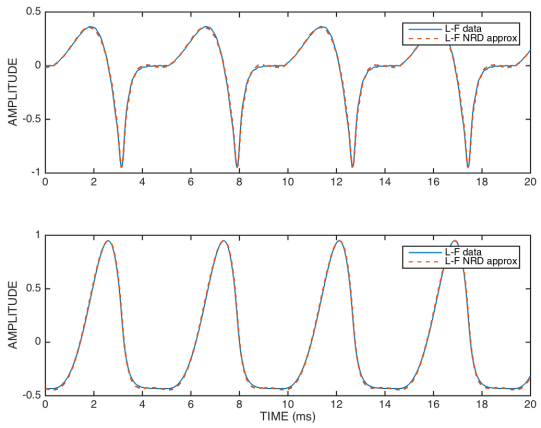


Figure 12: *L-F idealized glottal flow wave and its derivative using experimental data concerning the first 20 harmonic magnitudes and NRD coefficients. Versions of these signals are also shown that use a first-order NRD approximation. The Matlab code allowing to generate this figure is available.*

may then replace the accurate NRD coefficients LFNRD_ℓ by the approximate first-order model given by Eq. (5). The result of this approximation is also represented in Fig. 12. It can be concluded that the resulting wave is a faithful approximation to the original.

On the other hand, it is known from basic Fourier theory that if $X(j\Omega)$ is the Fourier transform of $x(t)$, then the Fourier transform of the integration of $x(t)$ is given by $X(j\Omega)/(j\Omega)$. This means that the magnitude of the Fourier transform is divided by the frequency, and $\pi/2$ is subtracted to the phase. Thus, the glottal

flow model, by integrating Eq. (8) and except for a scaling factor, is simply given by

$$\mathrm{gf}(t) = \sum_{\ell=1}^{L} \frac{\mathrm{LFmag}_\ell}{\ell} \cdot \sin\left(\frac{2\pi}{T}\ell t + 2\pi \cdot \mathrm{LFNRD}_\ell - \frac{\pi}{2}\right) . \quad (9)$$

This result, as well as its version when LFNRD_ℓ is approximated by its first-order model are also represented in Fig. 12. In order to facilitate the reproducibility of these results, the Matlab code generating Fig. 12 is available [1].

We have shown that we know how the holistic phase of the periodic part of the human glottal excitation looks like, future research will leverage on this result to more accurately estimate the spectral magnitude of the human glottal excitation.

6. CONCLUSION

We described in this paper how the NRD phase-related feature and that is extracted from the harmonics of a periodic waveform, effectively acts as an important holistic glottal feature that carries idiosyncratic information. NRD coefficients were shown to be moderately affected by the group delay of the vocal/nasal tract filters, or by the acoustic coupling between glottis and supra-laryngeal structures. We also identified several relevant first-order NRD approximation models, one of which represents the average NRD feature of the glottal excitation of a human speaker. Future work will include further research on phase effects of the vocal tract filter, the modeling of the glottal excitation spectral magnitude, and the application of the NRD features in such areas as speaker identification, high-quality parametric speech coding and dysphonic voice reconstruction.

7. REFERENCES

[1] J. M. Tribolet and R. E. Crochiere, "Frequency domain coding of speech," *IEEE Trans. on Acoustics, Speech and Sig. Proc.*, vol. 27, no. 5, pp. 512–530, Oct. 1979.

[2] Andreas S. Spanias, "Speech coding: A tutorial review," *Proc. of the IEEE*, vol. 82, no. 10, pp. 1541–1582, Oct. 1994.

[3] Eric Moulines and Jean Laroche, "Non-parametric techniques for pitch-scale and time-scale modification of speech," *Speech Com.*, vol. 16, pp. 175–205, 1995.

[4] Salim Roucos and Alexander M. Wilgus, "High quality time-scale modification of speech," in *IEEE ICASSP*, 1985, pp. 13.6.1–13.6.4.

[5] T. Painter and A. Spanias, "Perceptual coding of digital audio," *Proc. of the IEEE*, vol. 88, no. 4, pp. 451–513, 2000.

[6] M. R. Portnoff, "Time-scale modification of speech based on short time Fourier analysis," *IEEE Trans. on Ac., Speech and Sig. Proc.*, vol. 29, no. 3, pp. 374–390, June 1981.

[7] T. F. Quatieri and R. J. McAulay, "Shape invariant time-scale and pitch modification of speech," *IEEE Trans. on Signal Proc.*, vol. 40, no. 3, pp. 497–510, March 1992.

[8] Jean Laroche and Mark Dolson, "Improved phase vocoder time-scale modification of audio," *IEEE Trans. on Speech and Audio Proc.*, vol. 7, no. 3, pp. 323–331, May 1999.

[9] Aníbal J. S. Ferreira, "An Odd-DFT based approach to time-scale expansion of audio signals," *IEEE Trans. on Speech and Audio Proc.*, vol. 7, no. 4, pp. 441–453, July 1999.

[10] José M. Tribolet, "A new phase unwrapping algorithm," *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol. 25, no. 2, pp. 170–177, April 1977.

[11] Riccardo Di Federico, "Waveform preserving time stretching and pitch shifting for sinusoidal models of sound," in *COST-G6 Digital Audio Effects Workshop*, 1998, pp. 44–48.

[12] I. Saratxaga, I Hernaez, D. Erro, E. Navas, and J. Sanchez, "Simple representation of signal phase for harmonic speech models," *Electronic Letters*, vol. 45, no. 381, 2009.

[13] L. Wang, K. Minami, K. Yamamoto, and S. Nakagawa, "Speaker identification by combining MFCC and phase information in noisy environments," in *IEEE ICASSP*, 2010.

[14] P. Rajan, T. Kinnunen, C. Hanilci, J. Pohjalainen, and P. Alku, "Using group delay functions from all-pole models for speaker recognition," in *Proc. of Interspeech*, 2013.

[15] S. B. Davis and P. Mermelstein, "Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences," *IEEE Trans. on Ac., Speech and Sig. Proc.*, vol. 28, no. 4, pp. 357–366, August 1980.

[16] Aníbal Ferreira, "On the possibility of speaker discrimination using a glottal pulse phase-related feature," in *IEEE ISSPIT*, December 2014, Noida, India.

[17] Ricardo Sousa and Aníbal Ferreira, "Importance of the relative delay of glottal source harmonics," in *39th AES Int. Conf. on Audio Forensics*, 2010, pp. 59–69.

[18] Ricardo Sousa and Aníbal Ferreira, "Singing voice analysis using relative harmonic delays," in *Interspeech*, 2011.

[19] Sandra Dias and Aníbal Ferreira, "Glottal pulse estimation - a frequency domain approach," in *Speech Proc. Conf.*, July 2014, Tel-Aviv, Israel.

[20] Aníbal Ferreira and Deepen Sinha, "Advances to a frequency-domain parametric coder of wideband speech," *140th Convention of the AES*, May 2016, Paper 9509.

[21] Aníbal Ferreira, "Implantation of voicing on whispered speech using frequency-domain parametric modelling of source and filter information," in *ISIVC*, 2016, pp. 159–166, Tunis, Tunisia.

[22] I. Stylianou, *Harmonic + noise models for speech, combined with statistical methods, for speech and speaker modification*, Ph.D. thesis, École Nat. Sup. Télécom., France, 1996.

[23] Maurice Bellanger, *Digital Processing of Signals*, John Willey & Sons, 1989.

[24] Aníbal Ferreira and Deepen Sinha, "Accurate and robust frequency estimation in the ODFT domain," in *IEEE WASPAA*, Oct. 2005, pp. 203–206.

[25] Aníbal Ferreira and Vânia Fernandes, "Consistency of the F0, Jitter, Shimmer and HNR voice parameters in GSM and VOIP communication," in *DSP 2017*, 2017.

[26] G. Fant, *Acoustic Theory of Speech Production*, The Hague, 1970.

[27] Gunnar Fant, "Glottal flow: models and interaction," *Journal of Phonetics*, vol. 14, no. 3/4, pp. 393–399, 1986.

[28] G. Fant, J. Liljencrants, and Q. Lin, "A four-parameter model of glottal flow," *STL-QPSR*, vol. 26, no. 4, pp. 1–13, 1985.

[1] `http://www.fe.up.pt/~ajf/DAFx18_AJF_JMT.zip`

SOUND MORPHOLOGIES DUE TO NON-LINEAR INTERACTIONS : TOWARDS A PERCEPTUAL CONTROL OF ENVIRONMENTAL SOUND SYNTHESIS PROCESSES

Samuel Poirot

Aix Marseille Univ, CNRS, PRISM, Marseille, France
poirot@prism.cnrs.fr

Stefan Bilbao

Acoustics and Audio Group, James Clerk Maxwell Building, University of Edinburgh, EH9 3JZ, Edinburgh, United Kingdom
s.bilbao@ed.ac.uk

Mitsuko Aramaki

Aix Marseille Univ, CNRS, PRISM, Marseille, France
aramaki@prism.cnrs.fr

Richard Kronland-Martinet

Aix Marseille Univ, CNRS, PRISM, Marseille, France
kronland@prism.cnrs.fr

ABSTRACT

This paper is concerned with perceptual control strategies for physical modeling synthesis of vibrating resonant objects colliding non-linearly with rigid obstacles. For this purpose, we investigate sound morphologies from samples synthesized using physical modeling for non-linear interactions. As a starting point, we study the effect of linear and non-linear springs and collisions on a single-degree-of-freedom system and on a stiff strings. We then synthesize realistic sounds of a stiff string colliding with a rigid obstacle. Numerical simulations allowed the definition of specific signal patterns characterizing the non linear behavior of the interaction according to the attributes of the obstacle. Finally, a global description of the sound morphology associated with this type of interaction is proposed. This study constitutes a first step towards further perceptual investigations geared towards the development of intuitive synthesis controls.

1. INTRODUCTION

This paper is concerned with the perceptual control of environmental sound synthesis processes, based on the ecological approach to auditory events [1],[2]. This approach, adapted from the ecological approach to visual perception [3], supposes the existence of invariant structures (specific patterns in the perceived signal) that carry the necessary information for the recognition of sound events. These structures can be split in two groups: the structural invariants, which enable the recognition of properties of a sounding object and transformational invariants, that describe the transformations of the object. This theory was first exploited by Warren and Verbrugge concerning the auditory recognition of acoustic events [4]. Then, some studies have identified invariants containing sufficient information to discriminate the material [5] or the size [6] of impacted objects. More recently, This approach has inspired a conceptual description of sounds through an action-object paradigm [7],[8],[9].

Research into sound invariants is of great interest for the perceptual control of sound synthesis. Indeed, the definition of a morphology corresponding to an invariant allows for simplified control through the mapping of several synthesis parameters to one global parameter described perceptually. Thus, it allows for the control of sound synthesis processes using high level descriptors, according to perceptual measures.

This conceptual description has led synthesis processes based on the source-filter model. In [7],[8],[9] transformational invariants are responsible for the evocation of a sound-producing action (scratching, rolling), while structural invariants are responsible for the evocation of the exited object (shape, material, size). Hence, in the source-filter model the resulting sound is obtained by the convolution between the transformational invariant defining the source (action) and the structural invariant defining the filter (object).

One aim here is to develop new tools for sound designers, giving them an alternative to databases of recorded sounds for different applications such as video games. It leads to real-time synthesis of sounds in virtual or augmented reality environments directly controlled by the in-game events. In contrast to methods based on the use of a database of recorded sounds, such a synthesis procedure can adapt quickly to event occuring during gameplay. Also, it opens the perspective of generating unheard sounds that carry information contained in the sound invariants: "sound metaphors".

In previous studies, the mapping of perceptual features onto synthesis parameters for an intuitive control of sounds has been proposed [10]. Aramaki et al. developed an impact sound synthesizer intuitively controlled with semantic labels describing the perceived material, size and shape of the object [11],[12]. Here, the authors defined several labeled structural invariants (material, size and shape) in relation to signal properties (modes, damping). The impact synthesizer was extended to continuous-interaction sounds: rubbing, scratching, and rolling [9]. Here, transformational invariants are characterized as a statistical description of the excitation signal in relation to the perceived action. Also, Thoret et al. proposed a description of the non-linear transitions between squeaks and self-oscillation [13].

The aim of this study is to define the invariants relative to the disturbance undergone by a vibrating resonant object when it collides with a non-resonant obstacle. This kind of interaction are a regular occurrence in daily life. For example, in the case of electronic vibrating objects, one can hear a "buzzy" sound whenever they touch a stiff obstacle (washing machine, microwave, vibrating phone...). It occurs as well in various acoustic musical instruments. For instance, the particular timbre of the tanpura results from the collisions between the strings and the bridge [14]. Also, guitarists can produce a screaming tone by playing a pinched harmonic, and a large range of sounds can be generated using prepared pianos [15]. We can see here that this type of interaction includes a wide

range of phenomena from a perceptual point of view. Indeed, we perceive a vibrating phone on a table as a sequence of impacts, a natural harmonic on a string produces a short "buzzy" sound followed by a new modal state of the string, and in some other cases, it may produce harmonic distortion, affect the sustain, the modes' frequency or even change the type of interaction (e.g., transition from squeak to self-oscillation on a glass).

As a first step towards define these invariants, we consider a 1-D resonant object (a stiff string) colliding with a clamped stiff obstacle not located too close to the string ends. This would correspond to the action of choking a string or playing a natural harmonic if the obstacle is located at a specific position. In this case, there is no possible coupling between the string and the barrier as both of them are clamped to the ground and we do not study the specific behavior when the obstacle is close to the bridge.

There are recent investigations into the numerical modeling of collisions in musical instruments [16][17], but very little work on the perceptual characterization of the synthesized signal.

Our approach consists in first gathering a database representative of the diversity of sounds that can be produced with this type of interaction. We made the choice here to synthesize samples using a numerical solving of the differential equations that describe the physical behavior of the system, as it allows the synthesis of realistic sounds with a precise control over all the experimental parameters. We then propose an empirical description of the signal related to the type of interaction. Finally, we make hypotheses regarding the signal elements that seem to be significant for the perception of the phenomena to characterize the related sound invariant.

The next step is to validate these hypotheses through listening tests that consist in comparing reference sounds synthesised with the physical model to sounds synthesized with a signal model reproducing precisely the sound morphologies that seem to be important for the perception of the phenomena according to our observations. The sounds will be synthesized to evoke different spatial locations and structure of the obstacle.

Also, we may expand our study to include interactions close to the string ends (e.g. tanpura), with coupling between the objects (e.g. rattling elements) and apply these sound invariants to any type of objects (shape, material and size). This will lead to other perceptual tests and, it is hoped, to a real-time synthesis process controlled by perceptual features according to the action-object conceptual description of sounds. The final aim is to improve the design of the source-filter synthesis process and the related conceptual description of sounds to include this new type of interaction.

This article is organized as follows: To introduce how non-linear interactions modify the response of a system, a brief overview of the effects of non-linear springs and collisions on a single-degree-of-freedom system is presented in the next section. The following section details the effect of springs and collisions on a stiff string, and a description of signal morphology is proposed subsequently. Conclusions and perspectives are presented in the last section.

Sound examples are available at `https://drive.google.com/open?id=1sNUu6krfWO-rCZD_vJV4SrZ4RUyloJfq`

2. NON-LINEAR SPRINGS AND COLLISIONS ON A SINGLE DEGREE-OF-FREEDOM SYSTEM

In this section, we aim to describe the effects of collisions on the signal morphology for the simplest vibrating system: a 1 Degree-of-Freedom (DoF) mass/spring/damper system. This is the first step to understand how the signal is affected by collisions on a rigid barrier.

2.1. Single degree-of-freedom system

Consider a mechanical damped harmonic oscillator, of mass M, stiffness K_0 and damping constant σ_0, and with displacement $u\,(t)$ as a function of time t. The ordinary differential equation governing the displacement of the oscillator is

$$\frac{d^2u}{dt^2} = -\omega_0^2 u - 2\sigma_0 \frac{du}{dt} \tag{1}$$

where $\omega_0 = \sqrt{K_0/M}$. For underdamped conditions (as is usually the case in musical systems), the general solution is

$$u\,(t) = e^{-\sigma_0 t}\left(A\cos(\omega t) + B\sin(\omega t)\right) \tag{2}$$

where $\omega = \sqrt{\omega_0^2 - \sigma_0^2}$, and for some constants A and B determined by initial conditions.

In discrete time, consider the time series u^n, representing an approximation to $u\,(t)$ at time $t = nk$, where k is the time step (and $F_s = 1/k$ is the associated sample rate). An explicit finite difference scheme approximating (6) above may be written, in condensed operator form, as:

$$\delta_{tt} u^n = -\omega_0^2 u^n - 2\sigma_0 \delta_{t\cdot} u^n \tag{3}$$

where

$$\delta_{tt} u^n = \frac{1}{k^2}\left(u^{n+1} - 2u^n + u^{n-1}\right),\ \delta_{t\cdot} u^n = \frac{1}{2k}\left(u^{n+1} - u^{n-1}\right) \tag{4}$$

This scheme may be written more explicitly as a recursion allowing the calculation of u^{n+1} from u^n and u^{n-1}:

$$u^{n+1} = (u^n(2 - k^2\omega^2) + u^{n-1}(-1 + k\sigma_0))/(1 + k\sigma_0) \tag{5}$$

2.2. Effect of a non-linear spring

As we model the barrier as a unilateral non-linear spring, it is of interest to take a look at a classic non-linear spring (see figure1):

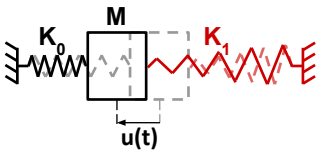


Figure 1: *Damped harmonic oscillator with a cubic spring of stiffness coefficient* K_1.

$$\frac{d^2u}{dt^2} = -\omega_0^2 u - 2\sigma_0 \frac{du}{dt} - H(t - t_0)\omega_1^4 u^3 \tag{6}$$

With $\omega_1 = \sqrt[4]{\frac{K_1}{M}}$, K_1 the stiffness of the cubic spring,$H(t)$ the Heaviside step function, and t_0 the time of appearance of the non-linear spring (here, we set $t_0 = 1s$).

We solve the problem with the following scheme[18]:

$$\delta_{tt}u^n = -\omega_0^2 u^n - 2\sigma_0 \delta_{t\cdot} u^n - H[n - \frac{t_0}{k}]\omega_1^4 (u^n)^2 \mu_{t\cdot} u \quad (7)$$

with $\mu_{t\cdot} u = (u^{n+1} + u^{n-1})/2$, $H[n]$ the discrete Heaviside step function.

The appearance of the non-linear part in a second time allows us to visualise the linear behavior (for $t < t_0$) and the non-linear behavior ($t > t_0$) on the same spectrogram.

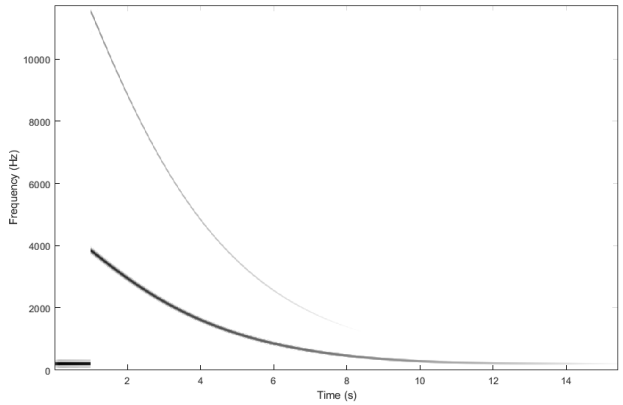


Figure 2: *Spectrogram of u for the single-degree-of-freedom system with a cubic spring term activated at $t = 1s$ with $\omega_1 = 300m^{-1/2}.s^{-1/2}$, initial conditions $u^0 = 1m$, $u^1 = 1m$.*

One can notice two phenomena related to the appearance of the non linear spring (see figure 2): the frequency of oscillation changes abruptly to an higher value then decreases and harmonic distortion appears (creation of the third harmonic). Thus, the frequency of non-linear modes varies with the amplitude of vibration of the spring (stiffness increases with amplitude, which is typical of springs of hardening type), and the waveform is no longer sinusoidal.

2.3. Effect of Collisions

The modeling of collisions with a rigid barrier may be written as the contact with a stiff unilateral non-linear spring (see figure 3), of restoring force $F_c = \frac{d\phi}{du}$, $\phi = \frac{K_c}{\alpha+1}[u]_+^{\alpha+1}$ ([16]).
With K_c the stiffness of the interaction, α the non-linear exponent, and $[u]_+ = (u + |u|)/2$ the positive part of u.

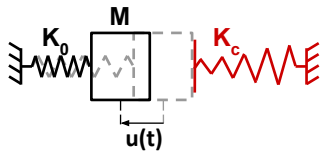


Figure 3: *Damped harmonic oscillator colliding with a barrier of stiffness K_c.*

$$\frac{\partial^2 u}{\partial t^2} = -\omega_0^2 u^n - \sigma_0 \frac{\partial u}{\partial t} - \frac{H(t - t_0)}{M}\frac{d\phi}{du} \quad (8)$$

We use the following scheme [16]:

$$\delta_{tt}u^n = -\omega_0^2 u^n - 2\sigma_0 \delta_{t\cdot} u^n - \frac{H[n - \frac{t_0}{k}]}{M}\frac{\delta_{t-}\phi^{n+\frac{1}{2}}}{\delta_{t\cdot} u^n} \quad (9)$$

with:

$$\phi^{n+\frac{1}{2}} = \frac{1}{2}\left(\phi(u^{n+1}) + \phi(u^n)\right) \quad (10)$$

It leads to the expression:

$$\mathcal{F}(r) = r + b + \frac{H[n - \frac{t_0}{k}]k^2}{M(1 + \sigma_0 k)}\frac{\Phi(r + a) - \Phi(a)}{r} = 0 \quad (11)$$

Given:
$r = u^{n+1} - u^{n-1}$,
$a = u^{n-1}$
$b = (-2u^n + 2u^{n-1} + \omega_0^2 k^2 u_0^n)/(1 + \sigma_0 k)$

This equation can be solved using a Newton-Raphson algorithm at each time step.

We use the approximation $\frac{\Phi(r+a)-\Phi(a)}{r} \approx \Phi'(a)$ when $r < \epsilon$.

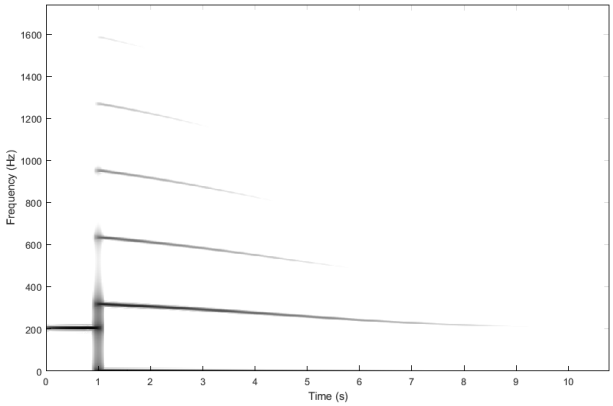


Figure 4: *Spectrogram of u for the single-degree-of-freedom system with a stiff unilateral non-linear spring appearing at $t_0 = 1s$. $K_c/M = 1.6 * 10^{10} m^{-1/2}.s^{-1/2}$, $\alpha = 1.9$ initial conditions $u^0 = 1m$, $u^1 = 1m$.*

The response for a unilateral non-linear spring is close to the classic non-linear spring (see figure 4). We observe a frequency-varying mode tending to the original mode as the amplitude tends to zero, and important harmonic distortion as the deformation of the waveform is abrupt.

To sum up, when a single-degree-of-freedom system collides with a non-resonant obstacle, it increases the frequency of its mode of vibration and creates harmonics. The frequency tends to its original value and the harmonic distortion disappears as the amplitude gets closer to zero.

As we switch the single-degree-of-freedom system to the stiff string, we can expect mode coupling when the descendant modes and their harmonics cross other modes of the structure, as we observe this kind of behavior on gongs and cymbals [19].

3. SPRINGS AND COLLISIONS ON A STIFF STRING

In this section, the linear model and the scheme used to synthesize realistic sounds of a stiff string is presented. Then, we study the

effects of linear and non-linear springs attached to the string on the frequency content of the samples in order to introduce how modes are impacted by perturbations. Finally, the collision model is implemented, and a description of the resulting morphology is presented.

3.1. Physical model of the stiff string and numerical scheme

The following partial differential equation describes the behavior of a stiff string subject to forces. This linear model does not take into account the variation of tension in the string (no pitch bending). it is commonly used for simulations and sound synthesis [20][21][18].

$$\frac{\partial^2 u}{\partial t^2} = c^2\frac{\partial^2 u}{\partial x^2} - \kappa^2\frac{\partial^4 u}{\partial x^4} - 2\sigma_0\frac{\partial u}{\partial t} + 2\sigma_1\frac{\partial^3 u}{\partial t \partial x^2} + \frac{1}{\rho S}\sum_m \delta(x - x_m)F_m \tag{12}$$

with:

- $u(x,t)$ the transversal motion of the string,
- $c = \sqrt{\frac{T}{\rho S}} = 404.02 m.s^{-1}$,
- $\kappa = \sqrt{\frac{EI_z}{\rho S}} = 1.297 m^2.s^{-1}$,
- $\delta(x)$ the Dirac function.

The signification and values of the parameters are defined in table1.

We can solve the equation using the following explicit finite difference scheme:

$$\delta_{tt}u = c^2\delta_{xx}u - \kappa^2\delta_{xxxx}u - 2\sigma_0\delta_{t\cdot}u + 2\sigma_1\delta_{t\_}\delta_{xx}u + J_m.F_m \tag{13}$$

The previous equation can be displayed in a matrix form:

$$\bar{\bar{A}}\bar{u}^{n+1} = -\bar{\bar{B}}\bar{u}^n - \bar{\bar{C}}\bar{u}^{n-1} + \bar{J}_m F_m \tag{14}$$

with:

- h the grid spacing, chosen at the stability limit, $h = \sqrt{\frac{(c^2k^2 + 4\sigma_1 k + \sqrt{(c^2k^2+4\sigma_1 k)^2 + 16\kappa^2 k^2})}{2}}$,
- k the time step interval, $k = 1/f_s$ with f_s the sampling frequency,
- $\delta_{xx}u = \frac{1}{h^2}(u^n_{l+1} - 2u^n_l + u^n_{l-1})$,
- u^n_l the discretized value of $u(x,t)$ at the n^{th} time step, and the l^{th} step of the string,
- $\delta_{xxxx}u = \frac{1}{h^4}(u^n_{l+2} - 4u^n_{l+1} + 6u^n_l - 4u^n_{l-1} + u^n_{l-2})$,
- $\delta_{t\_}u = \frac{1}{k}(u^n_l - u^{n-1}_l)$,
- $\bar{e}_m = [\ \underset{1}{0}\quad \underset{2}{0}\quad \underset{3}{0}\quad ...\quad \underset{i_{m-1}}{0}\quad \underset{i_m}{1}\quad \underset{i_{m+1}}{0}\quad ...\quad \underset{L}{0}\ ]$
- $\bar{J}_m = \bar{e}^T_m/h$
- F_m is the scalar value of the force m.

The boundary conditions are simply supported at the end points of the domain $u(x = \{0, L\}, t) = 0$; $\frac{\partial^2 u}{\partial x^2}|_{(x=\{0,L\},t)} = 0$.

As excitation force, we use a simple model for a plucked string at $x = x_{ex}$:

$$F_e(t) = \begin{cases} A_f * (-cos(\frac{\pi}{\Delta t}t) + 1) & if\ 0 \le t < \Delta t \\ 0 & else \end{cases}$$

String:	
Diameter	$\phi = 1mm$
Length	$L = 0.5m$
Density	$\rho = 7800 kg.m^{-3}$
Young Modulus	$E = 210GPa$
Tension	$T = 1000N$
Damping parameters:	$\sigma_0 = 0.05 rad.s^{-1}$
	$\sigma_1 = 0.002 rad.s^{-1}$
Sampling:	
Sampling frequency	$f_s = 176400Hz$
Recording duration	$t_{rec} = 10s$
Excitation:	
Position	$x_{ex} = L/10$
Duration	$\Delta t_{ex} = 1ms$
Amplitude	$A_f = 100N$
Behavior:	
Maximum amplitude	$U_{max} = 0.0103m$
Resonance frequencies	$f_n = \frac{nc}{2L}\sqrt{1 + \left(\frac{\kappa\pi n}{c}\right)^2}$
	$f_1 = 410.7Hz$

Table 1: Parameters used for the simulation

3.2. Spring on a stiff string

To observe the effects of a linear and a non linear spring on a stiff string, we use the string model presented eq. 12 with a linear and a cubic spring:

$$\frac{\partial^2 u}{\partial t^2} = c^2\frac{\partial^2 u}{\partial x^2} - \kappa^2\frac{\partial^4 u}{\partial x^4} - 2\sigma_0\frac{\partial u}{\partial t} + 2\sigma_1\frac{\partial^3 u}{\partial t \partial x^2} + \delta(x - x_s)F_s \tag{15}$$

with:

$$F_s = -\omega_0^2 u(x_s,t) - \omega_1^4 u(x_s,t)^3 \tag{16}$$

We use the following scheme (see eq.13 for the stiff string, eq. 7 for the non-linear spring):

$$\delta_{tt}u = c^2\delta_{xx}u - \kappa^2\delta_{xxxx}u - 2\sigma_0\delta_{t\cdot}u + 2\sigma_1\delta_{t\_}\delta_{xx}u + J_s.F_s \tag{17}$$

with:

$$F_s = -\omega_0^2 u_{i_s} - \omega_1^4 (u_{i_s})^2 \mu_{t\cdot} u_{i_s} \tag{18}$$

u_{i_s} is the element of the vector $\bar{u}$ at the point of application of the spring on the string (on the node $i = i_s$).

We observe a modification of the frequency of several modes of the string with a pure linear spring at $x = L/2$. These modifications remain constant as the signal evolves and follow a specific pattern (see figure 5): even harmonics remain approximately unchanged as they have a vibration node located at the application point of the spring, when odd harmonics get their frequency increased as they get stiffer around their vibration antinode. The frequency value of the odd harmonics gets greater with ω_0 but never exceed the next even harmonic, and the increase get lower as the rank of the harmonic get higher.
The quasi-harmonic string becomes in-harmonic, and those relatively low variations of the frequency content cause a categorical change of the perception: the system sounds like a linear plate or a shell.

For the cubic spring at $x = L/2$, the behavior is consistent with the previous observations: the even harmonics remain unchanged and

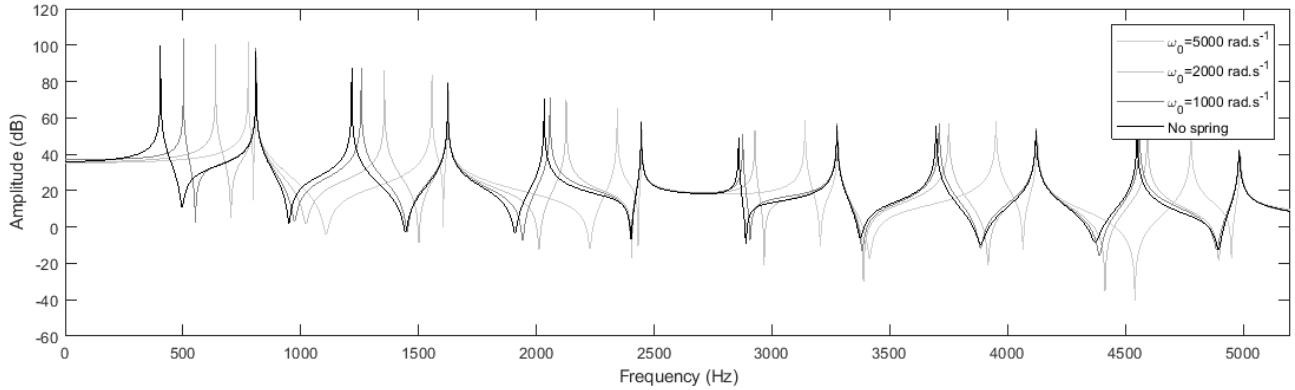


Figure 5: *FFT of the stiff string with a linear spring at $x = L/2$ for different values of ω_0.*

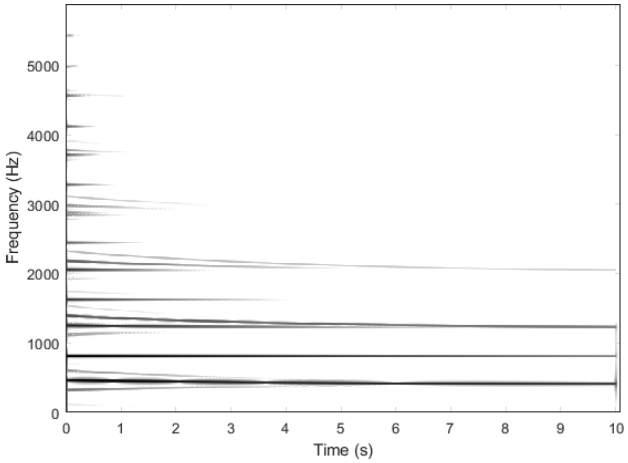


Figure 6: *Spectrogram of the stiff string with a cubic spring at $x = L/2$ for $\omega_1 = 300m^{-1/2}.s^{-1/2}$.*

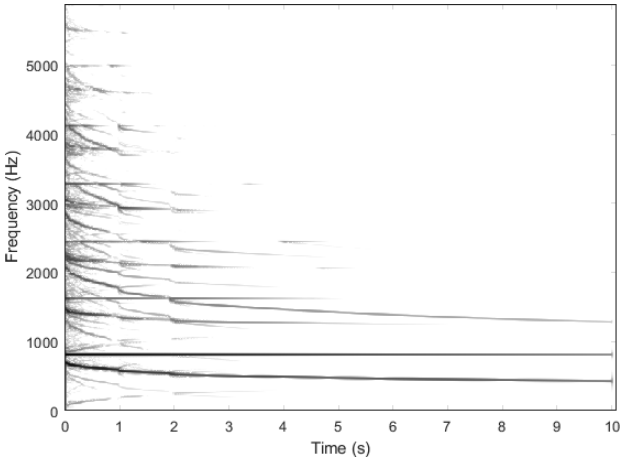


Figure 7: *Spectrogram of the stiff string with a cubic spring at $x = L/2$ for $\omega_1 = 1000m^{-1/2}.s^{-1/2}$.*

the frequency of the odd harmonics vary over time as the stiffness decrease with the amplitude (see figure 6).
But other effects caused by non-linearity appear. For $\omega_1 = 300m^{-1/2}.s^{-1/2}$, we can observe distinct frequency components around the original modes of the string, but in a large number due to harmonic distortion. Those new frequency components get closer to the original modes as the amplitude decrease over time. It is the same behavior that we observe on the single-degree-of-freedom system unless the modes are duplicated.
When the stiffness gets to high values (figure 7), a lot of components appear. This produces coupling between modes, resulting in fast variation of amplitude and frequency of several modes. It tends to chaotic behavior, creating a noisy-like signal at the beginning of the signal.
The perceived signal sounds like non-linear plates such as cymbals.

3.3. Stiff string colliding with a point rigid barrier

Considering the size of the article, we do not model the barrier as an object with its own dynamic but as a non-linear spring clamped to the ground. We control the appearance of the barrier with the Heaviside function, and we manage to turn on the collision function when $u(x_c) < 0$ (x_c define the location of the barrier). In this case, the appearance of the obstacle does not create any transient behavior.
We use the collision model (eq. 8910) with the model of the stiff string (eq. 12 13).

$$\frac{\partial^2 u}{\partial t^2} = c^2 \frac{\partial^2 u}{\partial x^2} - \kappa^2 \frac{\partial^4 u}{\partial x^4} - \sigma_0 \frac{\partial u}{\partial t} + 2\sigma_1 \frac{\partial^3 u}{\partial t \partial x^2} + \delta(x - x_c) F_c \tag{19}$$

$$F_c = -\frac{H(t - t_0)}{\rho S} \frac{d\Phi}{du}|_{(x=x_c,t)} \tag{20}$$

with $\Phi = \frac{K}{\alpha+1}[u_{i_c}]_+^{\alpha+1}$ and $[u]_+ = \frac{u+|u|}{2}$.

The corresponding scheme is presented bellow:

$$\delta_{tt} u = c^2 \delta_{xx} u - \kappa^2 \delta_{xxxx} u - 2\sigma_0 \delta_{t\cdot} u + 2\sigma_1 \delta_{t_-} \delta_{xx} u + J_c . F_c \tag{21}$$

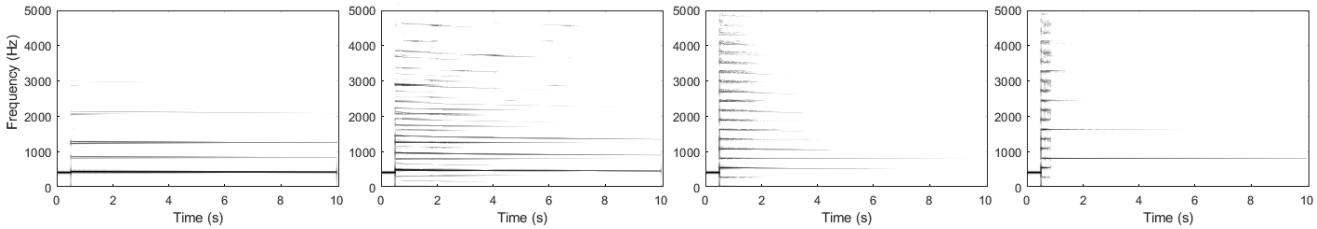


Figure 8: *Spectrogram of the stiff string ($u_l^0 = A_{max} sin(\pi \frac{lh}{L})$) colliding with a point rigid barrier ($\alpha = 1.6$) at $x = L/2$, from $t = 0.5s$, for different values of K_c. From the left: $K_c = 5 * 10^4 N.m^{-\alpha}$; $K_c = 5 * 10^5 N.m^{-\alpha}$; $K_c = 5 * 10^7 N.m^{-\alpha}$; $K_c = 5 * 10^8 N.m^{-\alpha}$.*

with:

$$F_c^n = -\frac{H[n - \frac{t_0}{k}]}{M} \frac{\delta_{t-} \phi^{n+\frac{1}{2}}}{\delta_{t\cdot} u^n} \ , \ \phi^{n+\frac{1}{2}} = \frac{1}{2} \left(\phi(u_{i_c}^{n+1}) + \phi(u_{i_c}^n) \right) \tag{22}$$

u_{i_c} is the element of the vector $\bar{u}$ at the point of application of the collisions (on the node $i = i_c$).
the finite difference scheme at the local point of the collision (node i_c) gives the following non-linear equation:

$$\mathcal{F}(r) = r(1 + \sigma_0 k) + b + \frac{H[n - \frac{t_0}{k}] k^2}{\rho S h} \frac{\Phi(r + a) - \Phi(a)}{r} = 0 \tag{23}$$

Given:
$r = u_{i_c}^{n+1} - u_{i_c}^{n-1}$,
$a = u_{i_c}^{n-1}$,
$b = < \bar{e}_c \ , \ (-2 - c^2 k^2 \delta_{xx} + \kappa^2 k^2 \delta_{xxxx} - 2\sigma_1 k \delta_{xx}) \bar{u}^n > + < \bar{e}_c \ , \ (2 + 2\sigma_1 k \delta_{xx}) \bar{u}^{n-1} >$

We use a Newton-Raphson algorithm to solve the scheme at this particular point.

4. SIGNAL MORPHOLOGIES DUE TO COLLISIONS ON STIFF STRINGS

4.1. Simulations and investigations

In order to make it easier to understand how the collisions modify the frequency response, we study the response of the system without excitation force with the following initial condition: $u_l^0 = U_{max} sin(\pi \frac{lh}{L})$ (see figure 9).

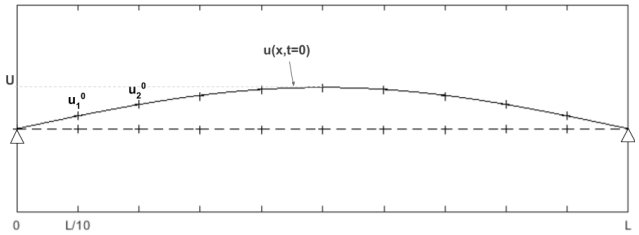


Figure 9: *Representation of the initial condition of the transverse displacement of the string.*

Here the string initially vibrates only on its first vibration mode until it collides with the barrier at $t_0 = 0.5s$. Then, we can observe a new distribution of the energy due to the frequency shift of the mode, harmonic distortion and mode coupling (figure 8). If we observe the left figure, we distinct clearly a frequency gap as the obstacle appears and a few harmonics of this mode are generated. Then, a few other modes of the string are excited due to mode coupling. This process expands itself as K gets higher. For a really stiff barrier, very high frequency components are generated (up to $25kHz$ for $K = 5 * 10^8 N.m^{-\alpha}$) causing important losses. Thus, all the components but the even harmonics disappear quickly, creating a vibration node at the location of the obstacle (here $L/2$).
The perceived sound is similar to a natural harmonic played on a guitar for high values of K. For low values of K, the string sounds like a bell as the barrier appear, and the frequency shift bring back the sound to a regular string.

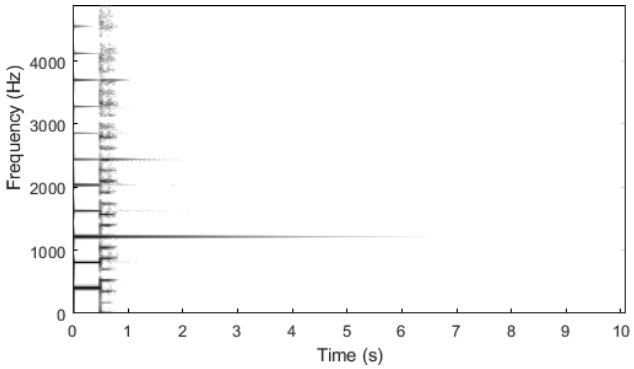


Figure 10: *Spectrogram of a plucked stiff string colliding with a point rigid barrier at $x = L/3$, from $t_0 = 0,5s$, for $K_c = 5 * 10^8 N.m^{-\alpha}$, and $\alpha = 1,6$.*

We study more specifically the cases that sound like a natural harmonic ($K \geq 1.10^8$) because it is a clearly identified type of interaction and we are able to specify a pattern corresponding to this behavior. The sounds generated for low values of K are peculiar and it is hard to recognize what is the source of it. We describe the pattern for the natural harmonic as following: if the considered mode of the string does not have a vibration node at the exact location of the obstacle, its frequency increases of a constant value ($\sim f_0/3$ for $x_c = L/2$) and a component appears in a symmetric way below with a lower amplitude. Harmonic distortion and mode coupling provoke the apparition of higher frequency components in the whole audible frequency band (and above) corresponding to other modes of the system {String + Rigid barrier}. These newly excited modes provoke the apparition of higher frequency compo-

nents themselves. This chain reaction induces important losses, and lasts for a short duration after which only the modes with a vibration node at the location of the obstacle remain.
From a perceptual point of view, the simultaneous presence of frequency components created by harmonic distortion and modes of the system create beats, roughness, and noisy-like signal at high frequency.

This pattern varies with the position of the obstacle. Generally, modes with a vibration node at the location of the obstacle keep their frequency unchanged but may undergo some variations of their amplitude (see the modes multiple of 3 for $x_c = L/3$ on figure 10). The modifications on the other modes depends on the ratio between the vibration amplitude and the proximity of the obstacle.

We introduce the scalar y, the transverse position of the rigid barrier. The expression of ϕ become $\phi = \frac{K_c}{\alpha+1}[u_{i_c} - y]_+^{\alpha+1}$.

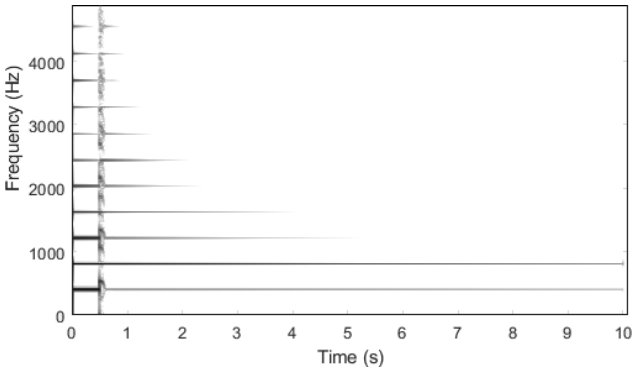


Figure 11: *Spectrogram of a plucked stiff string colliding with a point rigid barrier at ($x = L/2$;$y = 0.065 * U(t_0 = 0.5s)$), from $t = 0.5s$, for $K_c = 5 * 10^8 N.m^{-\alpha}$, and $\alpha = 1.6$.*

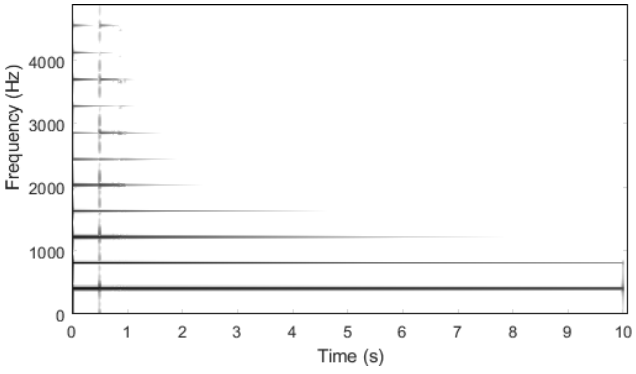


Figure 12: *Spectrogram of a plucked stiff string colliding with a point rigid barrier at ($x = L/2$;$y = 0.99 * U(t_0 = 0.5s)$), from $t_0 = 0.5s$, for $K_c = 5 * 10^8 N.m^{-\alpha}$, and $\alpha = 1.6$.*

As $y \neq 0$, the frequency components get back to the natural vibration modes of the string when the amplitude of u_{i_c} get below y. For instance, if $y = 0.065U(t_0)$ (with $U(t_0)$ the amplitude of u at the time of appearance of the obstacle), we observe a short

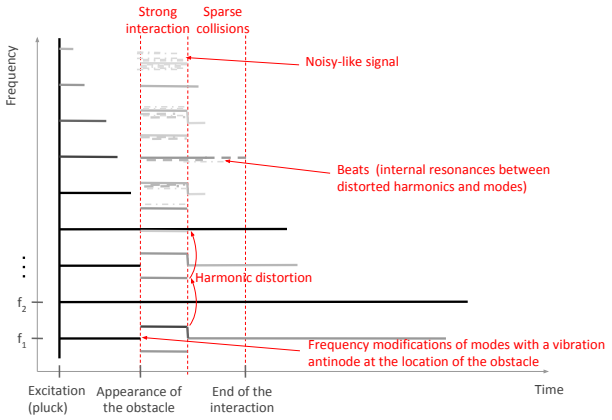


Figure 13: *Schematic Time-Frequency representation of a plucked stiff string colliding with a point rigid barrier at ($x = L/2$).*

time of interaction ($\sim 0.1s$), then the string gets back to its regular vibrations with new initial conditions (see figure 11).
In the case of a really light touch ($y_0 = 0.99 * U(t_0)$), the pattern is close to disappear, but we can notice a very short apparition of new frequency components for $t = 0.5s$ and some slight harmonic distortion inducing mode coupling due to sparse collisions for $0.5s < t < 1s$ (see figure 12).

4.2. Towards a non-linear interaction invariant

Based on the previous considerations, we propose a description of the morphology of the signal resulting of the collisions between a stiff string and a stiff barrier. The highly non-linear nature of this interaction induces complex phenomena such as frequency shift, harmonic distortion and mode coupling.
Still, it is possible to define a pattern that describes the time-frequency content of the signal regarding the location and the nature of the obstacle (see figure 13). One can notice two different interaction phases:

- If the transversal position of the barrier is distincly lower than the amplitude of vibration of the string, the interaction is strong. In this case, we observe important modifications of the modes' frequency and the generation of new partial tones due to harmonic distorsion and internal resonances. The partial tones are clearly distinguishable around the first modes, but it gets to noisy-like signal above the sixth mode.
- When the amplitude of vibration of the string is close to the transversal position of the barrier, we get to an other phase with sparse collisions. Here, we notice some harmonic distorsion that transfers energy from the first modes to the following ones, and it creates beats as the string is slightly inharmonic.

Hence, the transversal position of the obstacle has an influence on the duration of the strong interaction phase duration. The longitudinal position of the obstacle define which modes are modified. The material of the obstacle (stiffness and damping) will affect the energy distribution within the modes as the harmonic distorsion gets more important with the stiffness of the obstacle. This pattern is specific to a point rigid barrier, it may be of interest to expand it to a distributed contact model.

5. CONCLUSION

In this paper, we aimed at identifying sound morphologies due to nonlinear interactions between a stiff string and colliding objects. This is the first step towards the development of synthesis processes perceptually controlled. For that, we hypothesized that nonlinear interactions are perceived through morphological sound invariants. We based our investigations on a physical modeling of the interaction phenomena to synthesize realistic sounds with a perfect control of the experimental parameters. This led to an experimental sound data bank that we analyzed to observe the morphologies of the computed sounds in order to deduct typical signal behaviors. Eventually, we defined specific patterns linked to the nonlinear interaction that may be relevant perceptual cues for sound recognition. These patterns mainly rely on frequency shifts, harmonic distorsion and mode coupling that may be responsible for the perception of roughness occurring during the interaction.

The next step is to model the invariant from a signal point of view and to design a synthesis process with an intuitive control strategy. This signal model will be validated through formal listening tests, and will be possibly extended to more general sound textures.

6. REFERENCES

[1] William W Gaver, "What in the world do we hear?: An ecological approach to auditory event perception," *Ecological psychology*, vol. 5, no. 1, pp. 1–29, 1993.

[2] Stephen McAdams and Emmanuel Bigand, "Introduction to auditory cognition," 1993.

[3] James J Gibson, "The ecological approach to visual perception," 1979.

[4] William H Warren and Robert R Verbrugge, "Auditory perception of breaking and bouncing events: a case study in ecological acoustics.," *Journal of Experimental Psychology: Human perception and performance*, vol. 10, no. 5, pp. 704, 1984.

[5] Richard P Wildes and Whitman A Richards, "Recovering material properties from sound," *Natural computation*, pp. 356–363, 1988.

[6] Stephen Lakatos, Stephen McAdams, and René Caussé, "The representation of auditory source characteristics: Simple geometric form," *Perception & psychophysics*, vol. 59, no. 8, pp. 1180–1190, 1997.

[7] Mitsuko Aramaki, Mireille Besson, Richard Kronland-Martinet, and Sølvi Ystad, "Controlling the perceived material in an impact sound synthesizer," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 19, no. 2, pp. 301–314, 2011.

[8] Richard Kronland-Martinet, Sølvi Ystad, and Mitsuko Aramaki, "High-level control of sound synthesis for sonification processes," *AI & society*, vol. 27, no. 2, pp. 245–255, 2012.

[9] Simon Conan, Etienne Thoret, Mitsuko Aramaki, Olivier Derrien, Charles Gondre, Sølvi Ystad, and Richard Kronland-Martinet, "An intuitive synthesizer of continuous-interaction sounds: Rubbing, scratching, and rolling," *Computer Music Journal*, vol. 38, no. 4, pp. 24–37, 2014.

[10] Matthew D Hoffman and Perry R Cook, "Feature-based synthesis: Mapping acoustic and perceptual features onto synthesis parameters.," in *ICMC*. Citeseer, 2006.

[11] Mitsuko Aramaki, Richard Kronland-Martinet, Thierry Voinier, and Sølvi Ystad, "A percussive sound synthesizer based on physical and perceptual attributes," *Computer Music Journal*, vol. 30, no. 2, pp. 32–41, 2006.

[12] Mitsuko Aramaki, Charles Gondre, Richard Kronland-Martinet, Thierry Voinier, and Solvi Ystad, "Thinking the sounds: an intuitive control of an impact sound synthesizer," Georgia Institute of Technology, 2009.

[13] Etienne Thoret, Mitsuko Aramaki, Charles Gondre, Sølvi Ystad, and Richard Kronland-Martinet, "Eluding the physical constraints in a nonlinear interaction sound synthesis model for gesture guidance," *Applied Sciences*, vol. 6, no. 7, pp. 192, 2016.

[14] Maarten van Walstijn, Jamie Bridges, and Sandor Mehes, "A real-time synthesis oriented tanpura model," in *Proc. Int. Conf. Digital Audio Effects (DAFx-16)*, 2016, pp. 175–182.

[15] Stefan Bilbao et al., "Prepared piano sound synthesis," DAFx, 2006.

[16] Stefan Bilbao, Alberto Torin, and Vasileios Chatziioannou, "Numerical modeling of collisions in musical instruments," *Acta Acustica united with Acustica*, vol. 101, no. 1, pp. 155–173, 2015.

[17] Michele Ducceschi, "A numerical scheme for various nonlinear forces, including collisions, which does not require an iterative root finder," .

[18] Stefan Bilbao, *Numerical sound synthesis: finite difference schemes and simulation in musical acoustics*, John Wiley & Sons, 2009.

[19] KA Legge and NH Fletcher, "Nonlinearity, chaos, and the sound of shallow gongs," *The Journal of the Acoustical Society of America*, vol. 86, no. 6, pp. 2439–2443, 1989.

[20] Antoine Chaigne and Anders Askenfelt, "Numerical simulations of piano strings. i. a physical model for a struck string using finite difference methods," *The Journal of the Acoustical Society of America*, vol. 95, no. 2, pp. 1112–1118, 1994.

[21] Julien Bensa, Stefan Bilbao, Richard Kronland-Martinet, and Julius O Smith III, "The simulation of piano string vibration: From physical models to finite difference schemes and digital waveguides," *The Journal of the Acoustical Society of America*, vol. 114, no. 2, pp. 1095–1107, 2003.

GROUP DELAY-BASED ALLPASS FILTERS FOR ABSTRACT SOUND SYNTHESIS AND AUDIO EFFECTS PROCESSING

Elliot K. Canfield-Dafilou and Jonathan S. Abel

Center for Computer Research in Music and Acoustics,
Stanford University, Stanford, CA 94305 USA
kermit|abel@ccrma.stanford.edu

ABSTRACT

An algorithm for artistic spectral audio processing and synthesis using allpass filters is presented. These filters express group delay trajectories, allowing fine control of their frequency-dependent arrival times. We present methods for designing the group delay trajectories to yield a novel class of filters for sound synthesis and audio effects processing. A number of categories of group delay trajectory design are discussed, including stair-stepped, modulated, and probabilistic. Synthesis and processing examples are provided.

1. INTRODUCTION

Allpass filters are of particular interest for audio processing because they are defined to have a unit magnitude frequency response and exhibit time delays that vary with frequency. As passive, dispersive filters, they are found in a wide range of audio applications. For example, allpass filters are found in physical modeling of stiff strings and percussion [1,2], guitar bodies [3], pianos [4,5], and bells [6]. They are used for interpolation [7, 8] and decorrelation [9, 10]. The dispersive qualities of allpass filters are also useful for artificial reverberation [11] and spring modeling [12].

Allpass filters can be used for system measurement and identification [13, 14]. They can be found in shelving and equalization filters [15,16] as well as warped filters such as [17]. They have also been used in loudspeaker crossovers for time-alignment [18, 19].

In a more abstract sense, one can find allpass filters in a range of audio effects. Flanging, phasing, and chorus effects [20–22] can be implemented with allpass filters, and [23–25] have used allpass filters for distortion processing. Recently, [26] has used allpass filters for peak limiting. Further developing the work of [27], [28, 29] have shown uses of allpass filters for abstract sound synthesis by means of spectral delays and [30, 31] has shown methods for using allpass filters for distortion effects.

In this paper, we present a method for designing allpass filters from their group delay. We show that for an allpass filter, the group delay can be interpreted as a trajectory of frequency-dependent time delays, used here primarily for artistic effects. This group delay can be arbitrarily formed so long as each frequency has exactly one associated time delay. The result is a class of filters with a range of applications including abstract sound synthesis, decorrelation, steganography, impulse response measurement, and audio effects processing.

In section 2, we show how an arbitrary group delay trajectory can be used to drive the impulse response of an allpass filter. Section 3 introduces an equalization method to give the allpass filter a near constant amplitude envelope. Section 4 discusses details on the implementation of these filters and section 5 presents some applications. Finally, section 6 concludes the paper and suggests areas of future work.

2. METHOD

2.1. Allpass Filters

In discrete time, allpass filters are realized as having poles within the unit circle and zeros that are reciprocally reflected outside the unit circle at the same angle. A first-order allpass filter can be described by the transfer function

$$G(z) = \frac{-\rho + z^{-1}}{1 - \rho z^{-1}}\,, \tag{1}$$

and the difference equation

$$y(t) = -\rho x(t) + x(t-1) + \rho y(t-1)\,, \tag{2}$$

where ρ is the position of the pole. Its impulse response is therefore

$$g(t) = \begin{cases} 0, & t < 0 \\ -\rho, & t = 0 \\ (1-\rho)^2 \rho^{t-1}, & t \geq 1 \end{cases}\,, \tag{3}$$

Its magnitude is by definition

$$\left| G(e^{-j\omega}) \right| = 1\,, \tag{4}$$

and it has the phase response

$$\sphericalangle G(e^{-j\omega}) = -\arctan \frac{(1-\rho)^2 \sin(\omega)}{(1-\rho)^2 \cos(\omega) - 2\rho}\,. \tag{5}$$

The group delay of a filter is defined as the negative derivative of phase with respect to frequency,

$$\tau(\omega) = -\frac{d\phi(\omega)}{d\omega}\,. \tag{6}$$

The first order allpass filter has the group delay

$$-\frac{d\phi(\omega)}{d\omega} = \frac{1-\rho^2}{1+\rho^2-2\rho\cos(\omega)}\,. \tag{7}$$

Given a group delay trajectory, we can calculate the phase,

$$\phi(\omega) = -\int_0^\omega \tau(\omega) d\omega\,, \tag{8}$$

and since an allpass filter has unit magnitude, the time domain impulse response is simply

$$g(t) = \mathcal{F}^{-1}\left[e^{j\phi(\omega)} \right]\,. \tag{9}$$

It is important to note that cascading multiple allpass filters, even with different values for ρ, still creates an allpass filter.

2.2. Choosing a Group Delay Characteristic

If the group delay is set to a constant value,

$$\tau(\omega) = k\,, \tag{10}$$

(9) produces a band limited impulse. A linear chirp results if the group delay changes linearly, traversing a fixed frequency bandwidth in each time step,

$$\tau(\omega) = \eta\,\omega\,. \tag{11}$$

If the group delay trajectory traverses each octave in the same length of time, an exponential chirp is the result,

$$\tau(\omega) = \eta \ln(\omega)\,. \tag{12}$$

The observation here is that the group delay trajectory can be viewed as a function that maps the frequency axis to a set of time delays. Moreover, we can set this group delay trajectory to be any arbitrary function which determines this time/frequency relationship, so long as each frequency corresponds to one time delay value. Depending on the group delay trajectory, the resulting filters can be used as a type of novel audio effect. In some cases, the impulse responses themselves have interesting sounds and could stand on their own as a new synthesis method.

We will now discuss some of the myriad ways to set the group delay trajectory to produce musical effects and sounds.

2.3. Stair-Stepped Group Delays

We can discretize the linear sweep from above into n ascending segments, spaced by m in time with

$$\tau(\omega) = \frac{\lfloor n\,\omega \rfloor}{m}\,, \tag{13}$$

where ω is in normalized frequency ($\omega \in [0, 1]$). This discretized chirp can effectively be viewed as passing an impulse through a set of n bandpass filters that are each delayed by a multiple of m in time. For an example, see Fig. 1.

Since this stair-step equation has discontinuities introduced by the floor function, it might be desirable to suppress the time/frequency leakage by smoothing out the discontinuities. This can be done, for example, with the hyperbolic tangent function. The following expression,

$$\tau(\omega) = m\left(\frac{\tanh\left(\frac{\omega}{\alpha n} - \frac{1}{\alpha}\left\lfloor\frac{\omega}{n}\right\rfloor - \frac{2}{\alpha}\right)}{2\tanh\left(\frac{2}{\alpha}\right)} + \frac{1}{2} + \left\lfloor\frac{\omega}{n}\right\rfloor\right)\,, \tag{14}$$

produces a smoothed staircase group delay where n determines the number of segments and α is a smoothing parameter. When α is close to 0, (14) produces an output similar to (13). As α increases, the output of (14) becomes smoother and closer to the continuous chirp from (11). Fig. 2 shows an example of a smoothed stair-step group delay filter.

We can warp the frequency scale to control the frequency-dependent energy contained in each bandpassed-segment,

$$\tau(\omega) = \frac{\lfloor n\,\nu\{\omega\} \rfloor}{m}\,, \tag{15}$$

where $\nu\{\cdot\}$ is a function that determines the frequency axis warping, for example $\nu\{\omega\} = \omega^{1/2}$ would cause the higher frequency segments have a larger bandwidth.

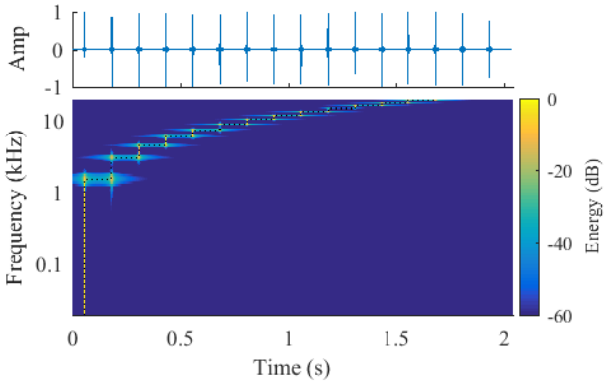


Figure 1: A sixteen-segment stair-stepped filter. The group delay is plotted on top of the spectrogram with a dotted black line like done by [29].

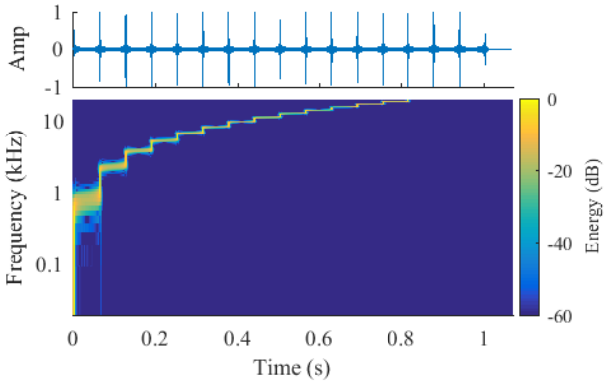


Figure 2: A smoothed sixteen-segment stair-stepped filter, with $\alpha = 0.05$.

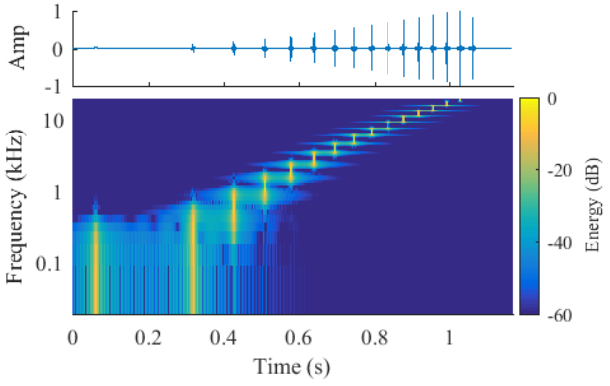


Figure 3: A sixteen-segment stair-stepped filter with time and frequency warping.

We could also warp the time scale for when these segments appear with the function $\zeta\{\cdot\}$

$$\tau(\omega) = \frac{\zeta\{\lfloor n\,\omega \rfloor\}}{m}\,. \tag{16}$$

For example, $\zeta\{x\} = x^2$ would compress the time interval between the early pulses and spread the later ones out in time.

We can naturally combine both time and frequency warping into the same expression,

$$\tau(\omega) = \frac{\zeta\{\lfloor n\,\nu\{\omega\} \rfloor\}}{m}\,. \tag{17}$$

Fig. 3 shows a filter with a group delay chosen to have even energy per octave and to compress the time interval in the high frequencies.

In (13), (15), (16), and (17), it may be beneficial to normalize the numerator to the interval $[0, 1]$ so the factor m does need to be modified to compensate for global timing changes introduced by time and frequency warping.

The group delay function can also be chosen to scramble the order of the frequency segments. For example,

$$\tau(\omega) = \begin{cases} d_1, & \omega \in [0, \omega_1) \\ d_2, & \omega \in [\omega_1, \omega_2) \\ \vdots & \vdots \\ d_N, & \omega \in [\omega_{N-1}, \omega_N] \end{cases}, \tag{18}$$

where $\{d_1, d_2, \cdots, d_N\}$ are the delays for each frequency region. An example of this "arpeggiated" group delay filter can be seen in Fig. 4. If the delay times are allowed to repeat one can create "chordal" structures, as seen in Fig. 5.

2.4. Modulated Group Delay

In addition to setting the group delay to create a "stair-step" function like described in 2.3, the group delay can also be modulated. For example, a group delay such as

$$\tau(\omega) = k\cos(2\pi\omega f + \phi)\,, \tag{19}$$

where k determines the total length of the filter, f the frequency of the modulator, and ϕ the initial phase, would create a filter that oscillates—or "chirps"—up and down simultaneously in different frequency bands f times across the audio band. When f is very small (below about 5), the individual chirp trajectories are audible (see Fig. 6). When f is between about $(5, 100)$, the filter sounds like a modulated signal (see Fig. 7). Above this modulator speed, the energy starts to pile up at discrete points in time, likely associated with the extrema of modulation, and the filter sounds like a sequence of clicks or "echoes" (see Fig. 8).

We can, again, warp the frequency axis to adjust how many oscillations occur within a certain frequency region. For example,

$$\tau(\omega) = k\cos(2\pi\,\nu\{\omega\} f + \phi)\,, \quad \nu\{\omega\} = \omega^{1/2} \tag{20}$$

would have an equal number of oscillations in each octave. Now, a large modulator f no longer stacks energy at discrete time points, but rather creates other perceptual chirp trajectories, as seen in Figs. 9 and 10.

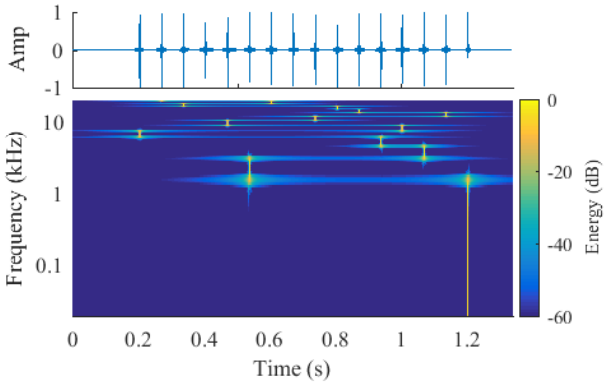


Figure 4: A sixteen-segment "arpeggiated" (scrambled) stair-stepped filter.

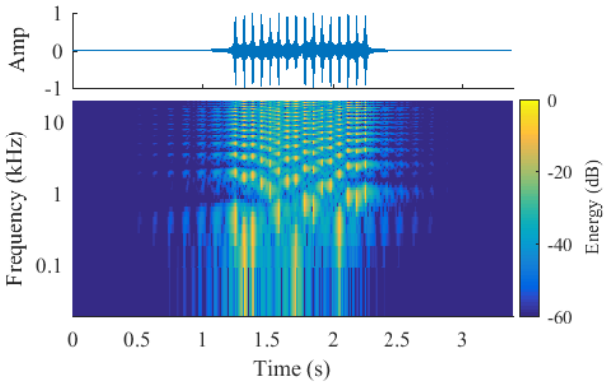


Figure 5: A sixteen-segment "chordal" (multiple frequencies at the same time) stair-stepped filter.

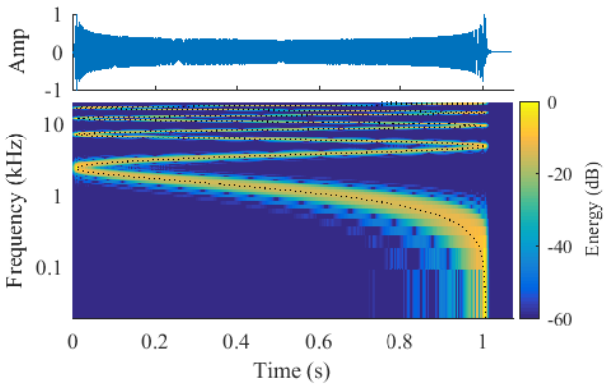


Figure 6: A slow (5 Hz) sine-modulated filter. The group delay is plotted on top of the spectrogram with a dotted black line.

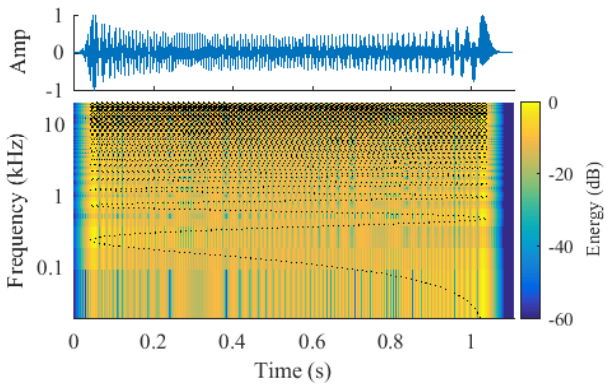


Figure 7: A medium (50 Hz) sine-modulated filter. The group delay is plotted on top of the spectrogram with a dotted black line.

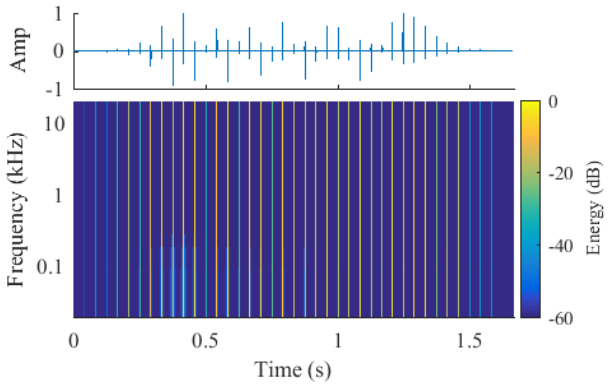


Figure 8: A fast (1 kHz) sine-modulated filter.

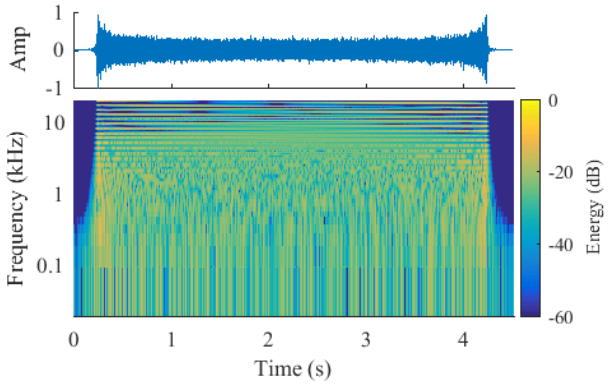


Figure 9: A sine-modulated filter with warped frequency axis.

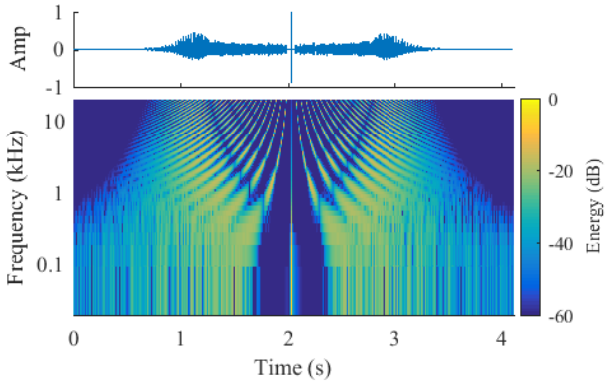


Figure 10: A sine-modulated filter with warped time and frequency axes.

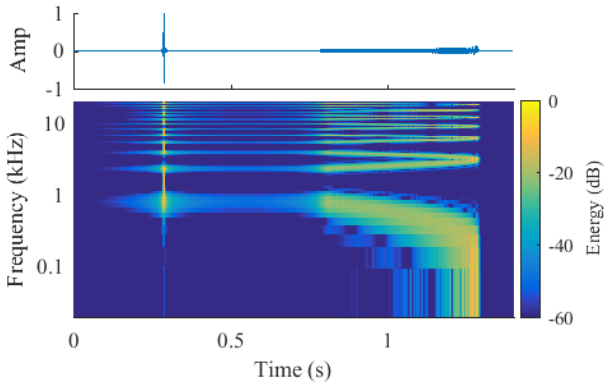


Figure 11: A filter created with a sinusoidally modulalated group delay that is soft-clipped on one side.

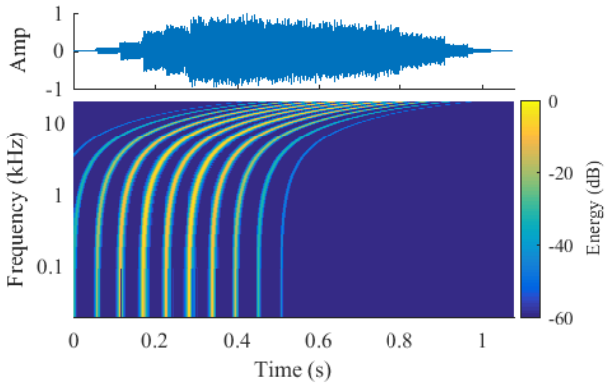


Figure 12: A "spring"-like group delay created by a fast, sinusoidal modulated group delay which is multiplied by an exponential ramp.

To (19), we can further add amplitude modulation with the modulator $m_a(\omega)$

$$\tau(\omega) = k\cos(2\pi\omega f + \phi)\, m_a(\omega)\,, \tag{21}$$

or frequency modulation with the modulating signal $m_f(\omega)$,

$$\tau(\omega) = k\cos(2\pi\omega f + m_f(\omega))\,. \tag{22}$$

Both of these methods can create interesting sounds with complex spectra. Naturally, the modulating function used as the group delay trajectory need not be a sinusoidal signal and there are many other possibilities. For example, Fig. 11 shows a modulated group delay filter with saturating distortion and Fig. 12 shows a sine modulated group delay with an additional exponential modulation used to create a spring reverb-like effect.

2.5. Probabilistic Group Delay

Another way to "draw" the group delay trajectory is probabilistically. If $\tau(\omega)$ is randomly drawn from a Gaussian distribution, the resulting impulse response will simply be a burst of enveloped noise with a duration proportional to the width of the Gaussian. We can also construct the group delay as a frequency dependent "drunk-walk" path.

Let there be N maximum-delay waypoints, each defined at some frequency. One such method would be to define the waypoints according to a perceptual criteria, like one waypoint per ERB-band center frequency. By smoothly interpolating between these waypoints, we define a frequency-dependent "area" within which we will draw our group delay curve.

To generate the actual group delay trajectories, we divide the maximum delay/frequency curve into β segments, distributed according to some function of frequency, $\zeta\{\omega\}$. If $\zeta\{\omega\}$ is linear, more segments will be in the high frequencies. If $\zeta\{\omega\}$ is an ERB warping, the segments will be approximately evenly distributed across the range of human hearing.

For each of these segments, we randomly choose a delay that falls within the maximum delay for that frequency. These segments are then either aligned along their leading or trailing edge (e.g., each segment can be between $[0, \texttt{maxdelay}]$, or centered about their midpoints $(-\texttt{maxdelay}/2, \texttt{maxdelay}/2)$. This now defines a set of discrete frequencies where the group delay is set. To define a continuous group delay function, we simply interpolate this set of points.

When β is small (see Fig. 13), there will be relatively few segments and the resulting filter may "sound chirpy," and when β is large (see Fig. 14), the result will sound more like enveloped noise. In between these extremes, filters designed like this can sound "metallic," like what is shown in Figs. 15 and 16. In all cases, the maximum frequency/delay curve defines an area that will be filled by the β segments. By defining these filters with a random process, we can generate a large number of mutually decorrelated allpass filters that have the same type of sound.

2.6. Hand-Drawn Group Delay

There are naturally many ways to design the group delay. One method which allows flexibility is to simply draw it by hand. Using a grid where the user sets a delay for each quantized frequency (potentially on a warped frequency axis), and then smoothly interpolating between the grid points and potentially resampling and scaling in time is one such method. Additionally, one could draw

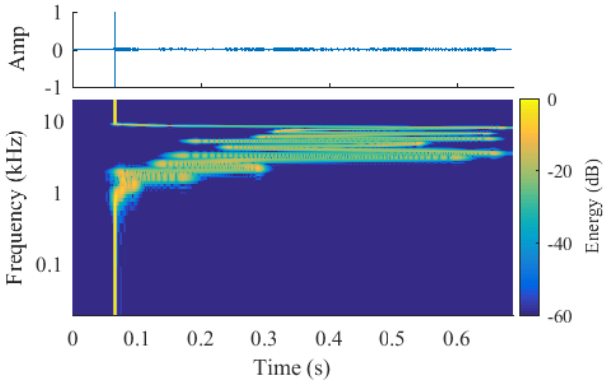


Figure 13: A probabilistic allpass filter with $\beta = 75$ and the segments aligned at $t = 0$. Note that a large frequency region was selected to have constant group delay so the filter only effects a narrow bandwidth.

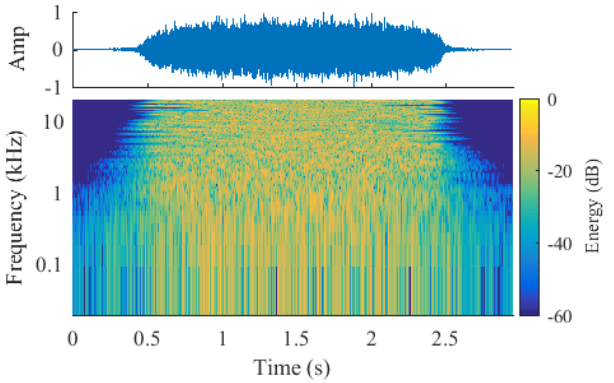


Figure 14: A probabilistic allpass filter with $\beta = 2000$ and the segments centered.

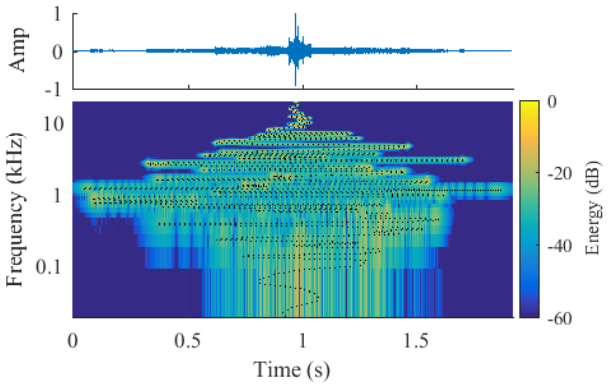


Figure 15: A probabilistic allpass filter with $\beta = 100$ and the segments centered. The group delay is plotted on top of the spectrogram with a dotted black line.

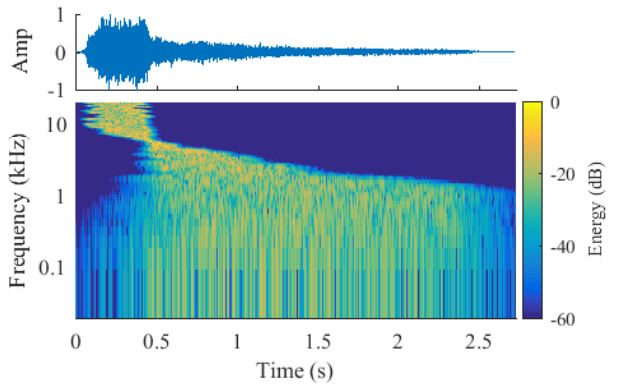


Figure 16: A probabilistic allpass filter with $\beta = 1000$ and the segments aligned at $t = 0$. Note that some of the time/frequency waypoints were set to have negative values.

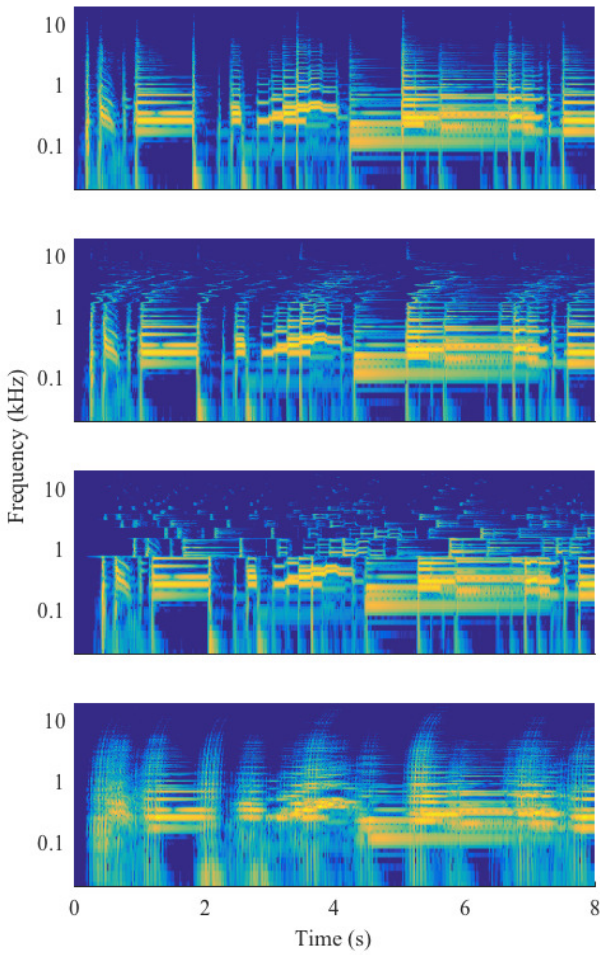


Figure 17: A guitar track unprocessed (top) and processed through the allpass filter shown in Fig. 13 (second), Fig. 4 (third), and Fig. 12 (bottom). Note how the various filters affect the timing of the spectral components of the guitar.

the maximum delay curve described in section 2.5 and then statistically generate the group delay.

2.7. Processing

Not only can these filters have interesting sounding impulse responses, they can provide the basis for interesting audio processes. For example, Fig. 17 shows the spectrogram of a guitar track processed through allpass filters such as the ones shown in Figs. 13, 4 and 12. In the first case, one hears the "tonal" components of the guitar accompanied by "chorus" of high-frequency chirps resulting from time-smeared transients. The second filter creates an "arpeggiated" sound. The last filter adds a spring reverb-like sound.

3. EQUALIZATION

In these filters, energy is conserved since these filters are allpass. However, the slope of the group delay determines the amount of time over which each frequency region is spread. In regions where frequencies are more spread in time, the instantaneous amplitude will be relatively lower than regions where the frequencies are less dispersed. In many cases, one would want the allpass filter that results from the methods above, but sometimes it is desirable to design a signal with a constant amplitude envelope. Our perception of the amplitude envelope of a signal is associated with a temporal time constant. Equalizing the amplitude envelope could help prevent unintentional changes in level.

Given a group delay characteristic, it is possible to calculate the amplitude envelope as a function of frequency, as shown in [29]. Denote by $\omega_\pm$ two close frequencies with difference Δ and mean ω,

$$\Delta = \omega_+ - \omega_- \qquad \omega = \frac{(\omega_- + \omega_+)}{2}. \tag{23}$$

An allpass filter will have Δ/π energy in the interval $[\omega_-, \omega_+]$. This energy is roughly equal to the signal energy in the time interval $[\tau(\omega_-), \tau(\omega_+)]$,

$$\frac{\Delta}{\pi} \approx |\tau(\omega_-) - \tau(\omega_+)| \, \frac{a^2(\omega)}{2}. \tag{24}$$

Taking the limit $\Delta \to 0$ and solving for the amplitude envelope a yields

$$a(\omega) = \left[\frac{\pi}{2}\left|\frac{d\tau(\omega)}{d\omega}\right|\right]^{-\frac{1}{2}}. \tag{25}$$

By approximating the inverse of (25), we have a good equalization filter $u(\omega)$ that yields a near constant crest factor

$$|u(\omega)| \approx \frac{1}{a(\omega)} = \left[\frac{\pi}{2}\left|\frac{d\tau(\omega)}{d\omega}\right|\right]^{\frac{1}{2}}. \tag{26}$$

Naturally, this equalized filter will change the amplitude relationships across frequency and brings out (by amplifying) the frequency regions with slowly changing group delays.

4. IMPLEMENTATION AND COMPLEXITY

These filters can be computationally expensive as the implementation of these filters with complex group delay characteristics require a large number of filter sections. A single biquad allpass filter

adds a cumulative 2π phase. Short filters without a large amount of integrated group delay could be implemented with the methods described by [32,33] or [34], however some of the filters described in this paper are temporally so long or have such a significant amount of integrated group delay that it would be impractical to implement them in the time domain. Moreover, a time-domain implementation would add a significant amount of pure delay due to the large number of filters in cascade. Instead, we implement these filters by finding their impulse responses with (8) and (9). We typically pre-compute the impulse responses offline and apply the filters to input in real-time with a fast convolution algorithm [35–37].

Since we are using the Discrete Fourier Transform (DFT) to find the impulse response related to a specific phase characteristic, it is necessary to use a DFT long enough to implement the filter. Since the group delay trajectory tells us how much each frequency is delayed, we simply need a DFT length longer than the maximum delay. If the DFT is not long enough, the specified delays will alias.

If the group delay changes slowly and smoothly, there will be little spectral leakage between the DFT bins. If the group delay changes quickly or there are large discontinuities between frequency indices of the sampled group delay, there is a higher likelihood of spectral leakage. This can be partially mitigated by using a longer DFT length. We typically use a DFT length that is twice as large as the maximum delay and trim the length of the resulting impulse response.

5. APPLICATIONS

The filters described above can be used in a variety of applications. When the total duration of the impulse response is long, these filters can be quite musical on their own. They can be, and have been, used as sound effects and musical components of electro-acoustic music [38, 39]. When the total IR duration is short, these filters can be useful for processing other sounds. For example, introducing chords and arpeggios to a piano, complex echoes and delays to drums, "birdies" to the transients of guitar strums, and inharmonic distortion to vocals. Some audio examples of these filters as sound effects and processors can be found online at `https://ccrma.stanford.edu/~kermit/website/gdapf.html`.

In addition to musical sounds, these filters have other practical uses. When the maximum group delay is shorter than about 30 ms, one does not necessarily perceive the frequency-dependent delays and the IR of the filter could sound like a click. If multiple, different filters are used together, these filters make effective decorrelators.

Like for decorrelation, if one generates many mutually decorrelated allpass filters, one could foreseeably use them for steganography, where a message or data is encoded with a set of filters that can only be decoded by correlating the code with the correct key.

These filter can also be used for impulse response measurement. While sine sweeps, Golay codes, and pseudo-random noise sequences are effective tools for probing systems, they are all unpleasant and aggressive sounds. One could use the filter design approach from this paper to create "musical" test signals that are less irritating to hear.

6. CONCLUSIONS AND FUTURE WORK

In this paper we have demonstrated a novel method for sound processing and synthesis which uses allpass filters formed by setting a group delay trajectory that sets frequency-dependent delays. By choosing the group delay characteristic, these filters can create a large variety of interesting sounds either on their own or for processing other sounds. These filters are passive and energy conserving, and are useful for abstract sound synthesis, audio effects processing, decorrelation, steganography, and impulse response measurement, among others.

We have shown several methods for constructing the group delay, including stair-stepped, modulated, probabilistic, and hand-drawn methods. We also showed a method for equalizing the allpass filter to have a near constant amplitude envelope. In addition to creating new sounds and effects, these filters can be used to produce new takes on classic audio effects.

The filters presented here have all been static. Moving forward, we would like to find an efficient method for implementing these filters that can accommodate time-varying designs.

7. REFERENCES

[1] John R. Pierce and Scott A. Van Duyne, "A passive nonlinear digital filter design which facilitates physics-based sound synthesis of highly nonlinear musical instruments," *The Journal of the Acoustical Society of America*, vol. 101, no. 2, pp. 1120–6, 1997.

[2] Jyri Pakarinen, Vesa Välimäki, and Matti Karjalainen, "Physics-based methods for modeling nonlinear vibrating strings," *Acta acustica united with Acustica*, vol. 91, no. 2, pp. 312–25, 2005.

[3] Matti Karjalainen and Julius O. Smith, "Body modeling techniques for string instrument synthesis," in *Proceedings of the International Computer Music Conference*, 1996.

[4] Jukka Rauhala and Vesa Välimäki, "Tunable dispersion filter design for piano synthesis," *IEEE Signal Processing Letters*, vol. 13, no. 5, pp. 253–6, 2006.

[5] Gianpaolo Borin, Davide Rocchesso, and Francesco Scalcon, "A physical piano model for music performance," in *Proceedings of the International Computer Music Conference*, 1997, pp. 350–3.

[6] Matti Karjalainen, Vesa Välimäki, and Paulo A. A. Esquef, "Efficient modeling and synthesis of bell-like sounds," in *Proceedings of the 5th International Conference on Digital Audio Effects*, 2002, pp. 181–6.

[7] David A. Jaffe and Julius O. Smith, "Extensions of the Karplus-Strong plucked-string algorithm," *Computer Music Journal*, vol. 7, no. 2, pp. 56–69, 1983.

[8] Timo I. Laakso, Vesa Välimäki, Matti Karjalainen, and Unto K. Laine, "Splitting the unit delay: FIR/all pass filters design," *IEEE Signal Processing Magazine*, vol. 13, no. 1, pp. 30–60, 1996.

[9] Elliot Kermit-Canfield and Jonathan Abel, "Signal decorrelation using perceptually informed allpass filters," in *Proceedings of the 19th International Conference on Digital Audio Effects*, 2016.

[10] Elliot K. Canfield-Dafilou and Jonathan S. Abel, "A group delay-based method for signal decorrelation," in *Proceedings of the 144th Audio Engineering Society Convention*, 2018.

[11] Manfred R. Schroeder, "Natural sounding artificial reverberation," *Journal of the Audio Engineering Society*, vol. 10, no. 3, pp. 219–23, 1962.

[12] Jonathan S. Abel, David P. Berners, Sean Costello, and Julius O. Smith III, "Spring reverb emulation using dispersive allpass filters in a waveguide structure," in *Proceedings of the 121st Audio Engineering Society Convention*, 2006.

[13] David Griesinger, "Impulse response measurements using all-pass deconvolution," in *Proceedings of the 11th International Audio Engineering Society Conference: Test & Measurement*, 1992.

[14] Elliot K. Canfield-Dafilou and Jonathan S. Abel, "An allpass chirp for constant signal-to-noise ratio impulse response measurement," in *Proceedings of the 144th Audio Engineering Society Convention*, 2018.

[15] Phillip Regalia and Sanjit Mitra, "Tunable digital frequency response equalization filters," *IEEE transactions on acoustics, speech, and signal processing*, vol. 35, no. 1, pp. 118–20, 1987.

[16] Federico Fontana and Matti Karjalainen, "A digital bandpass/bandstop complementary equalization filter with independent tuning characteristics," *IEEE Signal Processing Letters*, vol. 10, no. 4, pp. 119–22, 2003.

[17] Julius O. Smith and Jonathan S. Abel, "Bark and ERB bilinear transforms," *IEEE Transactions on speech and Audio Processing*, vol. 7, no. 6, pp. 697–708, 1999.

[18] Stephan Herzog and Marcel Hilsamer, "Low frequency group delay equalization of vented boxes using digital correction filters," in *Proceedings of the 17 International Conference on Digital Audio Effects*, 2014.

[19] Phillip Regalia and Sanjit Mitra, "A class of magnitude complementary loudspeaker crossovers," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 35, no. 11, pp. 1509–16, 1987.

[20] William M. Hartmann, "Flanging and phasers," *Journal of the Audio Engineering Society*, vol. 26, no. 6, pp. 439–43, 1978.

[21] Julius O. Smith, "An allpass approach to digital phasing and flanging," in *Proceedings of the International Computer Music Conference*, 1984.

[22] Jon Dattorro, "Effect design, part 2: Delay line modulation and chorus," *Journal of the Audio engineering Society*, vol. 45, no. 10, pp. 764–88, 1997.

[23] Jussi Pekonen, "Coefficient modulated first-order allpass filter as a distortion effect," in *Proceedings of the 11th Conference on Digital Audio Effects*, 2008, pp. 1–8.

[24] Joseph Timoney, Victor Lazzarini, Jussi Pekonen, and Vesa Välimäki, "Spectrally rich phase distortion sound synthesis using an allpass filter," in *IEEE International Conference on Acoustics, Speech and Signal Processing*, 2009, pp. 293–6.

[25] Joseph Timoney, Victor Lazzarini, Brian Carty, and Jussi Pekonen, "Phase and amplitude distortion methods for digital synthesis of classic analog waveforms," in *Proceedings of the 126th Audio Engineering Society Convention*, 2009.

[26] Julian Parker and Vesa Välimäki, "Linear dynamic range reduction of musical audio using an allpass filter chain," *IEEE Signal Processing Letters*, vol. 20, no. 7, pp. 669–72, 2013.

[27] David Kim-Boyle, "Spectral delays with frequency domain processing," in *Proceedings of the 7th International Conference on Digital Audio Effects*, 2004.

[28] Jussi Pekonen, Vesa Välimäki, Jonathan S. Abel, and Julius O. Smith, "Spectral delay filters with feedback and time-varying coefficients," in *Proceedings of the 12th International Conference on Digital Audio Effects*, 2009.

[29] Vesa Välimäki, Jonathan S. Abel, and Julius O. Smith III, "Spectral delay filters," *Journal of the Audio Engineering Society*, vol. 57, no. 7/8, pp. 521–31, 2009.

[30] Greg Surges and Tamara Smyth, "Spectral distortion using second-order allpass filters," in *Proceedings of the Sound and Music Computing Conference*, 2013, pp. 525–31.

[31] Greg Surges, Tamara Smyth, and Miller Puckette, "Generative audio systems using power-preserving all-pass filters," *Computer Music Journal*, 2016.

[32] Markus Lang, "Allpass filter design and applications," *IEEE Transactions on Signal Processing*, vol. 46, no. 9, pp. 2505–14, 1998.

[33] Zhongqi Jing, "A new method for digital all-pass filter design," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 35, no. 11, pp. 1557–64, 1987.

[34] Jonathan S. Abel and Julius O. Smith, "Robust design of very high-order allpass dispersion filters," in *Proceedings of the International Conference on Digital Audio Effects*, 2006, pp. 13–8.

[35] William G. Gardner, "Efficient convolution without latency," *Journal of the Audio Engineering Society*, vol. 43, pp. 2, 1993.

[36] Guillermo Garcia, "Optimal filter partition for efficient convolution with short input/output delay," in *Proceedings of the 113th Audio Engineering Society Convention*, 2002.

[37] Frank Wefers and Michael Vorländer, "Optimal filter partitions for real-time FIR filtering using uniformly-partitioned FFT-based convolution in the frequency-domain," in *Proceedings of the 14th International Conference on Digital Audio Effects*, 2011, pp. 155–61.

[38] Elliot K. Canfield-Dafilou, "Phasing cranes," 2018.

[39] Elliot K. Canfield-Dafilou, "1've c0unted 7hat 1 b4," 2018.

ASSESSING THE EFFECT OF ADAPTIVE MUSIC ON PLAYER NAVIGATION IN VIRTUAL ENVIRONMENTS

Manuel López Ibáñez
Department of Software Engineering and Artificial Intelligence
Complutense University of Madrid
Madrid, Spain
manuel.lopez.ibanez@ucm.es

Nahum Álvarez
Artificial Intelligence Research Center
National Institute of Advanced Industrial Science and Technology
Tokyo, Japan
nahum.alvarez@aist.go.jp

Federico Peinado
Department of Software Engineering and Artificial Intelligence
Complutense University of Madrid
Madrid, Spain
email@federicopeinado.com

ABSTRACT

Through this research, we develop a study aiming to explore how adaptive music can help in guiding players across virtual environments. A video game consisting of a virtual 3D labyrinth was built, and two groups of subjects played through it, having the goal of retrieving a series of objects in as short a time as possible. Each group played a different version of the prototype in terms of audio: one had the ability to state their preferences by choosing several musical attributes, which would influence the actual spatialised music they listened to during *gameplay*; the other group played a version of the prototype with a default, non-adaptive, but also spatialised soundtrack. Time elapsed while completing the task was measured as a way to test user performance. Results show a statistically significant correlation between player performance and the inclusion of a soundtrack adapted to each user. We conclude that there is an absence of a firm musical criteria when making sounds be prominent and easy to track for users, and that an adaptive system like the one we propose proves useful and effective when dealing with a complex user base.

1. INTRODUCTION

Most video game design challenges are related to the scope of possible player decisions. Current-generation open-world games offer an enormous variety of places to go and things to do, which makes designing specific player behaviour a daunting task. The problem of guiding a player through a big and complex virtual environment is frequently solved by adding extradiegetic information to the graphical user interface (GUI), thus reducing *presence* [1] and *immersion* [2]. Games like *Horizon: Zero Dawn*[1] overcome this problem by justifying the overabundance of head-up display (HUD) elements with an in-game excuse (in this particular case: a high-tech tracking device the main character wears). However, this is not always possible for every video game.

Through this article, we describe our study on how to guide a player in a virtual environment exclusively using audio. Our premise is that we can reduce the need of a cluttered GUI while retaining immersion and player performance, by letting participants firstly choose their preferred sound attributes and then adapting the soundtrack to these preferences.

In section 2, we start by analysing previously published work on adaptive music and player navigation using sound clues. In the next section, the experiment used to validate our proposal is described; its results are later summarized in section 4. Lastly, in sections 5 and 6, we include a brief discussion on the implications of our findings and several conclusions about how these ideas could be used in the design of a commercial adaptive music system for video games.

[1] https://www.guerrilla-games.com/play/horizon

2. PLAYER NAVIGATION AND ADAPTIVE MUSIC

The idea behind this article emerged from our previous work [3], which suggested there could exist a correlation between variations in the basic elements of a certain soundtrack and player decisions during an interactive experience: harmonic, high-pitched melodies seemed to attract users more efficiently than cacophonous, low-pitched ones. However, participant's reactions and behaviour varied greatly depending on the result each user achieved during the Bartle test [4]: certain groups of subjects were attracted to musical attributes which did not work as a lure for others. This led to the conclusion that personal auditive preference is important when using sound to orient players in video games.

We were also inspired by previous research with blind people [5], which acknowledges the existence of a conceptual level, in addition to a perceptual one, in the learning process associated with scouring an unknown environment in search for clues that allow to build mental, 3D "maps". This kind of perspective is of utmost importance for our research, because we base our work on the existence of culturally attained categories which relate to formal auditive parameters.

2.1. An adaptive music system

Adaptive music used in video games usually consists of an atmospheric, non-spatialised soundtrack which changes in response to specific events taking place in the virtual world. Said changes can happen procedurally or be previously scripted.

Due to the existence of very different social groups in terms of musical perception, we decided to build an adaptive, live music system, so as to respond in real time to player decisions while they play video games. This system is called LitSens [6, 7], and works by automatically combining short fragments of music composed by a human. LitSens was used in the present research as an audio foundation for the game we utilised in the experiment, which is described in section 3.

Our intention was to parameterise a series of basic musical attributes, so as to be able to modify them in real time with ease and efficiency. Our approach was similar to those of systems like ANTESCOFO [8], which go beyond pitch in terms of simple audio descriptors.

Furthermore, LitSens approaches adaptive music from an emotional perspective. The idea behind it is to adapt to player's emotional responses, in a way that allows a game designer to provide an adaptive soundtrack without taking into account every possible interaction or outcome. Wallis, Ingalls & Campana [9] approach this problem in a very similar way: they extract certain components, such as valence and arousal, from common emotional models, and apply them to music generation in real time.

2.2. Guiding players in virtual environments

The problem of guiding player movement in a virtual environment is a common issue in game design. Some authors, like Milam & El Nasr [10], have established a taxonomy of design patterns in 3D games, aiming to standardise these strategies, but academic work on player orientation through sound is scarce. Additionally, sound is not usually even taken into account when building these guiding techniques: none of the five patterns proposed by Milam & El Nasr make explicit use of sound. This opens an unexplored field of possibilities for game designers, who usually rely on visual clues.

It is not uncommon, however, to find alternative guiding techniques based on a video game's narratives. Earlier approaches, like the one presented by T. A. Galyean [11], rely on a path established by a narrative, which the user must follow to keep up. Recent commercial video games, like *The Stanley Parable*[2], *Dear Esther*[3] or *Gone Home*[4] all rely on narrative elements (e.g.: the voice of a narrator) to guide players to the next goal or important milestone. However, these techniques also rely on small or highly controlled virtual environments. Entangled paths or huge, open worlds require a different approach, and sound could be the key to solve the problem of subtle navigation assistance.

2.3. Auditive preference and meaningful variations

Additionally, Eisenberg & Forde[12] show that it is possible to establish a series of simple predictors, like creativity, complexity or technical goodness, which explain the variations in preference during a human evaluation of music. Though people's musical taste or preference is commonly measured and evaluated with musical genres in mind [13], we are interested in modifying simple auditive features, which allow for a more flexible approach and are consistent with an adaptive music system such as LitSens. We used complexity, pitch and rhythm as the three modifiable attributes during our experiment, as will be explained in section 3. This decision is consistent with previous uses of musical complexity (in and out-of-key notes, harmonic versus dissonant layering) [14], pitch (high and low tone) [15] and rhythm (slow or fast) [16] to produce perceptible changes when listening to audio fragments. The technique we used to increase complexity was simply to introduce layers of sound –formed by out of tune intervals and dissonant chords– that disrupted the harmony of a base track. This can be appreciated when comparing the two spectrograms depicted in figure 1. Pitch and rhythm modifications were made without adding any layer to the base mix; instead, we simply modified those values in real time for the whole track using commands from the game engine.

As for what makes a sound "stand out" over others, a very common opinion, based on classic works by Fletcher & Munson (the famed Fletcher-Munson curves) [17] is that a higher pitch – around 2000 and 5000 Hertz (Hz)– will usually dominate a mix in terms of perceived loudness. However, it has been known for a long time that listener's perception of several auditive attributes, included tone dominance, can be influenced by many different factors. Regarding pitch perception, in certain conditions[18], lower frequencies can be dominant. In the context of this research, dominance is a determinant factor when identifying and following spatialised sounds.

3. EXPERIMENT DESIGN

The following experiment had the objective of exploring the relationship between the presence or absence of adaptive music in a video game and player performance while solving a labyrinth-like orientation puzzle. It also measured the level of coherence between users' perception of sounds and their actual response to them.

3.1. Design

Before starting with the experiment, all participants were randomly distributed in two groups: A and B. Initially, both groups had the same size ($N = 17$), though group A lost a subject due to hearing health problems. Throughout the experiment, only two persons were in the area at a time: one participant and one test supervisor. There were four differentiated phases in every session: SAM test, attribute selection, game playing and sociological survey.

Subjects from group A started by taking a Self-Assessment Manikin (SAM) test [19, 20] about three pairs of sounds. Each pair was played consecutively, and had a strong relationship with one of the basic categories used to classify sounds in our test-bed game. The sounds in every pair represented the two opposed concepts for each of the following categories, presented in order in the test: tone (low-high), structure (simple-complex) and rhythm (slow-fast). The differences between the sounds of each category were big enough to be easily noticeable, as can be seen in figure 1, and during the test all sounds were evaluated separately after listening to each pair, in order to compare them.

The SAM test was passed in its 9-point scale version, by means of a digital form which contained all three measurements: emotional valence, arousal and dominance. This test uses the Semantic Differential [21] as a basis, and simplifies it. Thus, *emotional valence* measures "pleasure", and is strongly related to bipolar adjective pairs such as unhappy-happy, annoyed-pleased, unsatisfied-satisfied, melancholic-contented, despairing-hopeful or bored-relaxed. *Arousal*, on the other hand, is related to pairs like relaxed-stimulated, calm-excited, sluggish-frenzied, dull-jittery, sleepy-wide awake, unaroused-aroused. Lastly, *dominance* is related to adjectives like controlled-controlling, influenced-influential, cared for-in control, awed-important, submissive-dominant and guided-autonomous.

Subjects from group B were given the same test, but only evaluated one sound. This sound contained the default audio played by their version of the game, classified as: slow, low and simple. This evaluation was not taken into account later and it was performed to give the subject of this group the same insight than the subjects in group A about the auditive nature of the experiment, in order to avoid possible bias.

[2] https://www.stanleyparable.com/

[3] http://www.dear-esther.com/

[4] https://www.gonehome.game/

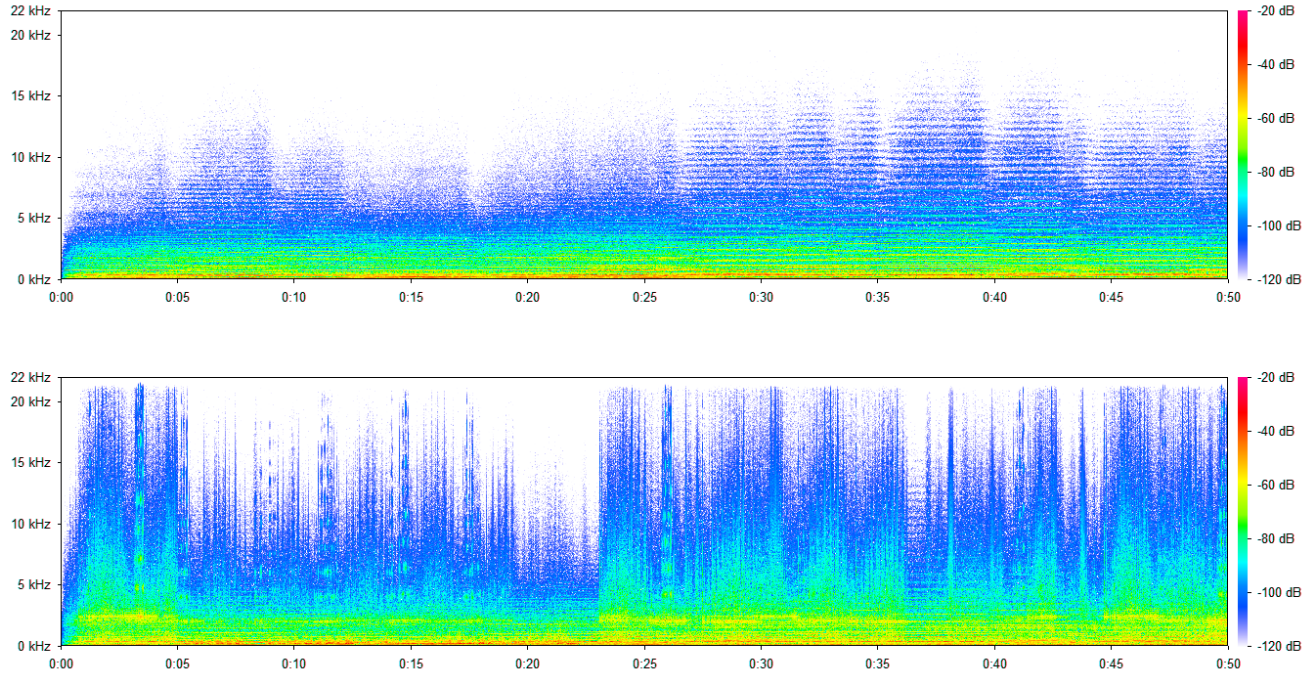


Figure 1: *Spectrogram of simple (top) and complex (bottom) variations of the same sound.*

Once finished with the test, subjects from both groups had to launch our software, which ran as a full-screen computer game developed with Unreal Engine 4 [5].

People in group A (experimental group) were asked to select, from an in-game menu with three categories (rhythm, tone and structure), the attribute for each of them (slow-fast, low-high simple-complex), which, from their point of view, would make a sound stand out over the rest. Thus, a total of 8 final outcomes were possible. People in group B (the control group) were not given this option, and played with default audio. No sound clues were included to help users from group A decide: the only previous reference was the SAM test. This was done in order to evaluate the coherence between subjects' perception of what sound suits them better and actual performance produced by their selection.

For group A, a personalized level was loaded after their preferences were specified. For group B, the level loaded with the default sounds (low tone, slow rhythm, simple structure). Said level consisted of a three dimensional labyrinth, played from a first-person perspective. From a logical standpoint, however, its structure can be considered two dimensional; it is depicted as a map in figure 2.

Players could move and look around using a keyboard (WASD keys) and a mouse. Every user was told to look for and recover a total of three statuettes inside this labyrinth, as quickly as possible. Elapsed time and number of statuettes recovered were shown on the screen permanently to keep the player informed at any time about his goal. The only way to recover a statuette was to step on it. Every time one of them was picked up, a measure of total elapsed time was stored in a log file. At the end of each session, this log was retrieved and tagged with the correspondent participant number. From now on, we will call the tree time measurements as follows, for convenience: t_1 (first statuette), t_2 (second statuette) and t_3 (third statuette, or total time).

Every statuette emitted a spatialised, monophonic music track which blended with a base stereophonic soundtrack. The base soundtrack was a low, synthetic drone, with no variations in tone or intensity. For users in group A, the emitted track was modified to adapt to their specified preferences in musical attributes. For users in group B, the track was always the default one. Once recovered, the statuette stopped emitting sound in every case, by means of a 2 second linear fade out. If the spatialised audio track was received by the camera listener through a wall, a low-pass filter with a cutoff frequency of 900 Hz was applied.

When all three objects were recovered, the game ended and the application was closed. After finishing with the game, every subject from both groups had to take a brief test to determine their sociological profile. Data retrieved included: age, sex, country of birth, level of education completed, presence of hearing problems, fondness for music and sound and performance when playing video games.

Subjects were also asked if sound was useful when trying to find the three statuettes inside the virtual labyrinth. Results from this question constitute a variable we named "help index" (h_i). A Likert 5-point scale [22, 23] was employed for this and all questions requiring gradation, except for the SAM test, where a 9-point scale was utilised.

Two leaflets with instructions were created, one for each group. Every subject had to read only the pertinent one while waiting to begin. These documents contained a detailed description of all actions every user would have to take during the experiment. Brief instructions on how to listen to the sounds and how to take the SAM test were included, as well as keyboard and mouse controls for the video game. All users were also told it was of utmost

[5] https://www.unrealengine.com/en-US/what-is-unreal-engine-4

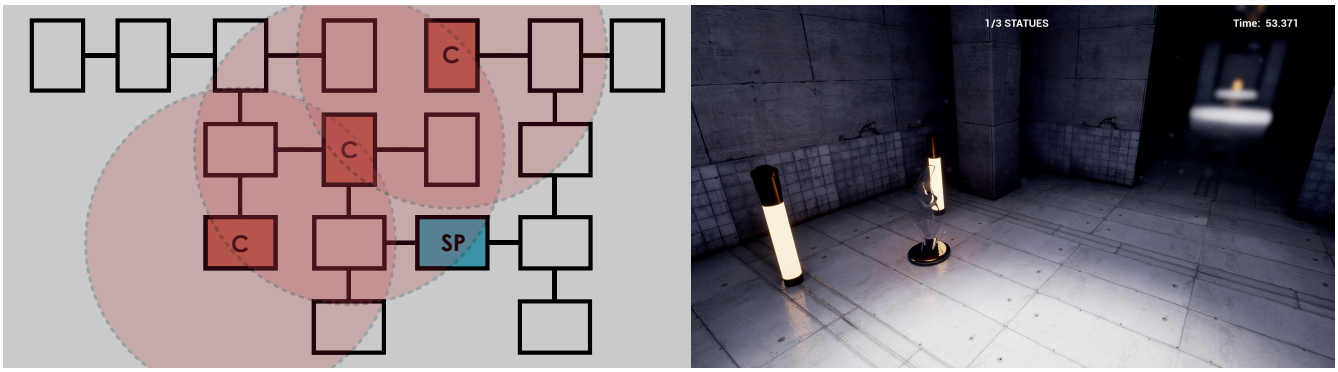


Figure 2: *Diagram showing the layout of the virtual environment utilised during the experiment. There is a starting point (SP) and three collectibles (C) in the form of statuettes. Red circles represent the area of influence of each sound. Walls applied occlusion through a low-pass filter, not depicted in the diagram. A capture of the game is also shown on the right.*

importance to complete the level in as few seconds as possible, and that they had to find three small statuettes to do so. The only difference between "A" and "B" versions was the lack of explanation on how to evaluate pairs of sounds (since this was not necessary for group B).

To evaluate results from both the 5 point Likert scales and the 9 point SAM scales a parametric, unpaired test (Student's t test) was utilised.

Finally, after finishing with the experiment, all subjects from group A were asked to explain, in their own words, the reasons for their attributes selection.

3.2. Hypothesis

Our hypothesis was that a statistical difference may be found between the two groups of users (A and B), in terms of performance (measured in total time, t_3), with the conditions established above. Our independent variable is the presence or absence of a preference selector at the beginning of the experiment that influences music played in the game. We also aimed to find a relationship between the initial selection of auditive features (available to participants in group A only) and t_3.

3.3. Demography

There existed two prerequisites participants had to meet so as to take the experiment: the ability to hear properly and having played at least one video game of the first person shooter (FPS) genre. All subjects met these requirements, and were students (graduate and postgraduate) or worked as lecturers in the field of Computer Science.

In total, 33 subjects participated in our experiment, of which 16 were assigned to group A (composed of 14 males and 2 females) and 17 (with 15 males and 2 females) to group B. From the total number, 29 were born in Spain, and the other 4 were from Colombia, Bolivia, Switzerland and Venezuela. All of them were native speakers of Spanish, which was the language used throughout the whole experiment. Also, they shared similar cultural features, and all but one had lived most of their lives in Spain.

Average ages in groups A and B were similar: 23,438 (A) and 24,059 (B) years. The mode was 18 in both cases, as most participants were first-year students.

68.8 % of the participants were undergraduate students, whereas 6.3 % were studying a master's degree at the moment. The rest were Ph. D. students (12.4 %) or university professors and researchers (12.5 %).

When asked if they played games frequently, most subjects in groups A and B answered positively, with a mode of 5 out of 5 in both cases, and a mean of 4.5 (A) and 4.412 (B). They also considered themselves good video game players, achieving a mode of 4 out of 5 for both groups and an average of 3.688 (A) and 3.824 (B). These scores were slightly lower when asking them if they were good with FPS games: the mode was 3 out of 5 in A and B, while the averages were 3.5 (A) and 3.353 (B).

As for self-evaluation of their hearing proficiency, when asked if they have good hearing, the modes were 5 (A) and 4 (B) out of 5, and the averages, 4.125 (A) and 4 (B). Besides, when told to answer if they have a "good ear" for music, the mode was 4 out of 5 in both cases, and the averages were 3.688 (A) and 3.353 (B).

Musicians were selected and distributed evenly between groups, with a total of 2 in each one. This was done to avoid possible bias due to their knowledge of music and audio, and they were the only participants which were not randomly distributed. This process occured before starting with the experiment, and the affected participants were unaware of it.

This leaves us with a surveyed sample which has a very good perception of their own hearing, but with an average-to-neutral confidence in their "musical ear".

4. RESULTS

The results of the previously described experiment (and its associated survey) point to several statistically significant differences between groups A and B in terms of both performance and self-assessment.

Subjects from group A achieved a total average time of completion (t_3) of 78.108 seconds, whereas participants in group B took an average of 132.987 seconds. The median in group A is 75.694, while in group B is 100.668. The lack of similarity between average and median times in group B can be explained by the presence of two clear outliers (as seen in figure 3), who completed the level in 369.250 and 367.020 seconds respectively. A parametric analysis of these results can be consulted in Table 1.

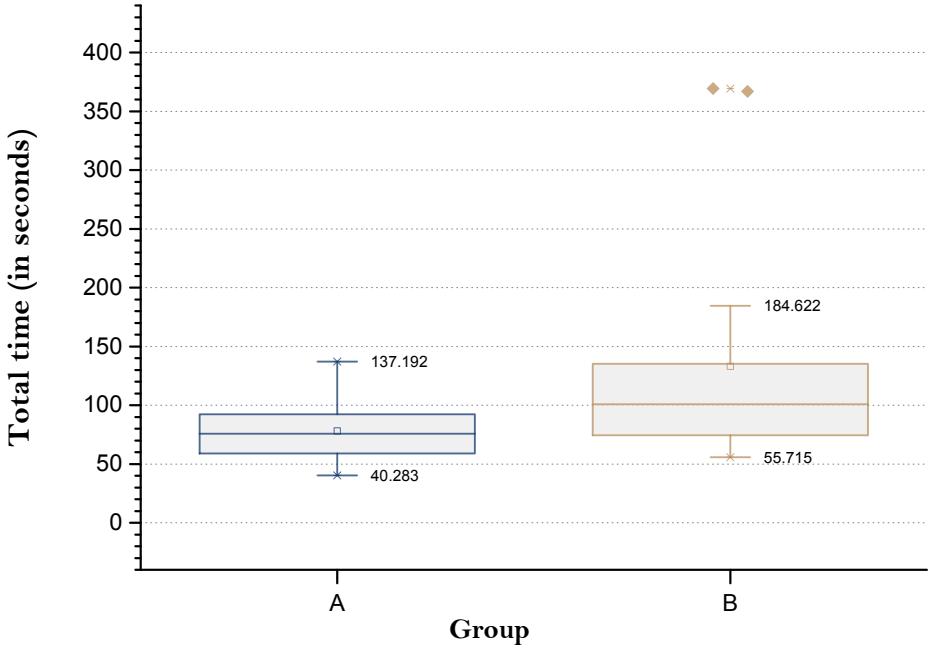


Figure 3: *Difference between groups A and B in total time (t_3).*

Table 1: *Student's t-Test for total time (t_3) in groups A and B.*

Group	A	B
N	16	17
Mean	78.108	132.987
Standard deviation	27.908	96.090
Two-tailed P value:	0.0356	

Table 2: *Student's t-Test for h_i values in groups A and B.*

Group	A	B
N	16	17
Mean	3.56	2.47
Mode	5	1
Standard deviation	1.46	1.59
Two-tailed P value:	0.0484	

Because time was measured when every statuette was picked up, not only total elapsed time gave an important insight about player behaviour during the experiment. It is also quite illustrative to look at how the difference in average time between the two groups increases as every object is taken. t_1 had an average value of 22.068 for group A, and of 25.637 for group B, so the difference between means equals 3.569 seconds. t_2 has an average value of 41.854 for group A and of 53.398 for group B, producing a difference of 11.544 seconds. Lastly, t_3 presents the biggest difference: 54.879 seconds.

As can be seen in Table 2, when it comes to the help index (h_i), there are also statistically significant differences between groups. Out of 5, group A has a mean of 3.56, while group B scores 2.47. The mode is particularly enlightening in this case: 5 in group A and 1 in group B.

There does not exist a strong statistical relationship between t_n and the initial selection of auditive features, which was only possible for members inside group A, as Table 3 shows. "High" (9) "fast" (9) and "simple" (11) were the most common options, however.

It is also worth mentioning that the attribute "complex" was the least selected (only 5 users chose it). However, this same attribute obtained the highest dominance score during the SAM test, with an average of 5.875 and a mode of 7 out of 9. It also had the highest excitement score, averaging 5.688 and with a mode of 7 out of 9. Additionally, general results from the SAM test were not consistent with player selections of attributes before playing the game (see Table 4).

It was not uncommon that, in spite of considering a complex sound more dominant, players chose a simple one instead before playing the game. Opinions around the very concepts of valence, dominance or arousal when it comes to detecting spatial sound were varied, and every user ended up choosing what appealed to them most. For example, when asked about the reason for their selection, 3 users mentioned "storms" or "thunder" as a reason for considering low tones useful when trying to orient themselves. They found those sounds "easy to track", "full of energy" or "very deep". The rest gave similar reasons for their decisions, based on personal experiences.

Table 3: *Features selected by group A participants and total time achieved (t_3).*

t_3	**Tone**	**Rhythm**	**Complexity**
40,283	High	Fast	Simple
42,159	High	Fast	Simple
45,318	High	Fast	Simple
57,027	Low	Slow	Simple
60,738	Low	Fast	Simple
66,320	Low	Slow	Simple
73,127	Low	Slow	Complex
73,483	High	Slow	Simple
77,904	High	Fast	Simple
81,639	High	Fast	Simple
84,315	High	Fast	Complex
92,305	Low	Slow	Complex
92,315	Low	Slow	Simple
95,468	Low	Fast	Complex
130,129	High	Fast	Complex
137,192	High	Slow	Simple

Table 4: *SAM test results in 9 point scale for variations of the same sound.*

Attribute	**SAM scale**	**Average**	**Mode**
1. High tone	Valence	5.938	7
	Arousal	3.625	2
	Dominance	4.5	5
2. Low tone	Valence	4.697	3
	Arousal	3.152	2
	Dominance	4.727	3
3. Simple structure	Valence	5.688	4
	Arousal	3.563	3
	Dominance	4.188	3
4. Complex structure	Valence	3.375	5
	Arousal	5.688	7
	Dominance	5.875	7
5. Fast tempo	Valence	6.063	7
	Arousal	5.25	7
	Dominance	5.188	5
6. Slow tempo	Valence	5.375	6
	Arousal	3.438	3
	Dominance	5.063	5

5. DISCUSSION

Considering the information retrieved through the previously explained results, we can extract a series of final deductions. First, the independent variable (the presence or absence of an attribute selector affecting music in the game) seems to be statistically related to the difference in total time (t_3) obtained by users during the experiment.

Besides, subjects in group A had a higher result in h_i, which means they perceived music as a helper more than participants in group B. Precedents for this effect have not been found in pertinent academic literature.

We have also observed that $t_n - t_{n-1}$ greatly increases with every measurement –whenever a statuette was gathered. This is inversely proportional to the number of statuettes present in the map. It is reasonable to think the amount of time elapsed in finding a statuette can increase when their remaining number is lower, because the probability of finding them by chance is also reduced. A need to backtrack and search more thoroughly also emerges when there are fewer objects to retrieve. However, the increasing variation in average t_n between groups A and B (see section 4) points to another, more important correlation. If we take into account that both prototypes (A and B) were identical except for the personalized music, it is possible to link the differences in mean time to the differences in audio.

Moreover, there were some counterintuitive aspects in the results. For example: the lack of consistency between SAM test results and player preference when selecting attributes inside the prototype could be happening due to multiple reasons. We have not retrieved enough information during our experiment to give a clear response to this particular matter, but several new and interesting lines of research are open as a result. Our main hypothesis for this unexpected behaviour is that the mere act of selecting sound attributes while already playing the game may not be in line with the mental state of the subject when answering the SAM test. While the test is a more relaxed experience, which is not limited by time constraints, the video game asks players to concentrate much more, and gives them a clear goal. As a consequence, it is possible that different attributes are found dominant in these different contexts, creating the mentioned variations in the results.

Another possible reason is that users learned to better identify dominant attributes through the duration of the SAM test, taking into account the specific variations in complexity, pitch and rhythm presented to them. This would mean the first answers would be less informed than the last ones, and that their decisions inside the final selector would imply a previous and meticulous "weighting up" of every possible option.

An appropriate new line of experimentation would involve distributing subjects in two groups in which the order of the test and the attribute selection would be inverted. Also, the SAM test only accounts for emotional scales (valence, arousal and dominance), and different measures could be needed to determine how easy to track a sound is for different persons, as sounds traditionally considered to be more dominant may not be easier to track for all users, and personal preference could be more important than dominance when it comes to finding sound sources in virtual environments.

Previous statements aside, user capacity to select attributes and user performance are, nevertheless, statistically related in our results. Consequently, we can state the mere ability to choose correlates with a lower average time of completion in group A, when compared to group B.

Other issues exist concerning data recovery and user distribution. For example: if we follow the Central Limit Theorem [24], our presumption of normal distribution would only solidly apply to groups with a number of participants (N) equal or greater than 30. However we want to note that our t_3 histogram forms a bell-like curve in both groups, even with less data, and the confidence interval of the mean is high enough (95 %) to trust the results. Nonetheless, a bigger sample would be needed to increase the reliability of the outcome. A similar problem is also the lack of women in our sample (only 4 out of 33 participants), which produces a genre bias and makes our retrieved data only strictly applicable to men. We aim to solve these predicaments in future iterations of this research.

6. CONCLUSIONS AND FUTURE WORK

The most relevant conclusion we can extract from the present research is the influence the mere act of selecting preferred attributes has over player performance when solving our 3D labyrinth. This effect, whilst somewhat predictable, was not verified in the past in any other research, so we open the path to further explore the consequences this causal relationship has in user behavior. For example, we observed that user preferences when selecting sounds differ from the ones chosen in the SAM test, so a future experiment would be necessary to analyse the rationale of this behavior.

Additionally, a similar experiment, based on player orientation, but without any kind of spatialisation or multichannel audio (that is: playing sound in mono in all channels), may help us elucidate if there are purely musical attributes that can make players take one path or another by themselves.

The increase in performance achieved when using adaptive, spatialised music may also make a preference selection system like the one we propose useful in different 3D environments where the inclusion of a GUI is not an option (such as virtual reality interactive experiences).

We noted also the surprising variety in attributes selected by subjects in group A. This attribute variety suggests the existence of a very complex population in terms of auditive preference when it comes to player orientation. Analysing this aspect in a bigger population and in more detail may prove useful for understanding what a "dominant" sound is in this context.

Another step we would like to take in the future to further validate our system would be to include LitSens in a commercial first-person video game and test whether we can guide players in bigger, more complex virtual environments.

Also, the development of an intelligent system integrated in LitSens, as a way to adapt to player musical preferences without a previous test, might improve immersion while reducing even more the amount of GUI elements needed. As the mentioned system already has the capacity to produce continuous adaptive music, only a new logic for the automatic selection process should be needed.

Lastly, it would also be useful to research how the level of presence achieved by systems which rely on GUI elements to guide a player varies when compared to systems using only sound to achieve similar results.

Consequently, the next experimental iteration for LitSens would have to take place in two separate steps: On one hand, the development of an intelligent system which would take into account player actions and camera movement to evaluate how users' context affects auditive predilections and consequently how this preferences impact performance.

On the other, an experimental validation with three groups of users ($N \geq 30$), which ought to include an implementation of the system in a commercial video game with open world environments and a presence test for all subjects (such as the Temple Presence Inventory [25]). Again, participants from group A would have access to adaptive, spatialised music, while group B would listen to a default, non-adaptive but spatialised soundtrack. Group C would listen to adaptive audio without any spatialisation (mono). Performance would be tested in terms of time when completing a navigation-related task, and camera movement would also be recorded.

We think that there is still much room for improvement in the field of intelligent management of sound systems for user navigation, especially when compared to the current state of image-based systems, which are much more developed. Audio is a less explored field in terms of semantic guidance, but it could substantially improve immersion and presence in virtual environments and be a useful tool for game designers. This is especially true when developing first-person or virtual reality experiences which cannot rely as heavily on GUI.

7. ACKNOWLEDGMENTS

This work has been partially supported by project *ComunicArte: Comunicación Efectiva a través de la Realidad Virtual y las Tecnologías Educativas*, funded by *Ayudas Fundación BBVA a Equipos de Investigación Científica 2017*, and project *NarraKit VR: Interfaces de Comunicación Narrativa para Aplicaciones de Realidad Virtual (PR41/17-21016)*, funded by *Ayudas para la Financiación de Proyectos de Investigación Santander-UCM 2017*.

We would also like to acknowledge the funding provided by Banco Santander, in cooperation with Fundación UCM, in the form of a predoctoral scholarship (CT2716 - CT2816) which contributed to the development of this research.

8. REFERENCES

[1] W. Barfield and S. Weghorst, "The sense of presence within virtual environments: A conceptual framework," *Advances in Human Factors Ergonomics*, vol. 19, pp. 699, 1993.

[2] C. Jennett, A. L. Cox, and P. Cairns, "Measuring and defining the experience of immersion in games," *International journal of human-computer studies*, vol. 66, no. 9, pp. 641–661, 2008.

[3] M. López Ibáñez, "Bartle Test Applications in Narrative Music Composition for Video Games," in *I Congreso Internacional de Arte, Diseño y Desarrollo de Videojuegos*, Madrid, 2015, pp. 1–13, ESNE.

[4] R. Bartle, "Hearts, Clubs, Diamonds, Spades: Players who suit MUDs," *Journal of MUD research*, vol. 6, no. 1, pp. 39, 1996.

[5] O. Lahav, "Improving orientation and mobility skills through virtual environments for people who are blind : Past research and future potential," *International Journal of Child Health and Human Development*, vol. 7, no. 4, pp. 349–355, 2014.

[6] M. López Ibáñez, N. Álvarez, and F. Peinado, "Towards an Emotion-Driven Adaptive System for Video Game Music," in *ACE 2017*, London, 2017.

[7] M. López Ibáñez, N. Álvarez, and F. Peinado, "LitSens: An Improved Architecture for Adaptive Music Using Text Input and Sentiment Analysis," in *C3GI 2017*, Madrid, 2017.

[8] A. Cont, "ANTESCOFO: Anticipatory Synchronization and Control of Interactive Parameters in Computer Music.," *Proceedings of the 2008 International Computer Music Conference, Belfast, Northern Ireland*, pp. 33–40, 2008.

[9] I. Wallis, T. Ingalls, and E. Campana, "Computer-Generating Emotional Music: the Design of an Affective Music Algorithm," *Proceedings of the 11th International Conference on Digital Audio Effects*, pp. 1–6, 2008.

[10] D. Milam and M. S. El Nasr, "Design Patterns to Guide Player Movement in 3D Games," *Proceedings of the 5th ACM SIGGRAPH Symposium on Video Games - Sandbox '10*, vol. 1, no. 212, pp. 37–42, 2010.

[11] T. A. Galyean, "Guided navigation of virtual environments," in *Proceedings of the 1995 symposium on Interactive 3D graphics - SI3D '95*, New York, New York, USA, 1995, pp. 103–ff., ACM Press.

[12] J. Eisenberg and W. F. Thompson, "A Matter of Taste: Evaluating Improvised Music," *Creativity Research Journal*, vol. 15, no. 2, pp. 287–296, jul 2003.

[13] M. Grimaldi and P. Cunningham, "Experimenting with music taste prediction by user profiling," *Proceedings of the 6th ACM SIGMM international workshop on Multimedia information retrieval - MIR '04*, p. 173, 2004.

[14] E. Fedorenko, A. Patel, D. Casasanto, J. Winawer, and E. Gibson, "Structural integration in language and music: Evidence for a shared system," *Memory and Cognition*, vol. 37, no. 1, pp. 1–9, 2009.

[15] J. H. McDermott and A. J. Oxenham, "Music perception, pitch, and the auditory system," *Current Opinion in Neurobiology*, vol. 18, no. 4, pp. 452–463, 2008.

[16] R. F. Day, C. H. Lin, W. H. Huang, and S. H. Chuang, "Effects of music tempo and task difficulty on multi-attribute decision-making: An eye-tracking approach," *Computers in Human Behavior*, vol. 25, no. 1, pp. 130–143, 2009.

[17] H. Fletcher and W. A. Munson, "Loudness, Its Definition, Measurement and Calculation," *Bell System Technical Journal*, vol. 12, no. 4, pp. 377–430, oct 1933.

[18] R. J. Ritsma, "Frequencies Dominant in the Perception of the Pitch of Complex Sounds," *The Journal of the Acoustical Society of America*, vol. 42, no. 1, pp. 191–198, jul 1967.

[19] M. M. Bradley and P. J. Lang, "Measuring emotion: The self-assessment manikin and the semantic differential," *Journal of Behavior Therapy and Experimental Psychiatry*, vol. 25, no. 1, pp. 49–59, 1994.

[20] B. Geethanjali, K. Adalarasu, A. Hemapraba, S. P. Kumar, and R. Rajasekeran, "Emotion analysis using SAM (Self-Assessment Manikin) scale," *Biomedical Research*, vol. 2, pp. 1–1, 2017.

[21] A. Mehrabian and J. A. Russell, *An approach to environmental psychology*, The MIT Press, Cambridge, 1974.

[22] R. Likert, "A technique for the measurement of attitudes," *Archives of Psychology*, vol. 22, no. 140, pp. 55, 1932.

[23] G. Norman, "Likert scales, levels of measurement and the "laws" of statistics," *Advances in Health Sciences Education*, vol. 15, no. 5, pp. 625–632, 2010.

[24] M. Rosenblatt, "A Central Limit Theorem and a Strong Mixing Condition," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 42, no. 1, pp. 43–47, jan 1956.

[25] M. Lombard, T. B. Ditton, and L. Weinstein, "Measuring Presence: The Temple Presence Inventory," in *Proceedings of the 12th Annual International Workshop on Presence*, 2009.

MODELING AND RENDERING FOR VIRTUAL DROPPING SOUND BASED ON PHYSICAL MODEL OF RIGID BODY

Sota Nishiguchi

Graduate School of Computer and Information Sciences
Hosei University
Tokyo, Japan
sota.nishiguchi.4b@stu.hosei.ac.jp

Katunobu Itou

Faculty of Computer and Information Sciences
Hosei University
Tokyo, Japan
katunobu.itou@hosei.ac.jp

ABSTRACT

Sound production by means of a physical model for falling objects, which is intended for audio synthesis of immersive contents, is described here. Our approach is a mathematical model to synthesize sound and audio for animation with rigid body simulation. To consider various conditions, a collision model of an object was introduced for vibration and propagation simulation. The generated sound was evaluated by comparing the model output with real sound using numerical criteria and psychoacoustic analysis. Experiments were performed for a variety of objects and floor surfaces, approximately 90% of which were similar to real scenarios. The usefulness of the physical model for audio synthesis in virtual reality was represented in terms of breadth and quality of sound.

1. INTRODUCTION

The sound in computer generated (CG) animation is created manually by the sound designer. Some sounds cannot be always realized, for example, the sound of a giant robot or a fictional weapon. Such sounds are prepared by processing or by using synthetic sounds that match the image in every single scene. Experience and the creative sense of the creator are important in the processing and synthesis of good sound. It is inconvenient to create a large number of sounds manually. In immersive content, it is necessary to create sounds to match the user's movement and the situation. However, it is difficult to always synchronize the sound with the video timing and impression. In this paper, the term "impression" indicates whether the sound matches the object and the phenomenon seen in the video. With a physical model, it is possible to generate a good sound from the physical information of the CG generation. In addition, irregular phenomenon, in which it is difficult to match the sound with the image, can be managed using synthesized sound. It can also be used for a fictional phenomenon. This study addresses the physical model for sound synthesis of the sound of a falling object, namely a huge sword, and the virtual dropping sound of the weapon with the aim of creating an automatic sound generation system corresponding to them.

There are two approaches to generate sounds of irregular movements such as dropping of objects. The first method is a statistical model that creates large quantities of transition models and attenuation models based on dropping sounds[1]-[4]. With this method, it is possible to generate dropping sounds that are very close to the real sounds and match the related images. In addition, by creating a model in advance, the generation process is completed in real time. Therefore, real-time sound generation is possible. However, it is not suitable for generating a virtual dropping sound because it is impossible to prepare a real thing. Another approach, which is a physical model, is a method for reproducing object vibration using physical simulation[5]-[11]. It is possible to generate dynamically natural sound based on rigid body simulation of CG animation and three-dimensional data in this method; it is also possible to generate virtual dropping sounds by appropriately setting the physical model and parameters.

The problem with the physical model is that the simulation cost is enormous and it takes time to generate the required sound. In the previous study[9], it was necessary to limit the frequency of vibration due to the cost of computation. As a result, high-frequency sound could not be generated and the sound had a boxy impression. Because it cannot handle huge objects due to high calculation cost, it can be said that it is insufficient as a synthesizer for virtual dropping sounds. In the above-mentioned research, the natural vibration mode is calculated by the finite element method (FEM) based on the three-dimensional data of the object, and the vibration of the object is precisely reproduced. The vibrations of all shapes such as complicated shapes and objects composed of a plurality of parts can be reproduced, whereas there are modes that do not need to be considered depending on the shape of the object; therefore, extra calculations are assumed to occur.

In this paper, we restrict the shape of the object to a bar, and generate a falling sound using a simple model that considers only sounds that can be heard. When limited to bars, we assume that the sonic vibration is only bending vibration. We simulate the vibration with a bending vibration model with reduced lattice points and dimensions when compared with the conventional method. Because the torsional vibration of the rod is smaller in amplitude than the bending vibration and the stretching vibration is an ultrasonic wave in the audible range or higher, it can be anticipated that the sound does not change considerably even when using a simple model with only bending vibration. We will construct a system that reduces the computational complexity while maintaining the quality of sound, and which can be used to generate the dropping sounds of huge objects.

2. PHYSICAL MODEL FOR DROPPING SOUND SYNTHESIS

2.1. Various dropping sounds

Various situations can be considered for the dropping phenomenon. The phenomenon also changes if the shapes and materials of the objects are different. In addition, the movement undergoes complex changes depending on the manner of dropping. Due to these factors, various dropping sounds having different tone pitch, volume, tone color, attenuation, and timing are generated. Particularly, it is difficult to control the movement at the time of fall; therefore, it is very difficult to retrofit or prepare in advance a

falling sound that matches the image. Figure 1 shows the spectrograms of sounds when the same object is dropped on different floor surfaces. The main physical phenomena that emit sounds are the collision between the floor surface and the object and the vibration of the object due to the collision.

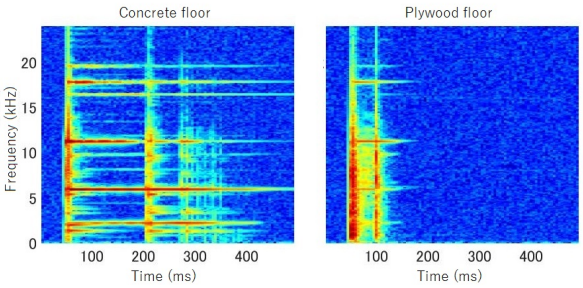


Figure 1: *Spectrograms of sounds of a falling aluminum rod. The rod is a round bar having a length of 15 cm and a diameter of 1 cm.*

Collision sounds are represented as pulse shock waves that vary depending on the physical properties of object, the floor surface, and the collision speed. A vibration sound is represented by a periodic function, which is the sum of a number of eigenmodes. There are three types of eigenmodes: bending vibration (transverse wave), stretching vibration (longitudinal wave), and torsional vibration. The intensity and attenuation of each eigenmode change depending on the position and intensity of the external force. These phenomena are repeated by the rebounding of the dropping object. Because speed and attitude of the object vary depending on each bounce, the collision sound and vibration sound also change accordingly. With recorded samples and general sound effects, it is difficult to reproduce dropping sounds that variously change depending on conditions. In a virtual space, objects are grasped and moved with high degree of freedom. Therefore, an acoustic generation engine without restrictions on movements is necessary.

2.2. Related researches

Sound source generation using physical equations and physical parameters has been studied for phenomena accompanying irregular movements such as the sound of fluids like water, the rubbing sound of clothes, and the sound of a flame [12]-[14]. There are also studies on the dropping sound of objects. Sound sources suitable for individual objects are generated by calculating the eigenmodes of falling objects by eigenvalue analysis[5]-[11].

As these studies use a three-dimensional vibration model, the amount of calculation is huge. Therefore, it is necessary to set the upper limit of the frequency, whereupon the generated sound becomes a muffled impression. Furthermore, it cannot handle huge objects that make the number of nodes extremely large. Three-dimensional vibration analysis can reproduce all eigenmodes bending, stretching, and torsional vibrations. However, depending on the shape of the object, there are eigenmodes whose frequencies are outside the audible range; therefore, it is necessary to select the vibration model that is most suitable for sound generation.

In addition, the initial condition of vibration is focused only on simple impulse excitation. Depending on the difference in the floor surface and the collision speed, the force applied to the object at the time of collision changes differently, and it is expected that it will also affect the subsequent vibration. To practically apply the physical model, it is necessary to consider a dropping sound generation model that can cope with a greater variety of phenomena and can ensure good sound quality.

3. PROPOSED METHOD

3.1. Overview of dropping sound generation system

The sound generation system consists of two processes. The first process is sound modeling, which generates sound sources using physical information and physical models. Another process is sound rendering, which generates an acoustic field based on the generated sound source and the spatial information. A desired sound is generated by the arrival sounds based on the sound field obtained through these processes and the observation point.

Physical information and spatial information are input to the system. In this system, the numerical values related to physical properties, shape, speed of the object, and the observation point are set. The sound generation is performed according to the phenomena in the image by quoting the shape and behavior data from the physical engine of the CG animation. Various parameters are assigned to the physical model of collision and vibration to generate the collision waveform and the vibration waveform. The vibration model is a bending model in which an object is regarded as a rod. The sound that reaches the ears is generated through sound rendering of these waveforms. In this paper, we synthesize the sound of a dropped object by solving the wave equation with the collision waveform and vibration waveform as the boundary conditions.

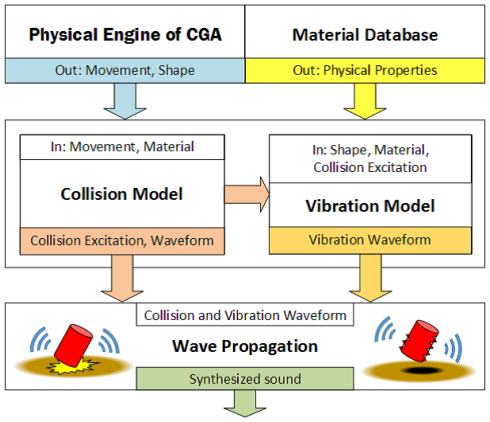


Figure 2: *Overview of a dropping sound generation system*

3.2. Bending vibration model

The Euler–Bernoulli beam is a model expressing the vibration of a bar only by the deflection deformation. Because of the bending deformation, the inside of the curve shrinks and the outside elongates in the axial direction, and the restoring force is created. For each material, the ratio of the restoring force to the deformation is given as the elastic modulus or Young's modulus E. The fundamental equation of the Euler–Bernoulli beam is obtained from

the elastic curve equation and the equation of motion. The elastic curve equation, which expresses the displacement when a bar receives an external force, is given as follows.

$$M = -EI\frac{\partial^2 w}{\partial x^2} \tag{1}$$

M is the bending moment. Let $w(x,t)$ be the displacement at position x and time t. Further, I is the moment of inertia of the area, ρ is the mass density of the object, and A is the cross-sectional area. The relationship between the bending moment M and the stress V applied to the object can be expressed by the following equation.

$$\frac{dM(x)}{dx} = V(x) \tag{2}$$

By applying this to the equation of motion, we obtain the fundamental equation of the Euler–Bernoulli beam. Considering the small section Δx of the bar, the external force is $F = \Delta x\frac{dV(x)}{dx} = \Delta x\frac{d^2M(x)}{dx^2}$ and weight is $\mu\Delta x$ (μ is the linear density of the bar); the acceleration can be expressed as d^2w/dt^2. Therefore, the equation of motion related to the bending of the bar is as follows.

$$\Delta x\frac{\partial^2 M(x)}{\partial x^2} = \mu\Delta x\frac{\partial^2 w}{\partial t^2} \tag{3}$$

By deleting Δx on both sides and substituting the expression (1) for M on the left side, the following equation is obtained.

$$\frac{\partial^2 w}{\partial t^2} + \frac{EI}{\mu}\frac{\partial^4 w}{\partial x^4} = 0 \tag{4}$$

The Euler–Bernoulli beam is a model that considers only the bending deformation of the object, so it is strictly different from the actual vibration. Therefore, when handling a short bar with a small slenderness ratio of the object, the frequency of vibration calculated by the above equation becomes higher than the actual value. This is an error, which occurs because the rotational inertia and the shear deformation of the entire bar are not considered. It is easier to rotate the entire object with the shorter bars, which increases the rotational inertia. Moreover, when the ratio of the cross section to the length increases in the short bar, the ratio of shear deformation to the bending deformation also increases and cannot be ignored. The Timoshenko beam is a model that considers these effects, and is therefore used in this paper.

The Timoshenko beam expresses the vibration of a bar by flexural deformation and shear deformation. Shear deformation is a displacement in the cross-sectional direction, and a restoring force against the displacement occurs. The elastic modulus is expressed by the rigidity rate G. Rotational inertia is applied to the Timoshenko beam to derive the basic equation.

$$\begin{aligned}\frac{\partial}{\partial x}\left[AG\kappa\left(\frac{\partial w}{\partial x} - \phi\right)\right] &= \rho A\frac{\partial^2 w}{\partial t^2}\\ AG\kappa\left(\frac{\partial w}{\partial x} - \phi\right) + EI\frac{\partial^2 \phi}{\partial x^2} &= \rho I\frac{\partial^2 \phi}{\partial t^2}\end{aligned} \tag{5}$$

Let $\phi(x,t)$ be the rotation angle of the section at position x and time t, and let the Timoshenko coefficient be κ.

Next, these partial differential equations are solved numerically by calculus of finite differences. The differential terms are substituted with the central difference of the second-order precision with respect to both temporal and spatial derivatives of the fundamental equation.

$$\begin{aligned}\frac{G\kappa}{\rho}\left(\frac{w_j^{i+1} - 2w_j^i + w_j^{i-1}}{\Delta x^2} - \frac{\phi_j^{i+1} - \phi_j^{i-1}}{2\Delta x}\right) &= \frac{w_{j+1}^i - 2w_j^i + w_{j-1}^i}{\Delta t^2}\\ \frac{AG\kappa}{\rho I}\left(\frac{w_j^{i+1} - w_j^{i-1}}{2\Delta x} - \phi_j^i\right) + \frac{E}{\rho}\frac{\phi_j^{i+1} - 2\phi_j^i + \phi_j^{i-1}}{\Delta x^2} &\\ = \frac{\phi_{j+1}^i - 2\phi_j^i + \phi_{j-1}^i}{\Delta t^2} &\end{aligned} \tag{6}$$

i is an index on space, and let w_j^i be the displacement at time step j. $\Delta x, \Delta t$ are discrete widths of space and time. In order to ensure the stability of the calculation, the set value of the discrete width needs to satisfy the following Courant–Friedrichs–Lewy(CFL) condition.

$$\begin{aligned}&|\alpha| < 1, \quad \alpha = \frac{G\kappa}{\rho}\left(\frac{\Delta t^2}{\Delta x^2} - \frac{\Delta t^2}{2\Delta x}\right)\\ &|\beta| < 1, \quad \beta = \frac{AG\kappa}{\rho I}\left(\frac{\Delta t^2}{2\Delta x} - \Delta t^2\right) + \frac{E}{\rho}\frac{\Delta t^2}{\Delta x^2}\end{aligned} \tag{7}$$

Because the propagation speed of the elastic vibration is different depending on the physical property data, the discrete width is set so that α, β become sufficiently small for any physical property.

The oscillation of the object is simulated by time evolution of the differentiated basic equation. The free edge boundary condition is used, as the object is free at the time of vibration after the collision. The free edge boundary condition in the Timoshenko beam model is given by the following equation.

$$\left.\frac{\partial \phi}{\partial x}\right|_{x=0,l} = 0, \quad \left[\frac{\partial w}{\partial x} - \phi\right]_{x=0,l} = 0 \tag{8}$$

The derivative terms in the boundary condition are also substituted with the difference formulas.

$$\begin{aligned}\frac{\phi_j^2 - \phi_j^0}{2\Delta x} = 0, \quad \frac{w_j^2 - w_j^0}{2\Delta x} - \phi_j^1 = 0\\ \frac{\phi_j^N - \phi_j^{N-2}}{2\Delta x} = 0, \quad \frac{w_j^N - w_j^{N-2}}{2\Delta x} - \phi_j^{N-1} = 0\end{aligned} \tag{9}$$

N is the end of the spatial index, and the elements of the indices 0 and N are dummy elements for the free boundary condition. For each calculation process of the explicit method, we update the displacement and rotation angle at both ends of the object using these boundary conditions.

Next, the vibration attenuation model will be explained. Rayleigh damping is used in this system. Rayleigh attenuation is a model that takes into consideration two types of attenuation: external damping (viscous damping) and internal damping (viscoelastic damping of the object). The following equation is obtained by adding the attenuation term to the Euler-Bernoulli beam in the expression .

$$\frac{\partial^2 w}{\partial t^2} + \frac{EI}{\mu}\frac{\partial^4 w}{\partial x^4} + \eta\frac{\partial}{\partial t}\left(\frac{EI}{\mu}\frac{\partial^4 w}{\partial x^4}\right) - \gamma\rho A\frac{\partial w}{\partial t} = 0 \tag{10}$$

γ and η are coefficients for external attenuation and internal attenuation. As approximate values of these attenuation coefficients can be defined for each material, a database of attenuation

coefficients is prepared along with physical property values such as density and Young's modulus.

Similarly, considering the attenuation in the Timoshenko beam, the fundamental equation becomes as follows.

$$\begin{aligned} &\frac{\partial}{\partial x}\left[AG\kappa\left(1+\eta\frac{\partial}{\partial t}\right)\left(\frac{\partial w}{\partial x}-\phi\right)\right]-\gamma\rho A\frac{\partial w}{\partial t}=\rho A\frac{\partial^2 w}{\partial t^2} \\ &AG\kappa\left(1+\eta\frac{\partial}{\partial t}\right)\left(\frac{\partial w}{\partial x}-\phi\right)+EI\left(1+\eta\frac{\partial}{\partial t}\right)\frac{\partial^2\phi}{\partial x^2}=\rho I\frac{\partial^2\phi}{\partial t^2} \end{aligned} \tag{11}$$

The relationship between γ and η and the loss factor ξ is expressed by the following equation.

$$\xi=\omega\gamma+\frac{\eta}{\omega} \tag{12}$$

ω is the angular frequency of vibration and the loss factor has frequency characteristics. The external attenuation shows a characteristic that is inversely proportional to the frequency, and the internal attenuation has a characteristic proportional to the frequency. In this study, these attenuation coefficients are cited from [10].

3.3. Collision model

Impact noise is the change in air pressure exerted by the deformation of the object occurring during the action of the impact force. The literature [16] is a study targeting collision sounds of steel balls. A steel ball with a diameter of 5 cm is used in the experiment, but in this case, the fundamental mode of the vibration sound is beyond the audible range and analysis is performed by considering only the impact sound. According to the literature [16], the impulsive sound is a pulse-like waveform, and the peak sound pressure of the pulse is proportional to the acceleration and volume of the object and is inversely proportional to the distance.

$$p(x,y,z;t)=\frac{\rho a^2}{4R}\frac{\partial}{\partial t}\left\{U\left(x',y',z';t-\frac{R}{c}\right)\right\} \tag{13}$$

p is the sound pressure at time t at the observation point (x,y,z), and U represents the velocity at the point (x',y',z') of the sound source. In addition, let a be the radius of the sphere, R be the distance between the sound source and the observation point, ρ be the mean density of air, and c be the sound velocity.

The object vibrates due to the collision. Therefore, the force applied to the object by the impact is calculated and used as the boundary condition of the vibration model. The force applied to the object at the time of collision can be predicted from Hertz's solid contact theory. The deformation $d(t)$ due to collision when the collision surface is spherical is obtained by the following formula.

$$d(t)=F(t)^{2/3}\left(\frac{C^2}{R}\right)^{1/3} \tag{14}$$

$F(t)$ is the force working at time t and R is the radius of the object. C is defined as follows.

$$C=\frac{3}{4}\left(\frac{1-{\nu_0}^2}{E_0}+\frac{1-{\nu_1}^2}{E_1}\right) \tag{15}$$

Let E_0 be the Young's modulus of the floor and nu_0 be Poisson's ratio of the floor. The constant with subscript 1 is the corresponding value of the object. The contact time τ can also be calculated from these parameters.

$$\begin{aligned} \tau&=\frac{4\sqrt{\pi}\Gamma(2/5)}{5\Gamma(9/10)}\left(\frac{m_r^2}{g^2 v_i}\right)^{1/5} \\ g&=\frac{4}{5C}\sqrt{R},\quad m_r=\frac{m_0 m_1}{m_0+m_1} \end{aligned} \tag{16}$$

m_r is the relative mass of the object and the floor surface, m_0 is the mass of the floor, and m_1 is the mass of the object. The following equation is obtained from the relationship between the momentum of the falling object $p=m_1 v_i$ and the excitation force $F(t)$.

$$F_{\mathbf{ave}}=\frac{2m_1 v_i}{\tau} \tag{17}$$

Using this $F_{\mathbf{ave}}$, the time waveform of the collision excitation force is modeled as a sine wave.

$$F(t)=F_{\mathbf{ave}}\left(1-\cos\left(\frac{2\pi t}{\tau}\right)\right) \tag{18}$$

The vibration corresponding to the change of collision can be generated by giving the time waveform of the external force as the boundary condition for vibration simulation. Even when the object has a shape other than a sphere, the external force waveforms can be calculated using appropriate models [17]. The pulse sound of the collision itself is reproduced by giving the time history $d(t)$ of deformation due to collision as the boundary condition of the sound propagation simulation.

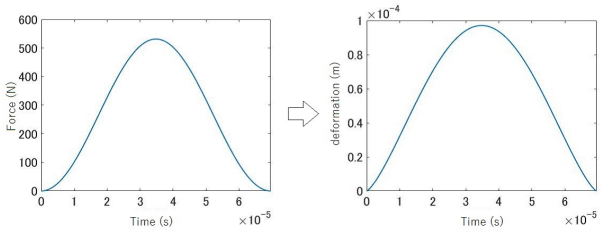


Figure 3: *Force history $F(t)$ and coordinate history $d(t)$*

3.4. Dropping rigid body simulation

The vibration model and the collision excitation model are applied to the dropping of the object. The collision speed can be obtained with $\sqrt{2gh}$ using the object's height h and the gravitational acceleration g. Next, using the coefficient of restitution of the floor and the object, the height after reflection is obtained and the speed at the time of re-impact is calculated. By repeating this, it is possible to obtain the vibration condition for the object bouncing from the floor surface. However, the coefficient of restitution is a numerical value involving complicated physical properties and shapes, and it is difficult to prepare a physical property such as density or Young's modulus for each material. Moreover, at the time of falling, the object rotates about its center of gravity; therefore, repulsion breaking processing is required.

In the generation of CG animation, the movements of a rigid body are reproduced by performing arithmetic based on the physical law so as to express falling and collision with more realistic behaviors. Bullet is a free physics engine, which is used in many 3DCG creation software (Maya, Blender, etc.). In order to obtain a sound consistent with the image, it is appropriate to use rigid body simulation information contained in the image. In this paper, we use rigid body simulation with Bullet. In Bullet, it is possible to set the coefficient of restitution and the friction coefficient for each object. Using these values as parameters for sound generation, we obtain the collision timing, velocity, and shape of the colliding surface from the rigid body simulation in Bullet.

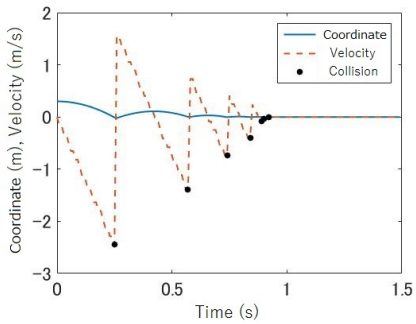


Figure 4: *Coordinate, velocity, and collision timing of an object obtained from Bullet*

3.5. Sound rendering

Acoustic processing based on the positional relationship between the sound source and the observation point is required to obtain the result of the vibration simulation as a sound. In the previous study [9], the amplitude of each mode was calculated from the positional relationship between the sound source and the observation point by FFAT(Far-Field Acoustic Transfer) map. An FFAT map is a model of the sound field around the object based on the Helmholtz equation, which is obtained by calculating the phase and amplitude for each mode by eigenmode analysis.

$$\nabla^2 p(\mathbf{x}) + k^2 p(\mathbf{x}) = 0 \tag{19}$$

$p(\mathbf{x})$ is the sound pressure at position $\mathbf{x}$, and the wavenumber is $k = \omega/c$ (sound speed c, each frequency ω).

With this differential equation, the sound field is estimated by assigning the displacement of the object surface obtained from sound modeling as the Neumann boundary condition. A sound field is created for each vibration mode, and the sound fields of all modes are synthesized to generate the sound field of the vibration sound.

In this research, sound wave propagation is reproduced by directly solving the wave equation, which is the basis of the Helmholtz equation in the FDTD method. The wave equation is given by the following equation.

$$\nabla^2 p(\mathbf{x}, t) + \frac{1}{c^2}\frac{\partial^2 p(\mathbf{x}, t)}{\partial t^2} = 0 \tag{20}$$

The boundary condition based on the vibration of the object is as follows.

$$\frac{\partial p(\mathbf{x}, t)}{\partial n} = -\rho_m a_n(\mathbf{x}, t), \quad \mathbf{x} \in \Omega \tag{21}$$

Let $\partial/\partial n$ be the normal derivative of the object surface Ω. ρ_m is the density of the medium. Air is usually the medium, and air vibration on the object surface can be obtained by using $rho_m = 1.2041 kg/m^3$. a_n is the acceleration of the surface of the object and can be obtained by differentiating the second-order displacement of the object surface with respect to time. The boundary conditions given as differential equations are differentiated, and conditional expressions for the sound pressure at the boundary between the object and the medium are obtained. We simulate sound propagation using this condition and the wave equation (Fig. 5).

The recording environment is reproduced for the evaluation of the generated sound. Because the recording environment is an anechoic room, a perfectly matched absorption boundary layer is set around the calculation area so that reflection of waves from the wall do not occur [18].

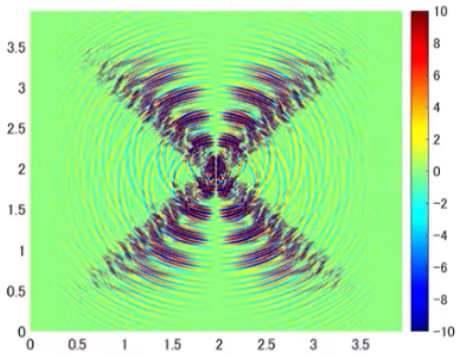


Figure 5: *The generated spatial acoustic field*

4. EVALUATION

In order to evaluate the bending vibration model, we simulated the vibration of the object and compared it with the recorded vibration sound. Next, dropping sounds were generated under various conditions, and spectrograms were compared with the actual sound sources. The shape and physical property parameters of the targeted object were set as shown in Tables 1 and 2.

4.1. Evaluation of vibration sound

For object A in Table 2, vibration was simulated with shock applied to the end of the rod as the initial condition. The air vibration was simulated at a position 5 cm away from the end of the rod in the direction perpendicular to the axis. It was found that although there was a numerical error from the actual sound, a sound with no incongruity in terms of the sound impression and height was generated. Figure 6 compares the spectrums of the actual sound and the generated sound. It was understood that the natural frequencies were roughly coincident, and the amplitude of the mode component was almost faithfully reproduced. For eigenmodes above 15,000 Hz, the actual sound and intensity were different and could not be reproduced satisfactorily, but because the frequency was

Table 1: *Physical property parameters of the material*

Material	Density (kg/m^2)	Young's modulus (GPa)	Rigidity ratio (GPa)	Poisson ratio	Internal damping	External damping
Aluminum	2698.9	70.3	26.1	0.345	3E-8	5
Brass	8411	100.6	37.3	0.35	3E-8	5
Iron	7874	211.4	81.6	0.293	4E-8	0.1
Wood	800	11	4.23	0.3	2E-6	60

Table 2: *Shape parameters*

Object name	Material	Length(m)	Width(m)	Thickness(m)	Cross section shape
Object A	aluminum	0.15	0.01	0.01	circle
Object B	aluminum	0.15	0.01	0.01	rectangle
Object C	brass	0.15	0.01	0.01	rectangle
Object D	iron	0.30	0.008	0.008	circle
Object E	wood	0.15	0.005	0.005	rectangle
Object F	iron	10	0.3	0.07	rectangle

close to the upper limit of the audible range, it was considered that there was no great influence on the sound impression; therefore, it was not evaluated this time.

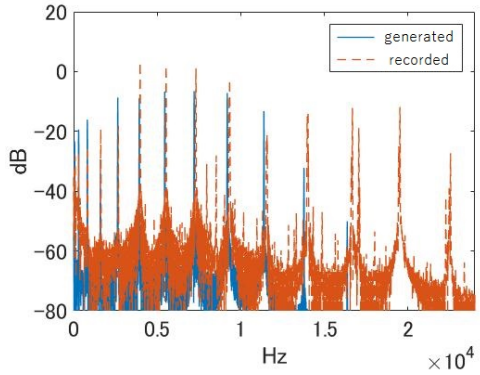


Figure 6: *Vibration sound spectrum of object A. Comparison of synthesized and recorded sounds.*

4.2. Evaluation of dropping sound

The sound of the falling object was evaluated for the objects B to E. We set various parameters and generated the dropping sounds. The object B had a metallic lightweight sound impression. The object C was heavier and softer than B, and the object D had a hard metallic sound. Natural dropping sounds were generated for aluminum, brass, and iron bars. Compared with the actual dropping sound, the height of the vibration sound of each object could almost be reproduced, and the vibration corresponding to the difference in material and shape could be generated. Furthermore, the object E which was a wooden rod, had a fast decaying dry sound, which was close to the impression of the actual sound. By defining the damping coefficient for each material, a natural falling sound was obtained even for an object made of material with great difference in hardness and weight.

Table 3: *Comparison of natural frequencies.*

Mode No.	Recorded	Generated	Relative error
1	303 (Hz)	304 (Hz)	0.33%
2	831	823	0.96%
3	1627	1606	1.29%
4	2675	2643	1.20%
5	3979	3928	1.28%
6	5532	5454	1.41%
7	7319	7211	1.48%
8	9330	9191	1.49%
9	11,568	11384	1.59%
10	14,006	13,779	1.62%
Average	—	—	1.26%

The objective evaluation of each generated sound is as follows. The recall ratio of the eigenmode is obtained by dividing the matching modes of the recorded sound and generated sound by the number of all modes. The relative error of the frequency is used as the condition for the match. A mode in which the relative error was less than 6% was regarded as the matching mode. The relative error of 6% was approximately the same as the chromatic scale, which was the minimum unit of the pitch.

We also generated the virtual falling sound of a huge sword using the system. The parameters correspond to object F in Table 2. The shape (the length, width and thickness) of a general Japanese sword was measured as a square bar and the value was magnified by 10. An image was created using the same physical parameters. A heavy metal sound was generated according to the movement of the drop, and a virtual falling sound with an impression suitable for the image was obtained.

Table 4: *Evaluation data*

Object	Generated mode	Recall ratio	Frequency relative error	Average power error
Object B	4/6	66.7% (4/6)	3.43%	5.65 dB
Object C	3/3	100% (3/3)	1.52%	17.12 dB
Object D	11/13	69.2% (9/13)	2.66%	12.15 dB
Object E	5/8	62.5% (5/8)	4.11%	8.16 dB

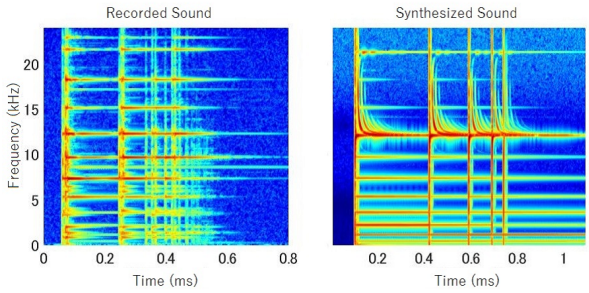


Figure 7: *Dropping sound spectrogram of object D. Comparison of recorded and synthesized sounds.*

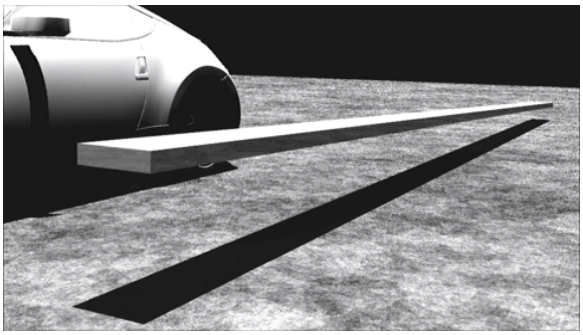


Figure 8: *A huge falling sword simulated by Ballet*

4.3. Discussion

For bars with a circular cross section, sufficient vibration sound reproduction is possible even with models with only flexural vibration. This is because the torsional vibration does not generate sound waves in a circular cross section. However, for bars with rectangular cross section, a strong torsional vibration mode occurred depending on the collision position. In the model with only bending vibration, a monotonous impression sound was generated rather than the actual vibration sound due to the lack of torsional vibration mode.

It is presumed that the power error is due to the mismatch between the attenuation and error of the excitation condition. Vibration suitable for the sound made by bars of various metals and wood was obtained by setting the damping coefficient for each material. However, because we did not consider energy absorption from the object to the floor, we could not sufficiently reproduce the drop to the soft floor surface. The floor, which was thin and easily vibrated, was not reproduced because of the same reason. We believe that the exchange of energy can be applied to the generation of the sound that causes the floor surface to vibrate, reproducing the rapid attenuation by the contact with the floor after the end of the bouncing phase.

By using the collision excitation force waveform for the vibration sound, the higher order mode of the vibration sound gradually weakened each time it bounced back the characteristics of the dropping sound. Moreover, by considering the collision sound caused by the collision deformation, strong feeling of attack on the falling sound was born. However, the spectrum did not change much in any of the generated collision sounds. In this system, the point-to-point collision model was applied to all collisions. The presumed reason is that it was not possible to reproduce the collision from line to point, line to line, and more. In addition, a strong collision generates a shock wave. There may also be a phenomenon wherein the falling object could not fully cope with only by atmospheric pressure change due to deformation and sound propagation.

The excessive attenuation of the component of 15,000 Hz or more of the generated sound is caused by the numerical dispersion. Numerical dispersion is the dispersion occurring due to change in the phase velocity depending on the wave number in the numerical solution. Actually, the phase velocity is constant irrespective of the wave number. As the wave number increases, the numerical dispersion increases. We consider that the components above 15,000 Hz could not be properly simulated with the mesh width used in this simulation. To suppress the numerical dispersion, it is necessary to set a discrete scheme where the CFL condition is sufficiently satisfied.

The simulation of sound wave propagation in a two-dimensional space with nothing around the object was performed for simplicity, but faithful sound generation is possible by further propagating sound waves in three dimensions considering shields and other objects. However, the amount of computation required is proportional to the power of the number of dimensions. A simulation can require extensive computations in three dimensions. In this paper, we adopted FDTD for solving the original partial differential equation for both vibration and propagation directly to clarify the relationship between the models. With regard to the amount of computation, it is considered that it is effective to use a radiation model [19] to simulate the propagation.

It is expected that sounds can be improved by implementing the attenuation due to absorption at the contact points and defining the attenuation rate of the vibration model by physical property parameters. Although we have implemented only the flexural vibration model this time, it is necessary to consider a framework to apply the optimal vibration model as compared with the model considering torsional vibration and stretching vibration.

5. CONCLUSION

We have proposed a simplified model of vibration in this study. As a result, it was found that the round bars were almost reproducible only by the bending vibration model. We could reduce the computation for round bars while maintaining the quality of the generated sound. A virtual falling sound of a huge object was generated from the physical model, and an appropriate sound was obtained. By considering sound waves caused by collision deformation, an impact effect was imparted to the dropping sound, leading to a more natural dropping sound generation.

It is necessary to conduct experiments with various parameters to confirm the versatility of the created system. During the evaluation, the generated sound was played along with the image, and the subjective evaluation of the degree of coincidence with the image and the sound quality were important criteria in the evaluation. To realize realistic sound generation, it is important to solve the problems inside the system such as expansion of the vibration model and collision model, use of shock wave propagation simulation, study of the discrete width of simulation, and the like.

6. REFERENCES

[1] Langlois, Timothy R., and Doug L. James. "Inverse-foley animation: Synchronizing rigid-body motions to sound." ACM Transactions on Graphics (TOG) 33.4 (2014): 41.

[2] Zheng, Changxi, and Doug L. James. "Rigid-body fracture sound with precomputed soundbanks." ACM Transactions on Graphics (TOG). Vol. 29. No. 4. ACM, 2010.

[3] Chadwick, Jeffrey N., Changxi Zheng, and Doug L. James. "Faster acceleration noise for multibody animations using precomputed soundbanks." Proceedings of the ACM SIGGRAPH/Eurographics Symposium on Computer Animation. Eurographics Association, 2012.

[4] Ren, Zhimin, Hengchin Yeh, and Ming C. Lin. "Synthesizing contact sounds between textured models." Virtual Reality Conference (VR), 2010 IEEE. IEEE, 2010.

[5] James F. O'Brien, Perry R. Cook, Georg Essl,"Synthesizing Sounds from Physically Based Motion," ACM SIGGRAPH, 2001.

[6] Van Den Doel, Kees, Paul G. Kry, and Dinesh K. Pai. "FoleyAutomatic: physically-based sound effects for interactive simulation and animation." Proceedings of the 28th annual conference on Computer graphics and interactive techniques. ACM, 2001.

[7] O'Brien, James F., Chen Shen, and Christine M. Gatchalian. "Synthesizing sounds from rigid-body simulations." Proceedings of the 2002 ACM SIGGRAPH/Eurographics symposium on Computer animation. ACM, 2002.

[8] Bonneel, Nicolas, et al. "Fast modal sounds with scalable frequency-domain synthesis." ACM Transactions on Graphics (TOG) 27.3 (2008): 24.

[9] Chadwick, Jeffrey N., Steven S. An, and Doug L. James. "Harmonic shells: a practical nonlinear sound model for near-rigid thin shells." ACM Trans. Graph. 28.5 (2009): 119-1.

[10] Zheng, Changxi, and Doug L. James. "Toward high-quality modal contact sound." ACM Transactions on Graphics (TOG). Vol. 30. No. 4. ACM, 2011.

[11] Langlois, Timothy R., et al. "Eigenmode compression for modal sound models." ACM Transactions on Graphics (TOG) 33.4 (2014): 40.

[12] Doel, Kees van den. "Physically based models for liquid sounds." ACM Transactions on Applied Perception (TAP) 2.4 (2005): 534-546.

[13] Langlois, Timothy R., Changxi Zheng, and Doug L. James. "Toward animating water with complex acoustic bubbles." ACM Transactions on Graphics (TOG) 35.4 (2016): 95.

[14] An, Steven S., Doug L. James, and Steve Marschner. "Motion-driven concatenative synthesis of cloth sounds." ACM Transactions on Graphics (TOG) 31.4 (2012): 102.

[15] Hideo Tsuru, "Numerical analysis of vibration of xylophone by Finite Difference Method," Acoustical Society of Japan, Vol. 67, no. 7, pp. 296-301 , 2011.

[16] Genrokuro Nishimura, Koichi Takahashi, "Impact Sound of Steel Ball Collision," Journal of the Society for Precision Mechanics of Japan, Vol. 28, no. 4, pp. 220-230 , 1962.

[17] S.P.Timoshenko and J.N.Goodier, "Theory of Elasticity THIRD EDITION," McGraw-Hill Book Company Inc., pp. 423-436, 1970.

[18] Yuan, Xiaojuen, et al. "Simulation of acoustic wave propagation in dispersive media with relaxation losses by using FDTD method with PML absorbing boundary condition." IEEE transactions on ultrasonics, ferroelectrics, and frequency control 46.1 (1999): 14-23.

[19] Eston Schweickart, Doug L. James and Steve Marschner, "Animating elastic rods with sound," ACM Transactions on Graphics 36(4):1-10, 2017.

OBJECTIVE EVALUATIONS OF SYNTHESISED ENVIRONMENTAL SOUNDS

David Moffat *

Centre for Digital Music,
Queen Mary University of London
London, UK
d.j.moffat@qmul.ac.uk

Joshua D. Reiss

Centre for Digital Music,
Queen Mary University of London
London, UK
joshua.reiss@qmul.ac.uk

ABSTRACT

There are a range of different methods for comparing or measuring the similarity between environmental sound effects. These methods can be used as objective evaluation techniques, to evaluate the effectiveness of a sound synthesis method by assessing the similarity between synthesised sounds and recorded samples. We propose to evaluate a number of different synthesis objective evaluation metrics, by using the different distance metrics as fitness functions within a resynthesis algorithm. A recorded sample is used as a target sound, and the resynthesis is intended to produce a set of synthesis parameters that will synthesise a sound as close to the recorded sample as possible, within the restrictions of the synthesis model. The recorded samples are excerpts of selections from a sound effects library, and the results are evaluated through a subjective listening test. Results show that one of the objective function performs significantly worse than several others. Only one method had a significant and strong correlation between the user perceptual distance and the objective distance. A recommendation of an objective evaluation function for measuring similarity between synthesised environmental sounds is made.

1. INTRODUCTION

The field of sound synthesis has seen significant work in a range of areas including effective and efficient replication of existing sounds or creation of new sounds. Sound synthesis evaluation can take many different forms. Ten different evaluation criteria for evaluation of synthesis techniques were presented by [1], in which half of the criteria are based on control and parameterisation, and only two evaluation criteria relate to the sonic properties of the synthesis. One of the key aims of sound synthesis is to produce a realistic sound, with the added ability to control or interact with the sound [2, 3]. Despite this, there is limited evaluation of sound synthesis systems and their ability to produce realistic convincing sounds [4, 5].

This paper proposes a comparison of sound similarity measures, through resynthesis. The aim is to identify an objective measure that can encapsulate the perceptual similarity of sounds. Optimization of this measure would then select appropriate parameters for a synthesis engine to match a given sound, Optimisation of synthesis parameters to evaluation of sound perception has been previously demonstrated [6]. Parameter selection can be viewed as an optimisation problem in which synthesis parameters are dimensions through a fitness landscape. In many cases, we are searching through highly nonlinear search spaces, and thus evolutionary optimisation functions are effective methods to use [7, 8, 9].

* This paper is supported by EPSRC Grants EP/L019981/1 and EP/M506394/1.

Table 1: Range of Objective Evaluation Metrics used in Current sound synthesis Research

Research	Objective Evaluation Methods
[11]	Fundamental Frequency
	Spectral Centroid
	First 4 Harmonics
	Zero Crossing Rate
[12]	Spectrogram
[13]	Spectrogram
	Num and Position of Harmonics
[14]	Spectrogram
	Magnitude Spectrum
[15]	Magnitude Spectrum
[16]	MFCC vector correlation
[17]	Spectrogram envelope
[18]	Error between STFT bins
[19]	PEAQ
[20]	Least Square Error (LSE) in FD
	Simultaneous Frequency Masking (SFM)
[21]	DCT of MFCC
	Spectral Shape
	Attack and Decay Characteristics
	Duration

Section 2 will present background literature and motivate the requirement for a generalisable objective measure for synthesised sounds. The objective metrics and evaluation framework will be presented in Section 3. The subjective listening test is presented in Section 4. Results of the subjective and objective measures are given in Section 5. Recommendations for synthesis evaluation metrics are presented in Section 6, and final comments and outline of impact in the community are presented in Section 7.

2. BACKGROUND

The research aims of sound synthesis are to produce realistic and controllable systems for artificially replicating real world sounds. Current research generally focuses on either implementation efficiency, interfacing control or physical modelling, and provides very limited evaluation. There is little or no research on comparison of existing synthesis techniques [5]. Subjective evaluation is occasionally used in current sound synthesis research [4, 10], however objective evaluation is rarely used and there is no consistency in metrics that are used. A summary of sound synthesis papers that use objective evaluation is presented in Table 1. The variety of different objective measures and methods used within Table 1, shows that there is a lack of inconsistency in method for objective

evaluation.

[2] and [22] both evaluated work based on its interactivity, which often measures the parameter mapping more than the quality of the sound synthesis. Within [23], comparison of two similarity measures was performed, the MFCC distance and an audio feature vector distance. The results were evaluated with a subjective listening test. [24] objectively compares different wavetable synthesis methods using "Relative Spectral Error", with no comparison to samples or perceptual evaluation. [18] also calculated the error of bins from the Short Time Fourier Transform (STFT), between the reference and the synthesised sample. [25] utilised sound texture statistics for resynthesis work by [6] by enforcing a set of statistics on an STFT representation of an audio signal.

[21] evaluated synthesis parameter selection using a range of low level audio features, such as Fundamental Frequency, Spectral Shape, Envelope Characteristics, and Overall Duration. [21] used the DCT of the MFCCs as a sound similarity measure, to determine how similar the synthesised sound was to a recorded sample. Similarly, [16] performed correlations between MFCC vectors within adjacent frames, as a similarity measure for audio textures. [11] compared a synthesis method to recorded samples, through visual comparison of spectrograms, and comparison of some low level audio features, such as fundamental and first 4 harmonic frequencies, spectral centroid and zero crossing rate. No comparison with other synthesis methods was undertaken and no perceptual evaluation. In contrast, [26, 27] builds a physically inspired model where the physical properties measured vs. estimated are compared. The output time domain and spectrogram signals are compared visually, including locations of fundamental and harmonics. [17] used the loudness curve weighted Equivalent Rectangular Bands (ERB) envelope to perform grain selection within a granular synthesis approach. [19] attempted to evaluate the perceptual similarity of a piano note synthesis method with a sample using PEAQ, an algorithm designed for determining the quality of audio compression codecs which analyses the sound on a sample by sample basis to determine any perceptual artifacts. Where perception was considered, the notes will never be exactly the same if played with slightly different attack or at a different sample time, thus resulting in a perceptual difference where none exists.

There have been a number of approaches to searching audio parameter spaces, within a synthesised environment. An iterative process to control parameters and minimise a set of perceptually motivated audio features was developed by [6, 28]. The results were subjectively evaluated based on participants identification and synthesis realism. Further approaches using genetic algorithms have attempted to modify musical parameters based on varying fitness functions. No other method performed any formal evaluation of the synthesis results, typically reporting their final distance measure. Fitness function methods are typically calculated as distances features such as between Mel Frequency Cepstrum Coefficients (MFCCs) [9], the Discrete Cosine Transform of the MFCCs [21]. The Perceptual Evaluation of Audio Quality (PEAQ [29]) distances were measured for piano string synthesis [19], where as the distance between Least Square Error(LSE) of time domain waveform, LSE of spectrograms and LSE of spectrograms with some masking weighting were all used as distance measures [7]. [8] used sets of different audio features to measure distances.

3. OBJECTIVE MEASURE THROUGH SYNTHESIS

In this section, the methodology of evaluating a range of objective measures will be presented. The principle is that evaluation of different objective measures can be compared through resynthesis. By using the objective measure as fitness function in an iterative synthesis process, we can identify which measure best encapsulates aspects of the perception of the sounds. Every synthesised sound will be produced with the intention of sounding as close to a recorded sample as possible, and if an objective measure is able to produce this sound, then the objective measure represents the perceptual similarity of the sounds.

3.1. Sound Synthesis Methods

Four different sound effects were used for evaluation purposes. All of them are available and hosted online as part of the FXive synthesis platform [30, 31]. All synthesis methods were originally derived from [32] and are all examples of physically inspired synthesis methods, as they are commonly available open source implementations of synthesis methods.

Fire The fire synthesis model is a noise shaping synthesis method. Individual sonic components of a fire, the hiss, crackle and lapping, are all modelled though filtered and envelope shaped noise signals. Three control parameters are exposed to the user, which are *lapping*, *hissing* and *crackling*.

Rain In the rain model, components of rain are broken into a number of categories. Ambience, which is modelled as constant shaped noise, droplets, rumble and drips. Three control parameters are exposed to the user, which are *density*, *rumble* and *ambience*.

Stream The stream is modelled entirely on the bubbling sounds that are made as water runs over substances, based on control of filtered chirp sounds. Three control parameters are exposed to the user, which are *bubbles*, *frequency* and *filter Q*.

Wind The wind model uses a varying filtered noise approach, where wind parameters control the overall envelope of the sound. Different wind hitting materials, such as door or branches/wires, select the timesteps over which the wind envelope shaping will occur. Ten parameters are exposed to the user: *Wind Speed*, *Gustiness*, *Squall*, *Buildings*, *Doorways*, *Branches*, *Leaves*, *Pan*, *Directionality* and *Gain*. The parameters *Pan*, *Directionality* and *Gain* were all left constant at their default values, as discussed in Section 3.1.

Parameters Not Changed Several parameters were not used, to limit the search space and as these parameters were considered to make no immediate impact to the synthesis of the sound. During analysis, all samples were loudness normalised, so output gain controls were redundant. As no evaluation metric used spatial aspects to evaluate synthesis, pan controls were also not considered. With each sound effect, there was the ability to apply a range of audio effects, including equalisation, distortion, delay, convolution reverb and HRTF spatialisation. However, because all of these controls can be added to every single synthesised sample, we felt this would significantly grow the search space without significant improvements in the synthesis. The impact of individual audio effects on the perceived realism of a synthesised sound is out of the scope of this work.

3.2. Parameter Optimisation

The parameters of each synthesis model were optimised using particle swarm optimisation. Particle swarm optimisation is an evolutionary inspired population based optimisation technique in which a swarm of particles iteratively propagate in a search space, where a weighting between individual and global preferences are modelled. Each particle is evaluated with a fitness function, and we use this fitness function to compare each of our objective functions presented in Section 3.3. Particle swarm is an effective optimisation method for highly nonlinear search spaces, and there are many examples of evolutionary algorithms applied to audio research [7, 8, 9, 33, 34]. A comprehensive overview of particle swarm optimisation is presented in [35].

3.3. Objective Function

The fitness functions were taken from literature, and their features used for evaluation are described in Table 2. To standardise implementations, all audio features were extracted using Essentia [36, 37].

Table 2: Attributes of Each Objective Function

Objective Function	Features and Attributes
Allamanche [38]	Loudness
	Spectral Flatness
	Spectral Crest Factor
Gygi [39]	Envelope Statistics
	Pitch
	Autocorrelation Waveform Peaks
	Spectral Centroid
	Spectral Moments
	Frequency Band Energy
	Modulation Statistics
	Subband Correlation
	Spectral Flux
MFCC [9]	MFCC
Moffat [40]	Loudness
	Pitch
	MFCC
	Envelope Statistics
	Spectral Contrast
	Spectral Flux
PEAQ [29]	Signal Bandwidth
	Masking Content
	Modulation Difference
	Distortion
	Harmonic Structure
Wichern [41]	Loudness
	Spectral Centroid
	Spectral Sparsity
	Harmonicity
	Temporal Sparsity
	Transient Index (ΔMFCC)

The MFCC's as a similarity was motivated as an anchor within the experiment, as we expected this method to underperform in comparison to other objective functions.

4. SYNTHESIS EVALUATION - LISTENING TEST

4.1. Participants

19 participants took part in the experiment, of which 12 were male and 7 female. The average age 29 and standard deviation of 3. The average test duration was 23 minutes, so fatigue was not an issue. The procedure was approved by the local ethics committee.

4.2. Experimental Setup

The experiment was set up as listening test, performed in Queen Mary Studio [42], and participants auditioned sounds over a pair of high quality calibrated PMC speakers. Participant were asked to adjust the volume of the audio to a comfortable level at the beginning of the test and refrain from adjusting it. All volume adjustments were recorded during the test. The listening test was set up using the Web Audio Evaluation Tool [43]. The listening test is available[1] with the same user interface and set of samples that were used by participants.

4.3. Materials

Participants were asked to evaluate sound samples for four categories (fire, rain, stream and wind). In each category six synthesised samples were provided and compared to a recorded sample reference. All samples were 48kHz wav files, and loudness normalised in accordance with [44]. Each category had one anchor, where random parameter values were used to generate a sample. The reference samples were all selected from a professionally available sound effects library[2].

The anchors were included to encourage participants to use the entire evaluation scale, and we could review how samples were distributed within that scale, in accordance with [45]. The anchor ensures that there is a lower limit sample to compare against. It also performs as a confirmation that a participant has fully understood the requirements for the experiment. If a participant rated the anchor as higher than the sample, then we would infer that the participant may not have fully understood the requirements, or may have some hearing defect.

4.4. Procedure

Participants were provided with instructions as to the experiment they were to undertake, and were asked to provide their native spoken language, whether they had previous experience of listening tests and whether they would consider themselves as accomplished musicians or audio engineers.

Participants were then asked to rate how similar they perceived a set of given samples to a provided reference. Participants were provided with a continuous linear scale on which to rate all sounds, labeled from "most similar" to "very different". All sounds were rated on a single horizontal scale, to encourage inter-sample comparison. Participants did not have any information regarding the samples, other than that they were all synthesised and the names of the four sound classes used in the experiment. Samples started off at a randomised position on the scale. Both the ordering of categories and the initial ordering of samples within a category were randomised, to remove bias effects.

[1] http://goo.gl/fusJv3

[2] https://www.prosoundeffects.com/hybrid-library/

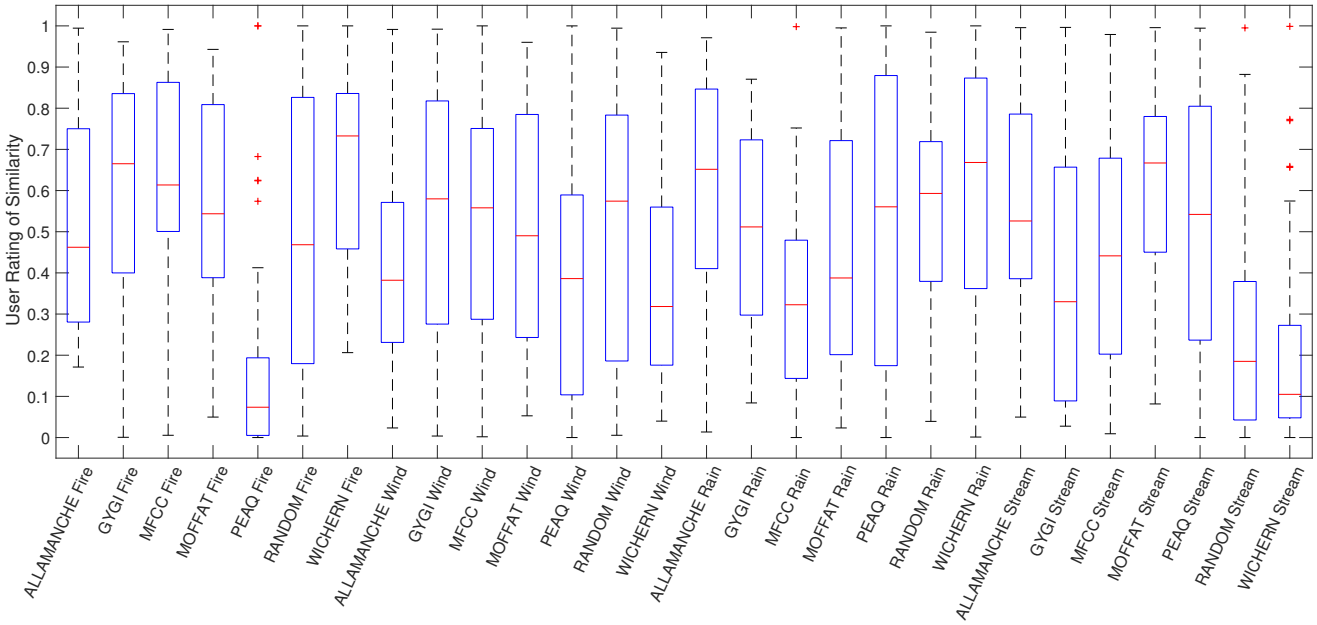


Figure 1: Distribution of User Similarity Ratings over Objective Function and Synthesis Model

5. RESULTS

One participant's results was identified as an outlier as over 30% of their answers was more than three scaled median absolute deviations from the median result. As such all results presented are of the remaining 18 participants. User similarity ratings are presented in Figure 2, where the distributions of the results can be seen.

A Shapiro-Wilk normality test showed that the data is not-normally distributed ($W = 0.95208$, $p < 2.2e\text{-}16$). A Kruskal Wallis test was performed to evaluate the impact of each objective function. A significant difference between the objective evaluation methods was found ($H=18.2$, $p=0.0057$). A post-hoc multiple comparison was performed, with results presented in Table 3.

5.1. Results per Synthesis Method

Table 3 shows that across all sound synthesis models, there is limited consistent variation. The PEAQ objective function is significantly worse than both Allamanche and Moffat. There are no further significant results at this level. To analyse the data further, we investigated the results per synthesis method, as shown in Figure 1. Kruskal Wallis tests were performed to identify the impact of each objective function for each synthesis method. The results show that there are significantly different grouping in three of the four sound synthesis methods. These results are presented in Tables 4-6. Within the wind synthesis method, no significant different in perceptual similarity to the reference sample were found between different objective synthesis methods ($H=11.72$, $p=0.069$).

As seen in Table 4, the PEAQ method is significantly worse than every other objective evaluation function with regards to fire sounds. But for rain sounds, in Table 5 MFCCs are significantly worse than Allamanche, PEAQ, random and Wicherni. For stream sounds, Table 6 shows that Allamanche, Moffat and PEAQ are all significantly better than both random and Wichern. MFCC is also significantly better than Wichern, and Moffat is significantly better than Gygi.

5.2. Comparison with Objective Function Results

Each of the objective functions also produced a distance measure, which is the value that was minimised as part of the synthesis. These distances indicate how successful the synthesis method believes it has performed in each case. The objective distances are compared with the perceptual distances, and are plotted in Figure 3, along with linear regression lines of best fit. The user similarity ratings were inverted to make the graphical representation easier to interpret, and correlations more clear. Each of the objective and subjective results were correlated, using a Spearman correlation, for non-parametric data, and the results presented in Table 7. Only the Wichern result is statistically significant, with a strong positive correlation.

6. DISCUSSION

Table 3 shows minimal significant variation in the distributions of similarity ratings. Overall Moffat performs as the best objective evaluation method, whereas Allamanche is a good options with a lower variance in the data. PEAQ performs the worst, and is significantly worse than both Allamanche and Moffat, which is the only significant generalised result.

For further analysis, we look into the breakdown per synthesis method. Within the fire sound, every objective function was significantly better than PEAQ. PEAQ is the only method that models distortion and bandwidth, and it is believed that these components of the objective function caused it to perform poorly for fire. A large portion of a fire sound is crackling and popping, and broadband noise. As PEAQ is designed for evaluation the quality of audio compression algorithms, it is designed to be sensitive to cracking and distortion artefacts. However, this is principally what makes up a fire sound. As such, it is expected that PEAQ failed to

Table 3: Multiple Comparisons Test Significance Results for All Synthesis Models, Kruskal Wallis Results (H=18.2, p=0.0057)

All Synthesis Methods	Allamanche	Gygi	MFCC	Moffat	PEAQ	Random	Wichern
Allamanche	.	o	o	o	**	o	o
Gygi	o	.	o	o	o	o	o
MFCC	o	o	.	o	o	o	o
Moffat	o	o	o	.	*	o	o
PEAQ	**	o	o	*	.	o	o
Random	o	o	o	o	o	.	o
Wichern	o	o	o	o	o	o	.

o > 0.05, * < 0.05, ** < 0.01, *** < 0.001, **** < 0.0001, . = no comparison made

Table 4: Multiple Comparisons Test Significance Results for Fire Synthesis Method, Kruskal Wallis Results (H=53.19, p=1.08e-9)

Fire	Allamanche	Gygi	MFCC	Moffat	PEAQ	Random	Wichern
Allamanche	.	o	o	o	***	o	o
Gygi	o	.	o	o	****	o	o
MFCC	o	o	.	o	****	o	o
Moffat	o	o	o	.	****	o	o
PEAQ	***	****	****	****	.	***	****
Random	o	o	o	o	***	.	o
Wichern	o	o	o	o	****	o	.

o > 0.05, * < 0.05, ** < 0.01, *** < 0.001, **** < 0.0001, . = no comparison made

appropriately model fire due to the wide-band, impulsive nature of the sound, which PEAQ is often identifies as a flaw. It is suspected that PEAQ will also fail to accurately model other sounds that are broadband and highly impulsive, such as applause [46] or gunshot [47] sounds.

Within the rain sounds, the MFCC evaluation metric performed significantly worse than Allamanche, PEAQ, Wichern and random. MFCCs are often used in music information retrieval as a descriptor for timbre. However, the variation in rain sounds are less timbral and more related to the ambient noise versus individual impulsive tones. The separation between constant noise tones and impulsive tones will not be identified by MFCCs. As MFCCs are no better than the random parameters, it is clear that MFCCs are not a good measure for parameter estimation within rain sounds. There is no other significant variation in objective evaluation functions. Wichern was the only method to perform better than random parameter selection, though this was not significantly better. This could be due to the random parameters being very good parameters selected by chance, or that there is limited variation within the synthesis method.

Regarding stream sounds, Figure 1 shows that Wichern and random both perform poorly, and are significantly worse than Allamanche, Moffat and PEAQ methods, and Alllamanche is significantly worse than MFCCs. It is suspected that this is due to Wichern primarily looking at harmonic content and transient sounds, where less attention was paid to broadband sound similarities. Within the stream model, most water noises will be highly broadband signals, and Wichern will most likely tend to produce more harmonic tuned sounds, than those present in a real signal. Wichern and random are not significantly worse than Gygi, which is most likely due to the large variation in the distribution of the Gygi results. This suggests that individuals were undecided or opinions were split on the result. Moffat was the best performing result and is significantly better than Gygi, along with random and Wichern. It is suspected that this is due to the inclusion of the spectral contrast feature. Spectral contrast is an audio feature that identifies the peaks and valleys in the magnitude spectrum, and performs dimensionality reduction on the result. Spectral contrast is often considered an effective method for evaluating audio masking and for identifying variations high contrast variations in frequency spectra.

The wind model failed to produce any significant difference between any objective metrics. Gygi performed the best, closely followed by random parameter allocation, but all methods are fairly similar to each other. This could be a failing of the synthesis model, as there were highly harmonic artefacts within the synthesis model, that no parameters could be removed. Further investigation of the synthesis model shows that a number of filter center frequencies are hard-coded into the model, which most likely led to inconsistent and inconclusive results. It is also possible that the number of parameters may also have influenced the results. Wind had more than twice the parameters to optimise compared to any other synthesis model, which the particle swarm algorithm may have had challenges optimising. The larger search space may have lead to issues in finding appropriate minima.

Each of the objective functions were compared and grouped in terms of how their effectiveness on a 1-5 rating scale, as presented in Table 8. It can be seen that the Gygi method performs best for both fire and wind sounds and fairly well for rain sounds, but is one of the worst objective measures for the stream sound. Gygi contains a large set of parameters relating to subband correlations and modulation statistics, which have been tied to the human auditory system [6]. As such, Gygi method seems to be the best overall performer, as consistently produced reasonable results in all cases, and between that and Moffat, it never produced the worst results. Moffat performed best overall, and was best for wind sounds, which it is suspected is due to the spectral contrast feature. It also performed reasonably well for fire and rain sounds, as the spectral contrast and spectral flux sounds will perform well for granular impulsive sounds. The Allamanche method performs best for rain sounds and reasonably well for stream sounds, but is

Table 5: Multiple Comparisons Test Significance Results for Rain Synthesis Method, Kruskal Wallis Results (H=26.81, p=1.57e-4)

Rain	Allamanche	Gygi	MFCC	Moffat	PEAQ	Random	Wichern
Allamanche	.	o	***	o	o	o	o
Gygi	o	.	o	o	o	o	o
MFCC	***	o	.	o	*	*	***
Moffat	o	o	o	.	o	o	o
PEAQ	o	o	*	o	.	o	o
Random	o	o	*	o	o	.	o
Wichern	o	o	***	o	o	o	.

o > 0.05, * < 0.05, ** < 0.01, *** < 0.001, **** < 0.0001, . = no comparison made

Table 6: Multiple Comparisons Test Significance Results for Stream Synthesis Method, Kruskal Wallis Results (H=54.91, p=4.84e-10)

Stream	Allamanche	Gygi	MFCC	Moffat	PEAQ	Random	Wichern
Allamanche	.	o	o	o	o	***	****
Gygi	o	.	o	*	o	o	o
MFCC	o	o	.	o	o	o	*
Moffat	o	*	o	.	o	****	****
PEAQ	o	o	o	o	.	**	**
Random	***	o	o	****	**	.	o
Wichern	****	o	*	****	**	o	.

o > 0.05, * < 0.05, ** < 0.01, *** < 0.001, **** < 0.0001, . = no comparison made

Table 7: Correlations of Objective Function Distance Measure with Mean User Similarity Rating

Objective Function	Correlations ρ	P-Value p
Allamanche	-0.3095	0.4618
Gygi	-0.0952	0.8401
MFCC	0.0238	0.9768
Moffat	-0.3095	0.4618
PEAQ	-0.4059	0.3155
Wichern	**0.7857**	**0.0279**

Table 8: Ratings of Success of each Objective Evaluation Method

	Overall	Fire	Rain	Stream	Wind
Allamanche	2	4	5	1	1
Gygi	2	1	1	3	4
MFCC	2	1	2	5	3
Moffat	1	3	3	4	1
PEAQ	5	5	5	3	2
Wichern	4	1	5	1	5

1 = Best, 5 = Worse. Ratings were created manually, based on ranking and clustering of results

one of the worse methods for wind and fire sounds. This suggests that the spectral characteristics are more complex for wind and fire sounds, as Allamanche only uses a spectral flatness and spectral crest factor as the evaluation, as all samples were loudness normalised before analysis. PEAQ performed worse overall, through performing worse in both fire and rain sounds, however performed reasonably we for stream and wind sounds. This demonstrates that PEAQ represents broadband noisy signals fairly well, however the low level textual and highly impulsive sounds are not effectively modelled by this method. The Wichern method is highly inconsistent as it performs best for fire and stream however is the worse for rain and wind sounds.

Wichern was the only objective evaluation method where the objective distance significantly correlated with the perceptual distance ratings. The correlations of the objective distance are a vital aspect of any objective evaluation function, where it is possible to predict how well the objective function performs and how effective the synthesised sound is.

7. CONCLUSION

A set of six different objective evaluation functions, for measuring similarity between environmental sounds, were tested and compared, through their ability to direct a resynthesis algorithm towards an appropriate parameter setting. In the general term, across four different types of sounds, there was no significant winner. The PEAQ method performed the worse, performing significantly worse than both Moffat and Allamanche. This demonstrates that PEAQ is not a suitable for evaluating sound similarity in a range of different cases, though it was effective for comparing broadband noisy signals, such as wind. The results demonstrate that there is currently no unilateral objective evaluation function, an consistently no method is a clear winner in most cases. One of the causes of this could be the failings or limitations of the synthesis models used. The limitation for each method to produce a wide range of sounds, could result in many different samples being challenging to synthesize, and thus cause all methods to underperform.

Despite this, the Wichern method results correlate significantly and strongly perceptual distance measures. This suggests that the

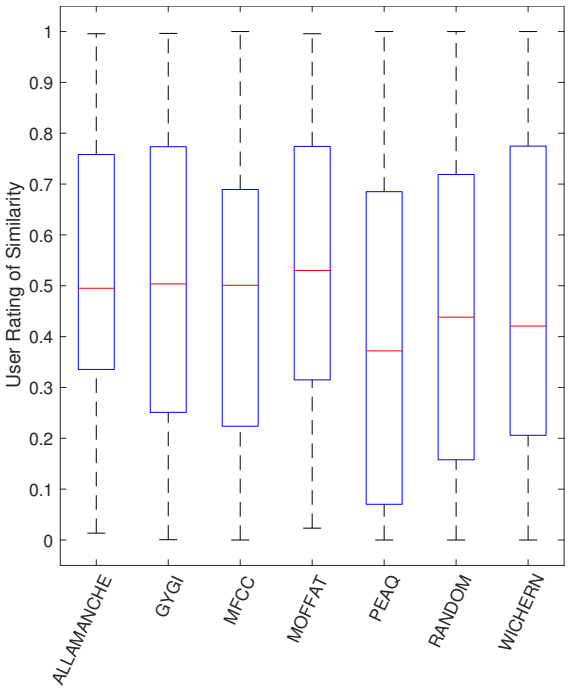


Figure 2: Distribution of User Similarity Ratings per Objective Function

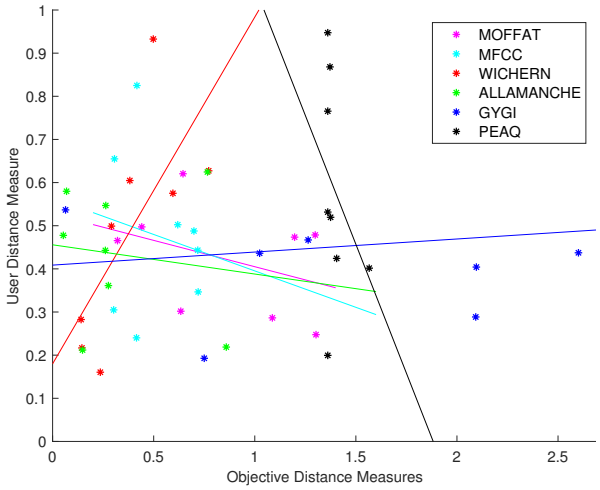


Figure 3: Inverse User Similarity Compared Against Objective Distance Metric for Each Objective Function, with Linear Best Fit Lines

Wichern method can be used as an effective distance metric, comparing similarity between different sets of sounds. Further evaluation with different synthesis methods is required to verify these results and to identify whether the synthesis methods themselves impacted the results.

The use of further different sounds samples and sound classes would also provide further data points, which would aid in correlating the objective results with the perceptual ratings. This would ensure that the results can be applied to a range of different sound types. Furthermore, there were some cases where the synthesis method was not capable of producing a very similar sample. In which case, careful improvement and selection of synthesis methods and samples could be made in future work. Further evaluation of different perceptual measures of similarity, and comparison of objective measures with expert human parameter modification could also be performed.

8. REFERENCES

[1] D. Jaffe, "Ten criteria for evaluating synthesis techniques," *Computer Music Journal*, vol. 19, no. 1, pp. 76–87, 1995.

[2] N. Böttcher and S. Serafin, "Design and evaluation of physically inspired models of sound effects in computer games," in *Audio Engineering Society Conference: 35th International Conference: Audio for Games*, London, 2009, AES.

[3] D. Moffat, R. Selfridge, and J. D. Reiss, "Sound effect synthesis and control," in *Foundations in Sound Design: an interdisciplinary approach*, Michael Filimowicz, Ed., vol. Volume 2: Interactive Media. Routledge, 2018.

[4] D. Moffat and J D. Reiss, "Perceptual evaluation of synthesized sound effects," *ACM Transactions on Applied Perception (TAP)*, vol. 15, no. 2, pp. 19, March 2018.

[5] D. Schwarz, "State of the art in sound texture synthesis," in *14th International Conference Digital Audio Effects (DAFx)*, Paris, France, 2011, pp. 221–231.

[6] J. McDermott and E. Simoncelli, "Sound texture perception via statistics of the auditory periphery: evidence from sound synthesis," *Neuron*, vol. 71, no. 5, pp. 926–940, 2011.

[7] R. Garcia, "Automating the design of sound synthesis techniques using evolutionary methods," in *COST G-6 Conference on Digital Audio Effects, Limerick, Ireland*, 2001.

[8] J. McDermott, N. Griffith, and M. O'Neill, "Evolutionary computation applied to sound synthesis," in *The Art of Artificial Evolution*, pp. 81–101. Springer, 2008.

[9] M. Yee-King and M. Roth, "A comparison of parametric optimization techniques for musical instrument tone matching," in *Audio Engineering Society Convention 130*, 2011.

[10] R. Selfridge, D. Moffat, and J. D. Reiss, "Real-time physical model for synthesis of sword swing sounds," in *International Conference on Sound and Music Computing (SMC)*, Espoo, Finland, July 2017.

[11] S. Hendry and J. D. Reiss, "Physical modeling and synthesis of motor noise for replication of a sound effects library," in *Audio Engineering Society Convention 129*, Los Angeles, CA, USA, 2010.

[12] M. Gasparini, P. Peretti, S. Cecchi, L. Romoli, and F. Piazza, "Real time reproduction of moving sound sources by wave field synthesis: Objective and subjective quality evaluation," in *Audio Engineering Society Convention 130*, 2011.

[13] R. Selfridge, D. Moffat, J. D. Reiss, and E. J. Avital, "Real-time physical model for an aeolian harp," in *International Congress on Sound and Vibration*, London, UK, July 2017.

[14] R. Selfridge, D. Moffat, and J. D. Reiss, "Physically derived sound synthesis model of a propeller," in *ACM Audio Mostly Conference*, London, UK, August 2017.

[15] R. Selfridge, D. Moffat, and J. D. Reiss, "Sound synthesis of objects swinging through air using physical models," *Applied Sciences*, November 2017.

[16] L. Lu, L. Wenyin, and H.-J. Zhang, "Audio textures: Theory and applications," *Speech and Audio Processing, IEEE Transactions on*, vol. 12, no. 2, pp. 156–167, 2004.

[17] S. O'Leary and A. Robel, "A montage approach to sound texture synthesis," in *22nd European Signal Processing Conference (EUSIPCO)*, Lisbon, Portugal, 2014, pp. 939–943.

[18] M. Athineos and D. P. W. Ellis, "Sound texture modelling with linear prediction in both time and frequency domains," in *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 2003, vol. 5, pp. 648–51.

[19] B. Hamadicharef and E. Ifeachor, "Perceptual modeling of piano tones," in *Audio Engineering Society Convention 119*, Barcelona, Spain, Oct 2005.

[20] R. A. Garcia, "Automatic generation of sound synthesis techniques," M.S. thesis, Massachusetts Institute of Technology, 2001.

[21] S. Heise, M. Hlatky, and J. Loviscach, "Automatic cloning of recorded sounds by software synthesizers," in *Audio Engineering Society Convention 127*, New York, USA, 2009.

[22] R. Nordahl, S. Serafin, and L. Turchet, "Sound synthesis and evaluation of interactive footsteps for virtual reality applications," in *IEEE Virtual Reality Conference*, Waltham, MA, USA, 2010, pp. 147–153, IEEE.

[23] D. Schwarz and S. O'Leary, "Smooth granular sound texture synthesis by control of timbral similarity," in *Sound and Music Computing (SMC)*, 2015, p. 6.

[24] A. Horner and S. Wun, "Evaluation of iterative matching for scalable wavetable synthesis," in *Audio Engineering Society Conference: 29th International Conference: Audio for Mobile and Handheld Devices*, Seoul, Korea, 2006.

[25] Wei-Hsiang Liao, Axel Roebel, and Alvin Su, "On the modeling of sound textures based on the stft representation," in *Proc. of the 16th Int. Conference on Digital Audio Effects (DAFx-13)*, 2013, p. 33.

[26] R. Selfridge, J. Reiss, E. Avital, and T. Xiaolong, "Physically derived synthesis model of an aeolian tone," in *141th Audio Engineering Society Convention*, Los Angeles, CA, USA, 2016.

[27] R. Selfridge, D. Moffat, J. D. Reiss, and E. Avital, "Creating real-time aeroacoustic sound effects using physically derived models," *Journal of the Audio Engineering Society (to appear)*, 2018.

[28] J. McDermott, D. Wrobleski, and A. Oxenham, "Recovering sound sources from embedded repetition," *Proceedings of the National Academy of Sciences*, vol. 108, no. 3, pp. 1188–1193, 2011.

[29] T. Thiede, W. Treurniet, et al., "PEAQ-The ITU standard for objective measurement of perceived audio quality," *Journal of the Audio Engineering Society*, vol. 48, no. 1/2, pp. 3–29, 2000.

[30] P. Bahadoran, A. Benito, T. Vassallo, and J. D. Reiss, "Sound effect synthesis," 2017, UK Patent App. Num. N411552GB HHG.

[31] P. Bahadoran, A. Benito, T. Vassallo, and J. D. Reiss, "FXive: A web platform for procedural sound synthesis," in *Audio Engineering Society Convention 144*, Milan, Italy, 2018.

[32] A. Farnell, *Designing sound*, MIT Press Cambridge, UK, 2010.

[33] T. Mäkinen, S. Kiranyaz, J. Pulkkinen, and M. Gabbouj, "Evolutionary feature generation for content-based audio classification and retrieval," in *Signal Processing Conference (EUSIPCO), 2012 Proceedings of the 20th European*. IEEE, 2012, pp. 1474–1478.

[34] D. Ronan, Z. Ma, P. Mc Namara, H. Gunes, and J. D. Reiss, "Automatic minimisation of masking in multitrack audio using subgroups," *ArXiv e-prints*, Mar. 2018.

[35] F. Marini and B. Walczak, "Particle swarm optimization (PSO). a tutorial," *Chemometrics and Intelligent Laboratory Systems*, vol. 149, pp. 153–165, 2015.

[36] Dmitry Bogdanov et al., "Essentia: An audio analysis library for music information retrieval," in *International Symposium on Music Information Retrieval (ISMIR)*, 2013, pp. 493–498.

[37] D. Moffat, D. Ronan, and J. Reiss, "An evaluation of audio feature extraction toolboxes," in *Proc. 18th International Conference on Digital Audio Effects (DAFx-15)*, November 2015.

[38] E. Allamanche, J. Herre, O. Hellmuth, et al., "Content-based identification of audio material using MPEG-7 low level description.," in *ISMIR*, 2001.

[39] B. Gygi, G. Kidd, and C. Watson, "Similarity and categorization of environmental sounds," *Perception & psychophysics*, vol. 69, no. 6, pp. 839–855, 2007.

[40] D. Moffat, D. Ronan, and J. D. Reiss, "Unsupervised taxonomy of sound effects," in *Proc. 20th International Conference on Digital Audio Effects (DAFx-17)*, Edinburgh, UK., September 2017.

[41] G. Wichern, H. Thornburg, B. Mechtley, et al., "Robust multi-features segmentation and indexing for natural sound environments," in *Content-Based Multimedia Indexing, 2007. CBMI'07. International Workshop on*. IEEE, 2007, pp. 69–76.

[42] M. Morrell, C. Harte, and J. Reiss, "Queen Mary's "Media and Arts Technology studios" audio system design," in *Audio Engineering Society Convention 130*, 2011.

[43] N. Jillings, B. De Man, D. Moffat, and J. Reiss, "Web audio evaluation tool: A browser-based listening test environment," in *Proc. Sound and Music Computing 2015*, Maynooth, Ireland, July 2015.

[44] ITU-R BS.1387-1, "BS. 1387, method for objective measurements of perceived audio quality," Tech. Rep., ITU-R, 1998.

[45] ITU-R BS.1534-3, "BS. 1534, method for subjective assessment of intermediate quality level of audio systems," Tech. Rep., ITU-R, 2015.

[46] A. Adami, A. Taghipour, and J. Herre, "On similarity and density of applause sounds," *Journal of the Audio Engineering Society*, vol. 65, no. 11, pp. 897–913, 2017.

[47] L. Mengual, D. Moffat, and J. D. Reiss, "Modal synthesis of weapon sounds," in *61st Audio Engineering Society International Conference: Audio for Games*, 2016.

[48] Marios Athineos and Daniel PW Ellis, "Autoregressive modeling of temporal envelopes," *IEEE Transactions on Signal Processing*, vol. 55, no. 11, pp. 5237–5245, 2007.

RESIZING ROOMS IN CONVOLUTION, DELAY NETWORK, AND MODAL REVERBERATORS

Elliot K. Canfield-Dafilou and Jonathan S. Abel

Center for Computer Research in Music and Acoustics,
Stanford University, Stanford, CA 94305 USA
kermit|abel@ccrma.stanford.edu

ABSTRACT

In music recording and virtual reality applications, it is often desirable to control the perceived size of a synthesized acoustic space. Here, we demonstrate a physically informed method for enlarging and shrinking room size. A room size parameter is introduced to modify the time and frequency components of convolution, delay network, and modal artificial reverberation architectures to affect the listener's sense of the size of the acoustic space taking into account air and materials absorption.

1. INTRODUCTION

Computational methods for simulating reverberant environments are well developed [1], and find application in fields ranging from music recording to virtual reality and film audio production. Room acoustics is an approximately linear and time-invariant process, and there are several widely used methods for room acoustics simulation, including direct convolution with an impulse response [2], delay network-based methods [3], and modal reverberation [4].

In a number of scenarios, it is desirable to manipulate or control the perceived size of a given acoustic space. In a virtual reality or film setting, for instance, the size of the room might be changing over time, and it is preferable that the acoustics of the space change accordingly. In a music recording, performance, or composition environment, different sizes of acoustic space convey different musical impressions, and it is useful to have a palette of room size options associated with a given room response for artistic purposes. Larger spaces tend to be more reverberant and "darker" than smaller ones, but there does not seem to be a systematic way to manipulate the perceived size associated with a given room response. Rafii and Pardo [5] proposed finding relationships between subjective terms and reverb parameters, and Chourdakis and Reiss [6] proposed an adaptive reverberation algorithm based on learning parameters from user actions. In both cases, these methods can be used to modify a reverberation algorithm based on subjective characteristics rather than the physics of changing the dimensions of a room.

In this paper, we introduce a room size parameter for modifying the time and frequency content of an artificial reverberator. We demonstrate a physically informed method for changing the size of a room, taking into account the changes in geometry, absorbing surface area, and volume. We then show how to implement this room size parameter in convolution, delay network, and modal reverberation architectures.

This paper is organized as follows: section 2 introduces the acoustical concepts necessary for modifying room size. Section 3 discusses the implementation of the room resizing parameter in convolution, delay network, and modal reverberaters. Finally, section 4 offers some concluding remarks.

2. ON THE ACOUSTICS OF ROOM SIZE

The room response to a transient sound is often described as a sequence of events over time, a direct path followed by early reflections that give way to late-field reverberation, as seen in Fig. 1. The direct path carries with it information about the source direction, and arrives with a time delay and amplitude fixed according to the source-listener distance. The early reflections contain information about the geometry of the space, and can be simulated using details of the architecture of the space [7]. The late-field reverberation brings to the listener information about the volume of the space and materials present in the space through the frequency dependent rates of sound energy decay. Roughly speaking, the reverberation time is proportional to the ratio of the room volume to the room absorbing surface area [8].

If the room size were doubled, with everything else remaining the same, then the timing of the direct path and early reflections would be stretched by a factor of two. Similarly, if the room size were doubled, then its volume would increase by a factor of eight, while its absorbing area would increase by a factor of four, thereby doubling the reverberation time.

Reasoning along these lines was used by Spandöck in building scale models of proposed concert halls to test how they might sound when built [9]. Spandöck argued that a scale model of a concert hall made with the appropriate materials and filled with a dried gas would respond to a given high-frequency sound the way the larger actual space would respond to a low-frequency sound having the same relative wavelength. Spandöck describes using a magnetic tape deck to play back a sound into the scale model sped up by a factor of, say, eight, while simultaneously recording the response in the model. The recording was then played back, slowed by the same factor. In this way, the original pitch was restored, and the reverberation time increased to match that of the hypothesized full-scale hall.

As described in [10], this approach was independently discovered by Walter Murch while working as a sound editor for motion pictures in the late 1960s, and was used to make long-lasting reverberation. Spratt, et al. present a digital technique for implementing a real-time version of the method, using a loudspeaker and microphone in a physical room [10].

The technique is described as being mathematically equivalent to stretching the room impulse response in time, which has the effect of increasing the reverberation time, and stretching the reflection arrival times. Spratt, et al. argue that the method is similar to slowing the speed of sound or increasing the room size. However, doing either of these will not result in proper reverberation time as a function of frequency, as the relative absorption of sound by air and room materials will not be taken into account.

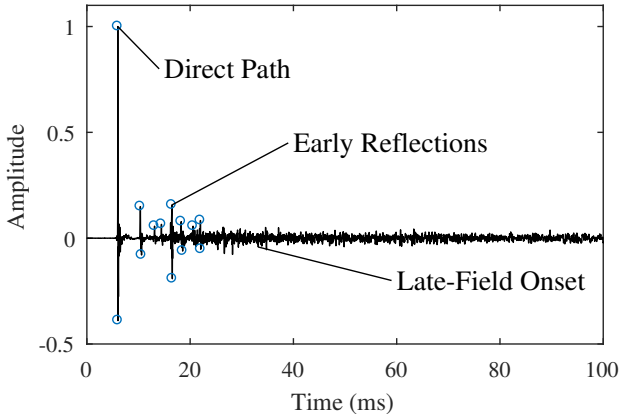


Figure 1: Example room impulse response showing the direct path, early reflections, and late-field reverberation onset.

If a room is proportionally scaled, the echo pattern will be linearly stretched or squished. As a result, the echo density [11], or rate of reflections, will also be linearly scaled. However, to be physically accurate, one must also take the surface area and volume changes into account. Air absorption is nonlinear across frequency, and high frequencies will typically decay faster in a larger room than a small one. Additionally, simply enlarging or shrinking the room via the method described in [10] also proportionally scales all the room materials. For example, the pores in a carpet would be scaled, changing its contribution to the frequency response in the room. Here, we suggest a method for taking the air absorption and materials absorption into consideration when scaling the size of rooms.

3. SCALING ROOM SIZE

Perceived room size may be manipulated in the context of a number of artificial reverberation methods. The idea is to warp the time and frequency axes and adjust the decay times of a given reverberation impulse response according to a desired room size. In addition, the source loudness and radiation pattern may be adjusted according to the room size.

3.1. Reverberation Time

We first describe the change in reverberation time in response to a changing room size as a result of different relative contributions of materials absorption and air absorption.

As described in [8] and elsewhere, the decay over time of well-mixed acoustic energy in a room can be approximated by examining a room with volume V and having objects and surfaces with absorbing area A. The energy density $w(t)$ as a function of time t is assumed to be well mixed and independent of position within the room. After a period of time Δt, the total energy in the room, the product of the energy density and the volume, $Vw(t+\Delta t)$, will be that at time t minus what is lost due to interactions with absorbing surfaces and objects and air propagation,

$$Vw(t+\Delta t) = Vw(t) - Acgw(t)\Delta t - Vaw(t)\Delta t\,, \qquad (1)$$

where the term $Acgw(t)\Delta t$ represents surface interaction absorption, and is proportional to the absorbing area A, sound speed c, a constant g, energy density $w(t)$, and time interval Δt, and the term $Vaw(t)\Delta t$ represents air absorption, and is proportional to the volume V, an absorption coefficient a, energy density $w(t)$, and time interval Δt. These absorption terms can be intuitively interpreted—the greater the time interval, the more energy that can be absorbed; the greater the energy density, the more energy that can "leave" the space during the time interval. Rearranging terms, and taking $\Delta t \to 0$, we have

$$\frac{w(t+\Delta t) - w(t)}{\Delta t} \to \frac{dw}{dt} = -\frac{1}{\tau}w(t), \qquad (2)$$

and

$$w(t) = w_0 e^{-t/\tau}, \qquad t \geq 0\,, \qquad (3)$$

with w_0 being the energy density at time $t = 0$, and τ being a time constant which increases with increasing volume, and decreases with increasing absorbing area,

$$\tau = \frac{V}{Acg + Va}\,. \qquad (4)$$

In other words, the energy density in a well mixed room will decay exponentially, decreasing by a factor of $1/e$ every τ units of time. It is typical to measure reverberation time in terms of the time taken for energy to decrease 60 dB, T_{60}, in which case we have

$$T_{60} = \frac{\log_{10} 10^6}{\log_{10} e}\,\tau\,, \qquad (5)$$

measured in units of seconds per 60 dB decay.

Energy density is also a function of frequency ω, $w(t,\omega)$, which was dropped from the discussion here for simplicity of presentation. It carries over to frequency-dependent materials and air absorption simply by making the absorbing area A and air absorption a frequency-dependent.

3.2. Adjusting Room Size

Consider a room described by a nominal length L_0. Now scale the room and all of its surfaces and objects to have a new characteristic length L. We want to understand how the decay time changes with a changing room size L. Using (4) and (5), and assuming that the room volume V is proportional to L^3 and the absorbing area A is proportional to L^2, the decay time of the resized room $T_{60}(L)$ is then

$$T_{60}(L) = \frac{L}{L_0\mu + L\alpha}\,, \qquad (6)$$

where μ has been introduced to represent the materials absorption for the nominally sized room, α has been introduced to represent the materials absorption. Both μ and α are expressed in terms of 60 dB decay per unit time. Note also that

$$\alpha = \frac{\log_{10} 10^6}{\log_{10} e}\,a\,. \qquad (7)$$

The decay time at the nominal room size,

$$T_0 = T_{60}(L_0)\,, \qquad (8)$$

may be estimated from the room impulse response or otherwise modeled, and that the air absorption α is known, derived assuming a given temperature, pressure, and humidity, or tabulated [12, 13].

Accordingly, setting $L = L_0$ and solving (6) for the unknown materials absorption μ gives

$$\mu = \frac{1}{T_0} - \alpha\,. \tag{9}$$

Due to errors in estimating decay times from measured impulse responses, the reverberation time T_0 might exceed the air absorption-only reverberation time $1/\alpha$, and (9) would produce a negative value for μ. In these cases (or at such frequencies that this is true), a value of $\mu = 0$ is preferably used, and the reverberation time will not be affected by room size. If it is desired to have a changing reverberation time with room size, a small value for μ could be selected.

Substituting for μ in (6) gives T_0, the decay time as a function of room size L. In the case that μ is given by (9) and not modified, a little algebra gives an expression for $T_{60}(L)$ in terms of the nominal decay time T_0 and the decay time if the only absorption of sound energy were due to air $T_{\text{air}} = 1/\alpha$,

$$T_{60}(L) = \frac{L \cdot T_0 T_{\text{air}}}{L_0 \cdot T_{\text{air}} + (L - L_0) \cdot T_0}\,. \tag{10}$$

As an example of a changing reverberation time as a function of room size, consider the reverberation time of a church with a 10-meter nominal size, shown as a line with markers in Fig. 2. Also shown are the reverberation times of hypothesized churches that are 2, 4, 8, and 16 times as large, and 2, and 4 times as small. For reference, the reverberation time associated with air absorption only, α for 50% humidity and 25° C is shown in Fig. 3. Generally speaking, a doubling of the room size doubles the reverberation time. However, for large rooms and high frequencies (where the air absorption and materials absorption are somewhat comparable), a doubling of the room size increases the reverberation time by a good bit less than the factor of two seen at low frequencies or for small rooms. Note that this is the case for high frequencies in Fig. 2.

The effect of a finite air absorption may be exaggerated or suppressed by reducing or increasing—or even replacing—the air absorption characteristic shown in Fig. 3. The idea is to have different frequency bands express different reverberation times, scaling with room size. In doing so, when solving (9) for the materials absorption $\mu(\omega)$, any frequencies ω producing values less than zero should be set to zero. That is,

$$\mu(\omega) = \max\left(0,\, \frac{1}{T_{60}(L_0, \omega)} - \alpha(\omega)\right)\,. \tag{11}$$

As an example of a changing room size with a modified air absorption characteristic, Fig. 4 shows the reverberation times of Fig. 3 with a wacky air absorption.

3.3. Implementation in Common Reverberation Architectures

As described in [1], there are many commonly used reverberation algorithms. Here, we show how to resize rooms using three common methods: direct convolution, feedback delay network, and modal reverberators. The room impulse response is stretched in time and its decay rate as a function of frequency is modified to properly account for the changing relative importance of air absorption and materials absorption.

When using a convolutional reverberator (see Fig. 5), the room impulse response is resampled in time according to a room size

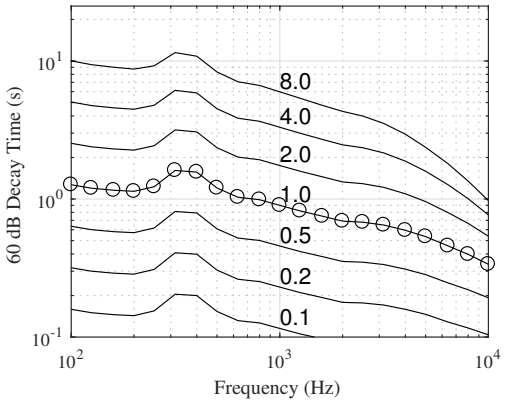


Figure 2: Example reverberation time of a small church as a function of room size taking air absorption into consideration. The markers show the measured reverberation times and the traces show the decay times when the nominal length of the church is scaled.

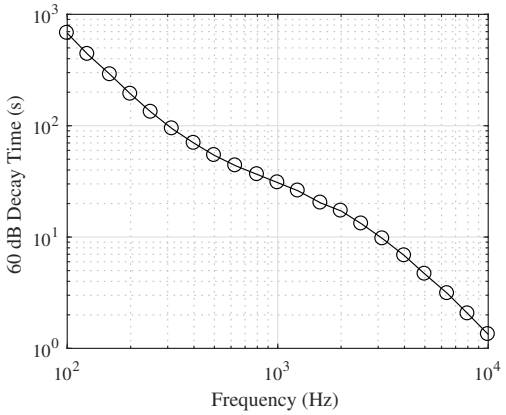


Figure 3: Reverberation time of a room with perfectly reflecting walls filled with STP air at 50% humidity.

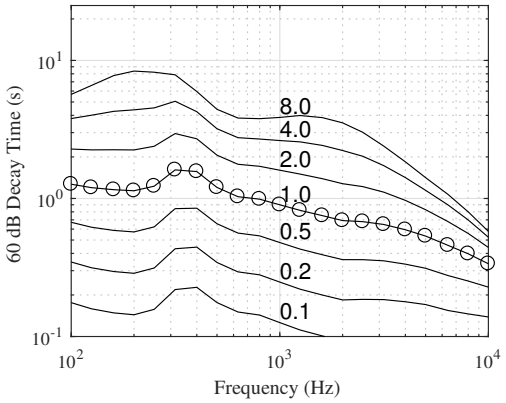


Figure 4: Example reverberation time of a small church as a function of room size, with a strange air absorption characteristic.

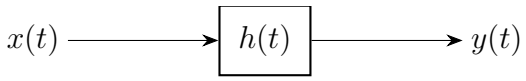


Figure 5: A convolutional reverberator showing an input signal $x(t)$, convolved with a room impulse response $h(t)$ to produce a reverberated output $y(t)$.

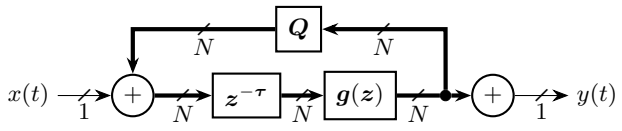


Figure 6: A feedback delay network reverberator, including a set of N delay lines z^{-T_n}, filters $g_n(z)$, $n = 1, 2, \ldots, N$, and an orthonormal mixing matrix $\boldsymbol{Q}$.

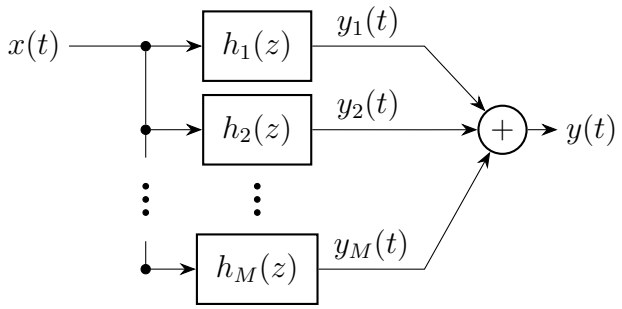


Figure 7: A modal reverberator having a parallel set of M mode filters $h_m(z)$, each characterized by a mode frequency ω_m, mode decay time τ_m, and mode amplitude γ_m.

control, and its decay rate as a function of frequency is modified. Depending on whether the room is being made larger or smaller, a high-frequency reverberant room response may be synthesized to extend the reverberation to frequencies which are warped into the audio band. A second method is described where an existing impulse response is resynthesized from its room-size-modified echo density.

For a reverberator implemented using a network of delay lines (see Fig. 6), the delay times are stretched according to the room size control, and the feedback filters are warped and scaled according to the new decay times and delay lengths. A second method is also described where the delay line lengths are not adjusted but the filters and mixing matrix are modified to account for the room resizing.

In a modal reverberator (see Fig. 7), the room size control modifies the mode frequencies and dampings. Additionally, high-frequency or low-frequency modes may need to be synthesized.

3.3.1. Convolution Reverberator

In the case of a convolutional reverberator [2, 14], the given or nominal room impulse response, $h_0(t)$, associated with a nominal room size L_0, may be resampled according to the new room size L to produce an adjusted impulse response $h_L(t)$,

$$h_L(t) = h_0\left(t \cdot \frac{L_0}{L}\right) . \tag{12}$$

As seen in Fig. 8, this adjusted impulse response may then be used to process an input signal $x(t)$ to produce a reverberated output $y(t)$ associated with the room of size L.

In the case that the room size L is smaller than the nominal room size L_0, the resampling will shorten the impulse response, thereby increasing its bandwidth. Preferably, the resampling would include the step of low-pass filtering so as to avoid aliasing if the increased bandwidth exceeds the Nyquist limit.

If L is larger than L_0, then the resampled (i.e., interpolated) impulse response will be longer than the original impulse response, and have decreased bandwidth. In this case, the adjusted impulse response may be extended to the Nyquist limit by first estimating reverberation characteristics such as decay times, equalization, echo density, and the like for that band. For instance, the decay times may be assumed to decrease in a manner typical of air absorption with increasing frequency above the original bandwidth. A trend could be fit to the decay characteristic of the nominal impulse response, and extended in frequency. Similarly, the equalization could be extrapolated to higher frequencies by noting the trend near the nominal band edge.

The mechanism of increasing the reverberation time by multiplying the reverberation impulse response by a growing exponential (as used in a number of commercially available convolutional reverberators) will generate unwanted artifacts, including a bloom in energy at the end of the impulse response [15, 16]. Resampling the impulse response as described above generally avoids this difficulty, though extending the impulse response to below the noise floor also would be of benefit. It should be noted that such a mechanism for lengthening reverberation time, even when applied to a properly extended room response, is not preferred, as the timing of temporal features, such as significant early reflections, are not appropriately modified.

As described above, the reverberation time of a room with with a modified size is roughly scaled by the relative change in size. It is affected by the different relative absorptions of air and materials, with materials absorption accounting for a greater portion of the decay in smaller rooms. The resampling of the impulse response described above has the effect of simultaneously stretching the reverberation time and compressing the associated frequency axis,

$$\tilde{T}_{60}(L, \omega) = \frac{L}{L_0}\, T_0\left(\omega \cdot \frac{L}{L_0}\right) , \tag{13}$$

where $\tilde{T}_{60}(L, \omega)$ is the frequency-dependent reverberation time of the stretched impulse response $h_L(t)$, and $T_0(\omega)$ is that of the given impulse response $h(t)$. For example, if a room impulse response were stretched by a factor of two, the reverberation time at 500 Hz would be twice that of the original impulse response at 1000 Hz. As a result, when the given reverberation time $T_0(\omega)$ is not relatively constant with frequency, the reverberation time produced by resampling $h(t)$ will differ from the desired one given by (10), and it is preferable to modify the reverberation time of the stretched impulse response accordingly.

As shown in Fig. 9, this may be accomplished by splitting the resampled room impulse response $h_L(t)$ into a set of frequency bands (for instance, half-octave-wide bands or ERB bands). Each band is then windowed with a growing or shrinking exponential function to give it the desired reverberation time. Then the windowed bands are summed to form a room response having the appropriate amplitude envelope as a function of frequency. This process could also be applied to the given impulse response $h(t)$

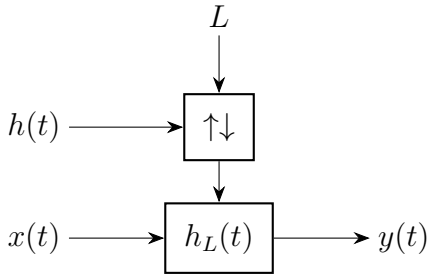


Figure 8: A convolutional reverberator showing a process operating on the impulse response $h(t)$ so that it is time-stretched (resampled) according to a room size control.

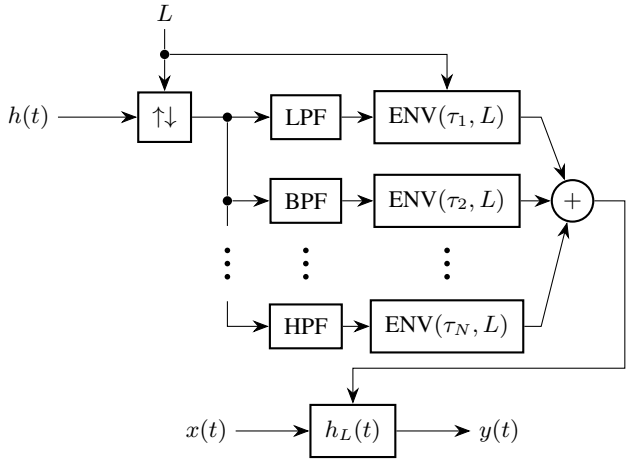


Figure 9: A convolutional reverberator in which a room size parameter modifies the resampling amount as well as modifying the frequency-dependent decay rates by $e^{-t/\boldsymbol{\tau}} \rightarrow e^{-(tL_0)/(\boldsymbol{\tau}L)}$.

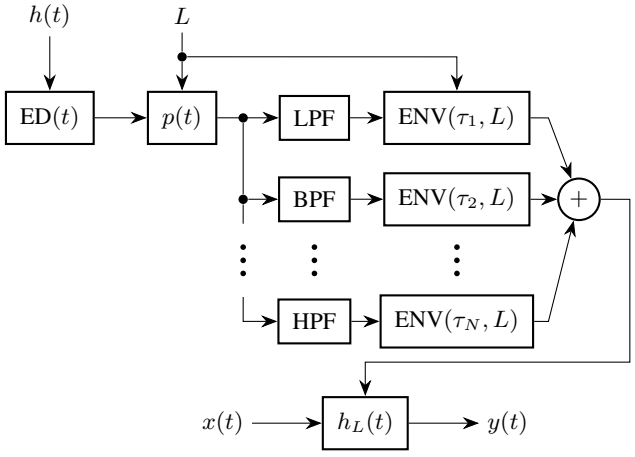


Figure 10: A convolutional reverberator in which a pulse sequence is synthesized from the echo density estimated from a desired room impulse response, split into frequency bands, and the bands windowed and summed to form an impulse response used in a convolutional reverberator. The timing of the pulses and duration of the band envelopes are adjusted according to room size.

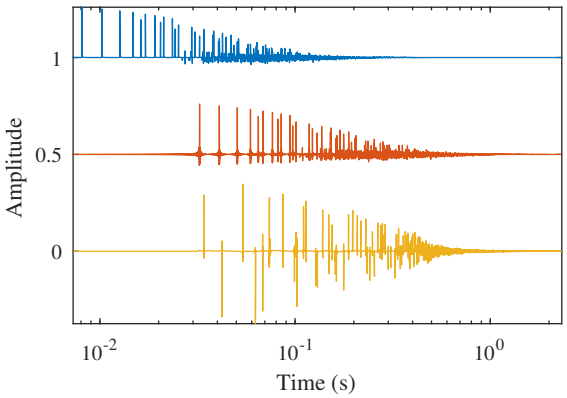


Figure 11: Time domain plots of the early reflections of impulse responses for use with a convolution reverberator showing the original IR (top), the IR stretched by a factor of 4 through resampling (middle), and resynthesized from its echo density, stretched by a factor of 4 (bottom). As a result of plotting these IRs on a logarithmic time axis, the stretched IRs appear shifted by an amount $\log_{10} 4$. Note that the ideal sinc interpolation used for the middle example filters each pulse, while the bottom example shows how resynthesizing the stretched impulse response from a statistical model does not preserve the exact echo sequence but does not have the same filtering as a result of the resampling.

before resampling, with the band windowing anticipating the reverberation time changes produced by the resampling.

It should be pointed out that while Spratt and Abel [10] describe resampling the room impulse response as similar to changing the sound speed or resizing the room, this is only true if everything about the room is resized, including materials absorption features. Here, we desire to scale the room size without modifying the materials or air properties, and it is thus preferred to correct the reverberation time produced by resampling as described above.

Finally, we note that the method described in [17] to synthesize impulse responses from balloon pop recordings may be adapted to synthesize room impulse responses at different room sizes. The process is shown in Fig. 10. Echo density is measured along the given impulse response $h(t)$, and the impulse response root energy over time (e.g., an amplitude envelope) in a set of frequency bands is estimated. A statistically independent, but perceptually identical, nominal impulse response $h_L(t)$ is then synthesized by randomly generating a set of full-bandwidth pulses, $p(t)$, according to the measured echo density, NED. (Note that in cases where the reverberation becomes quickly dense, white Gaussian noise may be used in place of the statistical pulse sequence.) This pulse sequence is then split into a set of frequency bands, and the estimated amplitude envelopes are imprinted on the pulse sequence bands before being summed to form the nominal impulse response.

To generate impulse responses of different room sizes, the same process is used, with the pulse times being scaled by the room size or with the echo density used to generate the pulse times being scaled by the inverse room size. This pulse sequence is processed as above, but with the band root energy envelopes resampled according to the room size ratio L/L_0, and preferably the envelopes modified to bring the band reverberation times in line with the desired $T_{60}(L, \omega)$ described by (10) or (9) and (6). Fig. 11

shows an impulse response resized to be twice as large through resampling and by generating a statistically similar, but stretched, pulse sequence from normalized echo density.

3.3.2. Feedback Delay Network Reverberator

Artificial reverberators are often implemented as networks of delay lines with filtering, mixing, and feedback. One such reverberator structure is the feedback delay network (FDN) [3]. The FDN reverberator employs a tapped delay line to generate the direct path and early reflections. A set of delay lines with output filtering and feedback through a unitary mixing matrix is used to produce the late-field reverberation.

Consider a FDN with N delay lines $z^{-\tau_n}, n = 1, 2, \ldots, N$ having delays τ_n and feedback filtering $g_n(z)$. The feedback filters are typically designed so that they produce similar dB attenuation per unit delay-time according to a desired decay time as a function of frequency [3]. The unitary matrix $\boldsymbol{Q}$ represents state mixing, and controls the rate of echo density increase. An identity mixing matrix $\boldsymbol{Q} = \boldsymbol{I}$ feeds each delay line to itself with no mixing between delay lines and produces a constant echo density. A Hadamard mixing matrix $\boldsymbol{Q} = \boldsymbol{H}$ generates significant mixing between delay lines, producing a rapidly increasing echo density.

To change the room size to L from a nominal L_0, the delay line lengths can be changed proportionately, as seen in Fig. 12,

$$\tau_n(L) = \frac{L}{L_0}\tau_n(L_0), \quad n = 1, 2, \ldots N\,. \tag{14}$$

Interpolated delay lines can be used to implement the desired early reflection delay times, but allpass filters are suggested to implement any fractional portion of delays used in the feedback loop so as to prevent unwanted magnitude filtering that would affect the resulting decay time.

The feedback filters $g_n(z)$ need not be modified, as the increased (or decreased) delay line lengths will result in proportionally longer (or shorter) decay times as the filters are, in effect, being applied less (or more) often. However, if desired, the feedback filters $g_n(z)$ can be modified so as to properly account for the effect of air absorption on the decay time. Additionally, note that by changing the feedback delay line lengths τ_n, the mixing matrix $\boldsymbol{Q}$ need not be modified in response to a changing room size, as the room mixing time will simply scale with the delay line lengths.

It might be the case that it is desired to leave the feedback delay lines fixed, independent of room size. In such scenarios, it is possible to change the apparent size of the room by adjusting the reverberation time and echo density profile (e.g., mixing time) by (i) modifying the feedback filters $g_n(z) \rightarrow (g_n(z))^{1/L}$, and (ii) modifying the mixing matrix $\boldsymbol{Q}$ so as to slow the state mixing, and therefore the rate of echo density increase, for larger rooms, and speed state mixing for smaller rooms as seen in Fig. 13. Fig. 14 shows the impulse response of a FDN resized by modifying the delay line lengths compared to modifying the mixing matrix and decay filters.

3.3.3. Modal Reverberator

As presented in [4], the modal reverberator implements reverberation as a parallel sum of resonant filters $h_m(t)$, each representing a room resonance or mode, and each characterized by a mode frequency ω_m, mode decay rate σ_m, and mode amplitude γ_m,

$$h(t) = \sum_{m=1}^{M} h_m(t), \tag{15}$$

where,

$$h_m(t) = \gamma_m e^{j\omega_m t - \sigma_m t}\,. \tag{16}$$

A number of options are described for implementing such filters in [4], including biquad structures, phasor filters, and heterodyning-modulation architectures.

To implement a changing room size in a modal reverberator, the mode parameters are adjusted accordingly. The mode frequen-

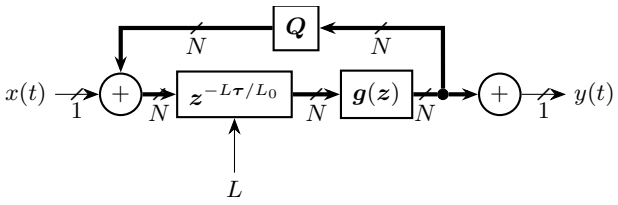


Figure 12: A delay network reverberator in which delay lengths are adjusted according to a room size control.

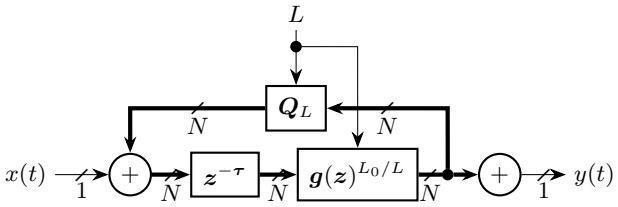


Figure 13: A delay network reverberator in which the feedback filters and mixing matrix are adjusted according to a room size control.

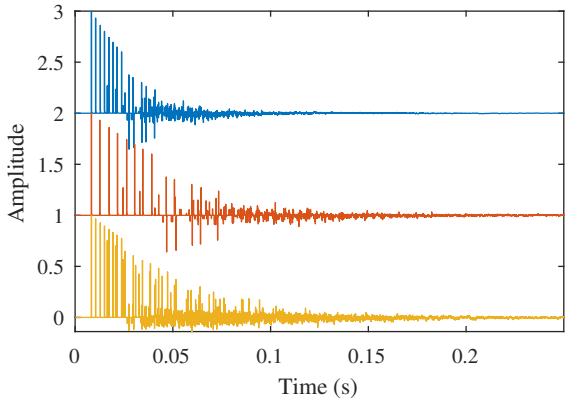


Figure 14: Time domain plots of the impulse response from a FDN showing the original IR (top), the IR stretched by a factor of 2 by modifying the delay lines (middle), and stretched by a factor of 2 by modifying the mixing matrix and decay filters (bottom). Note how the method that modifies the delay line lengths preserves the reflections exactly, just scaled by the room size parameter while modifying the mixing matrix and decay rates changes the echo pattern.

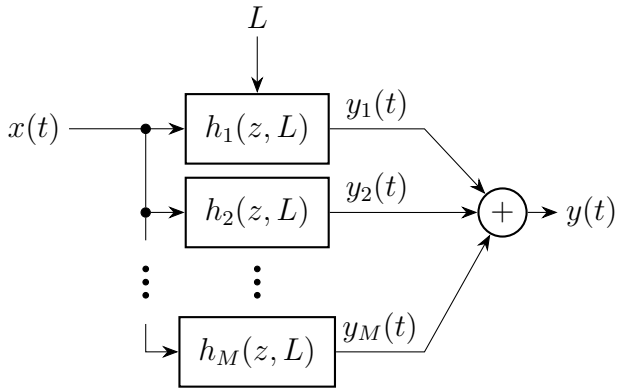


Figure 15: A modal reverberator having mode frequencies, decay times, and amplitudes modified according to a room size control.

cies would be changed in inverse proportion to the varying room size,

$$\omega_m(L) = \frac{L_0}{L}\omega_m(L_0), \quad m = 1, 2, \ldots M\,, \tag{17}$$

as seen in Fig. 15. One way to understand this is to consider a closed path among a set of reflecting surfaces that creates a resonance. If the path length were twice as long, the associated travel time would be twice as long, and the frequency reduced to half its original value.

The mode decay rates would be modified according to the scaled decay times at the new mode frequencies as described above in (6),

$$T_{60}(L, \omega_m(L)) = \frac{L}{L_0\mu(\omega_m(L)) + L\alpha(\omega_m(L))}\,, \tag{18}$$

where the decay times $T_{60}(L, \omega_m(L))$ can be found by interpolation if they are not directly available. The decay rates $\sigma_m(L)$ at room size L are then

$$\sigma_m(L) = \frac{\ln 1000}{T_{60}(L, \omega_m(L))}\,. \tag{19}$$

If the room size L is made smaller than the nominal room size L_0, then the mode frequencies will be increased. Those modes with frequencies that become larger than the Nyquist limit can be eliminated, for instance, not computed or their amplitudes reduced to zero.

If the room size L is made larger than the nominal room size L_0, then the mode frequencies will be decreased. Those modes with frequencies that become smaller than the audio band lower limit, or the lower limit of what can be reproduced with the target sound reproduction system, can be eliminated. As in the case of manipulating a convolution impulse response for changing room size, an increase in room size may significantly reduce the bandwidth of the modal reverberator response, and additional bandwidth would be preferably created. This may be done by synthesizing additional high-frequency modes, for example by statistically generating additional new high-frequency modes by extrapolating the density of mode frequencies and the decay rates from the known lower-frequency modes.

As an alternative to eliminating and synthesizing modes to accommodate a changing room size, the mode frequencies ω_m can

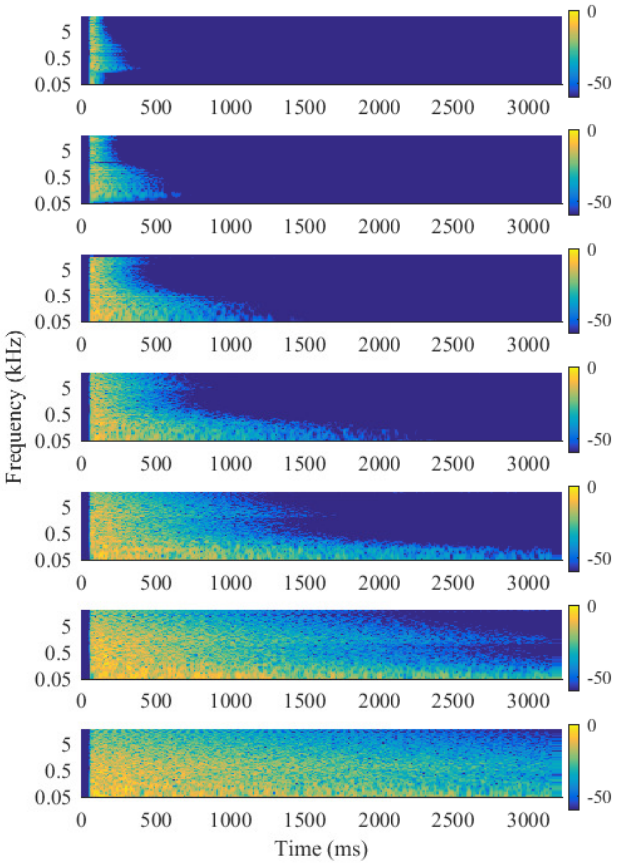


Figure 16: Spectrograms of a modal impulse response resized by factors of $1/4, 1/2, 1, 2, 4, 8$, and 16.

be warped within the audio band to generate new frequencies ν_m according to a first-order allpass characteristic,

$$e^{-j\nu_m} = \frac{\rho + e^{-j\omega_m}}{1 + \rho e^{-j\omega_m}}\,, \tag{20}$$

that is,

$$\nu_m = j \ln\left\{\frac{\rho + e^{-j\omega_m}}{1 + \rho e^{-j\omega_m}}\right\}\,. \tag{21}$$

Here, the allpass parameter ρ is chosen according to the room size ratio L/L_0 , and a little algebra gives

$$\rho = \frac{L - L_0}{L + L_0}\,. \tag{22}$$

Doing so will scale the low frequencies according to the desired linear characteristic

$$\nu_m(L) \approx \frac{L_0}{L}\omega_m(L_0), \quad |\omega_m| \ll 1\,, \tag{23}$$

with the high frequencies being warped to map the band edge ω onto the band edge ν.

Note that if it is desired to retain the original reverberation equalization, the mode amplitudes can be adjusted with room size to account for the changing equalization resulting from a changing modal density. Where the modal density is increased, the mode energy (the square of the mode magnitude) is proportionally increased. Fig. 16 shows spectrograms of the impulse response corresponding to a modal reverberator resized by various scale factors.

Finally, the circumstance in which only aspects of the room were made larger or smaller—say only a pair of walls being moved further apart—can be accommodated by having certain modes be unaffected or only modestly affected. Similarly, in the delay network reverberator structures above, only certain delay lines could be affected or others only modestly affected by a changing room size. This would be similar to changing the shape of the room.

3.4. Changing room size in real time

It may be desirable to modify the size of the room in real time. All three of the models presented here may experience undesirable pitch gliding artifacts if one were to modify the filter parameters in real time. Instead, it would be better to run multiple reverberators in parallel and cross-fade between them. In some situations, it may be beneficial to stretch the decay rates without modifying the modal frequencies to make the transitions across room size more smooth even though this is less physically accurate.

4. CONCLUSION

Here we have shown how a room size parameter can be introduced to scale the size of a virtual room in convolution, delay network, and modal reverberation algorithms. If a room is resized, the modal frequencies will be proportionally raised or lowered because of the scaling of the geometry of the space. Because resizing the room changes the surface area and volume, we must adapt the frequency dependent delay rates to account for these changes. Furthermore, we must also adapt the filtering to account for the fact that the material properties should remain unchanged. We do this by decoupling the modal frequencies and decay rates. There are clear trade offs in the complexity and sound of our various solutions, but these methods allow one to take an existing reverberant characteristic and stretch or shrink the size of the room with a physically informed method.

5. REFERENCES

[1] Vesa Välimäki, Julian D. Parker, Lauri Savioja, Julius O. Smith, and Jonathan S. Abel, "Fifty years of artificial reverberation," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 20, no. 5, pp. 1421–48, 2012.

[2] Guillermo Garcia, "Optimal filter partition for efficient convolution with short input/output delay," in *Proceedings of the 113th Audio Engineering Society Convention*, 2002.

[3] Jean-Marc Jot and Antoine Chaigne, "Digital delay networks for designing artificial reverberators," in *Proceedings of the 90th Audio Engineering Society Convention*, 1991.

[4] Jonathan S. Abel, Sean Coffin, and Kyle Spratt, "A modal architecture for artificial reverberation with application to room acoustics modeling," in *Proceedings of the 137th Audio Engineering Society Convention*, 2014.

[5] Bryan Pardo Zafar Rafii, "Learning to control a reverberator using subjective perceptual descriptors," in *Proceedings of the 10th International Society for Music Information Retrieval Conference*, 2009.

[6] Emmanouil Theofanis Chourdakis and Joshua D. Reiss, "Automatic control of a digital reverberation effect using hybrid models," in *Proceedings of the 60th International Audio Engineering Society Conference: DREAMS (Dereverberation and Reverberation of Audio, Music, and Speech)*, 2016.

[7] Jeffrey Borish, "Extension of the image model to arbitrary polyhedra," *Journal of the Acoustical Society of America*, vol. 75, no. 6, pp. 1827–36, 1984.

[8] Wallace Clement Sabine, *Collected papers on acoustics*, Peninsula Publishing, Los Alto, CA, 1993.

[9] Friedrich Spandöck, "Die Vorausbestimmung der Akustik eines Raumes mit hilfe von Modellversuchen," in *Proceedings of the 5th International Conference on Acoustics*, 1965, vol. 2, p. 313.

[10] Kyle Spratt and Jonathan S. Abel, "All natural room enhancement," in *Proceedings of the International Computer Music Conference*, 2009, pp. 231–4.

[11] Jonathan S. Abel and Patty Huang, "A simple, robust measure of reverberation echo density," in *Proceedings of the 121st Audio Engineering Society Convention*, 2006.

[12] American National Standards Institute, Committee S1, Acoustics, *Method for Calculation of the Absorption of Sound by the Atmosphere, ANSI S1.26-2009*, American National Standards Institute,, New York, NY, Sept. 1995.

[13] International Organization for Standardization, Committee ISO/TC 43, Acoustics, Sub-Committee SC 1, Noise, Acoustics, *Attenuation of sound during propagation outdoors-Part 1: Calculation of the absorption of sound by the atmosphere, ISO9613-1*, International Organization for Standardization, Geneva, Switzerland, 1993.

[14] William G. Gardner, "Efficient convolution without input-output delay," *Journal of the Audio Engineering Society*, vol. 43, no. 3, pp. 127–136, 1995.

[15] Jonathan S. Abel and Nicholas J. Bryan, "Methods for extending room impulse responses beyond their noise floor," in *Proceedings of the 129th Audio Engineering Society Convention*, 2010.

[16] Elliot K. Canfield-Dafilou and Jonathan S. Abel, "On restoring prematurely truncated sine sweep room impulse response measurements," in *Proceedings of the 20th International Conference on Digital Audio Effects*, 2017.

[17] Jonathan S. Abel, Nicholas J. Bryan, Patty P. Huang, Miriam Kolar, and Bissera V. Pentcheva, "Estimating room impulse responses from recorded balloon pops," in *Proceedings of the 129th Audio Engineering Society Convention*, 2010.

BIVIB: A MULTIMODAL PIANO SAMPLE LIBRARY OF BINAURAL SOUNDS AND KEYBOARD VIBRATIONS

Stefano Papetti
Institute for Computer Music
and Sound Technology
Zürcher Hochschule der Künste
Zurich, Switzerland
stefano.papetti@zhdk.ch

Federico Avanzini
Dipartimento di Informatica
Università di Milano
Milan, Italy
federico.avanzini@unimi.it

Federico Fontana
Dipartimento di Scienze Matematiche,
Informatiche e Fisiche
Università di Udine
Udine, Italy
federico.fontana@uniud.it

ABSTRACT

An extensive piano sample library consisting of binaural sounds and keyboard vibration signals is made available through an open-access data repository. Samples were acquired with high-quality audio and vibration measurement equipment on two Yamaha Disklavier pianos (one grand and one upright model) by means of computer-controlled playback of each key at ten different MIDI velocity values. The nominal specifications of the equipment used in the acquisition chain are reported in a companion document, allowing researchers to calculate physical quantities (e.g., acoustic pressure, vibration acceleration) from the recordings. Also, project files are provided for straightforward playback in a free software sampler available for Windows and Mac OS systems. The library is especially suited for acoustic and vibration research on the piano, as well as for research on multimodal interaction with musical instruments.

1. INTRODUCTION

The multisensory aspects of musical performance have been studied since long, particularly focusing on sound and vibration [1, 2, 3, 4], and are recognized to have a major role in the complex perception-action mechanisms involved in musical instrument playing [5]. Indeed, during instrumental performance the musician is exposed to visual, haptic (i.e., tactile and kinesthetic), and of course auditory cues. Research in this direction has substantially gained momentum in recent years, as attested by the birth of new keywords such as "musical haptics" [6].

This increased interest is partly due to the availability of novel compact, accurate, and low-cost sensors and actuators, which enable the development of complex experimental settings for measuring and delivering multisensory information in real-time on a musical instrument during the performance [7, 8, 9, 10]. On the one hand these technologies offer the possibility to investigate the perceptual role of different sensory modalities in the interaction with traditional musical instruments, while on the other they enable the design of novel digital musical interfaces and instruments in which richer feedback modalities can increase the performer's engagement, as well as the perceived quality and playability of the device [11, 12, 13, 14].

As a consequence, the availability of multimodal datasets combining and synchronizing different types of information (audio, video, MOCAP data of the instrument and the performer, physiological signals, etc.) is increasingly recognized as an essential asset for studying music performance and related aspects. Some recent examples include the "multimodal string quartet performance dataset" (QUARTET) [15], the "University of Rochester Multi-modal Music Performance dataset (URMP) [16], the "Database for Emotion Analysis using Physiological Signals" (DEAP) [17], as well as the RepoVizz initiative [18], which provides a system for storing, browsing, and visualizing synchronous multimodal data.

Within this general framework, the piano represents a relevant case study both for its prominence in the history of western musical tradition and for its potential in commercial applications (figures from the musical instrument industry[1] show a continuing growth of digital pianos and keyboard synthesizer sales).

When playing an acoustic piano, the performer is exposed to a variety of auditory, visual, somatosensory, and vibrotactile cues that combine and integrate to shape the pianist's perception-action loop. The present authors are involved in a long-term research collaboration around this topic, with particular focus on the following two aspects. The first one is the tactile feedback produced by keyboard vibrations that reach the pianist's fingers after keystrokes and holds until key release. The second one is the spatial auditory information contained in the sound field produced by the instrument at the performer's head location. For both research fields, the existing literature is scarce and provides mixed if not contradictory results about the actual perceivability and possible relevance of this multisensory information [3]. We provide extensive discussion of these aspects in previously published studies, regarding both vibration perception [14] and sound localization [19] on the acoustic piano. Moreover, a digital piano prototype was recently developed that reproduces various types of vibrations [20] – including those recorded on acoustic pianos.

As part of this research, an extensive amount of experimental data has been produced during the past years. The purpose of this paper is to present an extensive multimodal piano sample library consisting of binaural sounds and keyboard vibration signals, some of which have been used in previous works for acoustic analysis and psychophysical testing, and has now been further expanded with upright piano data and organized into a single coherent open-access dataset. Section 2 presents the main features of the library, including a description of the hardware and software recording setups, and the organization of the samples for use in a free software sampler. Section 3 discusses some key aspects involved in the usage of the library, including sample analysis, multimodal playback, and several application scenarios.

[1] https://www.namm.org/membership/global-report

2. BUILDING OF THE BiVib SAMPLE LIBRARY

The BiVib (**Bi**naural and **Vib**ratory) sample library is a collection of high-resolution audio files (.wav format, 24-bit @ 96 kHz) representing binaural piano sounds and keyboard vibrations, accompanied by project files for a free software sampler, and documentation. The dataset, whose core structure is illustrated in Tab. 1, is made available through an open-access data repository[2] and released under a Creative Commons (CC BY-NC-SA 4.0) license.

2.1. Recording procedure

The samples were recorded on two Yamaha Disklavier pianos – a grand model DC3 M4 located in Padova, Italy, and an upright model DU1A with control unit DKC-850 located in Zurich, Switzerland. Disklaviers are MIDI-compliant acoustic pianos equipped with sensors for recording keystrokes and pedaling, and electromechanical motors for playback. The grand piano is located in a large laboratory space (approximately 6×4 m), while the upright piano is in an acoustically treated small room (approximately 4×2 m).

Recordings were acquired for 10 velocity values on each of the 88 keys by means of automated software-driven procedures sending MIDI messages, as described in detail further below.

2.1.1. Hardware setup

Binaural recordings made use of dummy heads with simulated ears and ear canals mounting binaural microphones, with slightly different setups for the grand and upright pianos: a system based on the KEMAR 45BM was used in Padova (PD), and a Neumann KU 100 in Zurich (ZH). The mannequins were placed in front of the pianos at the height and distance of an average pianist (see Fig. 1). The two binaural microphones were connected to the microphone inputs of two professional audio interfaces, respectively a RME Fireface 800 (PD, gain set to $+40$ dB) and a RME UCX (ZH, gain set to $+20$ dB). The condenser capsules of the microphones were respectively fed by 26CB preamplifiers powered by a 12AL power module (PD), and powered by 48 V phantom provided by the audio interface (ZH).

Three lid configurations were adopted for each piano. The grand piano (PD) was measured with the lid completely *closed*, completely *open*, and *removed* (i.e., physically detached from the main body of the piano). The upright piano was recorded with the lid *closed*, *semi-open* (see Fig. 1), and completely *open*. The purpose of using different configurations was to gain additional insight about the possible role of the lid in modulating the sound field reaching the performer's ears and related lateralization/localization cues [19]. As a result, three sets of binaural samples were recorded for each piano.

Vibration recordings were performed with a Wilcoxon Research 736 piezoelectric accelerometer connected to a Wilcoxon Research iT100M Intelligent Transmitter, whose AC-coupled output fed a line input of a RME Fireface 800 interface and was recorded as an audio signal. The accelerometer was manually attached with double-sided adhesive tape to each key in sequence, as depicted in Fig. 2.

[2] https://doi.org/10.5281/zenodo.1213210

Figure 1: *The binaural recording setup used in Zurich. The piano lid is in 'semi-open' position*

2.1.2. Software setup

Two different software setups were used respectively for sampling sound and vibration. The same MIDI velocity values were used in both cases: 10 values between 12 and 111, evenly spaced by 11-point intervals. This choice was based on a previous study by the present authors that determined a reliable range resulting in consistent acoustic intensity [14]: in fact, the electromechanical motors of computer-controlled pianos fall short – to different extent depending on the model – of providing a consistent dynamic response, especially for the lowest and highest velocity values [21].

Binaural samples were recorded via a fully automated procedure programmed in SuperCollider.[3] The recording sessions took place overnight, thus minimizing unwanted noise from personnel working in the building. On the grand piano, note durations were determined algorithmically, based upon their dynamics and pitch – ranging from 30 s used for A0 at velocity 111, to 10 s used for C8 at velocity 12 – so as to cover their full decay while minimizing the amount of recorded data and the length of recording session (still amounting to about 6 hours each). Indeed, notes of increasing pitch and/or decreasing dynamics have shorter decay times. Unfortunately, on the upright piano an undocumented protection mechanism prevents the electromechanical system from holding down the keys longer than about 17 s, thus not allowing to fully cover the notes' decay. Therefore, for the sake of simplicity all notes were recorded for just as long as possible.

Vibration samples were recorded through a slightly less so-

[3] A programming environment for sound processing and algorithmic composition: http://supercollider.github.io/.

Table 1: *Dataset core structure. Lid configurations used for binaural recordings are reported in square brackets*

	Disklavier DC3 M4 (grand, Padova)	**Disklavier DU1A with DKC-850 (upright, Zurich)**
Sample sets (.wav files)	Binaural [closed]	Binaural [closed]
	Binaural [open]	Binaural [semi-open]
	Binaural [removed]	Binaural [open]
	Keyboard vibration	Keyboard vibration
Sampler projects (Kontakt *multis*)	Binaural [closed] + vibration	Binaural [closed] + vibration
	Binaural [open] + vibration	Binaural [semi-open] + vibration
	Binaural [removed] + vibration	Binaural [open] + vibration

Figure 2: *The vibration recording setup: A Wilcoxon Research 736 accelerometer is attached with adhesive tape to a key that is being played remotely via MIDI control*

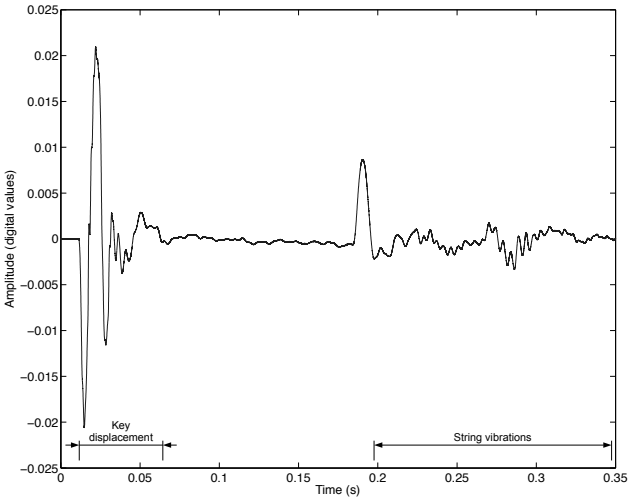


Figure 3: *Waveform of a vibration signal recorded on the grand Disklavier by playing the note A2 at MIDI velocity 12. Picture from [14]*

phisticated procedure. A DAW software was used to play back MIDI notes at the previously mentioned 10 velocity values while recording keyboard vibrations as audio signals. In this case, all notes had a fixed duration of 16 s that, considered the much weaker intensity of vibration signals as compared to sound, still allowed to describe the decay of vibration well beyond perceptual thresholds [14, 22].

2.2. Sample processing

Because of the intrinsic delay between sending MIDI messages from a computer and the mechanical actuation of the Disklavier pianos, the recorded samples started with a silent section, which we decided to remove especially in view of their use in a sampler (see 2.3). Given the large number of files (880 for each sample set), automated procedures were developed, tested and fine tuned, with the goal of removing the initial silence while leaving the rest unaffected.

Having been recorded through an accelerometer, vibration signals additionally had abrupt onsets in the attack, appearing in the first $200-250$ ms, and corresponding to the initial fly of the measured key followed by its impact with the piano keybed (see Fig. 3). As such, these onsets were not linked to sound-related vibratory cues at the keyboard, and therefore they had to be removed as well. Due the fact that onset profiles showed large variations, despite several tests made in MATLAB no reliable automated strategy could be found for editing the vibration samples. Therefore, a manual approach had to be employed instead: Files were imported in the Audacity sound editor, their waveform was zoomed in and auditioned, and the onset part was cut.

Sound recordings instead showed a more uniform shape, and an automated procedure programmed in SuperCollider was successfully used to cut the initial silence: For each sample, the program analyzes its amplitude envelope, detects the position of its largest peak, moves back by a few milliseconds, and finally applies a short fade-in.

2.3. Sampler projects and library organization

Project files are provided for use with the free 'Player' version of the software sampler Native Instruments Kontakt 5,[4] available for Windows and Mac OS systems. The full version of Kontakt 5 was instead used for developing the sampler projects. The library is organized into four folders named 'Documentation', 'Instruments',

[4] https://www.native-instruments.com/en/products/komplete/samplers/kontakt-5-player/

'Multis', and 'Samples'.

The 'Samples' folder – whose total size amounts to about 65 Gb – holds separate subfolders respectively for the binaural and vibration sample types, which in turn contain further subfolders for each sample set (see Table 1), for example 'grand-open' under the 'binaural' folder.

Independent of their type, sample files were named according to the following mask:

```
[note][octave #]_[lower MIDI velocity] ...
... _[upper MIDI velocity].wav
```

where `[note]` follows the English note-naming convention, `[octave #]` ranges from 0 to 8, `[lower MIDI velocity]` equals the MIDI velocity (range 12–111) used during recording and is the smaller velocity value mapped to that sample in Kontakt (see below), `[upper MIDI velocity]` is the greater velocity value mapped to that sample in Kontakt. For instance, a file `A4_100_110.wav` corresponds to the note A from the 4th octave (fundamental frequency $440\,\mathrm{Hz}$) recorded at MIDI velocity 100, and mapped to the velocity range 100–110 in Kontakt. Since the lowest recorded velocity value was 12, no samples were mapped to the velocity range 1–11 in Kontakt.

Following Kontakt's terminology, each of the provided *instruments* reproduces a single sample set (e.g., binaural recording of the grand piano with lid open), while each *multi* combines two *instruments* respectively reproducing one binaural and one vibration sample set belonging to the same piano. The two *instruments* in each *multi* are configured so as to receive MIDI input data on channel 1, thus playing back at once, while their respective outputs are routed to different virtual channels in Kontakt: binaural samples are routed to a pair of stereo channels (numbered 1-2), while vibration samples are played through a mono channel (numbered 3). In this way, when using audio interfaces offering more than two physical outputs, it is possible to render both binaural and vibrotactile cues at the same time by routing the audio signal respectively to headphones and vibration actuators.

In each *instrument*, sample mapping was implemented relying on the 'auto-map' feature found in the full version of Kontakt: this parses file names and uses the recognized tokens for assigning samples to e.g. a pitch and velocity range. The chosen file naming template made it straightforward to batch-import the samples.

The amplitude of the recorded signals was not altered, that is no dynamic processing or amplitude normalization was applied, and the volume of all Kontakt *instruments* was set to $0\,\mathrm{dB}$. Because of this and the adopted velocity mapping strategy, sample playback is made transparent for acoustic and vibratory analysis and experiments (see 3.1 and 3.2).

3. USING THE BiVib SAMPLE LIBRARY

The BiVib library is suited for both acoustic/vibratory analysis and interactive applications, for instance in experiments on musical performance and multisensory perception.

To our knowledge, no other existing piano datasets are fully comparable with what included with the BiVib library. Indeed, binaural piano sounds are offered by a few audio plugin developers (e.g., Modartt Pianoteq[5]) and digital piano manufacturers (e.g., Yamaha Clavinova[6]). Also, free binaural piano samples can be found, such as the "binaural upright piano" library,[7] which however offers only 3 dynamic layers as opposed to the 10 velocity levels provided by BiVib. Overall, such binaural sounds are conceived for use with virtual instruments, while they are not directly suitable for research purposes, due to non-reproducible and undocumented acquisition procedures and sample post-processing. Collections of haptic / vibrotactile data of musical instruments are even scarcer. To our knowledge, no other public dataset of piano keyboard vibrations is available.

3.1. Sample analysis

For many experimental purposes and applications it is essential to be able to reconstruct the physical values of the measured signals, that is acceleration in $\mathrm{m/s^2}$ for keyboard vibrations, and acoustic pressure in Pa for the binaural signals. Given the quality of the equipment used in the various stages of the acquisition chain, such reconstruction can be achieved with good accuracy by relying on the equipment's nominal specifications. These are summarized in a companion document included in the 'Documentation' folder.

For instance, accelerations in $\mathrm{m/s^2}$ can be computed from the acquired signals by making use of the nominal sensitivity parameters of the audio interface and the accelerometer: the digital signals, whose normalized values range between -1 and 1, are first converted to voltage values through the full scale reference of the RME Fireface 800 audio interface (for line inputs at the chosen sensitivity level, $0\,\mathrm{dBFS}$ @ $+19\,\mathrm{dBu}$, reference $0.775\,\mathrm{V}$), and then transformed into proportional acceleration values through the sensitivity constant of the Wilcoxon Research 736 accelerometer ($10.2\,\mathrm{mV/m/s^2}$). In a similar way, acoustic pressure values in Pa can be obtained from the binaural recordings, by making use of the nominal sensitivity levels of the audio interfaces' microphone inputs and of the binaural microphones.

Generally speaking, objective data computed from the library may help support results from psychophysical and quality evaluation studies focusing on the piano, as recently done by the authors in [14].

A more ambitious task could be that of extracting piano sounds free of the room response that affect the BiVib library. Methods exist to deconvolve common acoustic poles and zeros from samples that have been captured under invariant conditions [23], as it is in our case. However, in the case of BiVib care should be taken for preventing these methods from cancelling poles and zeros that are introduced by the mannequin, responsible of the binaural cues: Most such poles and zeros have frequencies higher than those associated to the dominant poles and zeros characterizing the recording rooms, in ways that at least the lower common modal resonances may be deconvolved safely from the samples. On the other hand, anechoic binaural sounds may not be suitable for the purpose of listening experiments in ecological settings.

3.2. Experiments and applications

We anticipate that this library will be useful for data analysis and experiments in music performance studies.

Acceleration values in $\mathrm{m/s^2}$ obtained from the vibration recordings as explained above can be used e.g. for comparison with the literature of touch psychophysics [22, 24], as shown in Fig. 4. In a recent article by the present authors, this allowed to

[5] https://www.pianoteq.com/

[6] https://europe.yamaha.com/en/products/musical_instruments/pianos/clavinova/

[7] https://www.michaelpichermusic.com/binaural-upright-piano

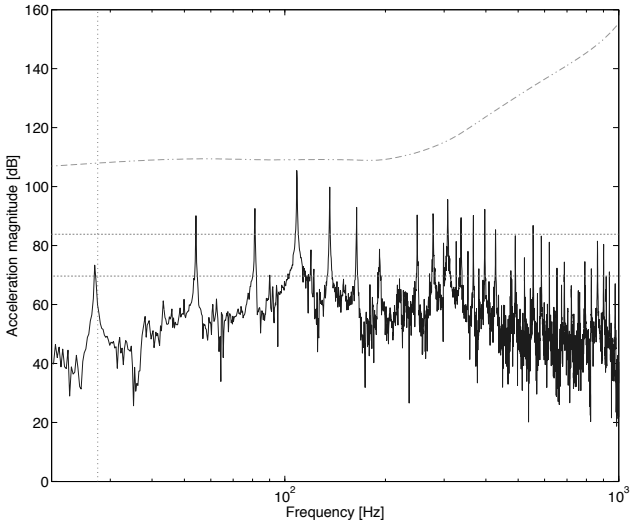


Figure 4: *Magnitude spectrum of the vibration signal at the A0 key, recorded with MIDI velocity 111 on the upright Disklavier. The dash-dotted curve depicts the reference vibrotactile threshold for passive touch [24], while the two horizontal dashed lines represent the minimum and maximum thresholds recently measured by one of the authors for active touch [22]. Picture adapted from [14]*

support the subjective results of a psychophysical experiment on the detection of vibration at the piano keyboard [14].

On a genuinely multisensory level, the relations in intensity existing between sound and vibration signals, recorded on the same instruments and provided by the database, may be used to investigate the presence of cross-modal effects occurring during piano playing. Such effects have been highlighted as part of a more general multisensory integration mechanism [25] that under certain conditions may increase the perceived intensity of auditory signals [26], or vice-versa can enhance touch perception [27]. The possibility to individually manipulate the magnitude of piano sounds and vibrations in experimental settings (e.g., using a digital keyboard that yields multimodal feedback) may lead to interesting observations on the perceptual consequence of this manipulation specifically for the pianist. In this regard, cross-modal effects resulting from varying the tactile feedback of the keyboard have been recently observed by the authors, however far from giving a systematic view about the impact of the different sensory channels to the pianist's playing experience [20].

The BiVib library has been previously used to investigate the presence of auditory lateralization cues for the acoustic piano, limited to sound samples. Although the recordings are not anechoic, their reproduction through headphones has unveiled the ability of pianists to localize tones in good accordance with the interaural level differences existing in the binaural material [28]. This ability was further supported by visual cues of self-moving keys producing the corresponding tones, as well as by somatosensory cues occurring during active piano playing of the same tones [19]. Interestingly, the supportive role of the visual and somatosensory channel ceased when the auditory feedback was subverted by swapping the left-right signals feeding the headphones. This evidence speaks in favor of the existence of a ventriloquist effect that affects piano listening and playing, which may be enabled only by a coherent multisensory experience as provided by an actuated piano [28].

One promising research direction that may also gain from using the BiVib library is represented by the use of methods from cognitive neuroscience (e.g., EEG and event-related potentials, brain imaging) to further investigate the role of multimodal audio-visuo-tactile processing in supporting musical abilities and triggering the activation of motor information in the brain of pianists.

Ultimately, all these studies can contribute to the perceptually and cognitively informed design of novel digital pianos, and to the understanding of perceived instrumental quality and playability. We provided initial results in an earlier study where we developed and tested a haptic digital piano prototype: various vibration signals, including grand piano vibrations from BiVib, were reproduced at the keyboard and compared to a non-vibrating condition [20]. Overall, vibrating condition was preferred over the standard non-vibrating setup in terms of perceived quality. However, when considering performance-related features such as timing and dynamics accuracy of performers, this initial study could not highlight significant differences between conditions.

Finally, the binaural recordings may be especially useful also for different research directions. One example in the field of music information retrieval is that of multipitch estimation and automatic transcription algorithms that exploit binaural information, whereas the datasets most commonly employed for these tasks are not binaural, such as the "MIDI Aligned Piano Sounds" (MAPS) database [29]. One further example, in the field of digital audio effects, is that of spatial enhancement effects (e.g., stereo enhancement): Piano sounds are typical examples of acoustic signals that are difficult to spatialize properly [30], and the BiVib samples may serve as a reference for the development/validation of novel effects.

4. CONCLUSIONS AND PERSPECTIVES

The BiVib sample library provides a unique set of multimodal piano data, acquired with high-quality equipment in controlled conditions through reproducible computer-controlled procedures.

Since the binaural samples in the library were meant for use in perceptual tests under ecological listening conditions, they currently include responses of the rooms where they were recorded. However we recognize that for acoustic research purposes this may be a relevant limitation, and therefore we have planned to add the respective (binaural) room impulse responses in a future version of the library, and possibly a complete new set of recordings in anechoic conditions.

We hope that the public availability of the library, in conjunction with this documentation and with the accompanying Kontakt sampler projects, will facilitate further research in the understanding and modeling of piano acoustics, performance, and related fields.

5. ACKNOWLEDGMENTS

This research was partially supported by project AHMI (Audio-haptic modalities in musical interfaces, 2014–2016), and HAPTEEV (Haptic technology and evaluation for digital musical interfaces 2018–2022), both funded by the Swiss National Science Foundation.

The Disklavier grand model DC3 M4 located in Padova was made available by virtue of the Sound and Music Processing Lab

(SaMPL), a project of the Conservatory of Padova funded by Cariparo foundation (thanks in particular to Nicola Bernardini and Giorgio Klauer).

The authors would like to thank several students and collaborators who contributed to the development of this work along the years, in chronological order: Francesco Zanini, Valerio Zanini, Andrea Ghirotto, Devid Bianco, Lorenzo Malavolta, Debora Scappin, Mattia Bernardi, Francesca Minchio, Martin Fröhlich.

6. REFERENCES

[1] K. Marshall and B. Genter, "The musician and the vibrational behavior of a violin," *J. of the Catgut Acoustical Society*, vol. 45, pp. 28–33, 1986.

[2] H. Suzuki, "Vibration and sound radiation of a piano soundboard," *J. Acoust. Soc. Am.*, vol. 80, no. 6, pp. 1573–1582, 1986.

[3] A. Askenfelt and E. V. Jansson, "On vibration sensation and finger touch in stringed instrument playing," *Music Percept.*, vol. 9, no. 3, pp. pp. 311–349, 1992.

[4] C. Saitis, "Evaluating violin quality: Player reliability and verbalization," Ph.D. dissertation, Dept. of Music Research, McGill University, Montreal, Canada, 2013.

[5] S. O'Modhrain and B. R. Gillespie, "Once more, with feeling: The dynamics of performer-instrument interaction," in *Musical Haptics*, S. Papetti and C. Saitis, Eds. Springer-Verlag, 2018, in press.

[6] S. Papetti and C. Saitis, Eds., *Musical Haptics*. Springer-Verlag, 2018, in press.

[7] S. O'Modhrain and C. Chafe, "Incorporating Haptic Feedback into Interfaces for Music Applications," in *Proc. of ISORA, World Automation Conf.*, 2000.

[8] M. T. Marshall and M. M. Wanderley, "Vibrotactile feedback in digital musical instruments," in *Proc. Int. Conf. on New Interfaces for Musical Expression (NIME)*, 2006, pp. 226–229.

[9] D. M. Birnbaum and M. M. Wanderley, "A systematic approach to musical vibrotactile feedback," in *Proc. Int. Computer Music Conf. (ICMC)*, 2007.

[10] D. Overholt, E. Berdahl, and R. Hamilton, "Advancements in actuated musical instruments," *Organised Sound*, vol. 16, no. 02, pp. 154–165, 2011.

[11] M. Keane and G. Dodd, "Subjective Assessment of Upright Piano Key Vibrations," *Acta Acust. united with Acust.*, vol. 97, no. 4, pp. 708–713, 2011.

[12] A. Galembo and A. Askenfelt, "Quality assessment of musical instruments - Effects of multimodality," in *Proc. Conf. of the European Society for the Cognitive Sciences of Music (ESCOM)*, Hannover, Germany, Sep 2003.

[13] I. Wollman, C. Fritz, and J. Poitevineau, "Influence of vibrotactile feedback on some perceptual features of violins," *J. Acoust. Soc. Am.*, vol. 136, no. 2, pp. 910–921, 2014.

[14] F. Fontana, S. Papetti, H. Järveläinen, and F. Avanzini, "Detection of keyboard vibrations and effects on perceived piano quality," *J. Acoust. Soc. Am.*, vol. 142, no. 5, pp. 2953–67, 2017.

[15] E. Maestre, P. Papiotis, M. Marchini, Q. Llimona, O. Mayor, A. Pérez, and M. M. Wanderley, "Enriched multimodal representations of music performances: Online access and visualization," *IEEE MultiMedia*, vol. 24, no. 1, pp. 24–34, 2017.

[16] B. Li, X. Liu, K. Dinesh, Z. Duan, and G. Sharma, "Creating a multi-track classical music performance dataset for multi-modal music analysis: Challenges, insights, and applications," *IEEE Trans. Multimedia*, 2018, submitted for publication.

[17] S. Koelstra, C. Muhl, M. Soleymani, J.-S. Lee, A. Yazdani, T. Ebrahimi, T. Pun, A. Nijholt, and I. Patras, "Deap: A database for emotion analysis; using physiological signals," *IEEE Trans. Affective Computing*, vol. 3, no. 1, pp. 18–31, 2012.

[18] O. Mayor, J. Llop, and E. Maestre Gómez, "Repovizz: A multi-modal on-line database and browsing tool for music performance research," in *Proc Int. Soc. for Music Information Retrieval Conf. (ISMIR 2011)*, Oct. 2011.

[19] F. Fontana, D. Scappin, F. Avanzini, M. Bernardi, D. Bianco, and G. Klauer, "Auditory, visual and somatosensory localization of piano tones: A preliminary study," in *Proc. Int. Conf. Sound and Music Computing (SMC)*, Espoo, Jul. 2017, pp. 254–260.

[20] F. Fontana, F. Avanzini, H. Järveläinen, S. Papetti, G. Klauer, and L. Malavolta, "Rendering and subjective evaluation of real vs. synthetic vibrotactile cues on a digital piano keyboard," in *Proc. Int. Conf. Sound and Music Computing (SMC)*, Maynooth, Ireland, Jul. 2015, pp. 161–167.

[21] W. Goebl and R. Bresin, "Measurement and reproduction accuracy of computer-controlled grand pianos," *J. Acoust. Soc. Am.*, vol. 114, no. 4, pp. 2273–83, 2003.

[22] S. Papetti, H. Järveläinen, B. L. Giordano, S. Schiesser, and M. Fröhlich, "Vibrotactile sensitivity in active touch: effect of pressing force," *IEEE Trans. on Haptics*, vol. 10, no. 1, pp. 113–122, Jan 2017.

[23] Y. Haneda, S. Makino, and Y. Kaneda, "Common Acoustical Pole and Zero Modeling of Room Transfer Functions," *IEEE Trans. on Speech and Audio Processing*, vol. 2, no. 2, pp. 320–328, Apr 1994.

[24] R. T. Verrillo, "Vibration sensation in humans," *Music Percept.*, vol. 9, no. 3, pp. 281–302, 1992.

[25] C. Kayser, C. I. Petkov, M. Augath, and N. K. Logothetis, "Integration of Touch and Sound in Auditory Cortex," *Neuron*, vol. 48, no. 2, pp. 373–84, oct 2005. [Online]. Available: http://www.ncbi.nlm.nih.gov/pubmed/16242415

[26] H. Gillmeister and M. Eimer, "Tactile enhancement of auditory detection and perceived loudness," *Brain Research*, vol. 1160, pp. 58 – 68, 2007. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0006899307006671

[27] T. Ro, J. Hsu, N. E. Yasar, L. C. Elmore, and M. S. Beauchamp, "Sound enhances touch perception," *Experimental Brain Research*, vol. 195, pp. 135–143, 2009.

[28] F. Fontana, F. Avanzini, and S. Papetti, "Evidence of lateralization cues in grand and upright piano sounds," in *Proc. Int. Conf. Sound and Music Computing (SMC)*, Cyprus, 2018, submitted.

[29] V. Emiya, R. Badeau, and B. David, "Multipitch estimation of piano sounds using a new probabilistic spectral smoothness principle," *IEEE Trans. Audio, Speech, and Language Processs.*, vol. 18, no. 6, pp. 1643–1654, 2010.

[30] D. Rocchesso, "Spatial effects," in *Digital Audio Effects*, U. Zölzer, Ed. Chirchester Sussex, UK: John Wiley & Sons, 2002, pp. 137–200.

[31] S. Soto-Faraco and G. Deco, "Multisensory contributions to the perception of vibrotactile events," *Behavioural Brain Research*, vol. 196, no. 2, pp. 145–154, 2009.

POSITION-BASED ATTENUATION AND AMPLIFICATION FOR STEREO MIXES

Luca Marinelli

Audio Communication Group
Technical University of Berlin
Berlin, Germany
luca.marinelli@campus.tu-berlin.de

Holger Kirchhoff

zplane.development GmbH & Co KG
Berlin, Germany
kirchhoff@zplane.de

ABSTRACT

This paper presents a position-based attenuation and amplification method suitable for source separation and enhancement. Our novel sigmoidal time-frequency mask allows us to directly control the level within a target azimuth range and to exploit a trade-off between the production of musical noise artifacts and separation quality. The algorithm is fully describable in a closed and compact analytical form. The method was evaluated on a multitrack dataset and compared to another position-based source separation algorithm. The results show that although the sigmoidal mask leads to a lower source-to-interference ratio, the overall sound quality measured by the source-to-distortion ratio and the source-to-artifacts ratio is improved.

1. INTRODUCTION

Over the past years, research on sound source separation and upmixing techniques has produced a vast body of literature. Non-negative matrix factorisation (NMF) [1], independent component analysis (ICA) [2], computational auditory scene analysis (CASA) [3] and time-frequency (TF) masking [4] appear to be the main families of blind audio source separation (BASS) methods. With regard to stereo recordings many different approaches have been proposed to model the mixing process and the nature of the sources. The derived techniques can be divided into blind or informed (guided) source separation [5].

This paper proposes a guided TF masking algorithm, assuming that the direction of the source can be approximately estimated by the user. As with other position-based source separation methods [4, 6], only the interaural intensity difference (IID) between the two channels (left and right) is taken into account to model the position of the sources. Our signal model is similar to the ones in [7] and [4] and assumes mono sources that have been positioned in the stereo image by a panorama potentiometer. Each TF bin is assumed to belong to a single source and we estimate its position as well as its mono magnitude assuming the energy-preserving panning law. Given a target azimuth range, we then compute a sigmoidal TF mask that weights the amplitudes with regard to their distance from the target azimuth range. In addition to source separation, our mask is able to perform source enhancement and attenuation with precise level indications. A binary mask as in [4] produces significant musical noise due to isolated non-zero TF bins. The sigmoidal mask has a smoother transition between the target and the adjacent azimuth ranges which reduces this kind of artifact.

In section 2 we briefly introduce our signal model, while in section 3 our method is presented in a closed analytical form. Finally, in section 4, we confirm the effectiveness of our approach.

2. FRAMEWORK

Commercial recordings are often instantaneous mixes of mono tracks combined through amplitude panning to generate a stereophonic effect [8].

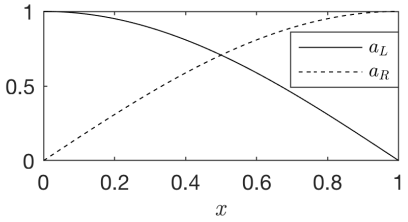


Figure 1: *Energy preserving panning coefficients*

2.1. Mixing Model

Given a set of mono sources $\{S_j\}_{j=1}^{J}$ and the relative amplitude panning gains a_j^L, a_j^R, a stereo mix can be modelled as:

$$\begin{aligned} L &= \textstyle\sum_j a_j^L S_j \\ R &= \textstyle\sum_j a_j^R S_j \end{aligned} \tag{1}$$

where L and R are the left and the right channels, respectively.

As reported in [8], the majority of analog and digital mixers approximate the *energy-preserving panning law* (Fig. 1), where the value of the panorama potentiometer takes on values $x_j \in [0, 1]$ and $(a_j^L)^2 + (a_j^R)^2 = C^2$:

$$\begin{aligned} a_j^L &= C \cdot \cos(x_j \cdot \pi/2) \\ a_j^R &= C \cdot \sin(x_j \cdot \pi/2) \end{aligned} \tag{2}$$

where $C = 1$ satisfies the energy preserving condition.

2.2. W-disjoint orthogonality

Our method is based on the W-disjoint orthogonality assumption, where two or more sources do not overlap in the short-time Fourier transform (STFT) domain. Mathematically, this condition can be expressed as:

$$S_i(k, m) \cdot S_j(k, m) = 0 \quad \forall i \neq j, \ \ \forall k, m \tag{3}$$

where $S_j(k, m)$ is the STFT of the j-th source at frame m and frequency bin k.

3. METHOD

In a first step, we estimate the panning position of each TF bin (sec. 3.1) as well as its mono magnitude (sec. 3.3). Given a target azimuth range, a sigmoidal mask is computed based on the estimated panning positions (sec. 3.2).

The sigmoidal mask is applied to the mono magnitudes which are then re-panned (sec. 3.4) and recombined with the phase from the original mixture.

3.1. Panning map

Given equation 3 and our assumptions from eq. 1 and 2, it is now possible to estimate the panning position for each element in the spectrograms:

$$x(k,m) = \arctan\left(\frac{|X_R(k,m)|}{|X_L(k,m)|}\right) \cdot 2/\pi \tag{4}$$

where X_L and X_R are the left and the right channel in the STFT domain. A similar estimation of the panning position has been used in [9].

3.2. Sigmoidal mask

The smoothness of sigmoidal functions has been proven useful in the post-processing of signal estimates coming from methods like ICA, CASA or NMF [10, 11, 12]. Those estimates can be then used to compute TF sigmoidal masks that are then applied on the original mixture. In this work we combine two sigmoids with the panning map to create a position-based mask that can control the level in a given azimuth range.

In order to attenuate or amplify the elements inside a target azimuth range, it is necessary to find a function that weights TF bins based on their estimated position. The target range is defined by its center position $T \in [0,1]$ and a width R. We define two complementary sigmoid functions that control the amount of attenuation and amplification both inside and outside the target azimuth range:

$$\begin{aligned}\sigma_L(x) &= \frac{1}{1+e^{-\beta(x-T+\frac{R}{2})}} \\ \sigma_R(x) &= \frac{1}{1+e^{+\beta(x-T-\frac{R}{2})}}\end{aligned} \tag{5}$$

In these equations, β defines the slope of the sigmoids. Choosing $\beta = \infty$ is equivalent to a binary mask as in [4], whereas lower values for β result in smoother transitions. In order to amplify the target azimuth range, we choose $\beta > 0$ and combine the two sigmoids as follows to get the sigmoidal mask:

$$M(x) = \min\left(\sigma_L(x), \sigma_R(x)\right) \tag{6}$$

For attenuation, we choose $\beta < 0$ and obtain the sigmoidal mask:

$$M(x) = \max\left(\sigma_L(x), \sigma_R(x)\right) \tag{7}$$

Finally, to control the level in decibels, one can simply rearrange one of the previous equations as follows:

$$M_{dB}(x) = 10^{(\alpha \cdot M(x) - \alpha)/20} \tag{8}$$

where $\alpha > 0$ is the desired attenuation in decibels (see Fig. 2).

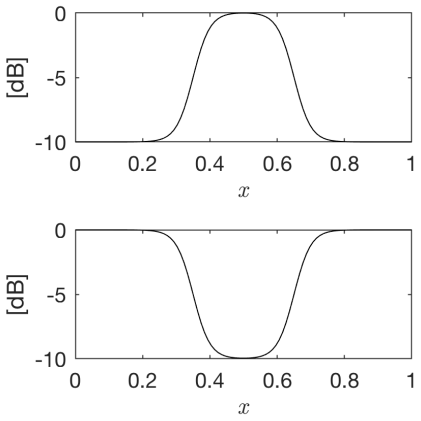


Figure 2: *Sigmoidal masks as in eq. 6 (upper) and eq. 7 (lower).* $\alpha = 10\ dB\ ,\ T = 0.5\ ,\ R = 0.3\ ,\ \beta = \pm 40$

3.3. Pre-panning magnitudes

The assumptions made in eqs. 1, 2 and 3 pose the *ideal* conditions to recover the mono magnitude of each source. Generally, the mono magnitude in the STFT domain can be computed as:

$$|S(k,m)| = \sqrt{|X_L(k,m)|^2 + |X_R(k,m)|^2} \tag{9}$$

3.4. Masking and re-panning

To synthesize the modified signal, the mono magnitudes are masked

$$|S_{out}(k,m)| = |S(k,m)| \cdot M_{dB}(x(k,m)), \tag{10}$$

and each component is re-panned to its original position.

$$\begin{aligned}|Y_L(k,m)| &= |S_{out}(k,m)| \cdot \cos(x(k,m) \cdot \pi/2) \\ |Y_R(k,m)| &= |S_{out}(k,m)| \cdot \sin(x(k,m) \cdot \pi/2)\end{aligned} \tag{11}$$

Finally, the phase from the original mixture has to be recombined:

$$\begin{aligned}Y_L(k,m) &= |Y_L(k,m)| \cdot e^{j \cdot \angle X_L(k,m)} \\ Y_R(k,m) &= |Y_R(k,m)| \cdot e^{j \cdot \angle X_R(k,m)}\end{aligned} \tag{12}$$

4. EVALUATION

4.1. Procedure

To evaluate our proposed method we use MedleyDB [13] a database of 122 royalty free multitrack recordings with a total length of 7:17 hours. The dataset provides stems (i.e. processed individual instrument tracks) for each song. For our purpose we eliminated tracks that were recorded in a live setting, due to their significant amount of spill between sources. We evaluated all tracks with a number

of stems greater than or equal to three and less than or equal to six, resulting in 43 tracks for a total of 199 sources. Due to the lack of metadata about the sources' spatial position, we created separate mixtures by downmixing the stems from stereo to mono and remixing them with random azimuth positions. The azimuth values were chosen from a uniform distribution over the whole azimuth range.

As a baseline, we compare our position-based sigmoidal source separation (PoSiS) method against the ADRess algorithm [4] which uses a different method for azimuth estimation and a binary mask instead of our proposed sigmoidal mask. We use an implementation written for the Csound system by Victor Lazzarini [14]. The ADRess algorithm was parameterized with 600 equally spaced azimuth positions and a target azimuth range of 60 azimuth positions for each source. To make our algorithm comparable, our mask was set as in equation 6 with $\beta = 30$, $R = 0.1$ and without rearranging it as in 8 to effectively emulate an attenuation of $-\infty$ dB for the TF bins outside the target range. For both algorithms, we opted for a 4096 points Hann window with 50% overlap.

To measure the quality of the separations, we used the MATLAB toolbox BSS_EVAL [15] distributed under GNU Public Licence. The computation of the criteria is performed in two steps. First, the estimated source signal is decomposed as:

$$\hat{s} = s_{\text{target}} + e_{\text{interf}} + e_{\text{noise}} + e_{\text{artif}} \tag{13}$$

where s_{target} is a modified version of the source through an allowed distortion (in this case a time invariant filter, with a 512 samples delay) and where e_{interf}, e_{noise} and e_{artif} are respectively the interference, noise and artifacts errors. From these terms, assuming no noise in our model, three numerical performance criteria are computed:

- the source-to-distortion ratio (SDR)
  that can be seen as a global quality assessment
- the source-to-artifacts ratio (SAR)
  in our case mainly related to musical noise
- the source-to-interference ratio (SIR)
  that measures the interference from other sources

4.2. Results

Figure 3 displays box plots of the measurements grouped by the number of sources present in the track.

In general, all quality measures for both algorithms show a decreasing trend when the number of sources increases, which can be attributed to the increased complexity and TF overlap of the sources when more sources are present. It can be observed that ADRess yields higher source-to-interference ratios than our proposed method, particularly when the number of sources increases. The sigmoidal mask provides a smoother transition between azimuth values inside and outside the target range and hence leads to a higher amount of contributions from other sources. On the other hand, however, PoSiS generally yields higher source-to-distortion and source-to-artifacts ratios which both capture the overall sound quality of the separated source signals. Artifacts — mainly musical noise — are reduced by the sigmoidal mask because it leads to less isolated TF bins in comparison with ADRess' binary mask.

The results suggest that the sigmoidal mask trades separation accuracy against artifacts, which can be controlled by the slope of the sigmoidal mask. With higher slopes, the mask approaches the binary mask, resulting in more artifacts and a better separation accuracy, whereas sigmoidal masks with lower slopes reduce musical noise artifacts but lead to more interference from other sources.

Δ_{SDR}	Δ_{SIR}	Δ_{SAR}
$\mu \simeq 3.3$ dB	$\mu \simeq 0.6$ dB	$\mu \simeq 3.0$ dB
$p \simeq 0.00$	$p \simeq 0.12$	$p \simeq 0.00$

Table 1: Paired difference t-test: μ is the average and p the p-value.

Assuming an underlying normal distribution of the source-wise differences of the performance measurements, where:

$$\begin{aligned} \Delta_{SDR} &= SDR_{PoSiS} - SDR_{ADRess} \\ \Delta_{SIR} &= SIR_{PoSiS} - SIR_{ADRess} \\ \Delta_{SAR} &= SAR_{PoSiS} - SAR_{ADRess} \end{aligned} \tag{14}$$

We then checked the statistical significance of our results by performing a paired t-test. With the resulting p-values in Table 1 we can, for the SDR and SAR, safely reject the null-hypothesis, while there is no statistically significant difference between the two methods in the SIR measurements.

5. CONCLUSION

We presented a system for position-based source separation from a stereo mixture. The algorithm first estimates a panning position and mono magnitude for each TF bin based on the energy-preserving panning law, assuming W-disjoint orthogonality. Given a target azimuth range, a sigmoidal mask is computed that enables attenuation and amplification of the audio within the target range. The attenuation/amplification level can be specified in dB. The mask is applied to the estimated mono magnitudes of each TF bin and the bins are re-panned to their estimated azimuth position. A resynthesis combining the magnitudes with the mixture phases yields the separated source signal.

We could confirm that using a sigmoidal mask, that is, a smoother transition between the target azimuth range and adjacent azimuth ranges, significantly reduces musical noise artifacts that occur in position-based algorithms that rely on binary masking. Binary masking often leads to isolated TF bins which cause perceptually disturbing musical noise. The sigmoidal mask smoothes the spectrogram of the separated source thereby trading musical noise artifacts against separation accuracy.

For certain use cases such as amplifying an instrument for the purpose of transcribing its musical performance, it is often not necessary to have a sharp separation and a complete suppression of interfering sources, but rather to provide a limited amplification that allows users to better listen to what has been played by the performer. In these cases an improved overall sound quality with less artifacts might be preferred.

Future work on position-based source separation will have to consider methods that do not assume W-disjoint orthogonality, which does not hold in general for professionally produced music mixtures. Even though it is possible to isolate sources under this assumption, a significant improvement in separation accuracy *and* sound quality will only be achieved if the TF contributions of each individual source can be estimated and reassigned to the corresponding source. Therefore monaural source separation methods

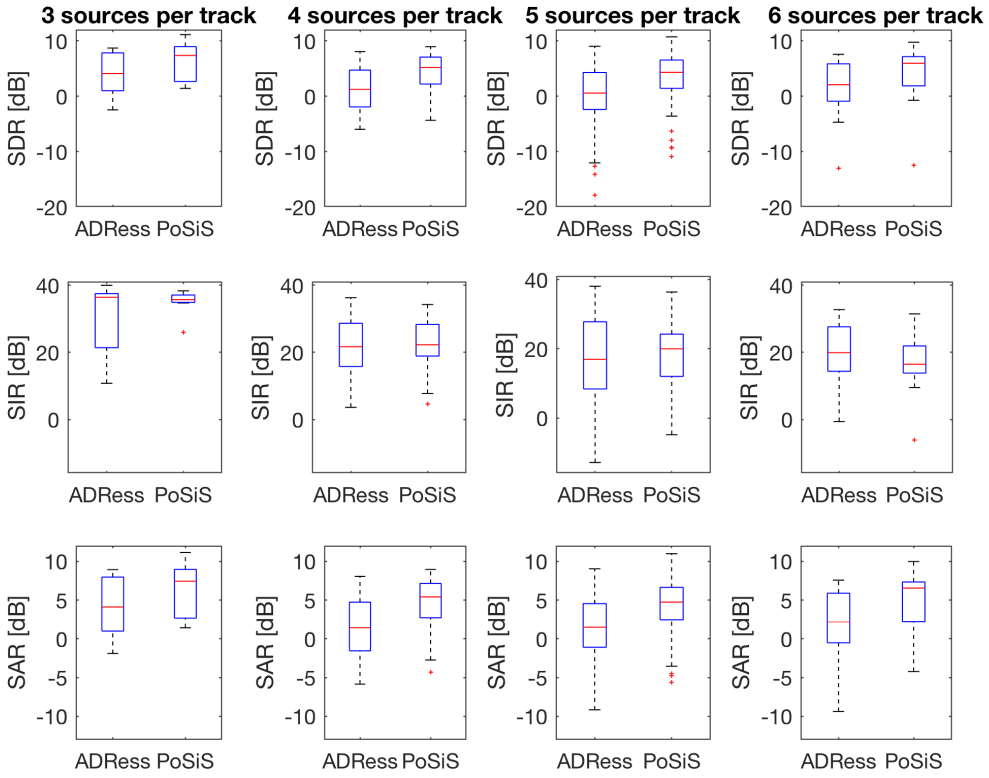


Figure 3: *SDR, SIR, SAR of ADRess and PoSiS grouped by number of sources in the mixture*

will have to be combined with position-based algorithms in order to improve sound source separation from stereo mixtures.

6. REFERENCES

[1] P. Smaragdis and J.C. Brown, "Non-negative matrix factorization for polyphonic music transcription," in *IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*, New Paltz, NY, Oct. 19-22, 2003.

[2] J.-F. Cardoso, "Blind source separation: statistical principles," in *Proceedings of the IEEE, vol. 9, no. 10*, Oct. 1998, pp. 2009–2025.

[3] D. Ellis, *Prediction-driven computational auditory scene analysis*, Ph.D. thesis, MIT, June 1996.

[4] E. Coyle D. Barry, B. Lawlor, "Sound source separation: Azimuth discrimination and resynthesis," in *7th Conference on Digital Audio Effects (DAFX 04)*, Neaples, IT, Oct. 5-8, 2004.

[5] Emmanuel Vincent, Nancy Bertin, Rémi Gribonval, and Frédéric Bimbot, "From blind to guided audio source separation: How models and side information can improve the separation of sound," *IEEE Signal Processing Magazine*, vol. 31, no. 3, pp. 107–115, May 2014.

[6] Maximo Cobos and J.Lopez, "Stereo audio source separation based on time-frequency masking and multilevel thresholding," *Digital Signal Processing*, vol. 18, no. 6, pp. 960 – 976, 2008.

[7] C. Avendano, "Frequency-domain source identification and manipulation in stereo mixes for enhancement, suppression and re-panning applications," in *2003 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*, Oct 2003, pp. 55–58.

[8] J. Bonada. A. Loscos M. Vinyes, "Demixing commercial music productions via human-assisted time-frequency masking," in *120th AES Convention*, Paris, FR, May 20-23, 2006.

[9] Y. Mitsufuji and A. Roebel, "Sound source separation based on non-negative tensor factorization incorporating spatial cue as prior knowledge," in *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, May 2013, pp. 71–75.

[10] Toby Stokes, Christopher Hummersone, Tim Brookes, and Andrew Mason, "Perceptual quality of audio separated using sigmoidal masks," in *137th Audio Engineering Society Convention 2014*, Oct. 2014.

[11] Dorothea Kolossa, Ramon Fernandez Astudillo, Eugen Hoffmann, and Reinhold Orglmeister, "Independent component analysis and time-frequency masking for speech recognition

in multitalker conditions," *EURASIP J. Audio Speech Music Process.*, vol. 2010, no. 1, Dec. 2010.

[12] Dorothea Kolossa and Reinhold Haeb-Umbach, *Robust Speech Recognition of Uncertain or Missing Data: Theory and Applications*, Springer Publishing Company, Incorporated, 1st edition, 2011.

[13] R. Bittner, J. Salamon, M. Tierney, M. Mauch, C. Cannam and J. P. Bello, "MedleyDB: A Multitrack Dataset for Annotation-Intensive MIR Research," in *15th International Society for Music Information Retrieval Conference*, Taipei, Taiwan, Oct. 2014.

[14] Victor Lazzarini, "pvsdemix - spectral azimuth-based de-mixing of stereo sources," Available at http://www.csounds.com/manualOLPC/pvsdemix.html, accessed March 19, 2018.

[15] Emmanuel Vincent, Rémi Gribonval, and Cédric Févotte, "Performance measurement in blind audio source separation," *IEEE Transactions on Audio, Speech and Language Processing*, vol. 14, no. 4, pp. 1462–1469, 2006.

DIMENSIONALITY REDUCTION FOR FEAR EMOTION DETECTION FROM SPEECH

*Safa Chebbi,**

University of Carthage
Higher School of Communication of Tunis
Research Lab. COSIM
safa.chebbi@supcom.tn

Sofia Ben Jebara

University of Carthage
Higher School of Communication of Tunis
Research Lab. COSIM
sofia.benjebara@supcom.tn

ABSTRACT

In this paper, we propose to reduce the relatively high-dimension of pitch-based features for fear emotion recognition from speech. To do so, the K-nearest neighbors algorithm has been used to classify three emotion classes: fear, neutral and 'other emotions'. Many techniques of dimensionality reduction are explored. First of all, optimal features ensuring better emotion classification are determined. Next, several families of dimensionality reduction, namely PCA, LDA and LPP, are tested in order to reveal the suitable dimension range guaranteeing the highest overall and fear recognition rates. Results show that the optimal features group permits 93.34% and 78.7% as overall and fear accuracy rates respectively. Using dimensionality reduction, Principal Component Analysis (PCA) has given the best results: 92% as overall accuracy rate and 93.3% as fear recognition percentage.

1. INTRODUCTION

Emotion is one of the main drivers of human thoughts and actions. It manifests itself through several modalities: speech, body gesture, facial expression, eyes contact,... As speech is a simple and natural way of communication, emotion recognition from speech is widely used (see for example [1][2]). In this paper, we deal with fear emotion recognition through the classification of speech into neutral, fear and other emotions. We are mainly interested in fear emotion because it has many applications. In our considered research, we aim to detect suspicious behavior which risks to be a terrorism attack, as part of civil safety. Therefore, we are particularly interested in detecting fear state which may characterize such person (before the action) in order to protect victims and limit damage [3].

In order to design a reliable emotion recognition system, the following questions should be answered: *i)* How to select appropriate features to extract from speech? *ii)*Which classification techniques to use? *iii)*How to select the most relevant and discriminatory features? and *iv)* How to reduce a high dimension feature set into a meaningful representation of reduced dimensionality? With regards to the first point, the speech production system consists of two principal organs: vocal folds, which are responsible for the production of sounds used for speech, and vocal tract related to the movement of the tongue tip, the jaw and the lip during the voice production. In our study, we are interested in studying vocal-folds related features and more precisely the pitch. Indeed, pitch expresses the vibration frequency of vocal folds during the production of voiced sounds. This choice is justified by the fact that, on the one hand, vocal folds vibrate, in a similar way, for all the phonemes unlike vocal tract, whose behaviour varies from one phoneme to another. On the other hand, the voice presents many modifications during the fear state such as oscillation, tremor, irregularity and stammering [4]. These changes are due to the vibration of vocal folds.

For the second point in the context of classification techniques, many classifiers are developped in the litterature based on machine learning approach. We quote for example Neural Network, K-nearest Neighbors, Random Forest, Decision Tree, Gaussian Mixture Model, among others [5]. In a previous work dealing with fear emotion detection [6], we performed the classification using four classifiers : Support Vector Machine (SVM) [7], Decision Tree (DT) [8], Subspace Discriminant [9] and K-nearest Neighbors (KNN) [10]. The highest fear emotion detection has been obtained using KNN. Therefore, KNN has been the classification tool of our study in this paper.

According to the third point related to discriminatory features selection, a large pool of techniques has been proposed for such purpose. We relate for example Fisher discriminant ratio, scatter matrices, statistical tests, the Receiver Operating Characteristic (ROC) curve, Bhattacharyya distance, RELIEF-F algorithm (see for example [11][12][13]). This has been the interest of our previous work for fear emotion detection [14]. In this work, four different relevance indexes have been used to select most relevant ones from a list of 27 features: Fisher Discriminant Ratio, probability divergence, scatter measure and ANOVA statistical test. Features with highest classification accuracy appearing in all relevance indexes are retained. Thanks to this approach, the fear emotion recognition results reached 86.7%.

Finally, the feature dimensionality reduction would be the objective of this paper. Indeed, when dealing with a high dimension data, classification problems become significantly harder and may lead to lower classification accuracy and poor quality of clusters. In the literature, this phenomenon is referred to as the curse of dimensionality [15]. This aspect has been a fertile field of research and development for over a century. In this context, many techniques have been proposed for this task. They are organized into two groups: linear methods such as principal component analysis [16], linear discriminant analysis [17], locality preserving projection [18], factor analysis [19], classical scaling [20] and non-linear ones including Kernel PCA [21], kernel discriminant analysis [22], Isomap [23] and multilayer autoencoders [24] among others.

The aim of this paper is the investigation of the effect of feature dimensionality reduction on classification performance. To this end, two approches have been adopted. The first one consists on performing the classification for all possible combinations of

* This work has been carried out as part of a federated research project entitled: Sensitive Supervision of Sensitive Multi-sensor Sites, funded by the Ministry of Higher Education and Scientific Research, Tunisia.

the whole pitch-based set of features using K-nearest neighbors algorithm. The approach is called 'N to N'. The goal here is to obtain the best combination of features and the suitable dimension range giving the best accuracy rate. The second one is to test many techniques for feature space reduction. They are *i)* Principal Component Analysis (PCA) and its variants (Kernel PCA denoted KPCA and probabilistic PCA denoted PPCA); *ii)* Linear Discriminant Analysis (LDA) and its kernelized version (Kernel DA denoted KDA); *iii)* Locality Preserving Projection (LPP) and its kernelized version (Projection denoted KLPP) and *iv)* many others which will be listed latter. The classification is carried out separately in the reduced space for each technique and the effect of dimension variation is analyzed. Finally, the best tradeoff between dimension reduction and classification performance is revealed.

The paper is organized as follows. Section 2 will give a brief description of the extracted pitch-based feature set, the considered emotional corpus and the emotion grouping adopted in this study. Section 3 will present a description of the 'N to N' approach as well as the classification results obtained using this process. Section 4 and 5 will provide an investigation about the use of correlation-based (resp. non-correlation based) techniques for dimensionality reduction and will display their classification results.

2. PRELIMINARIES

2.1. Features Set

The pitch is related to the vocal folds vibration, determining the periodicity of voiced sounds. More precisely, it translates the opening-closing frequency of vocal folds during the production of voiced sounds. Note that pitch is calculated only for voiced frames as vocal folds do not vibrate during the production of unvoiced sounds. In order to extract the set of pitch-based features, speech utterances are first decomposed into frames whose duration is 10ms. Next, voiced and unvoiced frames are identified and pitch values are calculated using the rapt algorithm [25]. Based on these pitch values, a whole set of global features are calculated. They are classified into four groups: usual measures, features related to pitch's derivative and second derivative as they are linked to the vibration speed and acceleration of vocal folds, features related to speech voicing since voicing rate varies from one emotion to another. The whole set of features has a 27 dimensionality. It is summarized in Table 1:

2.2. Emotional Database and Selected Emotion Classes

EMO database, which is a German emotional database publicly accessible, has been used during this study [26]. It includes 800 utterances simulated by 10 professional actors (5 males and females). It consists of seven emotion states namely: neutral, fear, anger, joy, sadness, disgust and boredom. Recordings were taken in an anechoic chamber, under supervised conditions with a sampling frequency of 48 kHz and later downsampled to 16 kHz. A human perception test to recognize various emotions with 20 participants resulted in a mean accuracy of 84.3%.
The adopted emotion grouping considers 3 groups: fear, neutral and other emotions. The 'Other emotions' class includes the five remaining states (joy, anger, disgust, sadness and boredom). The classes repartition through the corpus is the following: 14% for fear, 14% for the neutral class and 72% for other emotions.

Table 1: *Feature set.*

FAMILY	DESCRIPTION	ABREVIATION
Usual measures	Mean, Maximum, Minimum, Variance, Median, Normalised standard deviation	- Norm_STD
Speech voicing	Ratio of voiced frames on the total frames	Rat_Voic_tot
	Ratio of unvoiced frames on the total frames	Rat_UnVoic_tot
	Ratio of voiced frames on unvoiced frames	Rat_Voic_UnVoic
	First voiced frame	1st frm
	Second voiced frame	2nd frm
	Voiced frame in the middle frame	Middle frm
	Before last voiced frame	Bef_lst_frm
	Last voiced frame	Lst_frm
Pitch contour derivative	Mean of pitch's derivative	Mean_DRV
	Mean of the absolute value of pitch's derivative	Mean_ABS_DRV
	Variance of pitch's derivative	Var_DRV
	Variance of the absolute value of pitch's derivative	Var_ABS_DRV
	Maximum of pitch's derivative	Max_DRV
	Maximum of the absolute value of pitch's derivative	Max_ABS_DRV
	Mean of pitch's second derivative	Mean_Sec_DRV
	Maximum of pitch's second derivative	Max_Sec_DRV
Others	Ratio of pitch's mean on its maximum	flatness
	Ratio of pitch's mean on its minimum	Vehemence
	Ratio of peaks's number on total frames	Num_Peaks
	Minimum position	Min_Pos
	Maximum position	Max_Pos

2.3. Adopted Criteria for Evaluating the classification quality

In this study, we performed the classification using K-nearest neighbors (KNN) algorithm. KNN has been chosen according to our previous study dealing with a comparison between many classifiers [6]. This study has revealed that KNN is the best tradeoff between classification performance and computational cost. The database was trained and tested using the holdout validation method where 70% of the data were used for training while 30% were used for testing. The classification was judged by two criteria in order to have a clear idea of classification model performance:

✓**The overall accuracy rate** : it translates the percentage of well predicted emotion sequences among the total number of emotion speech sequences. It is calculated by dividing the number of well predicted samples on the total number of samples.

✓**The fear accuracy rate** : This rate indicates the proportion of fear recognition among others. It is calculated by dividing the number of well predicted fear samples on the total number of samples in fear class.

3. DIMENSIONALITY REDUCTION BASED ON 'N TO N' COMBINATION

3.1. Approach

The aim of this section is to extract the optimal feature list ensuring maximal overall emotion detection rate from the whole set of selected features. To this end, the adopted approach was to test all the possible combinations of the 27 features already extracted and to identify, as a result, the group giving the best classification accuracy. In the first iteration, we looked for the best accuracy reached by one feature. Then, we looked for the best combination of two features (2 by 2 among the 27 possible ones) giving the best accuracy. The process is re-iterated for all possible values of N (N = 1,..,27) until reaching the whole set of 27 features. This process for each value of N is called 'N to N' combination of features.

The 'N to N' dimensionality reduction technique requires laborious and complex calculation that has lasted many weeks. In-

deed, for each iteration of 'N to N' combinations among the 27 features, the classification algorithm is applied C_{27}^{N} times (where C is the combinatory operator). In order to make aware of the heaviness of computational cost, the number of combinations varies between 27 and 20 millions. The second line in Table 2 displays the total number of combinations for each 'N to N' combination of features. The run time of each 'N to N' combination is given in line 3 of Table 2 using a machine with an intel core CPU i3, 64 bits and having 1.70 GHz as a clock speed and 4 Go of RAM.

3.2. Classification Results

Figure 1 represents the evolution of the classification results for each feature vector size in terms of overall and fear accuracy rates. The solid line indicates the variation of overall accuracy rate while dashed line is reserved for fear accuracy. The first value indicates the best overall accuracy rate obtained using only one feature. The second provides the best accuracy rate obtained for the combination of 2 features among the 27 ones, and so on. Note that the best feature group has been extracted according to the overall accuracy rate optimization and not fear accuracy rate.

One can notice that the range of overall accuracy varies between

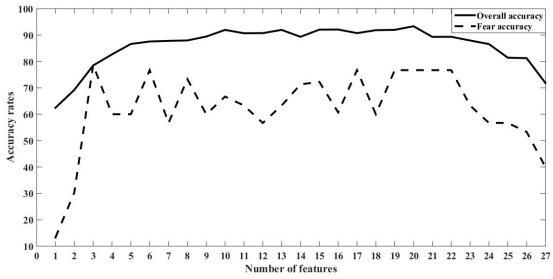


Figure 1: *Classification results according to 'N to N' approach.*

62% and 93,34%. The best value is obtained using 20 features for which the accuracy rate is equal to 93.34%. Also, we can note a stabilization at classification quality for a dimensionality between 10 and 22. On the other hand, fear recognition rate varies between 13.3% and 78.7%. The best one is obtained with a 3-features combination. Note that the quality varies enormously in the ascending and descending order for a dimension range between 3 and 19 features. Moreover, choosing a dimensionality range between 19 and 22 would be the best tradeoff between overall accuracy and fear accuracy rates. Indeed, the classification quality is among the best ones in that interval according to the two criteria.

3.3. Optimal features with reduced dimensionality

Table 3 indicates the list of relevant features giving the best accuracy rate obtained for each 'N to N' combination. The 20 features giving the best overall performance (93.34%) are: mean, median ,variance, normalised standard deviation, flatness, number of peaks, minimum, maximum, the ratio of voiced on unvoiced frames, the ratio of unvoiced frames on the total frames, mean of the absolute value of pitch's derivative, maximum position, variance of derivative, variance of the absolute value of derivative, maximum of the absolute value of derivative, mean of the second derivative, first, second, before last and last voiced frames.

However, one can notice that median, mean of second derivative, mean, last voiced frame and number of peaks are classed on the top-5 according to their presence as optimal features for the other combinations (ticked in bold in Table 3). This fact confirms their usefulness and relevance in discriminating between fear, neutral and other emotion states. If we deal with dimensionality reduction, the reduced vector size of dimensionalities varying from 19 to 22 is considered. Thus, eleven features are revealed as relevant common ones between these ranges. They are the mean, maximum, variance, Rat_Voic_UnVoic, Lst_frm, Mean_ABS_DRV, Var_ABS_DRV, Max_ABS_DRV, Mean_Sec_DRV, flatness and Num_Peaks. Thus, these features seem to be the most relevant ones.

4. CORRELATION-BASED DIMENSIONALITY REDUCTION

Whereas 'N to N' combination approach leads to very significant classification results reaching 93.3%, it remains difficult to apply them in practice because of their complexity and computational cost. Hence, the use of automatic dimension reduction techniques guaranteeing speed and performance are preferred. This section is devoted to investigate dimension reduction methods considering the correlation between features.

4.1. Correlation between features

Referring to the curse of dimensionality, dealing with a redundant and correlated features may lead to poor classification performance. In order to take an idea about the linear dependency between features, the correlation between them has been calculated pairwise and the results are displayed in Table 4. The features' names have been replaced by their corresponding number (1,2, ... 27) due to lack of space. The retained order is the same as the one adopted in Table 3. It means that 1 indicates mean, 2 indicates median and so on.

From Table 4, one can deduce that some pairs of features present strong correlation ($|\rho| > 0.7$). We relate for example the correlation between variance of derivative and mean of absolute value of derivative (0.92). Others are moderately correlated ($0.3<|\rho|<0.7$) such as variance and mean of absolute value of derivative. A good part of the features are weakly correlated. It means that they are quasi independent or totally independent ($|\rho|<0.2$). Thus, we decided to use a dimension reduction technique garanteeing features decorrelation and eliminating dependencies between them in order to obtain better classification results. The most used technique in the literature is Principal Component Analysis (PCA).

4.2. Traditional PCA and variants

PCA stills the most used technique for dimensionality reduction. It consists on using an orthogonal transformation to convert a set of possibly correlated features into uncorrelated ones called principal components. The new components of the embedded basis meet the following criteria: *(i)* they are linear combinations of the original features, *(ii)* they form an orthogonal basis that can be viewed as a rotation of the original one, and *(iii)* components are uncorrelated but preserve the maximum amount of variation in the data. In addition to traditional PCA [16], probabilistic PCA (PPCA) [27] and

Table 2: *Calculation complexity of 'N to N' combination approach.*

Features number	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**
Combinations number	27	351	2925	17550	80730	296010	888030	2220075	4686825	8436285	13037895	17383860	20058300	20058300
Overage calculation time	5.3s	223s	17.4 min	47min	23hours	18hours	3days	7days	15days	28 days	43days	43days	57days	66days
Features number	**15**	**16**	**17**	**18**	**19**	**20**	**21**	**22**	**23**	**24**	**25**	**26**	**27**	
Combinations number	17383860	13037895	8436285	4686825	2220075	888030	296010	80730	17550	2925	351	27	1	
Overage calculation time	43days	43days	28 days	15days	7days	3days	19hours	1day	56min	18min	250s	7s	12s	

Table 3: *Best feature combinations.*

Feature name \ Feature number	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**	**16**	**17**	**18**	**19**	**20**	**21**	**22**	**23**	**24**	**25**	**26**	**27**
Mean				✓	✓	✓	✓	✓	✓		✓	✓	✓	✓	✓	✓	✓		✓	✓	✓	✓	✓	✓	✓		✓
Median			✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓			✓	✓	✓	✓		✓	✓	✓	✓	✓
Maximum		✓														✓			✓	✓	✓	✓	✓		✓	✓	✓
Minimum												✓					✓			✓		✓	✓	✓	✓	✓	✓
Variance											✓	✓	✓	✓	✓	✓	✓		✓	✓	✓	✓	✓	✓		✓	✓
Norm_STD																✓		✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Rat_Voic_tot					✓		✓		✓			✓	✓	✓	✓		✓	✓	✓		✓	✓	✓	✓	✓	✓	✓
Rat_UnVoic_tot								✓				✓				✓		✓	✓	✓		✓	✓	✓	✓	✓	✓
Rat_Voic_UnVoic										✓	✓	✓	✓	✓	✓		✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
1st frm						✓	✓							✓			✓	✓	✓	✓	✓			✓	✓	✓	✓
2nd frm													✓			✓		✓	✓	✓	✓		✓	✓		✓	✓
middle frm			✓			✓							✓	✓		✓	✓	✓			✓	✓	✓	✓	✓	✓	✓
Bef_lst_frm									✓		✓	✓	✓	✓		✓	✓	✓	✓	✓	✓		✓		✓	✓	✓
Lst_frm					✓	✓		✓		✓		✓	✓	✓	✓	✓		✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Mean_DRV										✓					✓			✓			✓	✓				✓	✓
Mean_ABS_DRV								✓			✓			✓	✓				✓	✓	✓	✓	✓	✓	✓	✓	✓
Var_DRV									✓	✓	✓		✓		✓					✓	✓	✓	✓	✓	✓	✓	✓
Var_ABS_DRV									✓	✓			✓				✓		✓	✓	✓	✓	✓	✓	✓	✓	✓
Max_DRV				✓													✓	✓				✓	✓	✓	✓	✓	✓
Max_ABS_DRV								✓	✓			✓			✓	✓	✓		✓	✓	✓	✓	✓	✓	✓	✓	✓
Mean_Sec_DRV			✓		✓		✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓		✓	✓	✓	✓
Max_Sec_DRV																✓		✓			✓		✓	✓	✓	✓	✓
flatness	✓									✓				✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
vehemence		✓				✓	✓			✓	✓			✓	✓		✓	✓	✓		✓	✓		✓	✓	✓	✓
Num_Peaks							✓			✓	✓		✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Min_Pos				✓				✓	✓			✓			✓	✓	✓	✓				✓	✓	✓	✓	✓	✓
Max_Pos											✓					✓	✓			✓		✓	✓	✓	✓	✓	✓

kernel PCA (KPCA) [21] have been used for dimension reduction. KPCA is a non-linear reformulation of standard PCA. Indeed it uses a kernel trick to find principal components in a different space. In other words, it performs standard PCA in a new non-linear space. It is applicable for features presenting non-linear correlation between each other [21].

The PPCA is another formulation of standard PCA based upon a probability model [27]. The principal components are determined through maximum-likelihood estimation of parameters from the data principal components.

4.3. Classification Results

First, the embedded subspace is extracted for each technique. Then, the classification is performed with a different number of components each time. That is to say that first, the classification is performed using only the first component. Then the 2 first components are used and so on until using the whole set of componants. Hence, the suitable dimension range is the one giving the best classification performance. As for 'N to N' approach, it is judged using the overall and fear accuracy rates.

Classification results are provided in Figure 2 (resp. Figure 3) for each used technique from the PCA family in terms of overall accuracy rate (resp. fear accuracy rate). The two figures lead to the following interpretations:
✓Using traditional PCA, the best overall accuracy and fear accuracy rates reach 92% and 93.3% respectively with 19 components.

✓Using KPCA, the best overall accuracy reaches 82.7% with 6 components and the best fear accuracy reaches 86.7% with 6 components.

✓Using PPCA, the best overall accuracy and fear accuracy rates are worst. They are equal to 65.3% with 4 components and 33.3% with only one component. This approach should be discarded.

When dealing with tradeoff between accuracy and dimensionality reduction, KPCA seems to be better than PCA. In fact, the dimensionality is reduced to 6 (versus 19) with a loss of 10% for overall accuracy and 7% for fear rate. Moreover, KPCA has the advantage of presenting a stable variation of quality when dimen-

Table 4: *Correlation between features.*

Correlation	Feature pairs
0.9<|ρ|<1	(7,8);(7,9);(16,18);(17,18);(19,20)
0.8<|ρ|<=0.9	(8,9);(16,17)
0.7<|ρ|<=0.8	(1,12);(2,24);(3,22)
0.6<|ρ|<=0.7	(2,23);(3,17);(3,18);(5,6);(5,12);(5,16);(10,11);(10,15);(17,22);(18,22)
0.5<|ρ|<=0.6	(3,6);(3,16);(5,17);(5,18);(5,21);(6,16);(6,17); (6,18);(6,22);(14,15);(18,22);(23,24)
0.4<|ρ|<=0.5	(1,2);(1,5);(1,16);(1,23);(2,25);(3,5);(4,6);(16,22);(22,23);(23,25)
0.3<|ρ|<=0.4	(1,4);(1,11);(1,21);(2,5);(2,12);(2,13);(2,21);(3,23);(4,12);(5,24);(11,12);(11,16);(12,16);(12,21);(12,23); (12,24);(13,14);(13,17);(13,18);(13,24);(24,25)
0.2<|ρ|<=0.3	(1,3);(1,6);(1,7);(1,8);(1,9);(1,10);(1,13);(1,14);(1,17);(1,18);(2,4);(2,7);(2,8);(2,9);(2,10); (2,11);(2,14);(3,12);(3,21);(4,9);(4,22);(4,23);(4,25);(5,7);(5,8);(5,9);(5,11);(5,22);(6,12);(6,21); (6,23);(6,24);(8,12);(8,23);(8,24);(8,25);(9,12);(9,23);(10,12);(11,15);(11,23);(12,14);(12,21);(13,16);(13,23); (14,24);(14,27);(16,21);(16,24); (17,21);(17,23);(17,24);(17,27); (18,23);(18,27);(21,24);(22,27)
0.1<|ρ|<=0.2	(1,25);(2,16);(3,10);(3,11);(3,13);(3,14);(3,24);(3,25);(3,27);(4,5);(4,7);(4,8);(4,10);(4,11);(4,17); (4,18);(4,24);(5,10);(5,13);(5,14);(5,20);(6,7);(6,8);(6,13);(6,20);(6,25);(7,12);(7,16);(7,17);(7,18); (7,21);(7,23);(7,24);(7,25);(8,16);(8,17); (8,18);(8,21);(9,16);(9,17);(9,21);(9,24);(9,25);(10,13);(10,16); (10,17);(10,23);(10,24);(10,25);(10,26);(11,13);(11,14);(11,21);(11,24);(11,25);(11,26);(11,27);(12,13);(12,17);(12,18); (12,22);(12,25);(13,15);(13,22);(13,27);(14,16);(14,17);(14,18);(14,23);(14,25);(15,27); (16,19);(16,20);(16,23);(16,25); (16,27);(17,19);(17,20);(17,25);(18,20);(18,21);(18,24);(18,25);(20,25);(21,22);(21,23);(21,25);(22,25);(24,27)
|ρ|<=0.1	(1,15);(1,19);(1,20);(1,22);(1,24);(1,26);(1,27); (2,3);(2,6);(2,15);(2,17);(2,18);(2,19);(2,20);(2,22); (2,26);(2,27); (3,4);(3,7);(3,8);(3,9);(3,15);(3,19);(3,20);(3,26);(4,13);(4,14);(4,15);(4,16);(4,19); (4,20);(4,21);(4,26);(4,27);(5,15);(5,19);(5,23);(5,25);(5,26);(5,27);(6,9);(6,10);(6,11);(6,14);(6,15); (6,19);(6,26);(6,27);(7,10);(7,11);(7,13);(7,14);(7,15);(7,16);(7,17);(7,18);(7,19);(7,20);(7,22);(7,26); (7,27); (8,10);(8,11);(8,13);(8,14);(8,15);(8,19);(8,20);(8,22);(8,26);(8,27); (9,10);(9,11);(9,13);(9,14);(9,15); (9,18);(9,19);(9,20);(9,22);(9,26);(9,27); (10,14);(10,18);(10,19);(10,20);(10,21);(10,22);(10,27); (11,17);(11,18); (11,20);(11,19);(11,22);(11,25);(11,26);(11,27); (12,15);(12,19);(12,20);(12,26);(12,27);(13,19);(13,20);(13,21);(13,25); (13,26);(14,19);(14,20);(14,21);(14,22);(14,23);(14,25);(14,26);(15,16);(15,17);(15,18);(15,19);(15,20);(15,21);(15,22); (15,23);(15,24);(15,25);(15,26);(16,26);(17,26);(18,19);(18,26); (19,21);(19,22);(19,23);(19,24);(19,25);(19,26);(19,27); (20,21);(20,22);(20,23);(20,24);(20,26);(20,27);(22,24);(22,26);(23,26);(23,27);(24,26);(25,26);(25,27);(26,27)

sionality changes. Indeed, it appears as a horizontal line of accuracy rate for a feature number varying between 7 and 19.

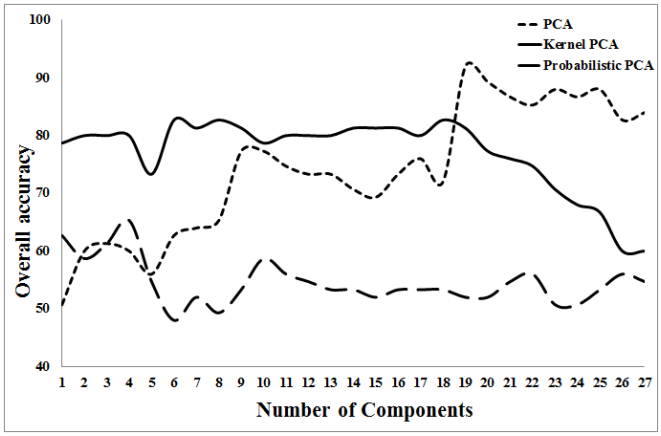


Figure 2: *Overall classification results according to PCA family techniques.*

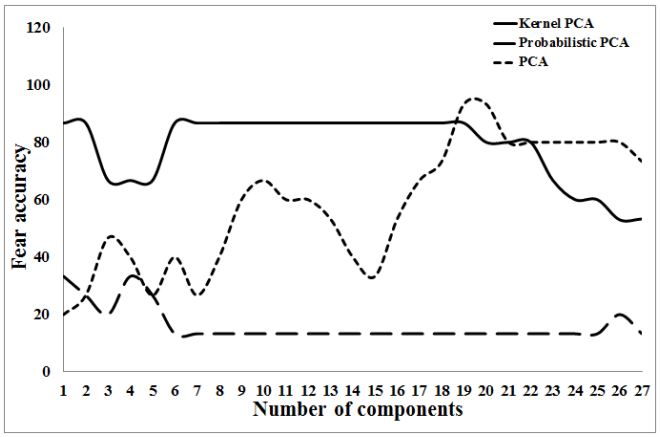


Figure 3: *Fear accuracy rates according to PCA family techniques.*

5. NON-CORRELATION BASED DIMENSIONALITY REDUCTION

5.1. Linear Discriminant Analysis Family

In contrast to most other dimensionality reduction methods, LDA is a supervised technique as it takes into consideration the class

labels when constructing the embedded feature space [17]. It attempts to find a new feature space to project the data in order to maximize classes separability. It is based on the concept of maximizing the Fisher ratio. This latter is calculated by dividing the between-class variability on the within-class variability.

Standard LDA and kernel LDA have been tested in this study in order to reduce feature space. Standard LDA attempts to maximize the linear separability between classes. It reduces dimensionality from original number of feature to C-1 features, where C is the number of classes. In our study, the new feature space will be only a 2-dimensional space as we have 3 emotion classes.

KDA is a kernelized version of LDA using the kernel trick [22]. Standard LDA is performed in a new feature space which allows non-linear mapping. Contrary to LDA, it has the advantage of allowing the variation of dimensionality from 1 to the total number of features (27 here).

Classification results are provided in Figures 4 and 5 for each used technique from the LDA family in terms of overall accuracy rate and fear accuracy rate respectively. It leads to the following results:

✓Using standard LDA, the best overall accuracy and fear accuracy rates reach 77.3% and 60% with 2 components.

✓Using KDA, the best overall accuracy reaches 80% with 9 components and the best fear accuracy rates reaches and 60% with 5 components.

Moreover, one can conclude that the LDA family seems to be not stable as the accuracy rate presents important variations when increasing the dimensionality. When dealing with trade-off between accuracy and dimensionality reduction, standard LDA seems to be better than KDA. In fact, the dimensionality is reduced to 2 (versus 9) with a loss of 3% for overall accuracy. For fear accuracy rate, they present the same accuracy rate with different dimensionality (2 for LDA versus 5 for KDA).

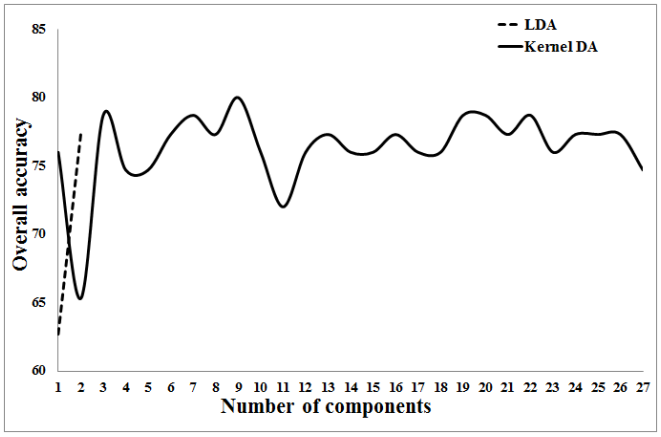


Figure 4: *Overall classification results according to LDA family.*

5.2. Locality Preserving Projection Family

LPP is an unsupervised family based on mapping the data in a low dimensional space preserving the neighborhood structure of the

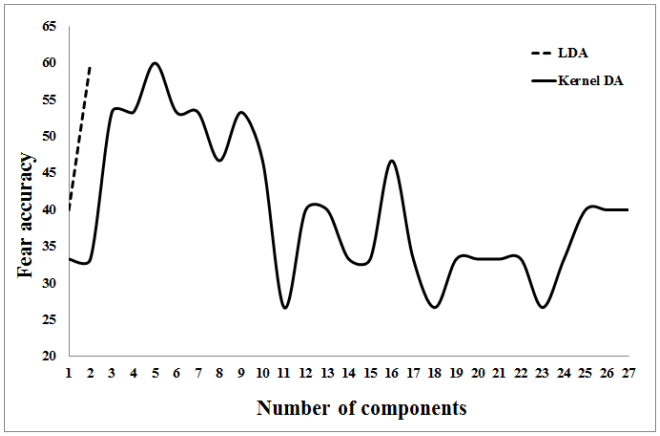


Figure 5: *Fear accuracy rates according to LDA family.*

dataset [18]. This mapping is obtained by constructing first the adjacency graph, then attempting to minimize an objective function. This latter ensures that if two data points are close in the original space, then their transformation in the embedded space are also close.

The linear property of classical LPP may lead to modeling failure when the data structure is non-linear. The basic idea of kernel LPP is to non-linearly map the data into a reduced feature space by using the non-linear structure of the features. To this end, the kernel trick is applied to extract nonlinear kernel model.

Classification results for LPP and KLPP are provided in Figures 6 and 7 and lead to the following results:

✓Using standard LPP, the best overall accuracy rate reaches 90.7% using 20 components. As for the fear accuracy, the best rate is obtained using 14 components reaching 86.7%.

✓Using KLPP, the best overall accuracy reaches 73.3% with 20 components and the best fear accuracy rate reaches 53.3% with 19 components.

One can deduce that the classification quality presents an increasing variation according to LPP as well as KLPP. Also, they both stabilize in the high dimensionality for which they reach their highest quality accuracies. Moreover, LPP seems to be better than KLPP in terms of classification performance for a fixed value of dimensionality greater than 9.

5.3. Other Techniques for dimension Reduction

In addition to the mentioned families, many other different techniques have been tested in this study namely Isomap, Landmark Isomap, Factor Analysis, Sammon Mapping, Locally Linear Embedding, Laplacian Eigenmaps, Local Tangent Space Alignment, Diffusion Maps, Stochastic Neighbor Embedding, Manifold Charting, Gaussian Process Latent Variable Model, Deep Autoencoders and Neighborhood Components Analysis. Their best classification results in terms of the overall accuracy rate and fear accuracy rate and their corresponding dimensions are summarized in Table 5. One can notice that they lead to worst results compared to previous ones.

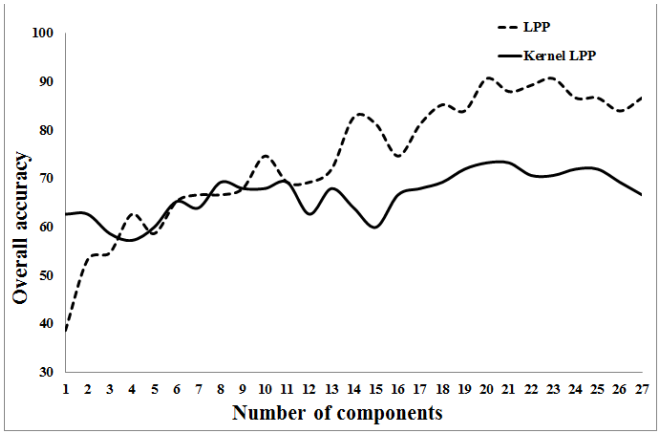


Figure 6: *Overall classification results according to LPP family techniques.*

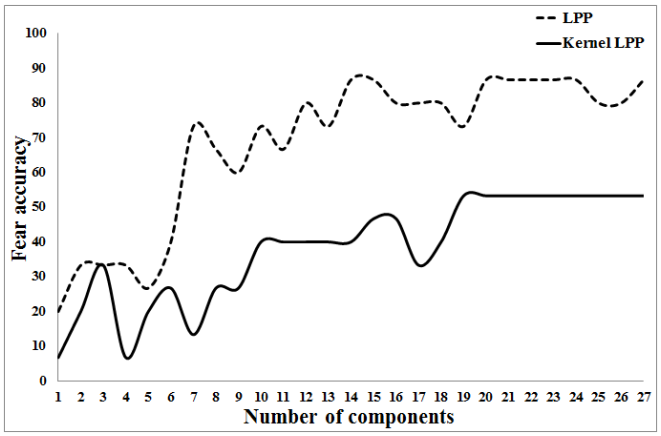


Figure 7: *Fear accuracy rates according to LPP family techniques.*

6. CONCLUSION

In this paper, we tested different techniques to reduce the relatively high dimensional feature set in order to guarantee a high overall classification rate and a high fear recognition rate. The first tested approach is manual and based on 'N to N' combinations. It leads to good results reaching 93.34% as an overall accuracy rate and 78.7% as a fear recognition rate. The other approaches are automatic. A comparative study between them was presented. It shows that the best fear recognition rate is obtained using principal components analysis reaching 93,3% using 19 components, which is practically the same result obtained for 'N to N' approach. If we aim to reduce more the dimensionality, we can use KPCA but we loose in terms of classification performance.

Table 5: *Best classification results using other dimensionality reduction techniques.*

Reduction Techniques	Overall detection		Fear detection	
	Overall Accuracy	*Dimension*	*Fear Accuracy*	*Dimension*
Isomap	45	2	40	3
Landmark Isomap	50	3	55	10
Factor Analysis	52	10	40	12
Sammon Mapping	54.7	22	53.3	20
Locally Linear Embedding	54.3	25	47.3	24
Laplacian Eigenmaps	49.3	25	46.3	22
Local Tangent Space Alignment	44.3	12	33.3	20
Diffusion Maps	40.3	6	33.3	8
Stochastic Neighbor Embedding	52.4	24	33.3	23
Deep Autoencoders	44.3	12	39.7	16
Neighborhood Components Analysis	53.3	22	55.7	24

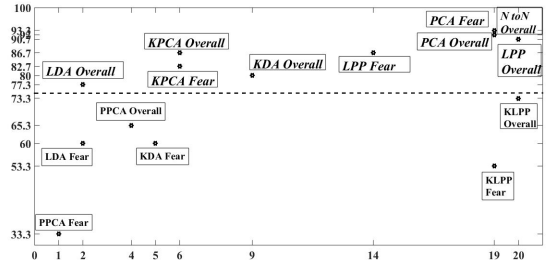


Figure 8: *Classification results according to all dimensionality reduction families and 'N to N' approach.*

7. REFERENCES

[1] Rupayan Chakraborty, Meghna Pandharipande, and Sunil Kumar Kopparapu, *Analyzing Emotion in Spontaneous Speech*, Springer, 2018.

[2] Moataz El Ayadi, Mohamed S Kamel, and Fakhri Karray, "Survey on speech emotion recognition: Features, classification schemes, and databases," *Pattern Recognition*, vol. 44, no. 3, pp. 572–587, 2011.

[3] Paul K Davis, Walter L Perry, Ryan Andrew Brown, Douglas Yeung, Parisa Roshan, and Phoenix Voorhies, *Using behavioral indicators to help detect potential violent acts*, RAND Corporation, 2013.

[4] Beatrice De Gelder, Jeffrey S Morris, and Raymond J Dolan, "Unconscious fear influences emotional awareness of faces and voices," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 102, no. 51, pp. 18682–18687, 2005.

[5] Yuichiro Anzai, *Pattern recognition and machine learning*, Elsevier, 2012.

[6] S. Chebbi and S. Ben Jebara, "On the use of pitch-based features for fear emotion detection from speech," in *4th International Conference on Advanced Technologies for Signal and Image Processing (ATSIP)*, 2018.

[7] Johan AK Suykens and Joos Vandewalle, "Least squares sup-

port vector machine classifiers," *Neural processing letters*, vol. 9, no. 3, pp. 293–300, 1999.

[8] S Rasoul Safavian and David Landgrebe, "A survey of decision tree classifier methodology," *IEEE transactions on systems, man, and cybernetics*, vol. 21, no. 3, pp. 660–674, 1991.

[9] Gaston Baudat and Fatiha Anouar, "Generalized discriminant analysis using a kernel approach," *Neural computation*, vol. 12, no. 10, pp. 2385–2404, 2000.

[10] Padraig Cunningham and Sarah Jane Delany, "k-nearest neighbour classifiers," *Multiple Classifier Systems*, vol. 34, pp. 1–17, 2007.

[11] Andrew R Webb, *Statistical pattern recognition*, John Wiley & Sons, 2003.

[12] Sergios Theodoridis, Aggelos Pikrakis, Konstantinos Koutroumbas, and Dionisis Cavouras, *Introduction to pattern recognition: a matlab approach*, Academic Press, 2010.

[13] Igor Kononenko, "Estimating attributes: analysis and extensions of relief," in *European conference on machine learning*. Springer, 1994, pp. 171–182.

[14] S. Chebbi and S. Ben Jebara, "On the selection of relevant features for fear emotion detection from speech," in *submitted in European Signal Processing Conference (EUSIPCO)*, 2018.

[15] Luis O Jimenez and David A Landgrebe, "Supervised classification in high-dimensional space: geometrical, statistical, and asymptotical properties of multivariate data," *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, vol. 28, no. 1, pp. 39–54, 1998.

[16] Svante Wold, Kim Esbensen, and Paul Geladi, "Principal component analysis," *Chemometrics and intelligent laboratory systems*, vol. 2, no. 1-3, pp. 37–52, 1987.

[17] Petros Xanthopoulos, Panos M Pardalos, and Theodore B Trafalis, "Linear discriminant analysis," in *Robust data mining*, pp. 27–33. Springer, 2013.

[18] Xiaofei He and Partha Niyogi, "Locality preserving projections," in *Advances in neural information processing systems*, 2004, pp. 153–160.

[19] Bruce Thompson, *Exploratory and confirmatory factor analysis: Understanding concepts and applications.*, American Psychological Association, 2004.

[20] Warren S Torgerson, "Multidimensional scaling: I. theory and method," *Psychometrika*, vol. 17, no. 4, pp. 401–419, 1952.

[21] Bernhard Schölkopf, Alexander Smola, and Klaus-Robert Müller, "Nonlinear component analysis as a kernel eigenvalue problem," *Neural computation*, vol. 10, no. 5, pp. 1299–1319, 1998.

[22] Sebastian Mika, Gunnar Ratsch, Jason Weston, Bernhard Scholkopf, and Klaus-Robert Mullers, "Fisher discriminant analysis with kernels," in *Neural networks for signal processing IX, 1999. Proceedings of the 1999 IEEE signal processing society workshop.* Ieee, 1999, pp. 41–48.

[23] Joshua B Tenenbaum, Vin De Silva, and John C Langford, "A global geometric framework for nonlinear dimensionality reduction," *science*, vol. 290, no. 5500, pp. 2319–2323, 2000.

[24] David DeMers and Garrison W Cottrell, "Non-linear dimensionality reduction," in *Advances in neural information processing systems*, 1993, pp. 580–587.

[25] David Talkin, "A robust algorithm for pitch tracking (rapt)," *Speech coding and synthesis*, vol. 495, pp. 518, 1995.

[26] Felix Burkhardt, Astrid Paeschke, Miriam Rolfes, Walter F Sendlmeier, and Benjamin Weiss, "A database of german emotional speech," in *9th European Conference on Speech Communication and Technology*, 2005.

[27] Sam T Roweis, "Em algorithms for pca and spca," in *Advances in neural information processing systems*, 1998, pp. 626–632.

IMMERSIVE AUDIO-GUIDING

Nuno Carriço
DETI
University of Aveiro
Portugal
carrico@ua.pt

Guilherme Campos
DETI / IEETA
University of Aveiro
Portugal
guilherme.campos@ua.pt

José Vieira
DETI / Institute of Telecommunications
University of Aveiro
Portugal
jnvieira@ua.pt

ABSTRACT

An audio-guide prototype was developed which makes it possible to associate virtual sound sources to tourist route focal points. An augmented reality effect is created, as the (virtual) audio content presented through headphones seems to originate from the specified (real) points.

A route management application allows specification of source positions (GPS coordinates), audio content (monophonic files) and route points where playback should be triggered.

The binaural spatialisation effects depend on user pose relative to the focal points: position is detected by a GPS receiver; for head-tracking, an IMU is attached to the headphone strap. The main application, developed in C++, streams the audio content through a real-time auralisation engine. HRTF filters are selected according to the azimuth and elevation of the path from the virtual source, continuously updated based on user pose.

Preliminary tests carried out with ten subjects confirmed the ability to provide the desired audio spatialisation effects and identified position detection accuracy as the main aspect to be improved in the future.

1. PROJECT MOTIVATION

Tourism and its economic impact have been growing markedly in recent decades [1][2]. The importance of enriching the visitor experience, promoting cultural tourism and adopting differentiation strategies are widely acknowledged [3], as well as the key role played in those efforts by digital information and communication technologies (ICT) [4][5].

Audio guides are increasingly popular in tourism applications (e.g. in museums, parks, historic sites and cities), both in- and outdoors. A variety of systems are commercially available. Some are intended as aids to improve intelligibility by avoiding noise and interference (especially important in heritage sites under intense visitor pressure) in otherwise conventional guided tours [6][7][9]. Others are designed to operate autonomously (i.e. without live human guiding), delivering pre-recorded (often multilingual) interpretation content [6][7][8][10][11][12][13]. The diagram in Figure 1 covers both cases. Autonomous systems can be triggered manually by the user [10] or automatically based on route sensing (GPS, infra-red and radio-frequency ID sensors being among the most common).

For example, 'hop-on hop-off' urban tour buses, now commonplace even in middle-sized cities, are invariably equipped with audio-guiding systems.

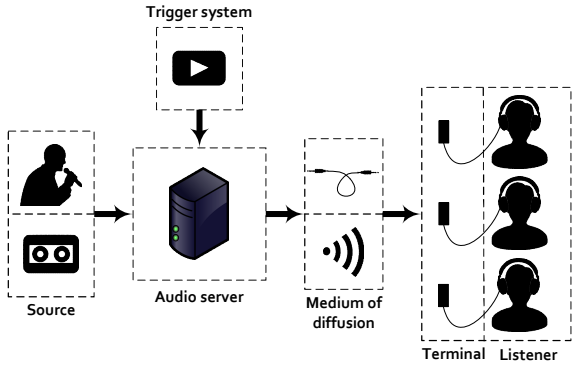


Figure 1: *Typical audio guiding system architecture*

Typically, operation is autonomous, with pre-recorded audio contents triggered at certain positions detected by GPS along the bus route; visitors are given a pair of disposable headphones (relatively 'low-fi' and uncomfortable) to be plugged into audio terminal units placed by each seat, as illustrated in Figure 2.

Figure 2: *Bus audio guide unit with language selection*

This project aims at radically improving the visitor experience provided by this kind of systems, making it as immersive as possible. The idea is to create binaural audio augmented/mixed reality (AR/MR) effects by using geo-location and applying auralisation and source spatialisation techniques. While not new, these techniques have been explored mainly in the context of computer games. As these are increasingly geared towards mobile devices, AR and MR gain ground over VR (see, for example, [14]) and geo-location becomes an essential feature. Geo-located spatial audio systems have been proposed for various applications, including artistic soundscaping (e.g. the *SoundDelta* system [15]) and guidance systems for the visually impaired (e.g. the *NAVIG* system [16]). The applicability to tour guiding

is obvious [13][17]. However, to the best of the authors' knowledge, there are no widespread commercial audio-guide models based on geo-location and incorporating spatialisation capabilities. The *USOMO* system [18] features binaural spatialisation, but is restricted to indoor usage.

2. SYSTEM OVERVIEW

A prototype was developed to address outdoor situations, taking the urban bus tour example mentioned above as the reference scenario. The goal is to turn focal points specified along the route (e.g. buildings, statues, trees…) into virtual sound sources, so that the interpretation content, delivered through headphones, be perceived by the visitor as originating from those focal points. This requires pre-recording appropriate content for each focal point, and processing this audio content in real time through filters capable of imprinting appropriate 3D directional cues according to listener pose (position and head orientation) relative to the corresponding source. Playback should be triggered when the vehicle enters route segments specified in the vicinity of the virtual source locations, as illustrated in Figure 3.

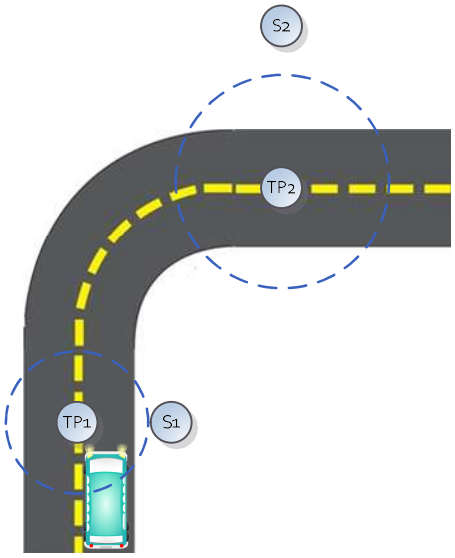


Figure 3: *Virtual audio source (S) locations and corresponding trigger point (TP) regions along a route*

Figure 4 represents the overall structure designed to achieve this goal. Its core element (***playback*** block) relies on an auralisation engine, as the binaural spatialisation effect is obtained by convolving the anechoic input sound with head-related transfer function (HRTF) filter pairs (to generate left and right channel output). The filter pair applied at a given moment must be selected (from an HRTF database) according to the azimuth and elevation of the virtual source relative to the listener. For real-time operation, this information (and thus the HRTF filter pair) must be continuously updated based on:

- Listener position – given by a GPS receiver (***GPS*** block);
- Listener head orientation – detected by an inertial head-tracking device attached to the headphone strap (***IMU*** block);
- Source position – specified at the route definition stage (***route manager*** block).

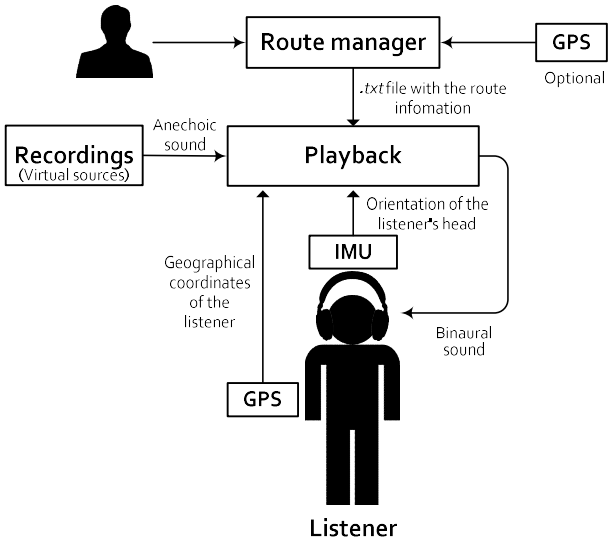


Figure 4: *System block diagram*

The following sections describe the implementation (based on C++ programming) and integration of these four blocks on a *Windows* environment.

3. PLAYBACK

3.1. Audio streaming and auralisation

The playback system was implemented with the help of the *PortAudio* [19] open-source library. As shown in Figure 5, it takes its input (44100Hz recordings of the virtual sound sources) from local memory files in 16-bit raw audio format and streams it through a real-time auralisation engine to generate output for binaural (i.e. headphone or earphone) presentation.

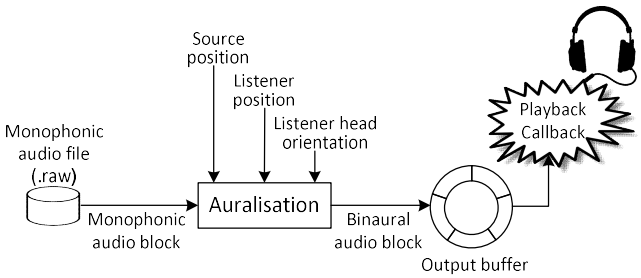


Figure 5: *Audio streaming through auralisation engine*

The auralisation engine was implemented using *LibAAVE*, a publicly available auralisation library [20] developed in a previous IEETA research project [21]. Its basic operation principles, described in [22], are illustrated in Figure 6.

.

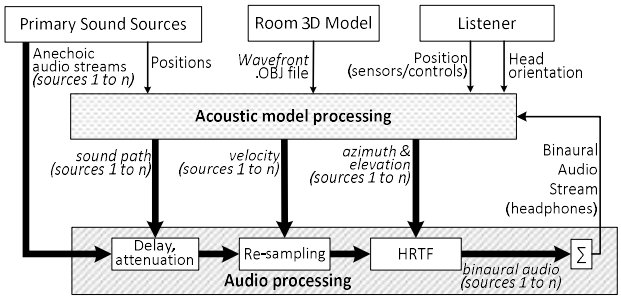


Figure 6: *LibAAVE operation structure* [21]

LibAAVE incorporates room acoustic modelling based on the mirror-image source (MIS) method. From the input data on 3D room configuration, primary source positions and listener position, the acoustic model works out the propagation paths reaching the listener considering wall reflections up to a user-defined order (this must be set low enough to allow real-time operation). The direction (azimuth and elevation) of each path relative to the listener head is also calculated considering the input information on head orientation (pitch, yaw and roll angles).

The audio processing block can then determine the appropriate delay, attenuation and HRTF filtering to be applied to the audio component transmitted through each path and generate the resulting binaural output by adding together all those contributions. Different HRTF sets can be selected, taken from public-domain databases, namely the KEMAR-based MIT Media-Lab set [23] and CIPIC [24]. The system allows arbitrary movement of both sources and listener. Cross-fading between successive audio output blocks is applied to avoid audible HRTF transition glitches.

Only a fraction of *LibAAVE*'s capabilities are utilised in the outdoor scenario explore here, as it does not involve a room model – the engine is configured to process only direct sound (no reflections). Also, a single primary source is considered at a time. Under these conditions, real-time operation is comfortably achieved. In a future extension to indoor scenarios, *LibAAVE* could be configured to take into account the acoustic influence of the room – albeit through a simplified model – without compromising real-time operation.

3.2. Playback control

Two playback trigger modes were defined. In both, audio tracks, once triggered, are played through without interruption, regardless of listener position. However, while in mode 1 tracks can be played only once along a route (i.e. are never re-triggered), in mode 2 they will be replayed if the listener re-enters the respective trigger region.

A program thread is constantly checking the current listener position, received from the GPS block, against the route information to detect if the listener has entered the trigger region of a playable virtual source. In that case, streaming is activated; each time the playback thread extracts an audio block from the output circular buffer, the auralisation engine processes a new one to refill it.

The number of samples per audio block and the size of the output buffer are configurable. To minimise latency, it is desirable to keep them as low as possible.

4. POSITION DETECTION (GPS)

The Global Positioning System (GPS) block is responsible for tracking listener position (amounting to bus position in the reference scenario) and continuously feeding the *playback* block with updated values of latitude and longitude – GPS measured altitude is not taken into account in this application. The chosen GPS receiver was a XUCAI *GD75* USB dongle – see Figure 7. Its main characteristics are listed in Table 1. Data is sent from the GPS dongle to the laptop in ASCII format using RS232 emulation.

Figure 7: *GPS receiver for position detection*

Table 1: *GPS receiver features*

Interface	USB
Communication protocol	NMEA 0183 (V3.0)
Maximum refresh rate	1Hz
Cold start time	<33s
Operating Temperature	-10º C a 70º C
Maximum error	5m (approx.)

To ensure correct integration, a simple C++ application was developed to test the device by displaying the received GPS position data. In addition to latitude, longitude and altitude, the application also extracted the number of satellites used by the receiver, since it is available from the same $GPGGA frames, constitutes an indicator of position measurement accuracy and may prove useful in scenarios to be explored in the future (e.g. transition to indoor situations).

5. HEAD-TRACKING (IMU)

For a given source position, sound perception depends not only on listener position but also on head orientation. This is normally specified by three rotation components:

- **Yaw**: around the vertical axis;
- **Pitch**: around the lateral (left-right) axis;
- **Roll**: around the longitudinal (back-front) axis.

If a virtual sound scene is to be recreated over headphones, head movements must be tracked and compensated for in real time. It is therefore necessary to use a head-tracking device capable of providing real-time pitch, yaw and roll angle data to the *playback* block. An inertial measurement unit (IMU) attached to the headphone strap is possibly the most appropriate choice for this purpose. An Intersense *InertiaCube3* unit was employed – see Figure 8. Its main characteristics are listed in Table 2.

Figure 8: *IMU for head-tracking*

Table 2: *IMU features*

Interface	USB
Latency	4 ms (via USB)
Maximum refresh rate	180Hz
Degrees of freedom	3 axis (Yaw, Pitch, and Roll)
Angular range	360º (all axis)
Precision	Yaw: 1º; Pitch and Roll: 0.25º (at the temperature of 25º C)
Maximum angular speed	1200 º per second

A software development kit is available to assist programmers using this device and provide examples regarding its operation, configuration and data acquisition.

To ensure correct integration, the IMU sensor was also tested with the help of a simple C++ application which displayed the received yaw, pitch, and roll values.

6. ROUTE MANAGER

A practical means of defining and configuring tourist routes is indispensable for efficient system operation. An application – whose user interface is presented in Figure 9 – was developed for this purpose. It allows the specification of a set of virtual sources, individually characterised in terms of (area 2 of Figure 9):

- Location (latitude and longitude);
- Height relative to a listener at the trigger region;
- Trigger region: centre point location (latitude and longitude) and radius;
- Corresponding anechoic audio file name.

This information is stored in a 'route file' (area 3 of Figure 9) under a very simple format (one text line per source) which is then passed to the *playback* block.

The latitude and longitude coordinates for the source and the trigger region centre can be entered manually (area 1 of Figure 9) but, as illustrated in Figure 4, there is also the option of acquiring them in-situ with the help of the GPS receiver.

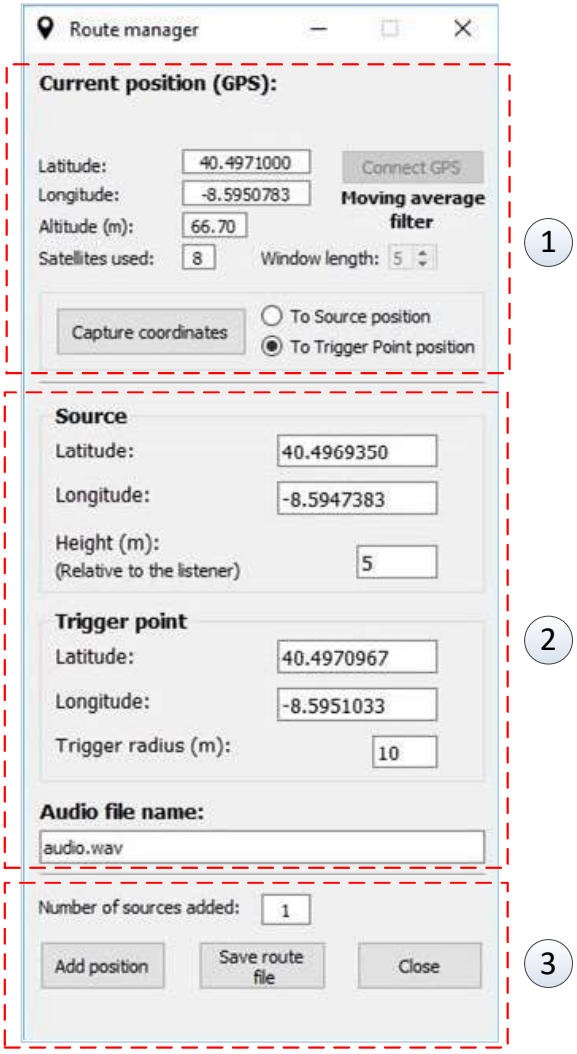


Figure 9: *Graphical user interface of the route manager*

7. VALIDATION

7.1. Test design and preparation

In order to obtain a preliminary assessment of system operation, a set of subjective tests was prepared on a short walking route with three virtual sound sources defined within the campus of the University of Aveiro, as depicted in Figure 10. Source locations are designated by 'S'; their corresponding trigger regions (interior of the dashed circles, centred at points TP) are shown to scale. Table 3 lists the audio files used (44.1kHz, 16-*bit* mono speech recordings regarding the chosen campus locations). Audio streaming (recall Figure 5 and section 3.2) was set for 1024-sample blocks and a 5-block output buffer. This choice of settings had seemed to ensure smooth audio playback and avoid any noticeable latency effects.

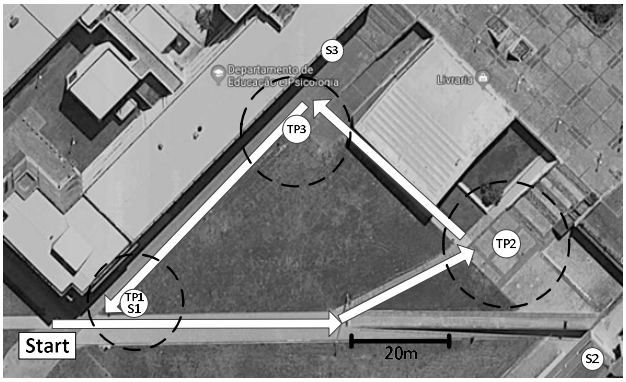


Figure 10: *Walking route for preliminary tests*

Table 3: Test route sources

Source	Audio file	Duration (s)	Visual Cue	Height (m)
S1	*Welcome_speech.wav*	38	Stone	0
S2	*Library.wav*	45	Corner	10
S3	*Media_Centre.wav*	19	Window	5

The definition and configuration of this test route was itself an opportunity to validate an important part of the system – the route manager application described in the previous section.

A walking route was preferred to a driving route (the system's reference usage scenario) because it simplified the logistics of the tests, seemingly without compromising their quality. In fact, as they involve shorter distances and less predictable user trajectories, walking routes would appear much more demanding in terms of position detection accuracy and precision.

Ten randomly chosen subjects (6 males and 4 females in the 20-35 age range, with no reported hearing problems) were invited to walk the route wearing the system. Figure 11 presents the equipment carried by the test subjects:

1. *Head-tracker* (Intersense *InertiaCube 3*).

2. *GPS receiver* (XUCAI *GD75* USB dongle).

3. *Headphones* (Sony *MDR-ZX110*).

4. *Processing unit* (laptop).

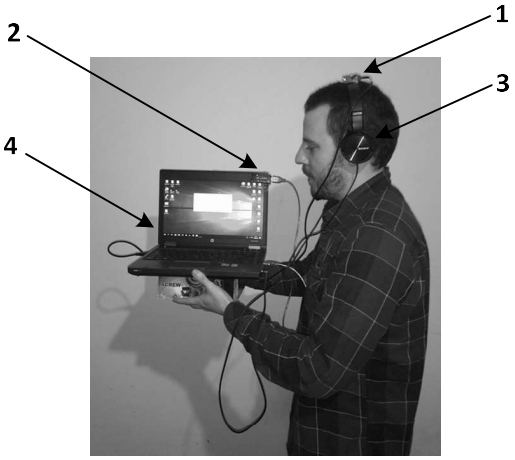


Figure 11: *Test equipment*

The subjects were briefed on the purposes and design of the tests and informed on the characteristics of the route: chosen source focal points (see Table 3), radius specified for each trigger region (respectively 10, 12 and 15m) and respective centre point locations.

7.2. Test execution and results

The first set of tests were carried out using trigger mode 1 (no re-triggering – recall trigger modes described in 3.2). The subjects were asked to use a three-point discrete scale [from 1 (bad) to 3 (good)] to rate the experience regarding triggering (Q1: 'does sound start at a seemingly correct distance?') and spatialisation (Q2: 'does sound appear to originate from the correct direction?'). The assessment – see Table 4 – was clearly positive in both regards for S2 and S3 and also positive for S1 regarding Q1, with no 'bad' ratings from any subject. However, the spatial effect of S1 was rated quite poorly; none of the subjects rated it 'good' and the majority considered it 'bad'.

Table 4: User ratings (first test set)

	Q1 – triggering		Q2 – spatialisation	
	Mean	**Std. Dev.**	**Mean**	**Std. Dev.**
S1	2.4	0.52	1.4	0.52
S2	2.7	0.48	2.9	0.32
S3	2.6	0.52	2.7	0.48

These bad results for S1 are not surprising, since shorter distances between source and listener are expected to amplify the ill effects of imprecise position detection. Unlike sources S2 and S3, placed well outside their respective trigger regions (see Figure 10), S1 was deliberately located at the centre of its trigger region to expose this effect. The spatialisation effect was very noticeably disrupted by the instability of GPS position readings (error up to 5m – see Table 1), causing abrupt changes in perceived source location. As expected, results for Q2 in S1 improved (from 1.4 to 2.4) when the subjects were asked to stop and make their assessment as soon as playback started (i.e. at the edge of the trigger region).

Under trigger mode 2, additional tests were conducted with the listeners asked to stand still for one minute inside the TP2 circle (15m radius) after the end of playback of source S2 in two situations: 1) more than 5m away from the trigger region limit and 2) less than 5m away from the trigger region limit. Obviously, playback re-triggering is not supposed to occur in either of them. However, it did in the second, again highlighting GPS position measurement errors. In this instance, they cause the listener to be occasionally detected outside the trigger region and subsequent position readings inside it are of course interpreted as a re-entry. In the first situation, re-triggering was never observed.

8. DISCUSSION AND FUTURE WORK

Whilst confirming the ability to provide the desired audio spatialisation effects, the preliminary tests identified lack of precision in GPS position detection as the main problem affecting the user experience. Although the impact of this problem may be significantly mitigated in the reference scenario (tour bus) for reasons pointed out in the previous section (higher predictability, larger distance to virtual source locations), solving it is essential for system versatility.

Simply applying moving-average filtering to the GPS output is not appropriate, as it would improve precision at the expense of responsiveness. Exploring sensor fusion techniques to combine IMU and GPS data is the most promising approach.

Work is under way to port the applications supporting the various blocks (playback, GPS, route manager…) to Android, as the system structure can be made simpler, lighter and more versatile by concentrating all the communication and processing functions on a smartphone or tablet – see Figure 12. As the figure suggests, operation would be completely autonomous, audio content being downloaded from the Internet according to the chosen route.

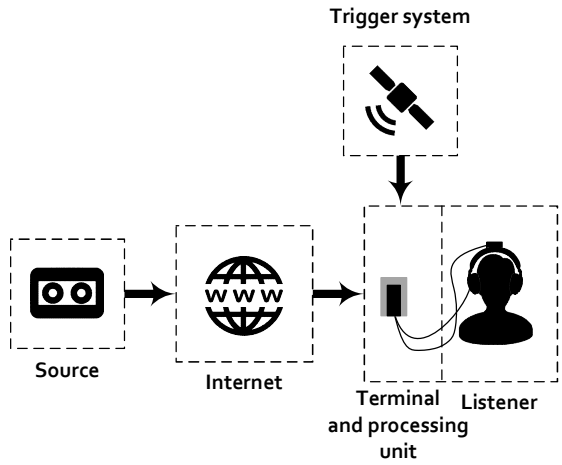


Figure 12: *Envisaged audio guiding system architecture*

Obviously, the IMU for head-tracking cannot be incorporated, as it must be attached to the headphone. The integration of a cost-effective, miniaturised head-tracking device, preferably with wireless connectivity, is another important future work front.

A wide variety of usage scenarios can be envisaged for a smartphone-based system. In the reference scenario, the triggering signal could be provided by a system installed on the bus, and hence constitute an added value of the ride. For indoor operation, position detection could no longer rely on GPS; alternative methods (e.g. based on radio-frequency ID tags, wi-fi sensors or ultrasonic beacons) would be required. By fully exploring *LibAAVE*'s capabilities, mentioned in section 3.1 (see Figure 6), the acoustic influence of the room could be modelled (early reflections and reverberation tail).

The perceived added value of the proposed audio AR effects in audio guides will no doubt be strongly influenced by other factors, namely:

• Quality of sound delivery – perfect-fit ear-enclosing high fidelity headphones are a must; comfortable, low-vibration bus seats would be desirable. Active noise cancellation systems may also prove indispensable.

• Content design – the added 3D audio dimension must be appropriately explored (e.g. for historic soundscape reconstruction). This requires expert story-telling based on appealing information and interpretation data brought to life by professional sound design/recording/editing.

To ensure the commercial success of audio guides incorporating the AR effects proposed here, these factors must be addressed simultaneously, which may impact on business models, possibly creating new (premium) market niche opportunities.

For this reason, establishing R&D partnerships with tour operators is also among the envisaged future work threads. The development of a demonstration route with excellent content design is key in this effort.

9. REFERENCES

[1] WTTC "Travel & Tourism Economic Impact," London, 2018. Avail. at https://www.wttc.org/-/media/files/reports/economic-impact-research/regions-2017/world2017.pdf, Accessed April, 16, 2018

[2] UNWTO "Tourism Highlights," Madrid, 2017. Available at https://www.e-unwto.org/doi/pdf/10.18111/9789284419029, Accessed April, 16, 2018

[3] UNWTO "Tourism and Culture Synergies," Madrid, 2018. Available at https://doi.org/10.18111/9789284418978, Accessed April, 16, 2018.

[4] D. Buhalis and Z. Yovcheva, "The digital tourism Think Tank report: 10 best practices in tourism". Available at https://thinkdigital.travel/wp-content/uploads/2013/04/10-AR-Best-Practices-in-Tourism.pdf, Accessed April, 16, 2018.

[5] D. Han, T. Jung and A. Gibson, "Information and Communication Technologies in Tourism," chapter Dublin AR: Implementing Augmented Reality in Tourism, Z. Xiang and I. Tussyadiah Eds. Springer, Cham, 2014.

[6] Tamo GPS Multilingual Commentary System. Available at http://www.tamotec.com/Product//8259674739.html, Accessed April, 15, 2018.

[7] toGuide TriggerPoint Wireless. Available at http://www.toguide.pt/pt/hardware/hardware_show/scripts/core.htm?p=hardware&f=hardware_show&lang=pt&idcont=123, Accessed April, 15, 2018.

[8] Soolai Bus Audio Guide System. Available at https://soolai.en.made-in-china.com/product/LCOndajPETVW/China-Bus-Audio-Guide-System.html, Accessed April, 15, 2018.

[9] Takstar WTG-500 Tour Guide System. Available at http://www.takstar.com/en/product/detail-13-38-0-400, Accessed April, 15, 2018.

[10] Mix Tech Polska, ATGS02. Available at http://www.mixtechpolska.pl/en/tour-guide-system-ATGS02.htm, Accessed April, 15, 2018.

[11] Mix Tech Polska, ATGS03. Available at http://www.mixtechpolska.pl/en/tour-guide-system-ATGS03.htm, Accessed April, 15, 2018.

[12] Acoustiguide. Available at http://www.acoustiguide.com/smartphone-applications, Accessed April, 15, 2018.

[13] ECHOES – Geolocated experiences. Available at https://echoes.xyz/ Accessed July, 3, 2018.

[14] N. Paterson, K. Naliuka, S. Jensen, T. Carrigy, H. Haahr and F. Conway, "Design, implementation and evaluation of audio for a location based augmented reality game," in *Proceedings of ACM Fun and Games*, Leuven, Belgium, 15–17 Sept. 2010.

[15] N. Mariette and B. Katz, "SoundDelta - Largescale, multi-user audio augmented reality," in *EAA Symp. on Auralization*, Espoo, Finland, 2009, pp. 1–6.

[16] B. Katz, S. Kammoun, G. Parseihian, O. Gutierrez, A. Brilhault, M. Auvray, P. Truillet, M. Denis, S. Thorpe and C. Jouffrais, "NAVIG: augmented reality guidance system for the visually impaired," *Virtual Reality*, vol. 16, no. 4, pp. 253–269, 2012.

[17] N. Paterson, G. Kearney, K. Naliuka, T. Carrigy, H. Haahr, and F. Conway, "Viking ghost hunt: creating engaging sound design for location–aware applications," *Int. Journal of Arts and Technology*, 6(1), pp. 61-82, Jan 2013.

[18] Usomo – The Immersive Sound system. Available at http://usomo.de/en/, Accessed April, 15, 2018.

[19] PortAudio: Portable Cross-Platform Audio I/O. Available at *http://www.portaudio.com*, Accessed April, 16, 2018.

[20] AcousticAVE: Auralisation Models and Applications in Virtual Reality Environments.
Available at https://code.ua.pt/projects/acousticave,
Accessed April 16, 2018

[21] G. Campos, P. Dias, J. Vieira, J. Santos, C. Mendonça, J. P. Lamas, N. Silva and S. Lopes, "AcousticAVE: Auralisation Models and Applications in Virtual Reality Environments," in *Proc. 8th Iberian Congress of Acoustics (Tecniacústica)*, Murcia, Spain, Oct. 29-31, 2014.

[22] A. Oliveira, G. Campos, P. Dias, D. Murphy, J. Vieira, C. Mendonça and J. Santos, "Real-Time Dynamic Image-Source Implementation for Auralisation," in *Proc. Digital Audio Effects (DAFx'13)*, Maynooth, Ireland, Sept. 2013, pp. 368-372.

[23] B. Gardner and K. Martin, "HRTF Measurements of a KEMAR Dummy-Head Microphone," MIT MediaLab, 2000. http://sound.media.mit.edu/resources/KEMAR.html, Accessed April, 16, 2018.

[24] V. Algazi, R. Duda and D. M. Thompson, "The CIPIC HRTF database," in *IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*, New York, Oct. 2001, pp. 99-102.

[25] J. Jacoby and M. S. Matell, "Three point Likert scales are good enough," Journal of Marketing Research, 8(4), pp. 495-500, 1971.

[26] J. Moutinho, D. Freitas and R. E. Araújo, "Indoor Sound Based Localization: Research Questions and First Results," in L. M. Camarinha-Matos, S. Tomic and P. Graça (eds) *Technological Innovation for the Internet of Things*. DoCEIS 2013. IFIP Advances in Information and Communication Technology, vol 394. Springer, Berlin, Heidelberg.

POWER-BALANCED MODELLING OF CIRCUITS AS SKEW GRADIENT SYSTEMS

Rémy Müller

IRCAM-STMS (UMR 9912)
Sorbonne University
Paris, France
remy.muller@ircam.fr

Thomas Hélie *

IRCAM-STMS (UMR 9912)
Sorbonne University
Paris, France
thomas.helie@ircam.fr

ABSTRACT

This article is concerned with the power-balanced simulation of analog audio circuits, governed by nonlinear differential algebraic equations (DAE). The proposed approach is to combine principles from the port-Hamiltonian and Brayton-Moser formalisms to yield a skew-symmetric gradient system. The practical interest is to provide a solver, using an average discrete gradient, that handles differential and algebraic relations in a unified way, and avoids having to pre-solve the algebraic part. This leads to a structure-preserving method that conserves the power balance and total energy. The proposed formulation is then applied on typical nonlinear audio circuits to study the effectiveness of the method.

1. INTRODUCTION

The need for stable, accurate and power-balanced simulation of nonlinear multi-physical systems is ubiquitous in the modelling of electronic circuits or mechanical systems and the natural setting for electronic circuits leads to Differential-Algebraic Equations.

Standard methods of solving electronic circuits are the State-variable [1], Modified Nodal Analysis [2], Sparse Tableau Analysis [3] and Wave Digital Filters (WDF) [4] according to the choice of variables the system is solved for. More recently, in the audio signal processing field, it has led to the Nodal DK method [5], nonlinear state-space [6] and extension of WDF to handle multi-port nonlinearities [7].

However, the underlying geometric structure and power-balance are often lost in the process. Furthermore, most numerical schemes either introduce or dissipate energy artificially, yielding unexpected, unstable or over-damped results.

To get rid of such artefacts, a very active research is focused on geometric numerical integration methods [8] that provide a theoretical framework for structure-preserving or invariant-preserving integration of dynamical systems. Among those methods, the Port-Hamiltonian (PHS) [9] [10] and Brayton-Moser (BM) [11] [12] formalisms are dual representations [13] [14] generalizing the Hamiltonian and Lagrangian formalisms to open dynamical systems with algebraic constraints (including dissipation).

PHS have been applied successfully to the modelling of the wah-wah pedal [15], Fender Rhodes [16], brass instruments [17] and loudspeaker nonlinearities [18]. Furthermore, automated generation of the PHS equations from the graph incidence matrix of a circuit's netlist has been investigated in [19] and leads to a skew-symmetric DAE form.

This paper considers this formulation as a starting point and proposes to combine the Brayton-Moser and Port-Hamiltonian viewpoints to represent all the constitutive laws as deriving from a single potential.

The presentation is organized as follows: first, in section 2, results about power balance, passivity, and duality of flow and effort spaces are recalled and it is shown how the power-balance can be represented by Dirac structures. Section 3 shows how, for both dynamic and algebraic components, the flow and effort variables can be derived from a single power potential involving the Hamiltonian and the algebraic content and co-content potentials [20] [21]. Section 4, then shows how to perform a power-balanced structure-preserving discretization of the system using a discrete gradient [22] [23]. Section 5 shows how to solve the resulting algebraic system using Newton iteration. Finally the method is applied to some example circuits in section 6 to show the effectiveness of the approach.

2. POWER BALANCE AND DIRAC STRUCTURES

For an electronic circuit, the Tellegen theorem [24] states that the sum of powers absorbed by all circuit elements is balanced.

$$P(\mathbf{e},\mathbf{f}) := \mathbf{e}^{\mathsf{T}}\mathbf{f} = \sum_n e_n f_n = 0 \tag{1}$$

where $\mathbf{e},\mathbf{f}$ are respectively the effort and flow variables of the circuit's branch components. This is an instance of the conservation of energy principle made famous by Lavoisier with the statement *nothing is lost, nothing is created, everything is transformed.*

This principle can be formalized mathematically by Dirac structures[1] that encodes the conservative power exchange in the circuit.

2.1. Power space

For an n-port element, let $\mathcal{F}$ be an n-dimensional real vector space and denote its dual $\mathcal{E} := \mathcal{F}^*$ (the space of linear functions on $\mathcal{F}$). We call $\mathcal{F}$ the space of flows $\mathbf{f}$ and $\mathcal{E}$ the space of efforts $\mathbf{e}$. On the product space $\mathcal{P} := \mathcal{F}\times\mathcal{E}$, power is defined by the non-degenerate bilinear form

$$P(\mathbf{e},\mathbf{f}) = \langle \mathbf{e} \,|\, \mathbf{f}\rangle, \quad \forall (\mathbf{f},\mathbf{e}) \in \mathcal{P} = \mathcal{F}\times\mathcal{E} \tag{2}$$

where $\langle \mathbf{e} \,|\, \mathbf{f}\rangle$ denotes the duality product, that is the linear function $\mathbf{e} \in \mathcal{E} = \mathcal{F}^*$ acting on $\mathbf{f} \in \mathcal{F}$. If $\mathcal{F}$ is equipped with an inner product $\langle\cdot,\cdot\rangle_{\mathcal{F}}$, then $\mathcal{E} = \mathcal{F}^*$ can be identified with $\mathcal{F}$ such that $\langle \mathbf{e} \,|\, \mathbf{f}\rangle = \langle \mathbf{e},\mathbf{f}\rangle_{\mathcal{F}}$, for all $\mathbf{f} \in \mathcal{F}$, $\mathbf{e} \in \mathcal{E} \sim \mathcal{F}$. If for example, $\mathcal{F}$ is the space of currents and $\mathcal{E}$ the space of voltages, then $\langle \mathbf{e} \,|\, \mathbf{f}\rangle = \langle \mathbf{e},\mathbf{f}\rangle_{\mathcal{F}} = \mathbf{e}^{\mathsf{T}}\mathbf{f}$ denote the electrical power.

* The author acknowledges the support of the ANR-DFG (French-German) project INFIDHEM ANR-16-CE92-0028.

[1] The Kirchoff Current and Voltage laws are special cases of Dirac structures when all the components share either the same current (series connection) or the same voltage (parallel connection).

2.2. Passivity and Dirac structures

In the $2n$-dimensional space $\mathcal{P}$, a passive linear n-port can be represented as an n-dimensional subspace $\mathcal{S} \subset \mathcal{P}$ defined by n linear constraints which admits the kernel representation

$$\mathcal{S} = \{(\mathbf{f}, \mathbf{e}) \in \mathcal{P} \mid \mathbf{F}\mathbf{f} + \mathbf{E}\mathbf{e} = 0\} \tag{3}$$

with $\mathrm{rank}([\mathbf{F}\ \mathbf{E}]) = n$. Furthermore, a linear subspace $\mathcal{D} \subset \mathcal{P}$ is said to be power-conserving if

$$\langle \mathbf{e} \,|\, \mathbf{f} \rangle = 0, \quad \forall (\mathbf{f}, \mathbf{e}) \in \mathcal{D} \tag{4}$$

It becomes a (constant) Dirac structure [25] [26] if and only if it is a maximal subspace of $\mathcal{P}$ with that property i.e. $\dim(\mathcal{D}) = \dim(\mathcal{F}) = \dim(\mathcal{E})$ and it admits the following matrix representations.

Definition 2.1 (Kernel representation)**.** *The kernel form of a Dirac structure is given by the subspace*

$$\mathcal{D} = \{(\mathbf{f}, \mathbf{e}) \in \mathcal{P} \mid \mathbf{F}\mathbf{f} + \mathbf{E}\mathbf{e} = 0,\ \mathbf{E}^\mathsf{T}\mathbf{F} + \mathbf{F}\mathbf{E}^\mathsf{T} = 0\} \tag{5}$$

where $\mathbf{F}, \mathbf{E} \in \mathbb{R}^{n\times n}$ *satisfy* $\mathrm{rank}([\mathbf{F}\ \mathbf{E}]) = n$.

Definition 2.2 (Hybrid skew-symmetric representation)**.** *Let* $\mathcal{D}$ *be given as in* (5)*, suppose there exists a permutation of the flow and efforts variables* $\pi : (\mathbf{F}, \mathbf{E}, \mathbf{f}, \mathbf{e}) \to (\tilde{\mathbf{F}}, \tilde{\mathbf{E}}, \tilde{\mathbf{f}}, \tilde{\mathbf{e}})$ *such that* $\tilde{\mathbf{F}}$ *is invertible then*

$$\mathcal{D} = \{(\tilde{\mathbf{f}}, \tilde{\mathbf{e}}) \in \mathcal{P} \quad | \quad \tilde{\mathbf{f}} = \mathbf{J}\tilde{\mathbf{e}}, \quad \mathbf{J} = -\tilde{\mathbf{F}}^{-1}\tilde{\mathbf{E}}\} \tag{6}$$

where $\mathbf{J} = -\mathbf{J}^\mathsf{T}$ *is skew-symmetric.*

Conversely, for any skew-symmetric matrix $\mathbf{J}$, the subspace $\mathcal{D}$ is a Dirac structure and one can verify that the power balance (1) is encoded by the skew-symmetry of $\mathbf{J}$:

$$P(\tilde{\mathbf{e}}, \tilde{\mathbf{f}}) = \tilde{\mathbf{e}}^\mathsf{T}\tilde{\mathbf{f}} = \tilde{\mathbf{e}}^\mathsf{T}\mathbf{J}\tilde{\mathbf{e}} = 0. \tag{7}$$

The skew-symetric form (6) will be used in the rest of the article.

3. GRADIENT DESCRIPTION OF COMPONENTS

Circuits are then categorized into dynamical, and algebraic components where algebraic components are further separated into dissipative and external sources because the later have degenerated constitutive laws. We show how the mixed effort $\tilde{\mathbf{e}}$ can be uniformly represented as the gradient of the scalar power potential (1).

3.1. Dynamic components: Hamiltonian potential

For dynamic components with state variable $\mathbf{x}$, flow variables are defined as the time-derivative of the state ($\mathbf{f} := \dot{\mathbf{x}}$) and the effort by a constitutive law $\mathbf{e} := \hat{e}(\mathbf{x})$. It is assumed that the constitutive law derives from the gradient of an energy storage function $H(\mathbf{x}(t))$ such that by definition $\hat{e}(\mathbf{x}) := \nabla H(\mathbf{x})$ and the power is

$$P(\mathbf{e}, \mathbf{f}) = \mathbf{e}^\mathsf{T}\mathbf{f} = \nabla H(\mathbf{x}) \cdot \dot{\mathbf{x}} = \frac{\mathrm{d}}{\mathrm{d}t} H(\mathbf{x}(t)). \tag{8}$$

The Hamiltonian function can then be found using the line integral.

$$H(\mathbf{x}) = \int \underbrace{\nabla H(\mathbf{x})}_{\mathbf{e}} \cdot \underbrace{\dot{\mathbf{x}}}_{\mathbf{f}} \,\mathrm{d}t = \int \nabla H(\mathbf{x}) \cdot \mathrm{d}\mathbf{x} \tag{9}$$

This idea is illustrated with the important cases of the linear capacitor and inductor. We then show how to handle a nonlinear component with an integrable constitutive law.

3.1.1. Capacitor

For a capacitor, the state variable is given by the charge $x_C = q$, with the flow $f = i_C = \dot{q}$, and effort $e = v_C = \frac{q}{C}$. This gives the Hamiltonian

$$H(q) = \int \frac{q}{C} \cdot \dot{q}\,\mathrm{d}t = \frac{1}{C}\int q\,\mathrm{d}q = \frac{q^2}{2C} \tag{10}$$

3.1.2. Inductor

Similarily for an inductor, the state variable is given by the flux-linkage $x_L = \phi$, the flow[2] by its time-derivative $f = \dot{\phi} = v_L$ and the dual effort by $e = i_L = \frac{\phi}{L}$ with an Hamiltonian function

$$H(\phi) = \int \frac{\phi}{L} \cdot \dot{\phi}\,\mathrm{d}t = \frac{1}{L}\int \phi \cdot \mathrm{d}\phi = \frac{\phi^2}{2L} \tag{11}$$

3.1.3. Nonlinear dynamic component

For a nonlinear dynamic component with state variable x, flow $f = \dot{x}$ and a constitutive law $e = \hat{e}(x) = \tanh(x)$, its Hamiltonian storage function is given by

$$H(x) = \int_0^t \hat{e}(x) \cdot \dot{x}\,\mathrm{d}t = \int_0^x \hat{e}(\bar{x}) \cdot \mathrm{d}\bar{x} = \ln(\cosh(x)) \tag{12}$$

3.2. Algebraic components: current and voltage potentials

If we consider the power differential $\mathrm{d}P$, using the product rule,

$$\mathrm{d}P(\mathbf{e}, \mathbf{f}) = \mathrm{d}(\mathbf{e} \cdot \mathbf{f}) = \mathbf{e} \cdot \mathrm{d}\mathbf{f} + \mathbf{f} \cdot \mathrm{d}\mathbf{e}. \tag{13}$$

Integration over a path Γ gives the integration by parts formula

$$\mathbf{e} \cdot \mathbf{f}\Big|_{\partial\Gamma} = \int_\Gamma \mathbf{e} \cdot \mathrm{d}\mathbf{f} + \int_\Gamma \mathbf{f} \cdot \mathrm{d}\mathbf{e}. \tag{14}$$

So, for components defined by algebraic constitutive laws $\Gamma = \{(\mathbf{e}, \mathbf{f}) \in \mathcal{P} \mid \mathbf{f} = \hat{f}(\mathbf{e})\}$, (respectively $\mathbf{e} = \hat{e}(\mathbf{f})$), the flow and effort potentials[3] are defined by the line integrals

$$D(\mathbf{f}) := \int_0^{\mathbf{f}} \hat{e}(\bar{\mathbf{f}}) \cdot \mathrm{d}\bar{\mathbf{f}}, \qquad D^*(\mathbf{e}) := \int_0^{\mathbf{e}} \hat{f}(\bar{\mathbf{e}}) \cdot \mathrm{d}\bar{\mathbf{e}}. \tag{15}$$

And according to (14), the instantaneous power is given, for $(\mathbf{e}, \mathbf{f}) \in \Gamma$, by (see figure 1 for a geometric interpretation and proof)

$$P(\mathbf{e}, \mathbf{f}) = \mathbf{e} \cdot \mathbf{f} = D(\mathbf{f}) + D^*(\mathbf{e}). \tag{16}$$

The flow and efforts can then be respectively obtained by partial derivatives of the power potential as

$$\mathbf{e} = \frac{\partial P}{\partial \mathbf{f}} = \nabla D(\mathbf{f}), \qquad \text{or} \qquad \mathbf{f} = \frac{\partial P}{\partial \mathbf{e}} = \nabla D^*(\mathbf{e}). \tag{17}$$

So in the case of a flow (resp. effort) controlled component the power can be expressed as a function of a single variable using either

$$P(\mathbf{e}) = \mathbf{e} \cdot \nabla D^*(\mathbf{e}) \qquad \text{or} \qquad P(\mathbf{f}) = \nabla D(\mathbf{f}) \cdot \mathbf{f}. \tag{18}$$

[2] Note that according to the energy domain (electric, magnetic, ...), the roles of flow and efforts need not necessarily be associated to the current and voltage. The convention adopted here, is that the flow of dynamic components is given by the time-derivative of the energy variable, while the effort is given by the gradient of the energy potential.

[3] These potentials are also called the content and co-content [20] [21].

3.2.1. Linear resistor

For a current-controlled (resp. voltage-controlled) resistor, the constitutive law is $v = \hat{e}(i) = Ri$ (resp. $i = \hat{f}(v) = v/R$). By consequence its current and voltage potentials are given by

$$D(i) = \int_0^i \hat{e}(f)\,\mathrm{d}f = \int_0^i Rf\,\mathrm{d}f = \frac{Ri^2}{2} \tag{19}$$

$$D^*(v) = \int_0^v \hat{f}(e)\,\mathrm{d}e = \int_0^v \frac{e}{R}\,\mathrm{d}e = \frac{v^2}{2R}. \tag{20}$$

Introduce function P as $P(v,i) = D(i) + D^*(v)$, then, for all (v,i) belonging on the characteristic curve, the power can be given by $v \cdot i$ (product-type), $P(v,i)$ (sum-type), $P(v,\hat{f}(v))$ (voltage-controlled) and $P(\hat{e}(i),i)$ (current-controlled), that is

$$P(v,i) = v{\cdot}i = D(i){+}D^*(v) = \frac{1}{2}\left(Ri^2 + \frac{v^2}{R}\right) = \frac{v^2}{R} = Ri^2. \tag{21}$$

In this particular case, we have $D(i) = D^*(v) = Ri^2$ because of linearity (for $v = Ri$) but this result should not be extrapolated as the next example will show.

3.2.2. P-N Diode

For a voltage controlled P-N diode, the constitutive law is given by

$$i = \hat{f}(v) = I_S\left(\exp\left(\frac{v}{nV_T}\right) - 1\right) \tag{22}$$

where I_S is the saturation current, n the ideality factor and V_T the thermal voltage. Its voltage potential is given by

$$D^*(v) = \int_0^v \hat{f}(e)\,\mathrm{d}e = nV_T I_S\left(\exp\left(\frac{v}{nV_T}\right) - \frac{v}{nV_T} - 1\right). \tag{23}$$

Direct integration for the current potential does not lead to an easily integrable primitive, however because of bijectivity, we can evaluate it indirectly by using the inverse map

$$v = \hat{e}(i) = \hat{f}^{-1}(i) = nV_T \ln\left(1 + \frac{i}{I_S}\right), \quad i > -I_S \tag{24}$$

and the Legendre transform $D(i) = \left[vi - D^*(v)\right]_{v=\hat{f}^{-1}(i)}$:

$$D(i) = nV_T I_S\left(\left(1 + \frac{i}{I_S}\right)\ln\left(1 + \frac{i}{I_S}\right) - \frac{i}{I_S}\right) \tag{25}$$

Using the above definitions, the current and voltage potentials being known, the component can be used as being either flow or effort-driven according to the constraints imposed by the circuit interconnections.

3.3. External sources

For external voltage (resp. current) sources, the constitutive laws $v = \hat{e}(i) = V$, (resp. $i = \hat{f}(v) = I$) are independent of the current (resp. voltage) variables and not bijective, with V (resp. I) being the source parameter. This gives the powers

$$P_V(v,i) = Vi = D(i), \qquad P_I(v,i) = vI = D^*(v). \tag{26}$$

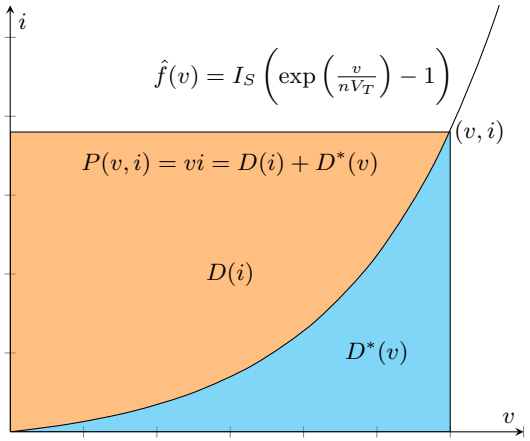


Figure 1: The areas occupied by the diode power $P(v,i)$ and the current and voltage potentials $D(i)$ and $D^*(v)$ are shown in the (v,i) plane for $I_S = 1$, $nV_T = 1$. It is geometrically clear that the current and voltage potentials are complimentary and their sum equals the power vi. It is also clear that in the nonlinear case $D(i) \neq D^*(v)$.

By consequence, for voltage (resp. current) sources, the voltage potential $D^*(v)$ (resp. current potential $D(i)$) is degenerate and null.

3.4. Summary

Using an appropriate permutation π (cf definition 2.2), the mixed flow $\mathbf{f}$ and its dual $\tilde{\mathbf{e}}$ can be parametrized by a state variable $\mathbf{x} \in \mathbb{R}^n$, a dissipative variable $\mathbf{w} \in \mathbb{R}^p$ and an output $\mathbf{y} \in \mathbb{R}^m$, where the potential $Z(\mathbf{w})$ (resp. $S(\mathbf{y})$) is an appropriate choice among the dissipative (resp. external) current and voltage potentials imposed by the permutation π. (Please refer to [19] for more details.)

$$\tilde{\mathbf{f}} := [\dot{\mathbf{x}}, \mathbf{w}, \mathbf{y}]^\mathsf{T} \tag{27}$$

$$\tilde{\mathbf{e}} := [\nabla H(\mathbf{x}), \nabla Z(\mathbf{w}), \nabla S(\mathbf{y})]^\mathsf{T} \tag{28}$$

The power potential[4] (1) can then be expressed as

$$P(\tilde{\mathbf{e}}, \tilde{\mathbf{f}}) = \tilde{\mathbf{e}}^\mathsf{T}\tilde{\mathbf{f}} = \underbrace{\nabla H(\mathbf{x})^\mathsf{T}\dot{\mathbf{x}}}_{P_c} + \underbrace{\nabla Z(\mathbf{w})^\mathsf{T}\mathbf{w}}_{P_d} + \underbrace{\nabla S(\mathbf{y})^\mathsf{T}\mathbf{y}}_{P_e}. \tag{29}$$

Combining the definitions (27) and (28), with the Dirac structure (6), leads to the skew-symmetric gradient form of Differential-Algebraic Port-Hamiltonian equations as

$$\underbrace{\begin{bmatrix}\dot{\mathbf{x}}\\ \mathbf{w}\\ \mathbf{y}\end{bmatrix}}_{\tilde{\mathbf{f}}} = \mathbf{J}\underbrace{\begin{bmatrix}\nabla H(\mathbf{x})\\ \nabla Z(\mathbf{w})\\ \nabla S(\mathbf{y})\end{bmatrix}}_{\tilde{\mathbf{e}}} \quad \Longleftrightarrow \quad \frac{\partial P}{\partial \tilde{\mathbf{e}}} = \mathbf{J}\frac{\partial P}{\partial \tilde{\mathbf{f}}} \tag{30}$$

[4] Note that because of the uniform usage of the *receiver convention* for each component (including sources), the power potentials represent the *absorbed* power by each component. This means that dissipative components will absorb *positive power*, while sources will, on average, absorb *negative power* to compensate for losses (but can temporarily receive power).

Integrating (29) over a time interval $[t_0, t_1]$ combined with the power balance (7), leads to the conservation of the total energy

$$\Delta E = H(\mathbf{x})\Big|_{t_0}^{t_1} + \int_{t_0}^{t_1} P_d(t)\,\mathrm{d}t + \int_{t_0}^{t_1} P_e(t)\,\mathrm{d}t = 0. \quad (31)$$

4. STRUCTURE-PRESERVING INTEGRATION SCHEME

The main objective of the numerical scheme is first and foremost, to provide a structure-preserving method that conserves the invariant (31) in discrete-time over each time-step. This offers the strong guarantee that no artificial energy is either consumed or created by the numerical scheme.To achieve this goal, thanks to the unified representation of DAE circuits as gradient systems introduced in section 3, it is now possible to generalize the usage of discrete gradient methods [22] [23] for *both* dynamic and algebraic components.

4.1. Discrete Gradients

Given a scalar potential $H : \mathbb{R}^n \mapsto \mathbb{R}$, a point $\mathbf{x} \in \mathbb{R}^n$ and a variation $\delta\mathbf{x} \in \mathbb{R}^n$, a necessary and sufficient condition for a function $\overline{\nabla} H(\mathbf{x}, \delta\mathbf{x}) : \mathbb{R}^n \times \mathbb{R}^n \mapsto \mathbb{R}^n$ to be a discrete gradient is given by

$$\overline{\nabla} H(\mathbf{x}, \delta\mathbf{x}) \cdot \delta\mathbf{x} = H(\mathbf{x} + \delta\mathbf{x}) - H(\mathbf{x}) \quad (32)$$

$$\overline{\nabla} H(\mathbf{x}, 0) = \nabla H(\mathbf{x}) \quad (33)$$

Definition 4.1 (Average Discrete Gradient)**.** *Let $\mathbf{x}, \delta\mathbf{x} \in \mathbb{R}^n$, and $H : \mathbb{R}^n \mapsto \mathbb{R}$ be a scalar potential. The average discrete gradient is defined for an affine trajectory model $\hat{\mathbf{x}}(\tau) = \mathbf{x} + \tau\delta\mathbf{x}$ by*

$$\overline{\nabla} H(\mathbf{x}, \delta\mathbf{x}) := \int_0^1 \nabla H(\mathbf{x} + \tau\delta\mathbf{x})\,\mathrm{d}\tau \quad (34)$$

Furthermore, using the gradient theorem, for separable potentials of the form

$$H(\mathbf{x}) = \sum_{i=1}^{N} H_i(x_i), \quad (35)$$

the discrete gradient can be computed *exactly* by finite differences on each scalar potential. It is given component-wise by

$$[\overline{\nabla} H(\mathbf{x}, \delta\mathbf{x})]_i := \begin{cases} \dfrac{H_i(x_i + \delta x_i) - H_i(x_i)}{\delta x_i} & \delta x_i \neq 0 \\ \dfrac{\partial H_i}{\partial x_i}(x_i) & \delta x_i = 0 \end{cases} \quad (36)$$

Finally, and *only in the case of quadratic potentials* of the form $H(\mathbf{x}) = \frac{1}{2}\mathbf{x}^T\mathbf{W}\mathbf{x}$ with $\mathbf{W} = \mathbf{W}^T \succeq 0$, does the discrete gradient correspond to evaluation of the gradient at the mid-point.

$$\overline{\nabla} H(\mathbf{x}, \delta\mathbf{x}) = \nabla H\left(\mathbf{x} + \frac{1}{2}\delta\mathbf{x}\right) = \mathbf{W}\left(\mathbf{x} + \frac{1}{2}\delta\mathbf{x}\right) \quad (37)$$

The following result will also be exploited in the next section.

Property 4.1. *Given a separable potential $H : \mathbb{R}^n \mapsto \mathbb{R}$, as in* (35) *of class $\mathcal{C}^2$, a point $\mathbf{x} \in \mathbb{R}^n$, a variation $\boldsymbol{\nu} \in \mathbb{R}^n$ and its discrete gradient $\overline{\nabla} H(\mathbf{x}, \boldsymbol{\nu})$ defined as* (36)*, the derivative of the discrete gradient with respect to the variation $\boldsymbol{\nu}$ is the diagonal matrix $\partial_{\boldsymbol{\nu}} \overline{\nabla} H : (\mathbf{x}, \boldsymbol{\nu}) \in \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}^{n\times n}$ with entries*

$$\left[\partial_{\boldsymbol{\nu}} \overline{\nabla} H\right]_{i,i} = \begin{cases} \dfrac{\nabla H_i(x_i + \nu_i) - \overline{\nabla} H_i(x_i, \nu_i)}{\nu_i} & \nu_i \neq 0 \\ \dfrac{1}{2}\dfrac{\partial^2 H_i}{\partial x_i^2}(x_i) & \nu_i = 0 \end{cases} \quad (38)$$

Proof. see Appendix A. □

4.2. Averaged System

Assuming over each time step $\Omega_n = [t_n, t_n + h]$, an affine trajectory model

$$\mathbf{z}(t_n + h\tau) = \mathbf{z}_n + \tau\delta\mathbf{z}_n \quad (39)$$

where $\mathbf{z} = [\mathbf{x}, \mathbf{w}, \mathbf{y}]^T$, and integrating (30) over Ω_n, we obtain the discrete structure-preserving system

$$\begin{bmatrix} \delta\mathbf{x}_n/h \\ \bar{\mathbf{w}}_n \\ \bar{\mathbf{y}}_n \end{bmatrix} = \mathbf{J}\begin{bmatrix} \overline{\nabla} H(\mathbf{x}_n, \delta\mathbf{x}_n) \\ \overline{\nabla} Z(\mathbf{w}_n, \delta\mathbf{w}_n) \\ \overline{\nabla} S(\mathbf{y}_n, \delta\mathbf{y}_n) \end{bmatrix} \quad (40)$$

where $\bar{\mathbf{w}}_n = \mathbf{w}_n + \delta\mathbf{w}_n/2$, $\bar{\mathbf{y}}_n = \mathbf{y}_n + \delta\mathbf{y}_n/2$. The DAE system (30) has been converted to an algebraic system that needs to be to solved for the average variation $\delta\mathbf{z}_n = [\delta\mathbf{x}_n, \delta\mathbf{w}_n, \delta\mathbf{y}_n]^\mathsf{T}$.

5. NEWTON ITERATION

Denote the variation $\boldsymbol{\nu} = \delta\mathbf{z}_n$, solving the discrete algebraic system (40) can be rewritten as the root-finding problem

$$F(\boldsymbol{\nu}^*) = 0 \quad (41)$$

where $\boldsymbol{\nu}^*$ is the looked for solution and F is defined by

$$F(\boldsymbol{\nu}) := \mathbf{D}_0\mathbf{z}_n + \mathbf{D}_1\boldsymbol{\nu} - \mathbf{J}\overline{\nabla}_{\tilde{\mathbf{f}}} P(\mathbf{z}_n, \boldsymbol{\nu}), \quad (42)$$

with $\mathbf{D}_0 = \begin{bmatrix} 0 & 0 & 0 \\ 0 & \mathbf{I}_p & 0 \\ 0 & 0 & \mathbf{I}_m \end{bmatrix}$, $\mathbf{D}_1 = \begin{bmatrix} \mathbf{I}_n/h & 0 & 0 \\ 0 & \mathbf{I}_p/2 & 0 \\ 0 & 0 & \mathbf{I}_m/2 \end{bmatrix}$, where $\mathbf{I}_n$ denote the $n \times n$ identity matrix and $\overline{\nabla}_{\tilde{\mathbf{f}}} P = [\overline{\nabla} H, \overline{\nabla} Z, \overline{\nabla} S]^\mathsf{T}$.

5.1. Newton update

For an estimate $\boldsymbol{\nu}_k$ and a perturbation $\Delta\boldsymbol{\nu}_k$, the true solution $\boldsymbol{\nu}^*$ of (41) can be written as $\boldsymbol{\nu}^* = \boldsymbol{\nu}_k + \Delta\boldsymbol{\nu}_k$. Taylor series expansion of F around $\boldsymbol{\nu}_k$, with $\|\Delta\boldsymbol{\nu}_k\|$ sufficiently small yields

$$0 = F(\boldsymbol{\nu}_k + \Delta\boldsymbol{\nu}_k) = F(\boldsymbol{\nu}_k) + [F'(\boldsymbol{\nu}_k)](\Delta\boldsymbol{\nu}_k) + \mathcal{O}(\|\Delta\boldsymbol{\nu}_k\|^2). \quad (43)$$

If the Jacobian F' is invertible, neglecting high-order terms and solving for $\Delta\boldsymbol{\nu}$ leads to the Newton update

$$\Delta\boldsymbol{\nu}_k := -F'(\boldsymbol{\nu}_k)^{-1}F(\boldsymbol{\nu}_k), \quad \boldsymbol{\nu}_{k+1} := \boldsymbol{\nu}_k + \Delta\boldsymbol{\nu}_k, \quad (44)$$

where the Jacobian of F is given by

$$F'(\boldsymbol{\nu}) = \mathbf{D}_1 - \mathbf{J}\left(\partial_{\boldsymbol{\nu}}\overline{\nabla}_{\tilde{\mathbf{f}}} P(\mathbf{z}_n, \boldsymbol{\nu})\right). \quad (45)$$

For a separable potential P, using property (4.1), $\partial_{\boldsymbol{\nu}}\overline{\nabla}_{\tilde{\mathbf{f}}} P$ is a diagonal matrix that can be computed from the knowledge of the gradient, Hessian and discrete gradient of the potential.

5.2. Convergence and stiffness

If the eigenvalues of the matrix $\mathbf{A} = \mathbf{D}_1^{-1}\mathbf{J}\left(\partial_{\boldsymbol{\nu}}\overline{\nabla}_{\tilde{\mathbf{f}}}P(\mathbf{z}_n,\boldsymbol{\nu})\right)$ are such that $\|\mathbf{A}\|_2 = \max(|\lambda_i|) < 1$, the fixed-point induced by (40) is contracting. The Banach fixed-point theorem guarantees existence and unicity of the solution. It is then possible to approximate the inverse of the Jacobian with the Neumann series identity

$$(\mathbf{I}-\mathbf{A})^{-1} = \sum_{k=0}^{\infty}\mathbf{A}^k \approx \mathbf{I}+\mathbf{A}+\mathbf{A}^2+\dots \tag{46}$$

to get the first (or any higher) order approximation

$$F'(\boldsymbol{\nu})^{-1} \approx \left(\mathbf{I}+\mathbf{D}_1^{-1}\mathbf{J}\left(\partial_{\boldsymbol{\nu}}\overline{\nabla}_{\tilde{\mathbf{f}}}P(\mathbf{z}_n,\boldsymbol{\nu})\right)\right)\mathbf{D}_1^{-1} \tag{47}$$

If $\max|\lambda_i| \geq 1$, the system is said to be stiff, the series (46) is divergent, and the approximation (47) is no longer valid. Solving the system then requires a matrix inversion for each iteration. Using the Newton-Kantorovich theorem, for a starting point $\boldsymbol{\nu}_0$, if there exists positive constants β_0, γ, h_0, such that $\|F'(\boldsymbol{\nu}_0)^{-1}\| \leq \beta_0$, $F'(\boldsymbol{\nu})$ is locally γ-Lipschitz and $h_0 := \|\Delta\boldsymbol{\nu}_0\|\,\beta_0\gamma < 1/2$, then the sequence $\{\boldsymbol{\nu}_k\}$ converges quadratically to some unique $\boldsymbol{\nu}^*$ such that $F(\boldsymbol{\nu}^*) = 0$. Please refer to [27] for more details.

6. CIRCUIT EXAMPLES

6.1. Envelope Follower

We consider the envelope follower circuit shown in figure 3 with parameters $C = 100$ pF, $I_S = 2.52$ nA, $V_T = 23$ mV and $n = 1.96$. Kirchoff laws leads to the following Dirac structure:

$$\underbrace{\begin{bmatrix} i_C \\ v_D \\ i_S \end{bmatrix}}_{\tilde{\mathbf{f}}} = \underbrace{\begin{bmatrix} 0 & 1 & 0 \\ -1 & 0 & 1 \\ 0 & -1 & 0 \end{bmatrix}}_{\mathbf{J}} \underbrace{\begin{bmatrix} v_C \\ i_D \\ v_S \end{bmatrix}}_{\tilde{\mathbf{e}}}. \tag{48}$$

For this circuit we have $\mathbf{x} = [q]$, $\mathbf{w} = [v_D]$, $\mathbf{y} = [i_S]$, $\tilde{\mathbf{f}} = [\dot{q}, v_D, i_S]^\mathsf{T}$ and the following potentials

$$H(q) = \frac{q^2}{2C}, \tag{49}$$

$$Z(v_D) = nV_TI_S\left(\exp\left(\frac{v_D}{nV_T}\right)-1\right) - v_DI_S, \tag{50}$$

$$S(i_S) = Vi_S. \tag{51}$$

Taking their gradients gives the right-hand side vector

$$\tilde{\mathbf{e}} = \begin{bmatrix} v_C \\ i_D \\ v_S \end{bmatrix} = \begin{bmatrix} \nabla H(q) \\ \nabla Z(v_D) \\ \nabla S(i_S) \end{bmatrix} = \begin{bmatrix} q/C \\ I_S\left(\exp\left(\frac{v_D}{nV_T}\right)-1\right) \\ V \end{bmatrix} \tag{52}$$

and the product $\tilde{\mathbf{e}}^\mathsf{T}\tilde{\mathbf{f}}$ gives the power balance potential

$$P(\tilde{\mathbf{e}},\tilde{\mathbf{f}}) = \underbrace{\nabla H(q)\dot{q}}_{P_C(q)} + \underbrace{\nabla Z(v_D)v_D}_{P_D(v_D)} + \underbrace{\nabla S(i_S)i_S}_{P_S(i_S)}. \tag{53}$$

For the capacitor and voltage source, we obtain the discrete gradients

$$\overline{\nabla}H(q,\delta q) = \frac{1}{C}\left(q+\frac{\delta q}{2}\right), \qquad \overline{\nabla}S(i,\delta i) = V, \tag{54}$$

and after some algebraic manipulations (see appendix B), the discrete gradient of the diode potential can be expressed as

$$\overline{\nabla}Z(v,\delta v) = I_S\left(\exp\left(\frac{v+\delta v/2}{nV_T}\right)\operatorname{sinhc}\left(\frac{\delta v}{2nV_T}\right)-1\right). \tag{55}$$

where the sinhc term ($\operatorname{sinhc} := \sinh(x)/x$) acts as a correction compared to evaluation of the gradient at the mid-point.

6.2. Diode Clipper

We consider the diode clipper circuit shown in figure 5 with parameters $R = 1$ kΩ, $C = 100$ nF, $I_S = 2.52$ fA, $V_T = 23$ mV and $n = 1$. For the two diodes, with $v_D := v_{D_1}$ and the diodes current $i_D := i_{D_1} - i_{D_2}$, the constitutive law is

$$i_D = \hat{f}(v_D) = 2I_S\sinh\left(\frac{v_D}{nV_T}\right). \tag{56}$$

Its integration gives the voltage potential

$$D_D^*(v_D) = \int_0^{v_D}\hat{f}(v)dv = 2nV_TI_S\left(\cosh\left(\frac{v_D}{nV_T}\right)-1\right). \tag{57}$$

Application of Kirchoff laws leads to the following Dirac structure:

$$\underbrace{\begin{bmatrix} i_C \\ v_R \\ v_D \\ i_S \end{bmatrix}}_{\tilde{\mathbf{f}}} = \underbrace{\begin{bmatrix} 0 & 1 & -1 & 0 \\ -1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \end{bmatrix}}_{\mathbf{J}} \underbrace{\begin{bmatrix} v_C \\ i_R \\ i_D \\ v_S \end{bmatrix}}_{\tilde{\mathbf{e}}}. \tag{58}$$

For this circuit, $\mathbf{x} = [q]$, $\mathbf{w} = [v_R, v_D]^\mathsf{T}$, $\mathbf{y} = [i_S]$, $\tilde{\mathbf{f}} = [\dot{q}, v_R, v_D, i_S]^\mathsf{T}$ and the potentials are

$$H(q) = \frac{q^2}{2C}, \quad Z(v_R,v_D) = \frac{v_R^2}{2R} + D_D^*(v_D), \quad S(i_S) = Vi_S. \tag{59}$$

Their gradients regenerates the mixed effort

$$\tilde{\mathbf{e}} = \begin{bmatrix} v_C \\ i_R \\ i_D \\ v_S \end{bmatrix} = \begin{bmatrix} \nabla H \\ \nabla Z_R \\ \nabla Z_D \\ \nabla S \end{bmatrix} = \begin{bmatrix} q/C \\ v_R/R \\ 2I_S\sinh\left(\frac{v_D}{nV_T}\right) \\ V \end{bmatrix} \tag{60}$$

and the product $\tilde{\mathbf{e}}^\mathsf{T}\tilde{\mathbf{f}}$ gives the power balance potential

$$P(\tilde{\mathbf{e}},\tilde{\mathbf{f}}) = \underbrace{\nabla H(q)\dot{q}}_{P_C(q)} + \underbrace{\nabla Z_R(v_R)v_R}_{P_R(v_R)} + \underbrace{\nabla Z_D(v_D)v_D}_{P_D(v_D)} + \underbrace{\nabla S(i_S)i_S}_{P_S(i_S)}. \tag{61}$$

Similarily as in the envelope follower case, we have the discrete gradients (54) for the capacitor and voltage source, with

$$\overline{\nabla}Z_R(v,\delta v) = \frac{1}{R}\left(v+\frac{\delta v}{2}\right) \tag{62}$$

for the resistor, and after some algebraic manipulations, the discrete gradient of the diodes potential can be expressed as

$$\overline{\nabla}Z_D(v,\delta v) = 2I_S\sinh\left(\frac{v+\delta v/2}{nV_T}\right)\operatorname{sinhc}\left(\frac{\delta v}{2nV_T}\right). \tag{63}$$

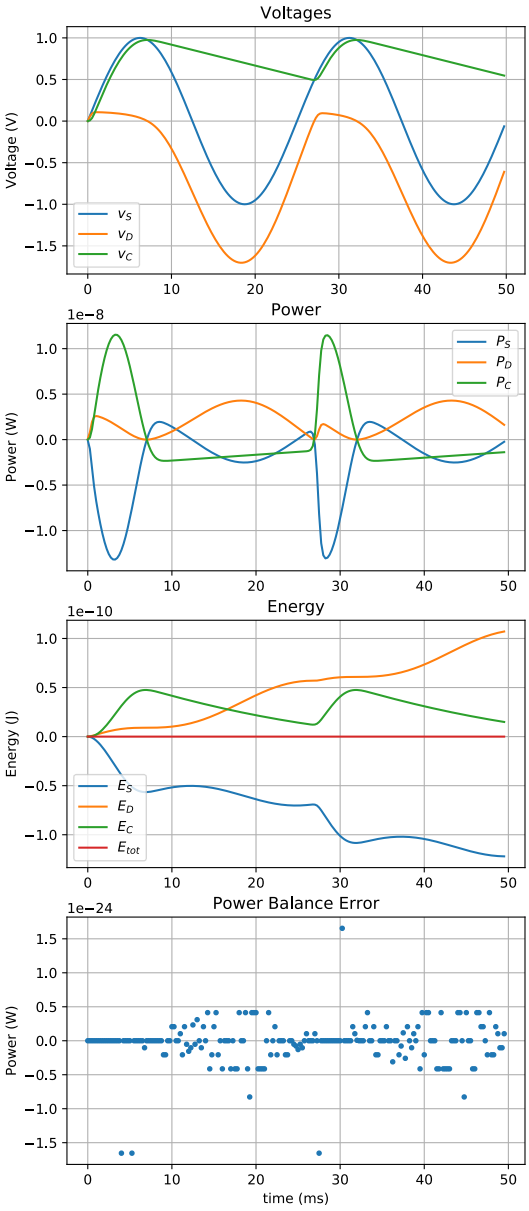


Figure 2: Envelope follower circuit driven by a 1V sinusoidal input with fundamental frequency $f = 40$ Hz, $f_s = 4$ kHz.

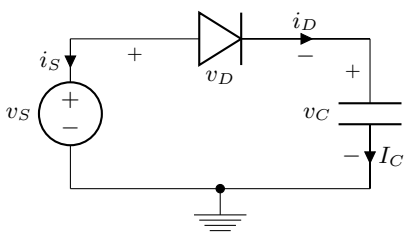


Figure 3: Envelope Follower circuit

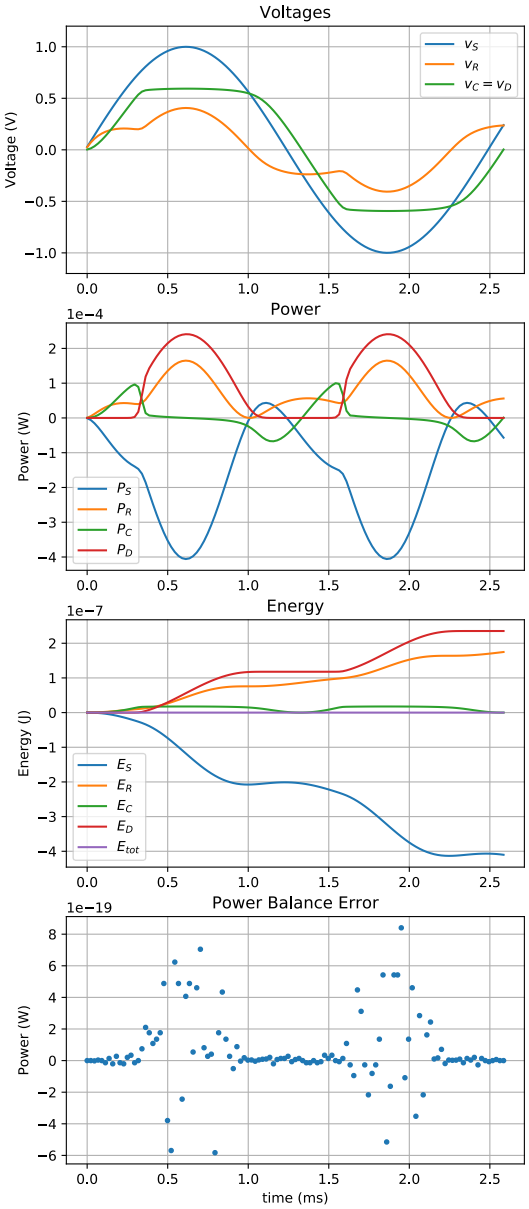


Figure 4: Diode clipper circuit driven by a 1V sinusoidal input with fundamental frequency $f = 400$ Hz, $f_s = 44.1$ kHz.

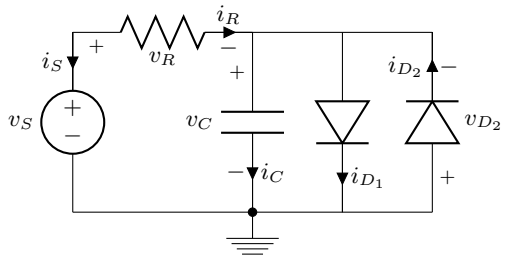


Figure 5: Diode Clipper circuit

6.3. Analysis

Simulation results for both circuits are shown in figure 2 and figure 4 with respective sampling frequencies 4 kHz and 44.1 kHz. We remark that in both cases, the power balance is satisfied with high precision. The relative error is of the order of the machine epsilon ($\epsilon = 2^{-53} \approx 1.11 \cdot 10^{-16}$). This results in a vanishing total energy variation.

For dissipative components, the absorbed power is always positive; the dissipated energy is thus monotonously increasing. For dynamic components and sources, the power is alternatively absorbed and released, the difference being that sources have a decreasing average energy trend to compensate for losses in the dissipative components.

Existence and uniqueness of the fixed points are guaranteed if $h < C/\gamma_D$ for the envelope follower and if $h < C/\max(\gamma_D, \gamma_R)$ for the diode clipper (proof is ommited) where γ_K stands for the local Lipschitz constants $\gamma_K = \max_\nu |\partial_\nu \overline{\nabla} Z_K(v_{K_0}, \nu)|$ of the diode and resistor components in a neighborhood around $\boldsymbol{\nu}_0$.

For the diode clipper circuit, the fixed-point does not converge, but the Newton iteration does. We can remark that each time the diodes are saturating, the precision of the power balance is slightly deteriorated. This can be explained by two facts: the dissipated power is also increasing during saturation and the system becomes stiff, thus the numerical conditioning of the Jacobian in the Newton iteration gets worse.

7. CONCLUSION

The main contribution of this paper consists in a) using the power-balance as the core object from which all quantities in the system are derived, b) generalizing the usage of potentials and their gradients to represent the flow and effort variables for both dynamic an algebraic components, c) keeping the sparse skew-symmetric structure matrix $\mathbf{J}$ until numerical simulation, d) integration of the system using the average discrete gradient. This leads to a consistent structure-preserving approximation that conserves the form of the original system in discrete-time.

It is also shown that the Jacobian of the Newton iteration has a special structure that only involves diagonal and skew-symmetric matrices. It can be computed only from the knowledge of the potentials associated with each component and stiffness can be inferred by inspection of the derivatives of the discrete gradient. Furthermore the structure-preserving approach offers a valuable tool to monitor the quality of our approximations with respect to the power balance.

The main drawback of the approach is a direct consequence from its strength. Indeed, the preservation of the power balance, prevents the use of L-stable integrators (which limit the stiffness by introducing artificial numerical dissipation) such as the Backward Difference Formulas or Radau IIa methods [28] [29]. This imposes some restrictions on the step size or the use of adaptive strategies. However, since the average integration of the system can be interpreted as a lowpass projector and first-order anti-aliasing filter [30], parasitic oscillations at the Nyquist frequency which are typical of stiff systems are attenuated during the simulation.

Further perspectives include the use of higher-order trajectory models, exponential integrators [31] which have shown to be effective in the simulation of stiff systems and more generally Lie-group integrators [32] [33] whose trajectories belong, by construction, to the system manifold.

8. ACKNOWLEDGMENTS

The second author acknowledges the support of the ANR-DFG (French-German) project INFIDHEM ANR-16-CE92-0028.

9. REFERENCES

[1] E. S. Kuh and R. A. Rohrer, "The state-variable approach to network analysis," *Proceedings of the IEEE*, vol. 53, no. 7, pp. 672–686, 1965.

[2] C.-W. Ho, A. Ruehli, and P. Brennan, "The modified nodal approach to network analysis," *IEEE Transactions on circuits and systems*, vol. 22, no. 6, pp. 504–509, 1975.

[3] G. Hachtel, R. Brayton, and F. Gustavson, "The sparse tableau approach to network analysis and design," *IEEE Transactions on circuit theory*, vol. 18, no. 1, pp. 101–113, 1971.

[4] K. Meerkotter and R. Scholz, "Digital simulation of nonlinear circuits by wave digital filter principles," in *Circuits and Systems*. IEEE, 1989, pp. 720–723.

[5] D. T. Yeh, J. S. Abel, and J. O. Smith, "Automated physical modeling of nonlinear audio circuits for real-time audio effects–part i: Theoretical development," *IEEE transactions on audio, speech, and language processing*, vol. 18, no. 4, pp. 728–737, 2010.

[6] M. Holters and U. Zölzer, "A generalized method for the derivation of non-linear state-space models from circuit schematics," in *Signal Processing Conference (EUSIPCO), 2015 23rd European*. IEEE, 2015, pp. 1073–1077.

[7] K. J. Werner, V. Nangia, J. O. Smith III, and J. S. Abel, "Resolving wave digital filters with multiple/multiport nonlinearities," in *Proc. 18th Conf. Digital Audio Effects*, 2015, pp. 387–394.

[8] E. Hairer, C. Lubich, and G. Wanner, *Geometric Numerical Integration: Structure-Preserving Algorithms for Ordinary Differential Equations; 2nd ed.* Dordrecht: Springer, 2006.

[9] A. van der Schaft and D. Jeltsema, "Port-hamiltonian systems theory: An introductory overview," *Foundations and Trends in Systems and Control*, vol. 1, no. 2-3, pp. 173–378, 2014.

[10] A. van der Schaft, "Port-hamiltonian systems: an introductory survey," in *Proceedings of the International Congress of Mathematicians Vol. III: Invited Lectures*, Madrid, Spain, 2006, pp. 1339–1365.

[11] R. Brayton and J. Moser, "A theory of nonlinear networks. i," *Quarterly of Applied Mathematics*, vol. 22, no. 1, pp. 1–33, 1964.

[12] ——, "A theory of nonlinear networks. ii," *Quarterly of applied mathematics*, vol. 22, no. 2, pp. 81–104, 1964.

[13] A. J. van der Schaft, "On the relation between port-hamiltonian and gradient systems," *IFAC Proceedings Volumes*, vol. 44, no. 1, pp. 3321–3326, 2011.

[14] D. Jeltsema and J. M. Scherpen, "A dual relation between port-hamiltonian systems and the brayton–moser equations for nonlinear switched rlc circuits," *Automatica*, vol. 39, no. 6, pp. 969–979, 2003.

[15] A. Falaize and T. Hélie, "Simulation of an analog circuit of a wah pedal: a port-Hamiltonian approach," in *135th convention of the Audio Engineering Society*, New-York, United States, Oct. 2013, pp. –.

[16] ——, "Passive simulation of the nonlinear port-Hamiltonian modeling of a Rhodes Piano," *Journal of Sound and Vibration*, vol. 390, pp. 289–309, Mar. 2017.

[17] N. Lopes and T. Hélie, "Energy Balanced Model of a Jet Interacting With a Brass Player's Lip," *Acta Acustica united with Acustica*, vol. 102, no. 1, pp. 141–154, 2016.

[18] A. Falaize and T. Hélie, "Passive simulation of electrodynamic loudspeakers for guitar amplifiers: a port- Hamiltonian approach," in *International Symposium on Musical Acoustics*, Le Mans, France, Jul. 2014, pp. 1–5.

[19] A. Falaize and T. Hélie, "Passive guaranteed simulation of analog audio circuits: A port-hamiltonian approach," *Applied Sciences*, vol. 6, no. 10, 2016.

[20] W. Millar, "Some general theorems for non-linear systems possessing resistance," *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, vol. 42, no. 333, pp. 1150–1160, 1951.

[21] C. Cherry, "Some general theorems for non-linear systems possessing reactance," *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, vol. 42, no. 333, pp. 1161–1177, 1951.

[22] R. I. McLachlan, G. Quispel, and N. Robidoux, "Geometric integration using discrete gradients," *Philosophical Transactions of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, vol. 357, no. 1754, pp. 1021–1045, 1999.

[23] E. Celledoni, V. Grimm, R. McLachlan, D. McLaren, D. O'Neale, B. Owren, and G. Quispel, "Preserving energy resp. dissipation in numerical PDEs using the 'average vector field' method," *Journal of Computational Physics*, vol. 231, no. 20, pp. 6770 – 6789, 2012.

[24] B. D. Tellegen, "A general network theorem, with applications," *Philips Res Rep*, vol. 7, pp. 256–269, 1952.

[25] I. Y. Dorfman, "Dirac structures of integrable evolution equations," *Physics Letters A*, vol. 125, no. 5, pp. 240–246, 1987.

[26] T. Courant and A. Weinstein, "Beyond poisson structures," *Action hamiltoniennes de groupes. Troisieme théoreme de Lie (Lyon, 1986)*, vol. 27, pp. 39–49, 1988.

[27] P. Deuflhard, *Newton methods for nonlinear problems: affine invariance and adaptive algorithms*. Springer, 2011, vol. 35.

[28] G. Wanner and E. Hairer, *Solving ordinary differential equations II: Stiff and Differential-Algebraic Problems*. Springer, 1991, vol. 14.

[29] J. C. Butcher, *Numerical methods for ordinary differential equations*. John Wiley & Sons, 2016.

[30] R. Müller and T. Hélie, "Trajectory anti-aliasing on guaranteed-passive simulation of nonlinear physical systems," in *Proc. 20th Conf. Digital Audio Effects*, 2017.

[31] M. Hochbruck and A. Ostermann, "Exponential integrators," *Acta Numerica*, vol. 19, pp. 209–286, 2010.

[32] E. Celledoni, H. Marthinsen, and B. Owren, "An introduction to lie group integrators–basics, new developments and applications," *Journal of Computational Physics*, vol. 257, pp. 1040–1061, 2014.

[33] A. Iserles, H. Z. Munthe-Kaas, S. P. Nørsett, and A. Zanna, "Lie-group methods," *Acta numerica*, vol. 9, pp. 215–365, 2000.

A. DISCRETE GRADIENT DERIVATIVE

Proof. To prove property 4.1 for $H(x)$ a scalar potential, when the variation $\nu \neq 0$, using a) the quotient rule, b) the chain rule and c) identification with the discrete gradient definition (36), we obtain

$$\begin{aligned}\frac{\partial\overline{\nabla}H}{\partial\nu} &\overset{a}{=} \frac{[\frac{\partial}{\partial\nu}(H(x+\nu)-H(x))]\nu - [H(x+\nu)-H(x)]\frac{\partial\nu}{\partial\nu}}{\nu^2}\\ &\overset{b}{=} \frac{1}{\nu}\left(\frac{\partial H}{\partial x}(x+\nu)\frac{\partial(x+\nu)}{\partial v} - \frac{H(x+\nu)-H(x)}{\nu}\right)\\ &\overset{c}{=} \frac{\nabla H(x+\nu)-\overline{\nabla}H(x,\nu)}{\nu}.\end{aligned}$$

When $\nu \to 0$, using a) the definition of the discrete gradient (36) with b) Taylor series expansion about x and neglecting high order terms when passing to the limit leads to

$$\begin{aligned}\frac{\partial\overline{\nabla}H}{\partial\nu}(x,0) &:= \lim_{\nu\to 0}\frac{\nabla H(x+\nu)-\overline{\nabla}H(x,\nu)}{\nu}\\ &\overset{a}{=} \lim_{\nu\to 0}\frac{\nabla H(x+\nu)}{\nu} - \frac{H(x+\nu)-H(x)}{\nu^2}\\ &\overset{b}{=} \lim_{\nu\to 0}\frac{H'(x)+H''\nu}{\nu} - \frac{H'(x)\nu+H''(x)\nu^2/2!}{\nu^2}\\ &= \frac{1}{2}\frac{\partial^2 H}{\partial x^2}(x)\end{aligned}$$

□

B. DISCRETE GRADIENT OF THE DIODE POTENTIAL

Proof. Using a) the definition of the discrete gradient (36), b) the definition of the diode potential (23) followed by c) factorization of the mid-point exponential term, then d) identification of the $\sinh$ and e) sinhc functions, the discrete gradient of the diode voltage potential can be expressed as

$$\begin{aligned}\overline{\nabla}D^*(v,\delta v) &:\overset{a}{=} \frac{D_D^*(v+\delta v)-D_D^*(v)}{\delta v}\\ &\overset{b}{=} \frac{nV_T I_S}{\delta v}\left(\exp\left(\frac{v+\delta v}{nV_T}\right)-\exp\left(\frac{v}{nV_T}\right)-\frac{\delta v}{nV_T}\right)\\ &\overset{c}{=} I_S\left(\frac{nV_T}{\delta v}\exp\left(\frac{v+\delta v/2}{nV_T}\right)\left(e^{\frac{\delta v}{2nV_T}}-e^{-\frac{\delta v}{2nV_T}}\right)-1\right)\\ &\overset{d}{=} I_S\left(\frac{2nV_T}{\delta v}\exp\left(\frac{v+\delta v/2}{nV_T}\right)\sinh\left(\frac{\delta v}{2nV_T}\right)-1\right)\\ &\overset{e}{=} I_S\left(\exp\left(\frac{v+\delta v/2}{nV_T}\right)\mathrm{sinhc}\left(\frac{\delta v}{2nV_T}\right)-1\right)\end{aligned}$$

and since $\mathrm{sinhc}(0)=1$, $\overline{\nabla}D^*(v,0)=\nabla D^*(v)$ satisfies eq (33). □

MODELING TIME-VARYING REACTANCES USING WAVE DIGITAL FILTERS

Ólafur Bogason

Genki Instruments
Reykjavik, Iceland
olafur@genkiinstruments.com

Kurt James Werner

The Sonic Arts Research Centre (SARC)
School of Arts, English and Languages
Queen's University Belfast, UK
k.werner@qub.ac.uk

ABSTRACT

Wave Digital Filters were developed to discretize linear time invariant lumped systems, particularly electronic circuits. The time-invariant assumption is baked into the underlying theory and becomes problematic when simulating audio circuits that are by nature time-varying. We present extensions to WDF theory that incorporate proper numerical schemes, allowing for the accurate simulation of time-varying systems.

We present generalized continuous-time models of reactive components that encapsulate the time-varying lossless models presented by Fettweis, the circuit-theoretic time-varying models, as well as traditional LTI models as special cases. Models of time-varying reactive components are valuable tools to have when modeling circuits containing variable capacitors or inductors or electrical devices such as condenser microphones. A power metric is derived and the model is discretized using the alpha-transform numerical scheme and parametric wave definition.

Case studies of circuits containing time-varying resistance and capacitance are presented and help to validate the proposed generalized continuous-time model and discretization.

1. INTRODUCTION

Time-varying lumped systems involve at least one parameter, e.g. the value of a resistor, that is changed over time. Many musical circuits are time-varying, including auto-wah pedals, phasers, and indeed most every circuit where a user can twist a knob on the fly. Some circuits may involve time-varying reactances, for instance ladder filters with variable inductors or stepped filters where reactances may be switched in and out. In virtual analog, stability and energy-preservation under time-varying conditions has been studied in, e.g., [1, 2, 3]. However in certain electrical devices, e.g. condenser microphones, the *dynamics* of a time-varying reactance (in that case, a capacitor) are the main operating principle of the device and the system may not actually be energy-preserving under time-varying conditions in continuous time. In virtual analog, modeling time-varying reactances is essential, both to accurately simulate time-varying phenomena in electrical systems and to develop principles for time-varying digital filters based on static analog filters, e.g. adaptive digital filtering.

Wave Digital Filters (WDFs) provide a computationally efficient way to simulate lumped element models [4] with excellent numerical properties. Recent developments in the field include topological advances in linear and nonlinear circuits [5, 6, 7], the introduction of new wave variable definitions, including parametric waves [8] and bi-parametric waves [9] and the development of new discretization schemes [10] applied to WDFs [8, 11, 12]. In the WDF literature there has been some research done on time-varying systems. By giving up guaranteed stability, Strube extended the paradigm to two dimensions to model vocal tracts [13, 14]. Stability of passive, time-varying circuits [15, 16] has been proven for WDF algorithms that employ power-normalized waves as signal variables and guaranteed stable approaches to varying the step-size on the fly have also been studied [17, 18].

The paper is structured as follows. In the rest of this section, we discuss notation and background information. In §2 we discuss continuous-time models of capacitors and inductors and propose novel generalized models of these reactances. In §3 we discuss discretization schemes for reactances in the WDF paradigm and discretize the generalized models. We use the newly discretized model to study the effects of time-varying resistance (§4) and reactance (§5) on the dc response of a RC circuit. §6 presents recommendations for time-varying WDF simulations and concludes.

1.1. Wave Variables

Instead of the Kirchhoff signal variables from circuit theory, voltage v and current i, in WDFs the *wave*-variables, a and b for incident and reflected waves, are used [4]. The parametric wave definition is a useful tool that was recently introduced [7, 8] as a parametrization of the traditional wave-variables. At port 0 in a circuit a linear transformation from the Kirchhoff domain $\mathcal{K}$ to the Wave-domain $\mathcal{W}$ is defined as

$$\begin{bmatrix} a_0 \\ b_0 \end{bmatrix} = R_0^{\rho} \begin{bmatrix} R_0^{-1} & 1 \\ R_0^{-1} & -1 \end{bmatrix} \begin{bmatrix} v_0 \\ i_0 \end{bmatrix} = \mathbf{\Phi}_{\mathcal{KW}} \begin{bmatrix} v_0 \\ i_0 \end{bmatrix}. \tag{1}$$

When $\det(\mathbf{\Phi}_{\mathcal{KW}}) = -2R_0^{2\rho-1} \neq 0$ (i.e. $R_0 \neq 0$), the inverse is

$$\begin{bmatrix} v_0 \\ i_0 \end{bmatrix} = \frac{1}{2} R_0^{-\rho} \begin{bmatrix} R_0 & R_0 \\ 1 & -1 \end{bmatrix} \begin{bmatrix} a_0 \\ b_0 \end{bmatrix} = \mathbf{\Phi}_{\mathcal{WK}} \begin{bmatrix} a_0 \\ b_0 \end{bmatrix}. \tag{2}$$

Note that $\mathbf{\Phi}_{\mathcal{WK}}\mathbf{\Phi}_{\mathcal{KW}} = \mathbf{\Phi}_{\mathcal{KW}}\mathbf{\Phi}_{\mathcal{WK}} = \mathbf{I}$ irrespective of the value of ρ, where $\mathbf{I}$ is the identity matrix. By varying the real parameter ρ, a family of transforms that include the standard voltage, power-normalized and current waves may be obtained

$$\rho \triangleq \begin{cases} 1 & \text{voltage waves} \\ 1/2 & \text{power-normalized waves} \\ 0 & \text{current waves} \end{cases}. \tag{3}$$

Plugging the definition (2) into the definition $p_0 = v_0\, i_0$ of instantaneous power at a port 0 gives the wave-domain power

$$p_0 = \frac{1}{4} R_0^{1-2\rho} (a_0^2 - b_0^2). \tag{4}$$

Note that the expression becomes independent of the port resistance when power-normalized waves are used ($\rho = 1/2$) [15].

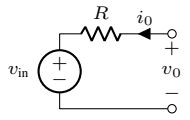


Figure 1: *Res. v. source.*

Table 1: *Source WDF mults.*

Adapted?	f	g
No	$\frac{R-R_0}{R+R_0}$	$\frac{2R_0^\rho}{R+R_0}$
$R_0 = R$	0	$R^{\rho-1}$

1.2. Resistive Voltage Source Derivation

As an example of how to derive wave-domain equations using the parametric wave definition, consider the resistive voltage source (Fig. 1). In the Kirchhoff domain, its constitutive equation is

$$v_{\text{in}} = v_0 - R\, i_0\,, \tag{5}$$

where v_{in} is the voltage source value and R is the resistor's value. Since this source represent an instantaneous geometric relationship, time indices in continuous and discrete time are suppressed.

Plugging in the parametric wave definition (2) and solving for b_0 yields the unadapted wave-domain equation

$$b_0 = \frac{R - R_0}{R + R_0} a_0 + \frac{2R_0^\rho}{R + R_0} v_{\text{in}}\,. \tag{6}$$

This wave-domain equation is adapted by setting $R_0 = R$, yielding the adapted wave-domain equation

$$b_0 = R^{\rho-1} v_{\text{in}}\,. \tag{7}$$

Note that ρ does not affect the adaptation criteria or reflectance (multiplication of incident wave a_0) but rather only contributes to scaling the input v_{in} [15, 8].

In the rest of the paper, we will need to refer directly to the adapted and unadapted multipliers in the resistive voltage source. To enable this we will define a generic wave-domain equation

$$b_0 = f\, a_0 + g\, v_{\text{in}}\,, \tag{8}$$

where f is the reflectance and g is the input scaling. These values are defined for unadapted and adapted resistive voltage sources in Tab. 1. The corresponding signal flow graphs are given in Fig. 2. Here triangles represent multiplications, $+$ symbols represent addition, unfilled semicircles represent wave sources, filled semicircles represent wave sinks. Throughout the paper a shaded background indicates that an element is the root of a WDF tree.

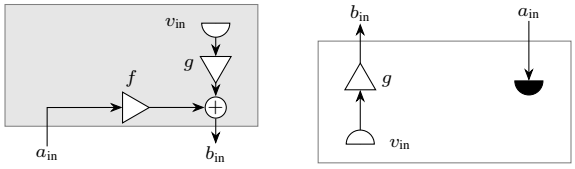
A full review of WDF elements defined with the parametric wave definition is beyond the scope of this paper. The reader is referred to [8] for a full catalog of WDF elements.

1.3. First-Order Difference Equation

In this paper we use multiplication coefficients of a first-order difference equation to show how parameters from continuous-time models, discretization schemes and the choice of wave-variables influence the WDF difference equation for reactive elements. We chose a difference equation of common form [19] and notate coefficients as d_1, n_0 and n_1 (d for "denominator" and n for "numerator") rather than the more common a and b to avoid confusion with wave variable notation [20]

$$b_0[n] = -d_1\, b_0[n-1] + n_0\, a_0[n] + n_1\, a_0[n-1]\,. \tag{9}$$

We use direct-form I [19] filter topologies in all realizations.



(a) *Unadapted resistive source.* (b) *Adapted resistive source.*

Figure 2: *WDF signal flow graphs for resistive sources.*

2. MODELING REACTIVE COMPONENTS

Here we review continuous-time capacitor and inductor models, including traditional LTI models, models proposed by Fettweis, and models used for time-varying components. Noting that they differ only in terms of which quantities are differentiated, we propose novel generalized continuous-time capacitor and inductor models that include the previous three models as special cases.

2.1. Models from Traditional WDF Theory

In traditional WDF theory [4], which is based on classical circuit theory, reactive elements were modeled as ideal. The constitutive equations for these elements are

$$i(t) = C(t)\frac{\mathrm{d}v(t)}{\mathrm{d}t}\,, \tag{10}$$

$$v(t) = L(t)\frac{\mathrm{d}i(t)}{\mathrm{d}t}\,, \tag{11}$$

where C is the capacitor's capacitance in Farads (F) and L is the inductor's inductance in Henries (H).

2.2. Fettweis' Lossless Models

In [21], Fettweis proposed following time-varying models

$$i(t) = \sqrt{C(t)}\,\frac{\mathrm{d}}{\mathrm{d}t}\left(\sqrt{C(t)}\,v(t)\right)\,, \tag{12}$$

$$v(t) = \sqrt{L(t)}\,\frac{\mathrm{d}}{\mathrm{d}t}\left(\sqrt{L(t)}\,i(t)\right)\,, \tag{13}$$

for a capacitor and inductor respectively. These models are lossless [16] as will be shown in §2.4.

2.3. Models from Circuit Theory

In circuit theory time-varying reactive models are given by

$$i(t) = \frac{\mathrm{d}q(t)}{\mathrm{d}t} = \frac{\mathrm{d}(C(t)v(t))}{\mathrm{d}t} = C(t)\frac{\mathrm{d}v(t)}{\mathrm{d}t} + \frac{\mathrm{d}C(t)}{\mathrm{d}t}v(t)\,, \tag{14}$$

$$v(t) = \frac{\mathrm{d}\phi(t)}{\mathrm{d}t} = \frac{\mathrm{d}(L(t)i(t))}{\mathrm{d}t} = L(t)\frac{\mathrm{d}i(t)}{\mathrm{d}t} + \frac{\mathrm{d}L(t)}{\mathrm{d}t}i(t)\,, \tag{15}$$

for a capacitor and inductor respectively [22, p. 40, 47].

2.4. Generalized Time-Varying Models

We propose a generalized time-varying model of a lumped reactive element by incorporating the real parameter λ

$$i(t) = C^{1-\lambda}(t)\,\frac{\mathrm{d}}{\mathrm{d}t}\left(C^{\lambda}(t)\,v(t)\right)\,, \tag{16}$$

$$v(t) = L^{1-\lambda}(t)\,\frac{\mathrm{d}}{\mathrm{d}t}\left(L^{\lambda}(t)\,i(t)\right)\,, \tag{17}$$

for a capacitor and inductor respectively. These models include the circuit-theoretic model ($\lambda = 1$), Fettweis model ($\lambda = 1/2$) and traditional model ($\lambda = 0$) as special cases.

Looking at the instantaneous power $p(t) = v(t)i(t)$ for these two models, we obtain the following expressions

$$p_{C,\lambda}(t) = \frac{\mathrm{d}}{\mathrm{d}t}E_C(t) + 2E_C(t)\left(\lambda - \frac{1}{2}\right)\frac{1}{C(t)}\frac{\mathrm{d}C}{\mathrm{d}t}\,, \quad (18)$$

$$p_{L,\lambda}(t) = \frac{\mathrm{d}}{\mathrm{d}t}E_L(t) + 2E_L(t)\left(\lambda - \frac{1}{2}\right)\frac{1}{L(t)}\frac{\mathrm{d}L}{\mathrm{d}t}\,, \quad (19)$$

where $E_C(t) = C(t)v^2(t)/2$ and $E_L(t) = L(t)i^2(t)/2$ are the non-negative energies of the capacitor and the inductor. Note that the instantaneous power reduces to the derivative of the energy in the case of Fettweis ($\lambda = 1/2$) and thus it is lossless [21, 16].

3. DISCRETIZATION

Here we review traditional LTI discretization via the bilinear transform (BLT) and discretize our proposed models using the new α-transform discretization scheme. The results of these discretizations (24)–(43) are collected in Tab. 2 at the end of the paper.

The traditional way to discretize an element in WDF theory [4] involves first transforming its constitutive equation to the wave domain via (2) and then discretizing it using the BLT. Using this discretization on a capacitor yields (24) and on an inductor yields (34). The BLT is derived from the unidirectional Laplace transform, which assumes LTI and steady-state [23]. For most audio circuits, neither of these assumptions hold true and BLT discretization will cause errors.

Instead of using the BLT we use the α-transform discretization scheme [10, 8]. The α-transform discretization scheme is a generalization that encompasses the trapezoidal discretization scheme ($\alpha = 1$), backward-Euler ($\alpha = 0$) and forward-Euler ($\alpha \to \infty$) as special cases. Like trapezoidal integration, it does not depend on time-invariance or steady-state.

3.1. Discretizing the Generalized Model

To demonstrate how the α-transform discretization scheme is applied, we discretize our generalized capacitor model (16). Before we go into the general case we show how to apply the trapezoidal numerical scheme ($\alpha = 1$). As in traditional WDF theory we begin by applying (2) to transform (16) into the wave-domain

$$\frac{a_0(t) - b_0(t)}{R_0^{\rho}(t)C^{1-\lambda}(t)} = \frac{\mathrm{d}}{\mathrm{d}t}\left(\frac{a_0(t) + b_0(t)}{R_0^{\rho-1}(t)C^{-\lambda}(t)}\right). \quad (20)$$

Each side is now integrated over the time interval $[T(n-1), Tn]$, where T is the sampling period and n is the discrete-time sample index. The trapezoidal rule [24] is used to approximate the integrated expression on the left

$$\int_{\mathrm{T}(n-1)}^{\mathrm{T}n} \frac{a_0(t) - b_0(t)}{R_0^{\rho}(t)C^{1-\lambda}(t)}\,\mathrm{d}t \approx \frac{T}{2}\left(\frac{a_0[n] - b_0[n]}{R_0^{\rho}[n]\,C^{1-\lambda}[n]} + \frac{a_0[n-1] - b_0[n-1]}{R_0^{\rho}[n-1]\,C^{1-\lambda}[n-1]}\right) \quad (21)$$

and the first fundamental theorem of calculus [25] to calculate the expression on the right over the same time interval

$$\int_{\mathrm{T}(n-1)}^{\mathrm{T}n} \frac{\mathrm{d}}{\mathrm{d}t}\left(\frac{a_0(t) + b_0(t)}{R_0^{\rho-1}(t)C^{-\lambda}(t)}\right)\mathrm{d}t = \frac{a_0[n] + b_0[n]}{R_0^{\rho-1}[n]C^{-\lambda}[n]} - \frac{a_0[n-1] + b_0[n-1]}{R_0^{\rho-1}[n-1]C^{-\lambda}[n-1]}. \quad (22)$$

Combining (21) and (22) and solving for $b_0[n]$ gives us the multiplication coefficients shown in equation (32), for $\alpha = 1$. The multiplication coefficients for the BLT based capacitor is shown in (24). These two sets of multiplier coefficients differ with respect to time indices of the port resistance, and scaling of capacitance, not dissimilar to the results obtained in [26].

The generalized difference equation can be obtained by discretizing (20) using the α-transform discretization scheme. As shown in [10] a time-varying system of the form $\dot{x}(t) = y(x,t)$ may be discretized using the difference equation

$$(1+\alpha)\,x[n] - (1+\alpha)\,x[n-1] = T\,y[n] + \alpha T\,y[n-1] \quad (23)$$

Here $\dot{x}$ is the right-hand side of (20) and y is the left-hand side. Carrying this out and solving for $b_0[n]$ yields the multiplier coefficients (32). The inductor can be discretized using the same method, yielding the multiplier coefficients (42). Tab. 2 shows general discretizations, the three special cases (Traditional, Fettweis, and Time-Varying) and the LTI discretization, as well as the *adapted* ($R_0[n]$ chosen to set $n_0[n] = 0$) versions of each, for both the capacitor and the inductor.

4. CASE STUDY: TIME-VARYING RESISTANCES

In this section we simulate a simple series RC circuit involving a time-varying resistance. Depending on whether the capacitor is at the root of the WDF tree or not, this may cause simulation inaccuracies using the BLT. By discretizing the capacitor using an α-discretization, these inaccuracies are avoided.

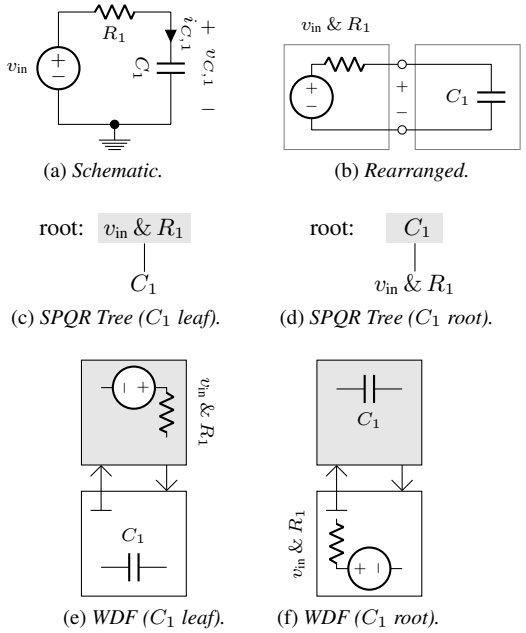
In both cases, we allow the circuits to settle to a dc solution, change the value of a resistor, and examine the effect on the output variables under different discretization schemes [3]. In each case we analytically derive the dc solution of the WDF so that we can set initial conditions "at dc" without any wait.

4.1. Circuit Description

Consider the series RC circuit whose schematic is shown in Fig. 3a. In this circuit, an ideal voltage source v_{in}, resistor R_1, and capacitor C_1 are connected in series. The capacitor is characterized by voltage $v_{C,1}$ and current $i_{C,1}$. Taking v_{in} as the input and $v_{C,1}$ as the output of the circuit, it forms a first-order (6 dB/octave) lowpass filter with a cutoff frequency of

$$f_{\text{cutoff}} = 1/(2\pi R_1 C_1)\ \text{Hz}\,. \quad (44)$$

Fig. 3b shows the series RC circuit decomposed into two one-port devices: the capacitor C_1 and a resistive voltage source composed of v_{in} and R_1. In such a simple circuit, the SPQR tree representing how the components are connected is trivial [27, 7]. However, this tree may be oriented in two different ways: with the resistive source at the root (Fig. 3c) or with C_1 at the root (Fig. 3d). These two tree orientations correspond to two different WDF diagrams: Figs. 3e and 3f respectively.



(a) *Schematic.* (b) *Rearranged.*

(c) *SPQR Tree (C_1 leaf).* (d) *SPQR Tree (C_1 root).*

(e) *WDF (C_1 leaf).* (f) *WDF (C_1 root).*

Figure 3: *Series RC circuit schematic, decomposition into ports, two possible SPQR trees, and two corresponding WDF diagrams.*

The signal flow graphs corresponding to these WDFs use the same notation as before (as well as delays, z^{-1}) and are shown in Fig. 4. In each case the output variables are formed by (1)

$$v_{C,1} = R_0^{1-\rho}(a_{C,1} + b_{C,1})/2\,, \tag{45}$$

$$i_{C,1} = R_0^{-\rho}(a_{C,1} - b_{C,1})/2\,. \tag{46}$$

Note however that the port resistance R_0 and multipliers n_0, n_1, d_1, f, and g will be different. For example, n_1 in Fig. 4a and Fig. 4b are not the same. Port resistance and multiplier values will be given later as we test out the different discretization techniques.

4.1.1. Description of DC Behavior

In continuous time, the dc behavior (assuming $v_{\text{in}}(t) = 1\,\text{V}, \forall t < 0$) of the series RC circuit (Fig. 3a) is easy to predict. At dc, capacitors "look like" open circuits (they have "infinite" resistance at dc). Since no current may flow through C_1 at dc, no current may flow through R_1 either, so no voltage may develop across R_1. This leads to a dc solution for the capacitor network variables:

$$V_{C,1} = 1\,\text{V}\,, \tag{47}$$

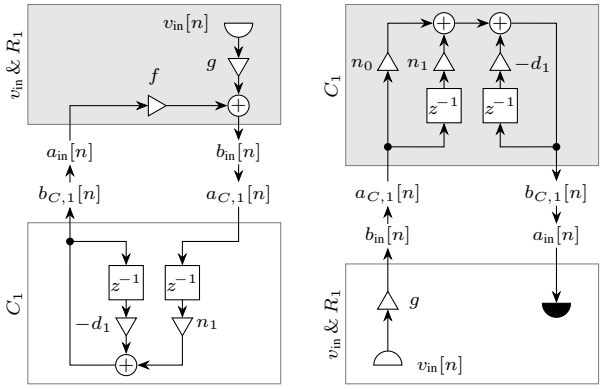
$$I_{C,1} = 0\,\text{A}\,. \tag{48}$$

Here and throughout, capital letters indicate dc quantities. In Fig. 5 a time-domain simulation of the circuit settling towards dc from zero initial conditions in response to the 1 V input is shown. Here and throughout, the sampling rate is $f_s = 44\,100$ Hz.

4.1.2. Finding DC Solution of WDFs

We will use simple signal flow graph manipulation techniques [28, 29] to solve the dc solution of our simple WDFs.[1] First we recall a few elementary transformations on signal flow graphs:

[1] In general, for more complicated circuits, it could be more convenient to use matrix-based techniques [30] to find dc solutions.



(a) C_1 *as leaf.* (b) C_1 *as root.*

Figure 4: *WDF signal flow graphs for different tree orientations.*

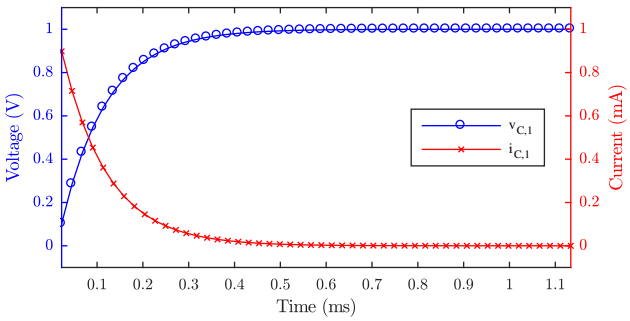


Figure 5: *RC circuit settling to dc. $R_1 = 1\,\text{k}\Omega$ and $C_1 = 0.1\,\mu\text{F}$.*

- Two multipliers χ and ζ in series may be replaced by a single multiplier $\chi\,\zeta$.
- Two multipliers χ and ζ in parallel may be replaced by a single multiplier $\chi + \zeta$.
- A self-loop though a multiplier χ may be replaced by a multiplier $1/(1-\chi)$. This creates a singularity when $\chi = 1$.

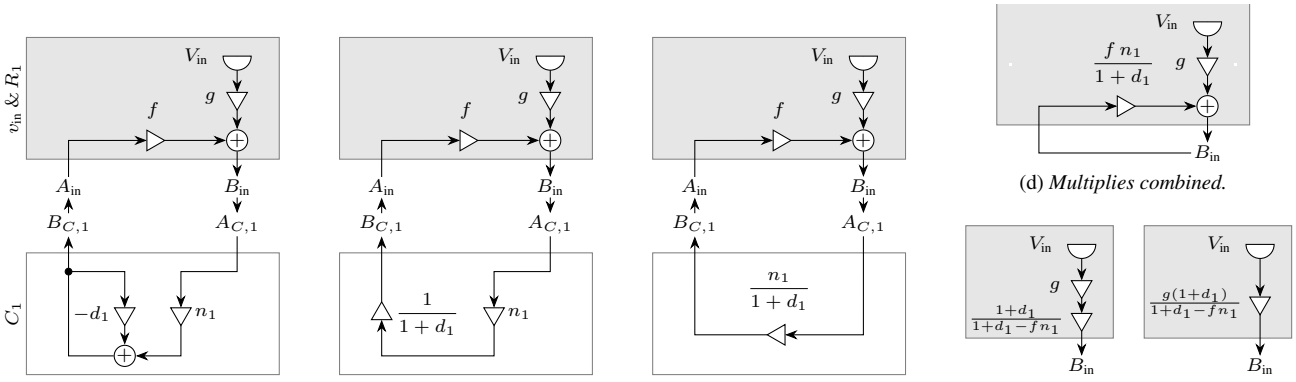
After a circuit has converged to dc, delays should be "transparent" so their outputs should equal their inputs, i.e., they can be replaced by a unity-gain, delay-free connection.

4.1.3. DC Solution with C_1 as Leaf

According to this logic, Fig. 4a at dc is shown in Fig. 6a. We see that a self-loop through $-d_1$ can be removed, giving Fig. 6b. Here the multiplier $1/(1+d_1)$ and n_1 can be combined, giving Fig. 6c. Here the multipliers $n_1/(1+d_1)$ and f can be combined, giving Fig. 6d. Here the self-loop through $fn_1/(1+d_1)$ can be removed, giving Fig. 6e. Finally, the gains g and $(1+d_1)/(1+d_1-fn_1)$ can be combined, giving Fig. 6f which solves for B_{in} in terms of V_{in}. This can be used to find the dc solution

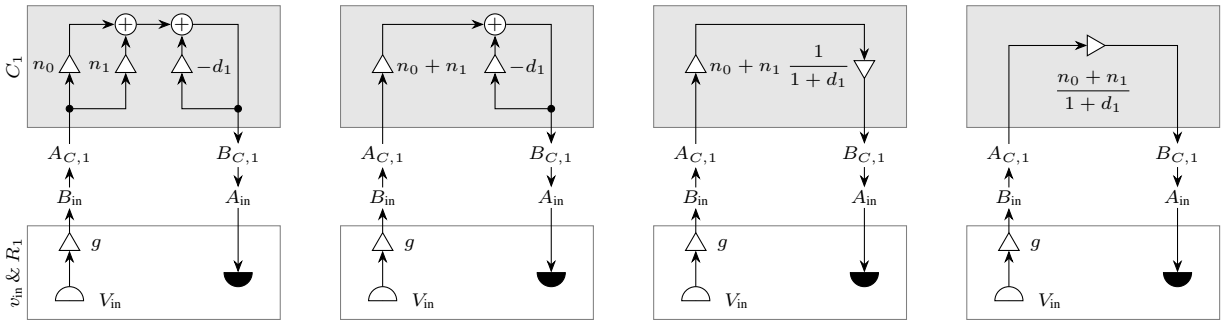
$$A_{C,1} = B_{\text{in}} = \frac{g(1+d_1)}{1+d_1-fn_1} V_{\text{in}} \tag{49}$$

$$A_{\text{in}} = B_{C,1} = \frac{n_1}{1+d_1} A_{C,1} = \frac{g\,n_1}{1+d_1-fn_1} V_{\text{in}}\,. \tag{50}$$



(a) *WDF at dc.* (b) *Self-loop removed.* (c) *Multiplies combined.* (d) *Multiplies combined.* (e) *Self-loop…* (f) *Mults…*.

Figure 6: *Finding dc solution for series RC WDF with source at root and C_1 at leaf.*



(a) *WDF at dc.* (b) *Multiplies combined.* (c) *Self-loop removed.* (d) *Multiplies combined.*

Figure 7: *Finding dc solution for series RC WDF with C_1 at root and source at leaf.*

Notice that $n_1/(1+d_1) = 1$. Recall that $B_{C,1}$ and $A_{C,1}$ are also the values stored in the two delays. That is, they are the specific values that should be stored in those delays to set up the WDF as if it has converged to steady-state. These quantities are combined using (2) to find the dc solution for the network in terms of the WDF multipliers and input:

$$V_{C,1} = \frac{g(1+d_1+n_1)R_0^{1-\rho}}{2(1+d_1-fn_1)}V_{\text{in}} \tag{51}$$

$$I_{C,1} = \frac{g(1+d_1-n_1)R_0^{-\rho}}{2(1+d_1-fn_1)}V_{\text{in}}\,. \tag{52}$$

Now we can check these values, making sure that they correspond to the continuous-time steady-state solution (47)–(48). This is done by plugging in the values for the multipliers from each discretization (see Tab. 2). We will do this in the general case only, since when R_1 and C_1 are not changing (remember we are finding a steady-state solution) then it can encompass all the other discretizations mentioned, including the LTI ones.

Plugging in the multiplier values from the adapted generalized model (33), the unadapted resistive voltage source (Tab. 1) and circuit input values (51)–(52)

$$V_{\text{in}} = 1\,\text{V} \qquad g = 2R_0^{\rho}/(R_1+R_0)$$
$$R_0 = T/(C_1(1+\alpha)) \qquad n_1 = (\alpha+1)/2$$
$$f = (R_1-R_0)/(R_1+R_0) \qquad d_1 = (\alpha-1)/2$$

yields the the dc wave solutions

$$A_{C,1} = B_{\text{in}} = A_{\text{in}} = B_{C,1} = R_0^{\rho-1}\,. \tag{53}$$

combining these solutions and the wave definition (2) yields the dc Kirchhoff solution

$$V_{C,1} = 1\,\text{V} \tag{54}$$

$$I_{C,1} = 0\,\text{A}\,. \tag{55}$$

This matches the continuous-time dc solution (47)–(48), which is expected because the entire family of α-discretizations (except the degenerate $\alpha = -1$) should be consistent (for the LTI versions, dc is matched). As a sanity check we can run a simulation for a long time (many times longer than the time constant of the circuit) to confirm both the wave-domain (53) and Kirchhoff-domain dc solutions (51)–(52).

4.1.4. DC Solution with C_1 as Root

Fig. 4b at dc is shown in Fig. 7a. We see that the parallel multipliers n_0 and n_1 can be combined, giving Fig. 7b. Here the self loop through $-d_1$ can be removed, giving Fig. 7c. Finally, the series multipliers n_0+n_1 and $1/(1+d_1)$ can be combined, giving Fig. 7d which solves for the dc wave variables:

$$A_{C,1} = B_{\text{in}} = g\,V_{\text{in}}\,, \tag{56}$$

$$A_{\text{in}} = B_{C,1} = \frac{n_0+n_1}{1+d_1}A_{C,1}\,. \tag{57}$$

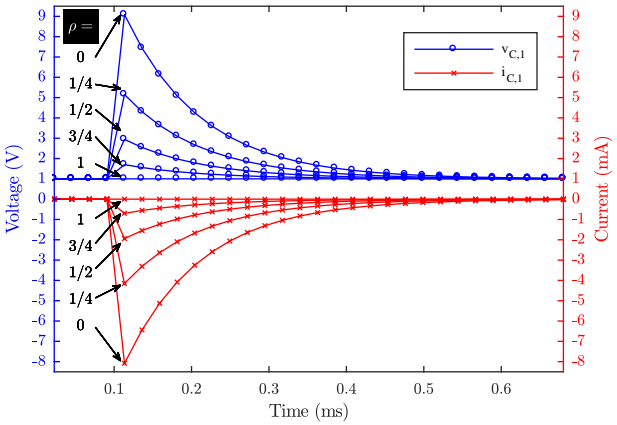


Figure 8: *Changing R_1 after 5 samples with various $\rho \in \{0, 1/4, 1/2, 3/4, 1\}$. C_1 at root of WDF tree. $\alpha = 1$.*

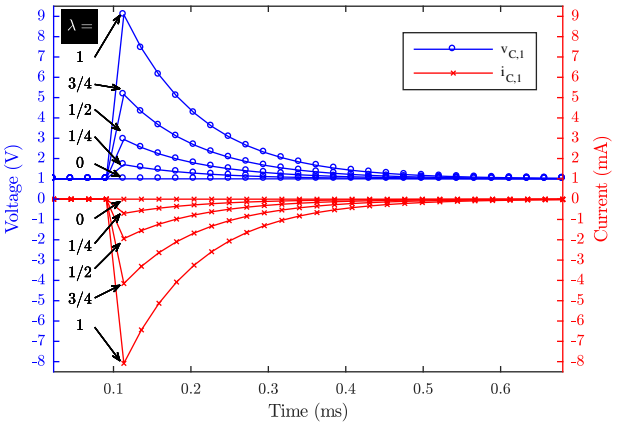


Figure 9: *Changing C_1 after 5 samples with various $\lambda \in \{0, 1/4, 1/2, 3/4, 1\}$. $\alpha = 1$ and $\rho = 1$.*

Notice that $(n_0+n_1)/(1+d_1) = 1$. These quantities are combined using (2) to find the dc solution for the network in terms of the WDF multipiers and input

$$V_{C,1} = \frac{(1 + n_0 + n_1)R_0^{1-\rho}}{2(1 + d_1)} V_{\text{in}}\,, \tag{58}$$

$$I_{C,1} = \frac{(1 - n_0 + n_1)R_0^{-\rho}}{2(1 + d_1)} V_{\text{in}}\,. \tag{59}$$

Plugging in the multiplier values for the adapted resistive source (Tab. 1) and discretized capacitor (32) as before

$$\begin{aligned} V_{\text{in}} &= 1\,\text{V} & n_0 &= (T - (1+\alpha)R_0C_1)/(T + (1+\alpha)R_0C_1) \\ R_0 &= R_1 & n_1 &= (T\alpha + (1+\alpha)R_0C_1)/(T + (1+\alpha)R_0C_1) \\ g &= R_1^{\rho-1} & d_1 &= (T\alpha - (1+\alpha)R_0C_1)/(T + (1+\alpha)R_0C_1) \end{aligned}$$

yields the dc wave solutions

$$A_{C,1} = B_{\text{in}} = A_{\text{in}} = B_{C,1} = R_1^{\rho-1}\,. \tag{60}$$

Note again that $B_{C,1}$ and $A_{C,1}$ are the values to be stored in the two delays. Combining (60) and the wave definition (2) yields the dc Kirchhoff solution

$$V_{C,1} = 1\,\text{V} \tag{61}$$

$$I_{C,1} = 0\,\text{A}\,. \tag{62}$$

Again this matches the continuous-time dc solution (47)–(48) and sanity check simulations also confirm this.

4.1.5. Time-Varying Simulations

We now run a simulation of the series RC circuit with time-varying resistor values. In this simulation, the resistor and capacitor values vary as a function of the sample index n according to

$$R_1[n] = \begin{cases} 100\,\Omega\,, & n < 5 \\ 1\,\text{k}\Omega\,, & n \geq 5 \end{cases} \tag{63}$$

$$C_1[n] = 0.1\,\mu\text{F}\,. \tag{64}$$

Recalling the equation for the filter's cutoff frequency (44), this circuit acts as a filter whose cutoff frequency varies with time

$$f_{\text{cutoff}} \approx \begin{cases} 15.9\,\text{kHz}\,, & n < 5 \\ 1.59\,\text{kHz}\,, & n \geq 5 \end{cases}\,. \tag{65}$$

WDF simulations of this circuit are made using the computational strutures in Fig. 4, using BLT discretizations and the generalized discretizations (with $\alpha = 1$). Because the capacitance does not change over time, the value of λ does not matter. We start the simulations at their dc solutions as calculated in the previous section. That is, the delay registers are loaded at $n = 0$ with the appropriate wave dc solutions (53) or (60).

For the case where C_1 is the leaf of the tree (Fig. 4b), there are no errors for any values of ρ for either the BLT or the generalized discretization. This can be explained by comparing (25) and (27), which are equivalent when the capacitor's value is static. Surprisingly, even though the BLT discretization should not be valid for time-varying circuits, it is acceptable for all values of ρ when C_1 is a leaf. The generalized transform, for all values of α and λ, has no errors since it has been discretized correctly.

In Fig. 8 simulations are shown using BLT discretizations (24) for the case where C_1 is the root of the tree (Fig. 4b). For voltage-wave BLT discretization ($\rho = 1$) we get the correct response, but for BLT discretization for any other ρ there are spurious transients. This discrepancy can be explained by comparing (24) and (26). Even for a static capacitor value, the BLT does not match the trapezoidal rule for the n_1 and d_1 coefficients except for the case $\rho = 1$ (voltage waves). Notice that for the inductor equations, this property would only hold for $\rho = 0$ (current waves).

5. CASE STUDY: TIME-VARYING REACTANCES

Now we study the series RC circuit with a time-varying capacitor value. In this simulation, the capacitor value vary as a function of the sample index n according to

$$R_1[n] = 1\,\text{k}\Omega \tag{66}$$

$$C_1[n] = \begin{cases} 1.0\,\mu\text{F}\,, & n < 5 \\ 0.1\,\mu\text{F}\,, & n \geq 5 \end{cases}\,. \tag{67}$$

Recalling the equation for the filter's cutoff frequency (44), this circuit acts as a filter whose cutoff frequency varies as

$$f_{\text{cutoff}} \approx \begin{cases} 0.159\,\text{kHz}\,, & n < 5 \\ 1.59\,\text{kHz}\,, & n \geq 5 \end{cases}\,. \tag{68}$$

WDF simulations of this circuits are made using the computational structures in Fig. 4, using the generalized discretization

(with $\alpha = 1$). Again we assume that the simulation has converged to a dc solutions (53) or (60) by time $n = 0$.

For both cases, where C_1 is the root of the tree (Fig. 2b) or the leaf of the tree (Fig. 2b), Fig. 9 shows simulations using (32) or (33). Since the generalized discretization is used correctly, there is no difference in behavior between the two configurations. By varying λ a family of responses are obtained. For $\lambda = 0$ (traditional model), there is no transient, i.e., the capacitor's state is maintained under time-varying conditions. For $\lambda = 1/2$ (Fettweis model), the energy is maintained but there *is* a transient. For $\lambda = 1$ (time-varying circuit theory model), the transient is the largest. We cannot necessarily say that one behavior is intrinsically the best; the appropriate choice of λ will depend on the desired behavior.

6. CONCLUSIONS

In this paper we argue for the use of proper numerical schemes since the theory that lies at the foundation of the discretization methods used in WDFs is invalid under time-varying conditions. Numerical schemes like the α-transform discretization and trapezoidal rule, however, have no problems with time-varying systems when applied properly.

If your goal is to model LTI circuits, traditional bilinear transform based discretizations generally suffice. In the case where a reactance is placed at the root of a tree but other elements may change its port resistance, you should discretize it using the trapezoidal method in order to avoid inaccuracies in the simulation.

When faced with the problem of simulating time-varying reactances we would recommend to gather data and use any number of optimization methods to get an estimate for a suitable value of λ. For some circuits the effects caused by time-varying reactances make up an important part of the sound, such as is the case with stepped filters, or even the intrinsic operation of the device, e.g. condenser microphone. Conversely in other cases the "smoothness" and/or reduction of transient may be a desired behavior. In any case, the parameter λ in our proposed continuous-time model allows the algorithm designer to control the behavior.

In closing, the combination of the novel generalized continuous-time capacitor and inductor models, α-discretizations, and parametric wave definition gives new tools that may be useful when creating audio effects and gives audio dsp designers more control over how energy is stored in discretized reactances.

7. REFERENCES

[1] S. Bilbao, "Time-varying generalizations of allpass filters," *IEEE Signal Process. Lett.*, vol. 12, no. 5, pp. 376–379, May 2005.

[2] J. Laroche, "On the stability of time-varying recursive filters," *J. Audio Eng. Soc. (JAES)*, vol. 55, no. 6, pp. 460–471, June 2007.

[3] A. Wishnick, "Time-varying filters for musical applications," in *Proc. 17th Int. Conf. Digital Audio Effects (DAFx-14)*, Erlangen, Germany, Sept. 2014, pp. 69–76.

[4] A. Fettweis, "Wave digital filters: Theory and practice," *Proc. IEEE*, vol. 74, no. 2, pp. 270–327, Feb. 1986.

[5] K. J. Werner, J. O. Smith III, and J. S. Abel, "Wave digital filter adaptors for arbitrary topologies and multiport linear elements," in *Proc. 18th Int. Conf. Digital Audio Effects (DAFx-15)*, Trondheim, Norway, Nov. 2015, pp. 379–386.

[6] K. J. Werner, V. Nangia, J. O. Smith III, and J. S. Abel, "Resolving wave digital filters with multiple/multiport nonlinearities," in *Proc. 18th Int. Conf. Digital Audio Effects (DAFx-15)*, Trondheim, Norway, Nov. 2015, pp. 387–394.

[7] K. J. Werner, A. Bernardini, J. O. Smith III, and A. Sarti, "Modeling circuits with arbitrary topologies and active linear multiports using wave digital filters," *IEEE Trans. Circuits. Syst. I: Reg. Papers*, June 2018, In Press, DOI: 10.1109/TCSI.2018.2837912.

[8] K. J. Werner, *Virtual Analog Modeling of Audio Circuitry Using Wave Digital Filters*, Ph.D. diss., Stanford Univ., CA, USA, Dec. 2016.

[9] A. Bernardini and A. Sarti, "Biparametric wave digital filters," *IEEE Trans. Circuits Syst. I: Reg. Papers*, vol. 64, no. 7, pp. 1826–1838, July 2017.

[10] F. G. Germain and K. J. Werner, "Design principles for lumped model discretisation using Möbius transforms," in *Proc. 18th Int. Conf. Digital Audio Effects (DAFx-15)*, Trondheim, Norway, November 2015, pp. 371–378.

[11] D. Fränken and K. Ochs, "Synthesis and design of passive Runge-Kutta methods," *Int. J. Electron. Commun. (AEÜ)*, vol. 55, no. 6, pp. 417–425, 2001.

[12] Ó. Bogason, "Modeling audio circuits containing typical nonlinear components with wave digital filters," M.A. thesis, McGill Univ., May 2018.

[13] H. W. Strube, "Time-varying wave digital filters for modeling analog systems," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. 30, no. 6, pp. 864–868, Dec. 1982.

[14] H. W. Strube, "Time-varying wave digital filters and vocal-tract models," in *IEEE Int. Conf. Acoust., Speech, Signal Process. (ICASSP)*, May 1982, vol. 7, pp. 923–926.

[15] G. Kubin, "On the stability of wave digital filters with time-varying coefficients," in *Proc. 7th Europ. Conf. Circuit Theory Design (ECCTD-85)*, Prague, Czecho-Slovakia, Sept. 1985, pp. 499–502.

[16] S. Bilbao, *Wave and Scattering Methods for Numerical Simulation*, John Wiley and Sons, Ltd, New York, 2004.

[17] D. Fränken and K. Ochs, "Automatic step-size control in wave digital simulation using passive numerical integration methods," *Int. J. Electron. Commun. (AEÜ)*, vol. 58, pp. 391–401, 2004.

[18] M. J. Olsen, K. J. Werner, and F. G. Germain, "Network variable preserving step-size control in wave digital filters," in *Proc. 20th Int. Conf. Digital Audio Effects (DAFx-17)*, Edinburgh, UK, Sept. 2017, pp. 207–207.

[19] J. O. Smith, *Introduction to Digital Filters with Audio Applications*, W3K Publishing, https://ccrma.stanford.edu/ jos/filters/, 2007.

[20] T. S. Stilson, *Efficiently-Variable Non-Oversampled Algorithms in Virtual-Analog Music Synthesis—A Root-Locus Perspective*, Ph.D. diss., Stanford Univ., California, USA, June 2006.

[21] A. Fettweis, "Robust numerical integration using wave digital concepts," in *Proc. 5th DSPS Educators Conf.*, Tokyo, Japan, Sept. 2003, pp. 23–32.

[22] C. A. Desoer and E. S. Kuh, *Basic Circuit Theory*, McGraw-Hill, New York, NY, USA, 1969.

[23] M. Vetterli, J. Kovačević, and V. K. Goyal, *Foundations of Signal Processing*, Cambridge Univ. Press, 2014, Online: `http://fourierandwavelets.org/`.

[24] M. Parviz, *Fundamentals of Engineering Numerical Analysis*, Cambridge Univ. Press, Cambridge, UK, 2nd edition, 2010.

[25] A. Howard, *Calculus: A New Horizon*, Wiley, New York, NY, USA, 6th edition, 1998.

[26] J. S. Abel and D. P. Berners, "The time-varying bilinear transform," in *Proc. 141st Conv. Audio Eng. Soc. (AES)*, Los Angeles, CA, Sept. 2016, pp. 9686–9697, conv. paper #9686.

[27] D. Fränken, J. Ochs, and K. Ochs, "Generation of wave digital structures for networks containing multiport elements," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 52, no. 3, pp. 586–596, Mar. 2005.

[28] S. J. Mason, "Feedback theory—further properties of signal flow graphs," *Proc. IRE*, vol. 44, no. 7, pp. 920–926, July 1956.

[29] K. J. Werner, J. S. Abel, and J. O. Smith, "More cowbell: a physically-informed, circuit-bendable, digital model of the TR-808 cowbell," in *Proc. 137th Conv. Audio Eng. Soc. (AES)*, Los Angeles, CA, USA, Oct. 2014, conv. paper #9207.

[30] A. Sarti and G. De Sanctis, "Systematic methods for the implementation of nonlinear wave-digital structures," *IEEE Trans. Circuits Syst. I: Reg. Papers*, vol. 56, no. 2, pp. 460–472, Feb. 2009.

Eqn.	Model	Discret.	Adapted?	n_0	n_1	d_1
(24)	LTI	BLT	No	$\frac{T-2R_0[n]C[n]}{T+2R_0[n]C[n]}$	1	$\frac{T-2R_0[n]C[n]}{T+2R_0[n]C[n]}$
(25)	LTI	BLT	$R_0[n] = \frac{T}{2C[n]}$	0	1	0
(26)	Traditional	Trap.	No	$\frac{T-2R_0[n]C[n]}{T+2R_0[n]C[n]}$	$\frac{T+2R_0[n-1]C[n-1]}{T+2R_0[n]C[n]} \left(\frac{R_0[n]}{R_0[n-1]}\right)^{\rho} \left(\frac{C[n]}{C[n-1]}\right)$	$\frac{T-2R_0[n-1]C[n-1]}{T+2R_0[n]C[n]} \left(\frac{R_0[n]}{R_0[n-1]}\right)^{\rho} \left(\frac{C[n]}{C[n-1]}\right)$
(27)	Traditional	Trap.	$R_0[n] = \frac{T}{2C[n]}$	0	$\left(\frac{C[n]}{C[n-1]}\right)^{1-\rho}$	0
(28)	Fettweis	Trap.	No	$\frac{T-2R_0[n]C[n]}{T+2R_0[n]C[n]}$	$\frac{T+2R_0[n-1]C[n-1]}{T+2R_0[n]C[n]} \left(\frac{R_0[n]}{R_0[n-1]}\right)^{\rho} \sqrt{\frac{C[n]}{C[n-1]}}$	$\frac{T-2R_0[n-1]C[n-1]}{T+2R_0[n]C[n]} \left(\frac{R_0[n]}{R_0[n-1]}\right)^{\rho} \sqrt{\frac{C[n]}{C[n-1]}}$
(29)	Fettweis	Trap.	$R_0[n] = \frac{T}{2C[n]}$	0	$\left(\frac{C[n]}{C[n-1]}\right)^{-\rho} \sqrt{\frac{C[n]}{C[n-1]}}$	0
(30)	Time-Varying	Trap.	No	$\frac{T-2R_0[n]C[n]}{T+2R_0[n]C[n]}$	$\frac{T+2R_0[n-1]C[n-1]}{T+2R_0[n]C[n]} \left(\frac{R_0[n]}{R_0[n-1]}\right)^{\rho}$	$\frac{T-2R_0[n-1]C[n-1]}{T+2R_0[n]C[n]} \left(\frac{R_0[n]}{R_0[n-1]}\right)^{\rho}$
(31)	Time-Varying	Trap.	$R_0[n] = \frac{T}{2C[n]}$	0	$\left(\frac{C[n]}{C[n-1]}\right)^{-\rho}$	0
(32)	Generalized	α	No	$\frac{T-(1+\alpha)R_0[n]C[n]}{T+(1+\alpha)R_0[n]C[n]}$	$\frac{T\alpha+(1+\alpha)R_0[n-1]C[n-1]}{T+(1+\alpha)R_0[n]C[n]} \left(\frac{R_0[n]}{R_0[n-1]}\right)^{\rho} \left(\frac{C[n]}{C[n-1]}\right)^{1-\lambda}$	$\frac{T\alpha-(1+\alpha)R_0[n-1]C[n-1]}{T+(1+\alpha)R_0[n]C[n]} \left(\frac{R_0[n]}{R_0[n-1]}\right)^{\rho} \left(\frac{C[n]}{C[n-1]}\right)^{1-\lambda}$
(33)	Generalized	α	$R_0[n] = \frac{T}{(1+\alpha)C[n]}$	0	$\frac{1}{2}\left(\frac{C[n]}{C[n-1]}\right)^{1-\rho-\lambda}(1+\alpha)$	$\frac{1}{2}\left(\frac{C[n]}{C[n-1]}\right)^{1-\rho-\lambda}(1-\alpha)$

(a) Capacitor discretizations and coefficients.

Eqn.	Model	Discret.	Adapted?	n_0	n_1	d_1
(34)	LTI	BLT	No	$\frac{2L[n]-TR_0[n]}{2L[n]+TR_0[n]}$	-1	$-\frac{2L[n]-TR_0[n]}{2L[n]+TR_0[n]}$
(35)	LTI	BLT	$R_0[n] = \frac{2L[n]}{T}$	0	-1	0
(36)	Traditional	Trap.	No	$\frac{2L[n]-TR_0[n]}{2L[n]+TR_0[n]}$	$-\frac{2L[n-1]+TR_0[n-1]}{2L[n]+TR_0[n]} \left(\frac{R_0[n]}{R_0[n-1]}\right)^{\rho} \left(\frac{L[n]}{L[n-1]}\right)$	$-\frac{2L[n-1]-TR_0[n-1]}{2L[n]+TR_0[n]} \left(\frac{R_0[n]}{R_0[n-1]}\right)^{\rho} \left(\frac{L[n]}{L[n-1]}\right)$
(37)	Traditional	Trap.	$R_0[n] = \frac{2L[n]}{T}$	0	$-\left(\frac{L[n]}{L[n-1]}\right)^{\rho}$	0
(38)	Fettweis	Trap.	No	$\frac{2L[n]-TR_0[n]}{2L[n]+TR_0[n]}$	$-\frac{2L[n-1]+TR_0[n-1]}{2L[n]+TR_0[n]} \left(\frac{R_0[n]}{R_0[n-1]}\right)^{\rho} \sqrt{\frac{L[n]}{L[n-1]}}$	$-\frac{2L[n-1]-TR_0[n-1]}{2L[n]+TR_0[n]} \left(\frac{R_0[n]}{R_0[n-1]}\right)^{\rho} \sqrt{\frac{L[n]}{L[n-1]}}$
(39)	Fettweis	Trap.	$R_0[n] = \frac{2L[n]}{T}$	0	$-\left(\frac{L[n]}{L[n-1]}\right)^{\rho} \sqrt{\frac{L[n-1]}{L[n]}}$	0
(40)	Time-Varying	Trap.	No	$\frac{2L[n]-TR_0[n]}{2L[n]+TR_0[n]}$	$-\frac{2L[n-1]+TR_0[n-1]}{2L[n]+TR_0[n]} \left(\frac{R_0[n]}{R_0[n-1]}\right)^{\rho}$	$-\frac{2L[n-1]-TR_0[n-1]}{2L[n]+TR_0[n]} \left(\frac{R_0[n]}{R_0[n-1]}\right)^{\rho}$
(41)	Time-Varying	Trap.	$R_0[n] = \frac{2L[n]}{T}$	0	$-\left(\frac{L[n]}{L[n-1]}\right)^{\rho-1}$	0
(42)	Generalized	α	No	$\frac{(1+\alpha)L[n]-TR_0[n]}{(1+\alpha)L[n]+TR_0[n]}$	$-\frac{(1+\alpha)L[n-1]+T\alpha R_0[n-1]}{(1+\alpha)L[n]+TR_0[n]} \left(\frac{R_0[n]}{R_0[n-1]}\right)^{\rho} \left(\frac{L[n]}{L[n-1]}\right)^{1-\lambda}$	$-\frac{(1+\alpha)L[n-1]-T\alpha R_0[n-1]}{(1+\alpha)L[n]+TR_0[n]} \left(\frac{R_0[n]}{R_0[n-1]}\right)^{\rho} \left(\frac{L[n]}{L[n-1]}\right)^{1-\lambda}$
(43)	Generalized	α	$R_0[n] = \frac{(1+\alpha)L[n]}{T}$	0	$-\frac{1}{2}\left(\frac{L[n]}{L[n-1]}\right)^{\rho-\lambda}(1+\alpha)$	$-\frac{1}{2}\left(\frac{L[n]}{L[n-1]}\right)^{\rho-\lambda}(1-\alpha)$

(b) Inductor discretizations and coefficients.

Table 2: *Summary of inductor and capacitor discretizations and difference equation coefficients.*

EXPERIMENTAL STUDY OF GUITAR PICKUP NONLINEARITY

Antonin NOVAK, Bertrand LIHOREAU, Pierrick LOTTON, Emmanuel BRASSEUR, Laurent SIMON

Laboratoire d'Acoustique de l'Université du Mans (LAUM, UMR CNRS 6613),
72085 Le Mans, France
antonin.novak@univ-lemans.fr

ABSTRACT

In this paper, we focus on studying nonlinear behavior of the pickup of an electric guitar and on its modeling. The approach is purely experimental, based on physical assumptions and attempts to find a nonlinear model that, with few parameters, would be able to predict the nonlinear behavior of the pickup. In our experimental setup a piece of string is attached to a shaker and vibrates perpendicularly to the pickup in frequency range between 60 Hz and 400 Hz. The oscillations are controlled by a linearizion feedback to create a purely sinusoidal steady state movement of the string. In the first step, harmonic distortions of three different magnetic pickups (a single-coil, a humbucker, and a rail-pickup) are compared to check if they provide different distortions. In the second step, a static nonlinearity of Paiva's model is estimated from experimental signals. In the last step, the pickup nonlinearities are compared and an empirical model that fits well all three pickups is proposed.

1. INTRODUCTION

The beautiful sounds created by musical instruments, whether acoustic or electro-acoustic, relies very often on a nonlinear mechanism and the electric guitar is obviously no exception. The heart of an electric guitar is a pickup, a nonlinear sensor that captures the string vibrations and translates them into an electric signal [1, 2, 3, 4]. A magnetic pickup is basically composed of a set of permanent magnets surrounded by an electric coil (see Figure 1). A ferromagnetic string vibrating in the vicinity of the pickup results in a variation of the magnetic flux through the coil, and, according to the Faraday's law, an electrical voltage is then induced across the coil [3].

Since first pickups appeared almost a century ago, there have been thousands of pickup models, each of them providing different output. Almost all the electric-guitar players have probably asked the puzzling question of what distinguishes one particular pickup from another. Why is it that some sound warmer, some cleaner and some more distorted ? The answer to this question is important not only for guitar players but also for pickup manufactures and for digital audio effects engineers, especially those working with instrument synthesis [5, 6, 7]. A few models of pickup available in the literature may help to find the answer to this tricky question.

Some of these models are based on physical approaches using either integral equations [3, 8] or port-Hamiltonian systems [9, 10] while others are based on block-oriented models combining linear and nonlinear blocks together [11, 12]. In [11] Paiva shows that the sound of a pickup is influenced by three main properties: 1) the pickup position and width which are closely related to the string vibration, 2) the pickup high impedance which together with the input impedance of the device to which the guitar is plugged forms a linear filter, and 3) a nonlinear behavior of the

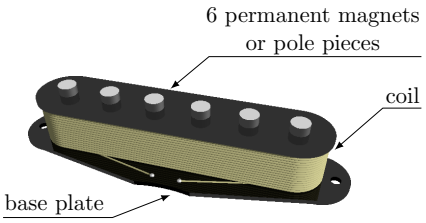


Figure 1: *Schematic representation of a "single-coil" pickup.*

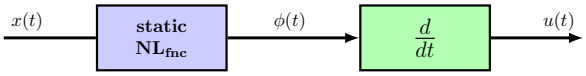


Figure 2: *Block diagram of the pickup nonlinear model, with $x(t)$ the string displacement, $\phi(t)$ the magnetic flux, and $u(t)$ the output voltage of the pickup.*

pickup. The main core of the Paiva's model [11] describing the magneto-electric conversion is based on a simple static nonlinearity representing the nonlinear relation between the string displacement and the magnetic flux, followed by a time derivative (see Figure 2). In [13, 14] we have experimentally shown, that this simple model, called Hammerstein system, is sufficiently precise for pickup nonlinear modeling and that more complicated models, such as a Generalized Hammerstein model, converge back to the simple Hammerstein system.

This paper focuses on experimental measurement of the static nonlinear block of Paiva's model and on comparison of nonlinear behavior of several pickups. Three different pickups of brand Seymour Duncan are chosen: 1) "SSL-5" - a single-coil pickup, 2) "SH-2N" - a humbucker (double-coil) pickup and 3) "STHR-1B Hot Rails" - a humbucker rail pickup using a rail in place of a row of six pole pieces. After presenting our experimental setup in section 2, a preliminary comparative measurement of harmonic distortion of each pickup is presented for two different pickup/string distances (section 2.1). Even if distortions of these three pickups are of the same nature, the difference in distortion between each pickup is visible not only in the spectra but also in the time-domain waveforms of the voltage output. To find the origin of this difference, we focus on the experimental identification of the static nonlinearity of the pickup. In section 3, the pickup nonlinear behavior is characterized experimentally leading to estimation of the input-output curve representing the static nonlinearity of the pickup. Finally, in section 4, all tested pickups are compared and an empirical law that fits well all the measurements at different pickup/string distances is proposed.

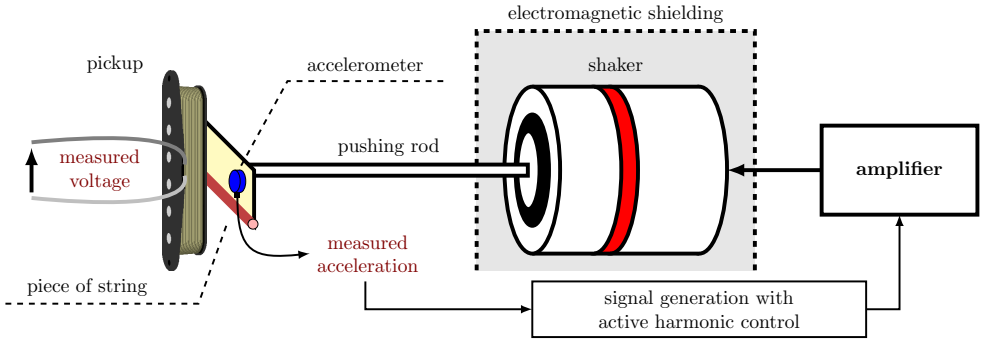


Figure 3: *Configuration of the measurement setup.*

2. EXPERIMENTAL SETUP & HARMONIC DISTORTION TEST

On the one hand, the identification of linear and nonlinear systems is generally based on a knowledge of input and output signals where the input signal is perfectly controlled (usually a random or deterministic signal such as sine, swept-sine, multi-tone, ...). On the other hand, the input signal of the pickup is the string displacement, guided by the laws of vibrations, which is difficult to control.

In order to overcome this problem, a specific measurement setup depicted in Figure 3 is used. A piece of string (8 cm long and 1 mm in diameter) is glued to a composite plate (3 x 8 cm) which is rigidly connected to an electrodynamic shaker (Brüel & KjæLDS V406). The shaker, driven by a Devialet D-premier amplifier, R.M.E Fireface 400 sound card, and a personal computer, is used as a source of the string displacement. The string is then placed next to the pickup's 6th pole piece (low E string position) at a distance d_0 so that the string can oscillate around d_0 with amplitude $\pm d_{max}$ (Figure 4). To avoid a possible disturbance by the electromagnetic field of the shaker, an electromagnetic shielding cage is placed around the shaker. An accelerometer PCB 352C22 is fixed to the composite plate (firmly fixed to the string). The sound card R.M.E Fireface 400 is then used to acquire both the signal $a(t)$ from the accelerometer (through a PCB sensor signal conditioner 482C series) and the output voltage $u(t)$ from the pickup (directly connected to the sound card instrument input with input impedance of 470 kΩ).

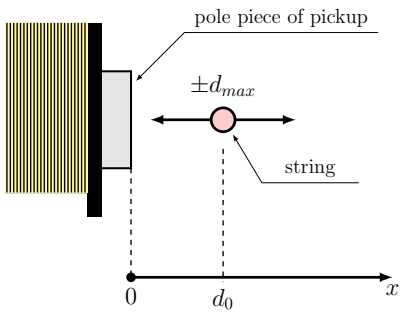


Figure 4: *Schematic representation of the pickup/string distance d_0 given by the distance between the string rest position and the pickup magnet (or its pole piece), and of the amplitude d_{max} of the string excursion, defining the total displacement $d_0 \pm d_{max}$.*

2.1. Measurement of pickup's harmonic distortion

Such a measurement setup can be used to control the input signal (displacement of the string - deduced from the measured acceleration) and to analyze the behavior of the pickup. However, in order to analyze the pickup from the nonlinear point of view, one would desire that the shaker, used to displace the string, behaves linearly. Otherwise, the displacement would be contaminated by the nonlinearities of the shaker which would make the identification of the nonlinear behavior of the pickup much more difficult. In [13, 14], a procedure based on swept-sine measurement [15], that allows to post-process the measured data and to identify the nonlinear system under test in terms of Generalized Hammerstein model, has been used. While efficient, this technique cannot fix the excitation signal in real time.

Recently, a simple and robust adaptive technique that can predistort the input signal in a real time to create a perfect periodical signal at the output of the shaker (with spectral purity up to 100 dB) has been proposed in [16]. Using this technique, we can generate a pure harmonic displacement even for large amplitudes, canceling completely the nonlinearity of the shaker. Therefore, if the measured acceleration, and consequently the string displacement, is ensured to be purely harmonic, the harmonic distortion observed at the output voltage of the pickup can be associated only to the nonlinearity of the pickup.

The results of this "harmonic excitation" experiment are depicted in Figure 5 for all three tested pickups and for two different pickup/string distances $d_0 = 3$ and 5 mm. The harmonic excitation with $d_{max} = 2$ mm and frequency 80 Hz, chosen in accordance with free vibrations of an E-string, is imposed to the string. One can see from Figure 5 that the nonlinear distortion of all three tested pickups has a similar character, but each output differs. The difference is visible not only in the frequency domain but also in the time domain. It is of no surprise that both humbucker pickups (SH-2N and the "STHR-1B Hot Rails") provide signals with higher level and higher distortion compared to a single-coil "SSL-5". The humbucker pickups also exhibit much stronger distortion when placed closer to the string ($d_0 = 3$ mm). When placed further from the string ($d_0 = 5$ cm), the level and distortion are surprisingly much more similar. In order to understand the origin of these differences in distortion generated by the pickups, we provide the following set of experiments, all of them conducted using the experimental setup described at the beginning of this section.

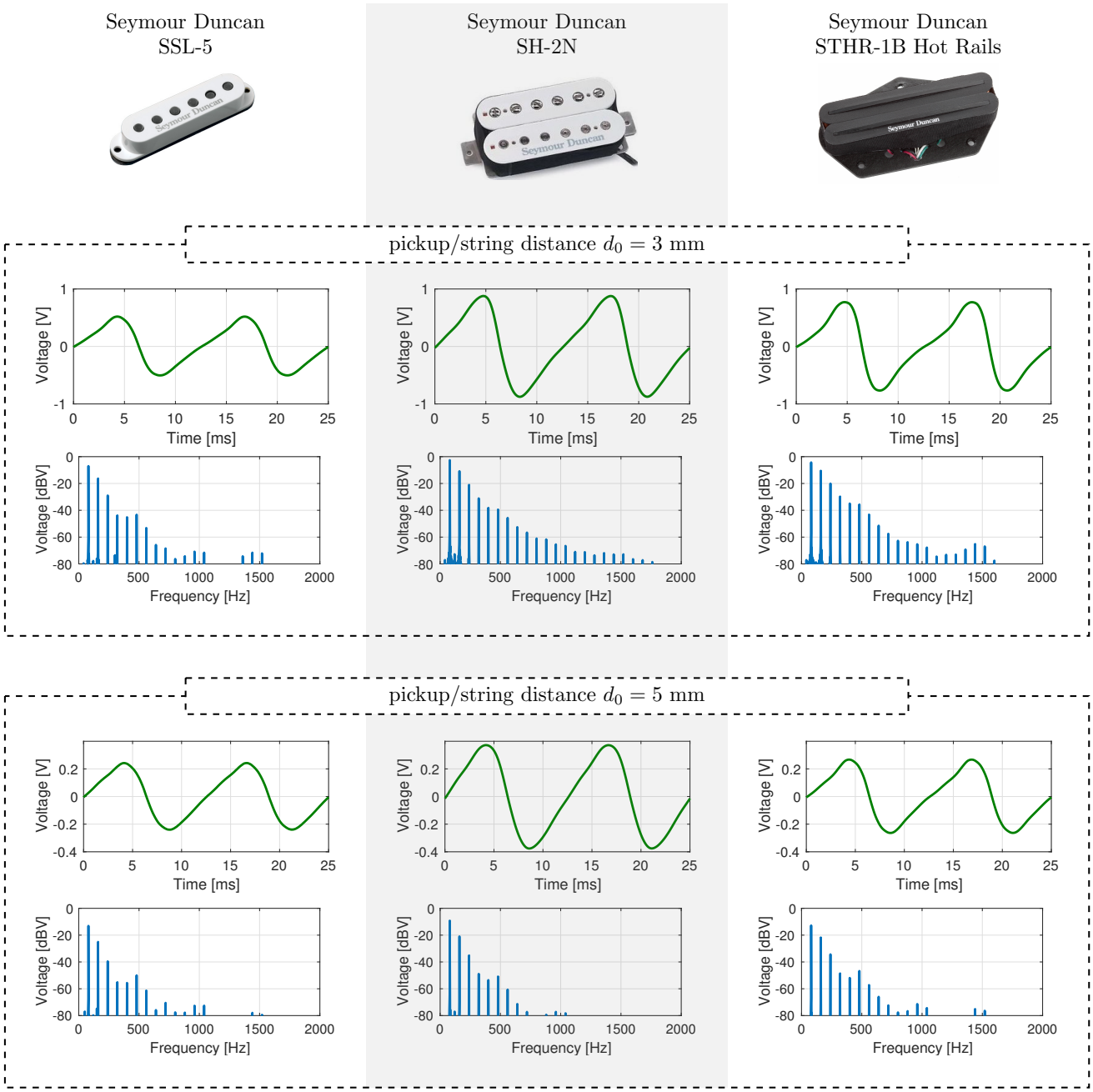


Figure 5: *Waveforms (green) and spectra (blue) of the output voltage of three pickups under test: a single-coil pickup (SSL-5) on the left, a humbucker double-coil pickup (SH-2N) in the middle (with gray background), and a rail pickup (STHR-1B Hot Rails) on the right. Results depicted for two different pickup/string distances $d_0 = 3$ mm and $d_0 = 5$ mm. The string is placed in front of the pickup's 6th pole piece (low E string position).*

3. EXPERIMENTAL ESTIMATION OF THE PICKUP NONLINEARITY

Since the Paiva's block-model [11] (static nonlinearity followed by the time derivative, see Figure 6(a)) has been experimentally verified in [13, 14], following experiments are focused on identification of the static nonlinearity directly from experimental signals.

To make a link between the physics and the block-model from Figure 6(a), we recall the Faraday's law of induction that defines the voltage $u(t)$ generated at the output of a coil with N turns as

$$u(t) = -N\frac{d\Phi_c(t)}{dt}, \tag{1}$$

$\Phi_c(t)$ being the magnetic flux passing through the coil. Comparing this law with the block-model, we can see, that the signal $\Phi(t)$ (time integral of the voltage $u(t)$) has the dimensions of the magnetic flux [V · s] and differs from the real flux $\Phi_c(t)$ of the coil by sign and by number of turns N. The signal $\Phi(t)$ can be easily deduced from the measured voltage $u(t)$ simply by integrating

$$\Phi(t) = \int_{-\infty}^{t} u(t')dt' + C. \tag{2}$$

Note, that an unknown constant of integration C, inherent in the construction of anti-derivatives, appears at the end of Equation (2). This constant is related to the direct component (DC) of the magnetic flux passing through the voice coil.

The block-model then assumes that there exists a direct static nonlinear relation between the time integral of the voltage $\Phi(t)$ and the string displacement $x(t)$

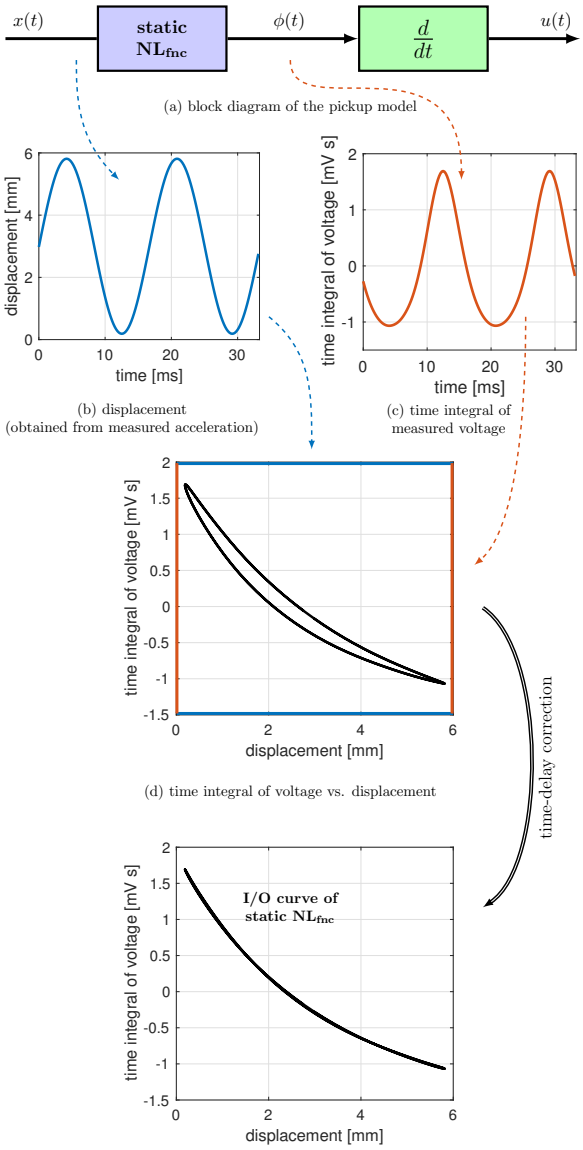
$$\Phi(t) = \mathrm{NL}_{\mathrm{fnc}}\Big\{x(t)\Big\}. \tag{3}$$

In the following, these two quantities $\Phi(t)$ and $x(t)$, derived from the measured voltage and acceleration[1], are used to estimate the input-output (I/O) relation of the static function $\mathrm{NL}_{\mathrm{fnc}}$. The term "static" means, that it is independent of frequency, and that an I/O relation depicted in a graph (translated to Matlab language as `plot(x,phi)`) should exhibit no area inside the closed curve.

In Figure 6, the procedure explained above is depicted using the experimental results ("SSL-5" pickup, $d_0 = 3$ mm, and $d_{max} = 2.8$ mm). Since the I/O relation should not depend on frequency, we use 60 Hz (resonant frequency of the shaker) to achieve larger amplitude d_{max} of displacement. The validation of this assumption through an experiment is provided in section 3.2. From the results depicted in Figure 6 we can conclude that the signal $\Phi(t)$, depicted in Figure 6(c), is very distorted and that, contrary to the assumption of a static nonlinearity, the I/O relation (Figure 6(d)) gives a curve with a non-negligible area inside the closed curve. The area can be simply explained by a small phase shift due to a time delay between the measured voltage and acceleration (e.g. due to the sensor signal conditioner). After applying a time delay (the actual time delay of 0.23 ms is the same for all measurements and independent of frequency), the I/O relation (Figure 6(e)) shows a nice smooth curve (with no closed curve area) that represents an estimate of the static nonlinear function $\mathrm{NL}_{\mathrm{fnc}}$.

As shown in the experiment with the harmonic distortion presented in Figure 5 the distortion differs for different pickup/string positions d_0. The following two experiments are thus focused on: 1) the influence of the pickup/string position d_0 on the static nonlinear function $\mathrm{NL}_{\mathrm{fnc}}$, and 2) the frequency dependence.

[1] integrating and differentiating in the frequency domain



(a) block diagram of the pickup model

(b) displacement (obtained from measured acceleration)

(c) time integral of measured voltage

(d) time integral of voltage vs. displacement

(e) time integral of voltage vs. displacement after time-delay correction

Figure 6: *(a) A block diagram of the pickup model with the measured signals (b) string displacement (obtained from the measured acceleration), (c) time integral of the measured voltage, and (d-e) a plot of time integral of voltage vs. string displacement in an I/O graph to estimated the form of the static nonlinear function; (d) without any correction, (e) a time-delay compensation is applied. Measurements performed on a SSL-5 pickup with a string placed at $d_0 = 3$ mm from the pickup's 6th pole piece (low E string position) and oscillating harmonically with amplitude $d_{max} = 2.8$ mm at 60 Hz.*

3.1. Influence of the pickup/string distance d_0

The previous experimental result shows that the I/O relation of the static function NL_{fnc} representing the static nonlinear block can be obtained directly from the string displacement (derived from measured acceleration) and from the time integral of the measured voltage of the pickup's output. The physical model of the guitar pickup proposed in [8] suggests that the nonlinear behavior of the pickup is influenced by the pickup/string distance d_0.

To study the influence of d_0 on the static nonlinear function NL_{fnc} of the model (see Figure 2), the following experiment is made for pickup/string distances d_0 = 3 mm, 5 mm, 7 mm, and 10 mm. The amplitude of the string displacement is $d_{max} = 3.5$ mm for all the measurements except for $d_0 = 3$ mm, for which d_{max} is set to 2.8 mm to avoid the string hitting the pickup.

The resulting I/O curves representing the NL_{fnc} are shown in Figures 7(a-d). The nonlinear I/O relation between the string displacement and the time integral of the voltage is much more nonlinear when the string is closer to the pickup (e.g. for $d_0 = 3$ mm, see Figure 7(a)) than when the string is much further ($d_0 = 10$ mm, see Figure 7(d)) even if, in this particular case with d_0 = 3 mm, the amplitude d_{max} of the string displacement is larger for the measurement made at other pickup/string distances.

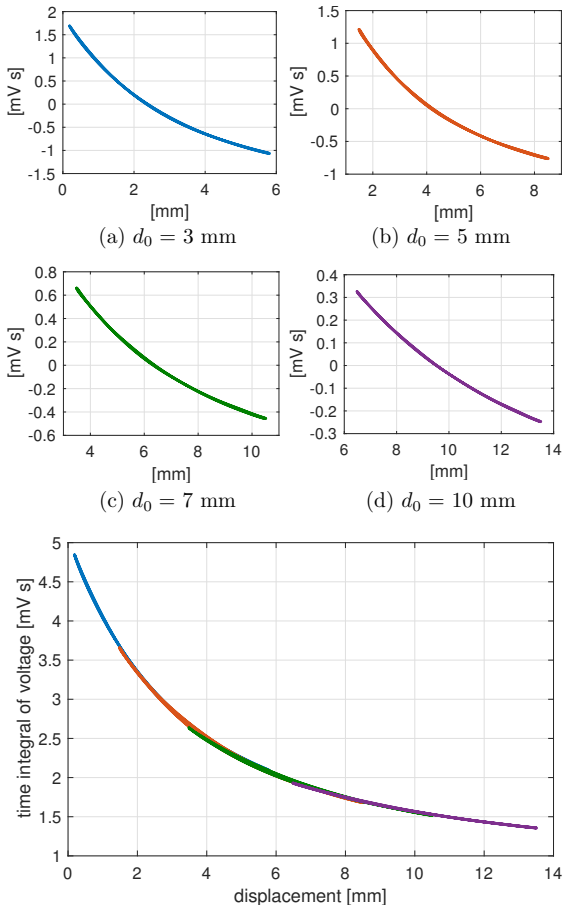
We can see from Figures 7(a-d) that the nonlinear behavior of the pickup (I/O curve) varies a lot with the pickup/string distance d_0. One could consequently conclude that when a guitar player changes the distance d_0 of the string, a different static nonlinear function NL_{fnc} applies. Indeed, as shown in Equation (2), the signal $\Phi(t)$, obtained as a time integral of the measured voltage $u(t)$, is missing the unknown constant of integration C. Consequently, the I/O curves can be offset (shifted vertically) with the same relative result. The offset has no consequence on the output voltage $u(t)$ due to the time derivative block. Figure 7(d) shows each I/O curve from Figures 7(a-d) plotted with an offset to achieve the best superposition. The superposition of the I/O curves, measured at different pickup/string distances d_0, is almost perfect, indicating that there is only one static nonlinear function NL_{fnc} no matter the pickup/string distance d_0. Therefore, when a guitar player changes the distance d_0 of the string, the same static nonlinear function NL_{fnc} applies.

3.2. Independent of frequency ?

To verify that the nonlinear function NL_{fnc} is really static, i.e. independent of frequency, a similar measurement is conducted on the SSL-5 pickup for different frequencies (60 Hz, 80 Hz, and 400 Hz) for a given pickup/string distance $d_0 = 4$ mm. The amplitude d_{max} differs for each measurement due to the physical limits of the shaker at different frequencies. While at low frequencies (e.g. 60 Hz) the shaker can provide a d_{max} close to 3 mm, for the same driving voltage at 400 Hz it provides a d_{max} lower than 1 mm.

Each I/O relation estimated from the measured signals for different frequencies is provided in Figure 8. The I/O curve obtained for 60 Hz (Figure 8(a)) is more curved than the one obtained for 80 Hz (Figure 8(b)), indicating a higher nonlinear distortion at 60 Hz. The I/O curve for 400 Hz (Figure 8(c)) is almost a perfect straight (linear) line.

Similarly to the previous results, one could conclude that the pickup behavior vary a lot with frequency (high distortion for low frequencies and low distortion for high frequencies). Nevertheless, it must not be forgotten that the string displacement have not the same value of amplitude d_{max} (see the x-axis in Figures 8(a-c)).



(a) $d_0 = 3$ mm (b) $d_0 = 5$ mm

(c) $d_0 = 7$ mm (d) $d_0 = 10$ mm

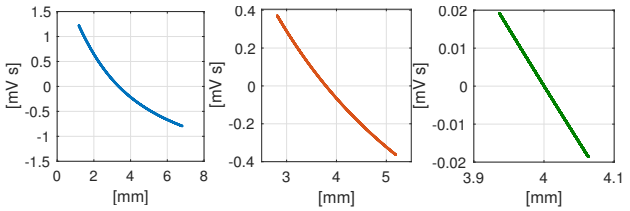
(e) superposition of measurements with $d_0 = 3$ mm, 5 mm, 7 mm, and 10 mm

Figure 7: *I/O graphs (time integral of measured voltage vs. string displacement) for four different pickup/string distances d_0 (a) 3 mm, (b) 5 mm, (c) 7 mm, and (d) 10mm. All the four I/O graphs are superposed in (e) where each time integral of measured voltage is offset by an unknown constant of integration. Measurements performed on a SSL-5 pickup with a string oscillating harmonically with amplitude $d_{max} = 2.8$ mm around $d_0 = 3$ mm and with amplitude $d_{max} = 3.5$ mm around $d_0 = 5$ mm, $d_0 = 7$ mm, and $d_0 = 10$ mm. The frequency is chosen to be 60 Hz in order to maximize the displacement of the shaker. The string is placed in front of the pickup's 6th pole piece (low E string position).*

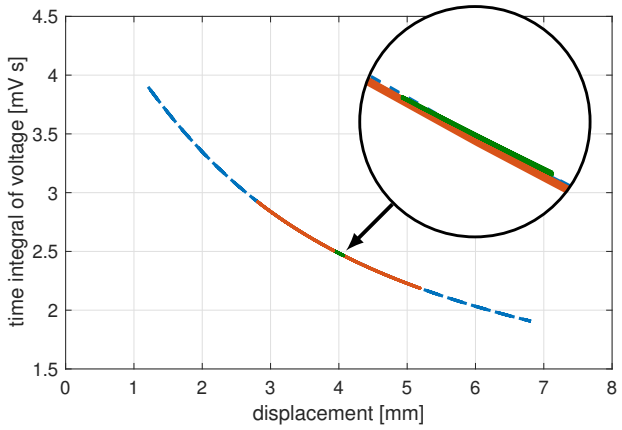
As in the previous experiment, each of the I/O curves can be offset (shifted vertically) to compensate for the unknown constant of integration (Equation (2)). The superposition of the offset I/O curves measured at different frequencies, depicted in Figure 8(d), results in an almost perfect superposition confirming the hypothesis of non frequency dependent static nonlinear function NL_{fnc}.

3.3. Discussion

These experiments conducted on the SSL-5 pickup show that a single static nonlinear function NL_{fnc} can be used, no matter the distance d_0 to which the guitar player sets the string. In other



(a) 60 Hz (b) 80 Hz (c) 400 Hz



(d) superposition of 60 Hz, 80 Hz, and 400 Hz curves

Figure 8: *I/O graphs (plot of the time integral of measured voltage vs. string displacement) for three different excitation frequencies (a) 60 Hz, (b) 80 Hz, (c) 400 Hz, and (d) all the three I/O graphs for three different frequencies superposed in one I/O graph. Each time integral of measured voltage is offset by an unknown constant of integration. Measurements performed on a SSL-5 pickup with a string placed at $d_0 = 4$ mm in front of the pickup's 6th pole piece (low E string position).*

words, d_0 and d_{max} join together into a single variable x representing the instantaneous distance of the string from the pickup (see the schematic representation in Figure 4). Consequently, the parameters d_0, d_{max}, and the frequency of string vibration, are parameters associated to the string displacement (input of NL_{fnc}), not to the pickup nonlinearity NL_{fnc} or its parameters.

Note also that the output voltage of the pickup is proportionally related to the gradient of the I/O curve, or, i.e. to the gradient of the magnetic flux (time integral of the voltage). Indeed combining Equations (1 - 3), one can write

$$u(t) = \frac{d\Phi(t)}{dt} = \frac{d\text{NL}_{\text{fnc}}\{x(t)\}}{dt} = \frac{\partial \text{NL}_{\text{fnc}}}{\partial x}\frac{dx(t)}{dt}. \quad (4)$$

The output voltage is thus proportional both to the velocity of the string and to the instantaneous gradient of the static nonlinear function NL_{fnc}. It is thus straightforward to guess from Figure 7 how the string position d_0 influences the level of output signal and the nonlinear distortion. For string position $d_0 = 3$ mm, close to the pickup, the I/O curve is very steep indicating high induced voltage at the output of the pickup. It is also curved due to the gradient variation, indicating a high nonlinear distortion. In the opposite way, for larger distance from the pickup (e.g. $d_0 = 10$ mm) the slope is weak and the curve flatter, resulting in a smaller voltage output with less distortion.

4. A SINGLE EMPIRICAL MODEL FOR ALL PICKUPS

In this section we provide the comparative results of three different pickups of brand Seymour Duncan: "SSL-5" - a single-coil pickup, "SH-2N" - a humbucker (double-coil) pickup, and "STHR-1B Hot Rails" a humbucker rail pickup. The goals of this section are threefold: (1) to verify that the findings proposed in the previous experiment on SSL-5 pickup also apply to the other pickups, (2) to see if there is any difference between the nonlinear I/O curves of each pickup and, if yes, what makes this difference, and (3) to provide an empirical model (other than the polynomial one) that, with minimal amount of parameters, would be able to predict the pickup nonlinear behavior.

4.1. Experiments on different pickups

All the three tested pickups are measured in the same way as the SSL-5 pickup in the previous section. The comparative table provided in Figure 9 shows the estimated NL_{fnc} of these three pickups. Observing the I/O graphs created by superposing the four measurements for different d_0 one can note that the conclusions proposed in the previous section for the single-coil (SSL-5) also apply to the humbucker (double-coil) pickups SH-2N and STHR-1B Hot Rails. Roughly speaking we can also predict the behavior of the pickups by observing the shapes of each nonlinear function NL_{fnc}. Following Equation (4), in which the pickup output voltage is proportional to the instantaneous value of the gradient of the NL_{fnc}, we can deduce that the SH-2N and STHR-1B will produce higher output level than the SSL-5 when the string is placed close (e.g. $d_0 = 3$ mm) to the pickup since the slope (gradient) of the NL_{fnc} is much higher. On the other hand when the string is placed at the distance of $d_0 = 5$ mm, the slopes of the NL_{fnc} are smaller and similar for all the three pickups, thus the amplitude of the output voltage should be smaller and similar for all pickups. This is perfectly correlated with the results presented in Figure 5 in which the signals and spectra of the pickups' output voltage are depicted. Similarly, the level of distortion of these signals is well correlated with the variations of the slope of the static nonlinear functions NL_{fnc} from Figure 9.

4.2. Single empirical model

To replicate the laws of physics that describe the nonlinear behavior of the pickup, represented by the static nonlinear function NL_{fnc}, it is desirable to find a fitting function that would fit the I/O law of the NL_{fnc} using few parameters. It could be used not only for sound synthesis of an electric guitar but also to quantitatively differentiate pickups through these parameters.

A polynomial fit (based on Taylor series) is usually the most common way to model a static nonlinear function when the analytical formula is not known or simply to reduce the computational cost of a platform on which the model of the pickup is implemented [1, 12]. The main disadvantages of a polynomial fit are, first, a missing physical interpretation and, second, the need of a high number of parameters in order to fit the curve correctly. Note, that the spectra of the SH-2N voltage output measured for a sinusoidally oscillating string around $d_0 = 3$ mm with amplitude $d_{max} = 2$ mm (see Figure 5) contains more than 20 harmonics. The polynomial fit would thus need at least 20 coefficients to reproduce a similar result which would be very impractical. Reducing the number of coefficients would lead to lower precision of the model. Moreover, all the measurement I/O curves presented in

this paper for different kinds of pickups rather exhibit a law similar to an exponential decay one which is difficult to fit a polynomial with few coefficients. Our attempt to fit the I/O curves with one exponential law revealed to be unsuccessful (high deviation on extremities of the I/O curves). A sum of two exponentials seemed to fit better the I/O behavior, but two exponentials require too many parameters for the fitting and do not seem to be really justified from the point of view of the physical laws of electromagnetism.

On the other hand, the basic magnetic field $B(x)$ of a cylindrical magnet (or a solenoid) along its x-axis is analytically described as [8]

$$B(x) = \frac{B_r}{2}\left(\frac{x+L}{\sqrt{r^2+(x+L)^2}} - \frac{x}{\sqrt{r^2+x^2}}\right), \quad (5)$$

where B_r is the remanent flux density of the magnet, r its radius, and L its length. Despite the fact that this relation is not describing the magnetic flux of the coil as a response to oscillating string in its proximity, we tried to find the best fit to the I/O curves using Equation (5) ... with no success (similar results that the exponential model). Inspired by this simple model, we tried to modify equation (5) empirically to find a better fit. Indeed, replacing both square roots by cube roots surprisingly led to very successful fit for all three studied pickups. Then, the following equation

$$\mathrm{NL_{fnc}}(x) = A\left(\frac{x+L_{eq}}{\sqrt[3]{r_{eq}^2+(x+L_{eq})^2}} - \frac{x}{\sqrt[3]{r_{eq}^2+x^2}}\right), \quad (6)$$

provides an empirical model of the static nonlinear function $\mathrm{NL_{fnc}}$ with only three parameters. Moreover, since the model is based on the physical basis, even if modified empirically, we can associate the model parameters to an equivalent cylindrical magnet with an equivalent radius r_{eq}, and an equivalent length L_{eq}. The constant A then includes the remanent flux density B_r as well as the string characteristics such as diameter and material properties.

The measured I/O curves of the static nonlinear function $\mathrm{NL_{fnc}}$ depicted in Figure 9 are fitted using Equation (6). The best fit is plotted in a black & white dashed curve in the same figure and the estimated parameters A, L_{eq}, and r_{eq} are provided for each pickup under each I/O curve. Note that the parameters L_{eq} and r_{eq} correspond to credible values of the length and radius of an equivalent magnet.

The output $\Phi(t)$ of the static nonlinear block (Figure 2) can be easily derived from Equation (6) as

$$\Phi(t) = A\left(\frac{x(t)+L_{eq}}{\sqrt[3]{r_{eq}^2+[x(t)+L_{eq}]^2}} - \frac{x(t)}{\sqrt[3]{r_{eq}^2+x^2(t)}}\right). \quad (7)$$

This equation can be used to directly calculate the output $\Phi(t)$ of the static nonlinear function to any string displacement $x(t)$, sinusoidal ($x(t) = d_0 + d_{max}\sin(2\pi f_0 t)$) or musical (offset by d_0). One can then simply calculate the time-derivative of $\Phi(t)$ to directly deduce the output voltage $u(t)$ of the pickup. Another possibility is to provide directly the voltage output $u(t)$ as a function of input string vibration $x(t)$ (still offset by d_0) using the gradient of the static nonlinear function $\mathrm{NL_{fnc}}$ (see Equation (4)) as

$$u(t) = \frac{\partial \mathrm{NL_{fnc}}}{\partial x}\frac{dx}{dt}, \quad (8)$$

with

$$\frac{\partial \mathrm{NL_{fnc}}}{\partial x} = A\left(\frac{(x(t)+L_{eq})^2+3r_{eq}^2}{3\left([x(t)+L_{eq}]^2+r_{eq}^2\right)^{4/3}} - \frac{x^2(t)+3r_{eq}^2}{3\left(x^2(t)+r_{eq}^2\right)^{4/3}}\right). \quad (9)$$

5. CONCLUSIONS

In this paper, the pickup nonlinear behavior is studied from an experimental point of view considering three different pickups: a single coil pickup, a humbucker, and a rail pickup. The experimental setup using a piece of string attached to a shaker, whose displacement is actively controlled to provide a spectrally pure (without distortion) sinusoidal excitation, shows that the output of each studied pickup differs and that the distance between the string and the pickup plays an important role in voltage distortion. It is next shown, that the model proposed by Paiva, consisting of a static nonlinear function followed by a time derivative, corresponds to the experiments and that the static nonlinear function is independent of frequency and follows the same rule no matter the pickup/string distance. Moreover, an empirical model describing the pickup nonlinear behavior is proposed.

Future works on this topic will focus on the measurements of string displacement along x and y axes, on comparison between the same types of pickups of different brands, on the dependence on the string properties (width, material, ...), as well as on analytical modeling that could justify (or find better) the empirical model proposed in this paper.

6. ACKNOWLEDGMENTS

The measurements, discussions, and redaction of this paper have been conducted mainly in a free time of all the authors, motivated by their passion for guitars and nonlinear systems. We would very much like to thank our wives and families for their understanding.

7. REFERENCES

[1] Thomas Jungmann, "Theoretical and practical studies on the behavior of electric guitar pick-ups," M.S. thesis, Helsinki Univ. of Tech., Espoo, Finland, 1994.

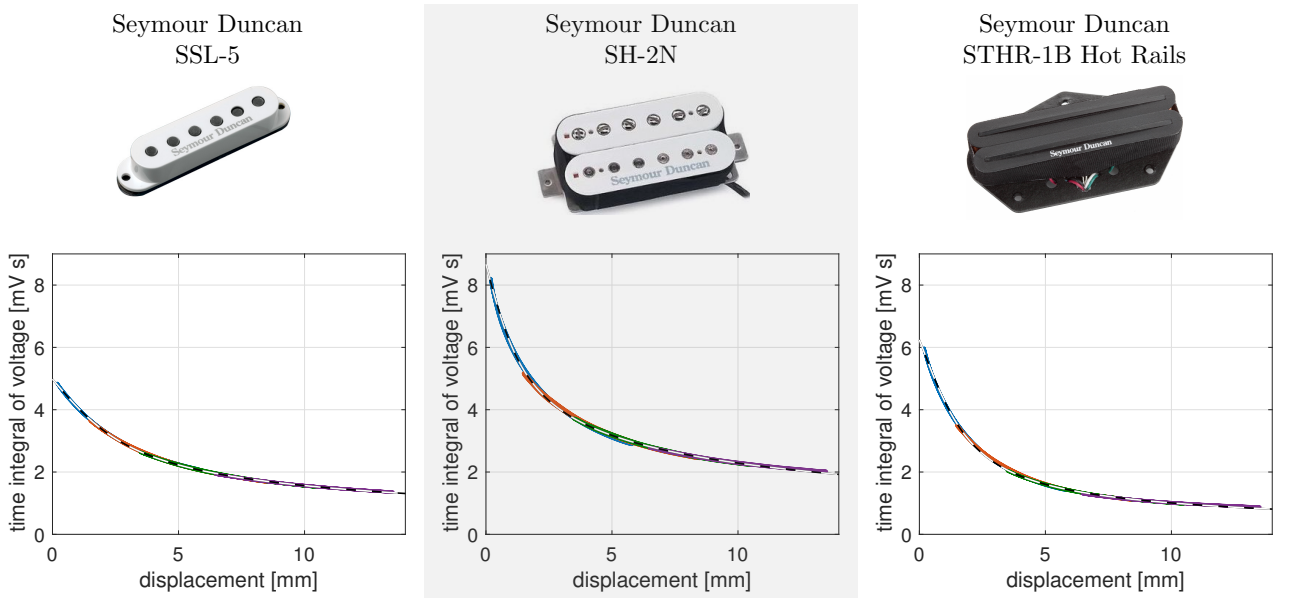
[2] Dave Hunter, *The Guitar Pickup Handbook: The Start of Your Sound*, Hal Léonard Corporation, 2008.

[3] Nicholas G Horton and Thomas R Moore, "Modeling the magnetic pickup of an electric guitar," *American Journal of Physics*, vol. 77, no. 2, pp. 144–150, 2009.

[4] Mirko Mustonen, Dmitri Kartofelev, Anatoli Stulov, and Vesa Välimäki, "Experimental verification of pickup nonlinearity," in *Proceedings International Symposium on Musical Acoustics (ISMA 2014), Le Mans, France*, 2014, vol. 1.

[5] Vesa Välimäki, Jyri Huopaniemi, Matti Karjalainen, and Zoltán Jánosy, "Physical modeling of plucked string instruments with application to real-time sound synthesis," *J. Audio Eng. Soc*, vol. 44, no. 5, pp. 331–353, 1996.

[6] Matti Karjalainen, Henri Penttinen, and Vesa Välimäki, "Acoustic sound from the electric guitar using dsp techniques," in *Acoustics, Speech, and Signal Processing, 2000. ICASSP'00. Proceedings. 2000 IEEE International Conference on*. IEEE, 2000, vol. 2, pp. II773–II776.



estimated parameters of the model from Equation (6)		
$A = 21.51 \cdot 10^{-3}$	$A = 40.67 \cdot 10^{-3}$	$A = 47.46 \cdot 10^{-3}$
$L_{eq} = 12.98$ mm	$L_{eq} = 9.86$ mm	$L_{eq} = 13.11$ mm
$r_{eq} = 2.77$ mm	$r_{eq} = 1.34$ mm	$r_{eq} = 1.88$ mm

Figure 9: *Plot of the static nonlinear functions (i.e time integral of voltage vs. string displacement depicted as an I/O graph) of three pickups under test: a single-coil pickup (SSL-5) on the left, a humbucker double-coil pickup (SH-2N) in the middle, and a rail pickup (STHR-1B Hot Rails) on the right. The data obtained from measurements are plotted in color (blue for $d_0 = 3$ mm, red for $d_0 = 5$ mm, green for $d_0 = 7$ mm, and violet for $d_0 = 10$ mm), and the fit using the empirical expression (6) is depicted in black & white dashed line. The string is placed in front of the pickup's 6th pole piece (low E string position).*

[7] Matti Karjalainen, Teemu Mäki-Patola, Aki Kanerva, and Antti Huovilainen, "Virtual air guitar," *Journal of the Audio Engineering Society*, vol. 54, no. 10, pp. 964–980, 2006.

[8] Léo Guadagnin, Bertrand Lihoreau, Pierrick Lotton, and Emmanuel Brasseur, "Analytical modeling and experimental characterization of a magnetic pickup for electric guitar," *Journal of the Audio Engineering Society*, vol. 65, no. 9, pp. 711–721, 2017.

[9] Antoine Falaize and Thomas Hélie, "Guaranteed-passive simulation of an electro-mechanical piano: A port-hamiltonian approach," in *Proc. of the 18 th Int. Conference on Digital Audio Effects (DAFx-15)*, 2015.

[10] Antoine Falaize and Thomas Hélie, "Passive simulation of the nonlinear port-hamiltonian modeling of a rhodes piano," *Journal of Sound and Vibration*, vol. 390, pp. 289–309, 2017.

[11] Rafael C.D. Paiva, Jyri Pakarinen, and Vesa Välimäki, "Acoustics and modeling of pickups," *Journal of the Audio Engineering Society*, vol. 60, no. 10, pp. 768–782, 2012.

[12] Luca Remaggi, Léonardo Gabrielli, Rafael C.D. Paiva, Vesa Välimäki, and Stefano Squartini, "A pickup model for the clavinet," in *Proc. of the 15 th Int. Conference on Digital Audio Effects (DAFx-12)*, 2012.

[13] Antonin Novak, Léo Guadagnin, Bertrand Lihoreau, Pierrick Lotton, E Brasseur, and Laurent Simon, "Non-linear identification of an electric guitar pickup," in *Proceedings of the 19th International Conference on Digital Audio Effects (DAFx-16), Brno, Czech Republic*, 2016, pp. 5–9.

[14] Antonin Novak, Léo Guadagnin, Bertrand Lihoreau, Pierrick Lotton, Emmanuel Brasseur, and Laurent Simon, "Measurements and modeling of the nonlinear behavior of a guitar pickup at low frequencies," *Applied Sciences*, vol. 7, no. 1, pp. 50, 2017.

[15] Antonin Novak, Balbine Maillou, Pierrick Lotton, and Laurent Simon, "Nonparametric identification of nonlinear systems in series," *IEEE Transactions on Instrumentation and Measurement*, vol. 63, no. 8, pp. 2044–2051, 2014.

[16] Antonin Novak, Laurent Simon, and Pierrick Lotton, "A simple predistortion technique for suppression of nonlinear effects in periodic signals generated by nonlinear transducers," *Journal of Sound and Vibration*, vol. 420, pp. 104–113, 2018.

WAVESHAPING WITH NORTON AMPLIFIERS: MODELING THE SERGE TRIPLE WAVESHAPER

Geoffrey Gormond *

Phelma
Grenoble Institute of Technology
Grenoble, France

Fabián Esqueda, Henri Pöntynen

Dept. of Signal Processing and Acoustics
Aalto University
Espoo, Finland

Julian D. Parker

Native Instruments GmbH
Berlin, Germany

ABSTRACT

The Serge Triple Waveshaper (TWS) is a synthesizer module designed in 1973 by Serge Tcherepnin, founder of Serge Modular Music Systems. It contains three identical waveshaping circuits that can be used to convert sawtooth waveforms into sine waves. However, its sonic capabilities extend well beyond this particular application. Each processing section in the Serge TWS is built around what is known as a Norton amplifier. These devices, unlike traditional operational amplifiers, operate on a current differencing principle and are featured in a handful of iconic musical circuits. This work provides an overview of Norton amplifiers within the context of virtual analog modeling and presents a digital model of the Serge TWS based on an analysis of the original circuit. Results obtained show the proposed model closely emulates the salient features of the original device and can be used to generate the complex waveforms that characterize "West Coast" synthesis.

1. INTRODUCTION

In the early 1970s, during the heyday of companies like Moog, ARP, and Buchla, access to modular synthesizers was mostly restricted to renowned musicians and members of the academic community. In those days a decently-equipped modular synthesizer, such as the Moog System 55[1], could easily cost tens of thousands of dollars. Frustrated by the high price tags of these instruments, Serge Tcherepnin, a then-professor of music composition at California Institute of the Arts (CalArts) decided to design a modular synthesizer that would be both affordable and powerful. With the support of a few CalArts students and faculty members, Serge set up a scheme in which people would pay $700 up front for parts and work on an improvised assembly line building their own six-module system [1, 2]. Serge's synthesizers became so successful that in 1975 he decided to leave his teaching position at CalArts to found *Serge Modular Music Systems*.

Since the beginning, Serge's approach to synthesizer design was heavily inspired by the work of Don Buchla on what is now known as "West Coast" synthesis. West Coast synthesis explored the use of non-traditional interfaces, such as step sequencers, and focused on timbre manipulation at waveform level via some form of nonlinear waveshaping [3]. In particular, Serge proposed expanding the signal path used in traditional subtractive synthesis by adding a "Wave Processor" stage between the oscillator and the voltage-controlled filter (VCF) [1]. Modules such as the Serge Wave Multipliers (VCM) [4], and the Triple Waveshaper (TWS) were designed for this purpose.

In this work we study the internal design of the Serge TWS module and propose a model for its digital implementation. The TWS is a processing module designed in 1973 as part of the first generation of Serge modules. As explained by Rich Gold[2] in his book *An Introduction to the Serge Modular Music System*, "the TWS module contains three identical devices which can be used to convert sawtooth waves into sine waves and can provide a wide range of other forms of sound and timbre modification. The timbre can be affected by a manual pot and two different VC inputs which operate on the sound in two different ways. It is a useful module for producing interesting and changing sound timbres, something difficult to achieve in other synthesizers" [5].

The motivation behind this study is to provide a better understanding of the Serge TWS, of which there is very little information available in the public domain, and to produce a "virtual analog" (VA) model that can be incorporated into a software-based synthesis environment. VA modeling is a popular area of study dedicated to emulating the behavior of vintage analog audio devices in the digital domain. This is highly desirable partly because nearly fifty years later, vintage analog synthesizers are still prohibitively expensive and hard to have access to. In a way, the motivation behind this kind of research is not much different from that of Serge Tcherepnin's when he started designing musical instruments.

Previous research on VA modeling of synthesizer circuits has concentrated on VCFs [6, 7, 8, 9, 10], oscillators [11, 12, 13], and effects processors [4, 14, 15]. Of related interest to this study is the pioneering work done during the 1970s on digital waveshaping synthesis [16, 17, 18]. This type of synthesis (much like West Coast synthesis) exploited the use of nonlinear waveshaping, e.g., via Chebyshev polynomials, to create harmonically-rich sounds from sinusoidal waveforms. These techniques are, in turn, closely related to popular digital synthesis methods such as frequency modulation and phase distortion synthesis [19, 20], which also relied on spectral manipulation via attribute modulation.

This paper is organized as follows. Section 2 provides an overview of Norton amplifiers, the component around which the module is based. Section 3 focuses on the analysis of the Serge TWS circuit. In Section 4 we observe the time- and frequency-domain behavior of the proposed model. Finally, Section 5 provides concluding remarks.

* Correspondence related to this work should be addressed to geoffrey.gormond@gmail.com

[1] https://www.moogmusic.com/products/modulars/system-55

[2] Rich Gold was part of the group of CalArts affiliates who worked on the design of the first Serge synthesizers. He was also responsible for the design of many of the emblematic Serge panels, which featured geometrical shapes instead of text labels [1].

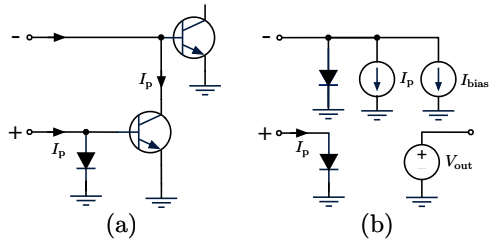


Figure 1: Circuit diagram for (a) the input stage and (b) a simplified equivalent circuit of a typical Norton amplifier. Figures adapted from [22].

2. NORTON AMPLIFIERS

Whereas the output voltage of the common operational amplifier is proportional to the voltage difference across its input terminals, the output of a Norton amplifier is proportional to the difference in the currents flowing into its input terminals [21]. Accordingly, Norton amplifiers are said to operate on a current differencing principle. This functionality is achieved by replacing the typical differential opamp input stage with a transistor configuration employing a current mirror at the positive input terminal to drain current from the negative input terminal, as shown in Fig. 1(a). As a first large-signal approximation, and as suggested in [22], a Norton amplifier can be modeled by the circuit shown in Fig. 1(b). Here, the transistor at the negative input terminal has been abstracted to a single base–emitter junction diode. Similarly, the current mirror has been reduced to a diode at the positive terminal and a current source that drains a replica of the positive input current from the negative terminal. A bias current source is added to the negative terminal and the output is represented as a voltage source that depends on the input currents. When negative feedback is applied, the output of the device settles at a voltage that minimizes the current difference between the input terminals [23]. This behavior is similar to that of conventional op-amps that seek to minimize the voltage difference across the inputs under negative feedback.

In typical applications, the diode at the negative input terminal remains forward biased if the small bias current $I_{\text{bias}} = 30\,\text{nA}$ is supplied to the negative input. Usually, this current is available when negative feedback is applied. The diode associated with the current mirror at the positive input terminal is commonly biased separately with a resistive connection to the power supply [24, 25]. With the diodes forward biased, the input terminals of the device are clamped to a diode drop above ground potential and the input pins can be treated as fixed voltage nodes. This fixed voltage assumption is the basis for many of the circuit design equations associated with Norton amplifiers [22, 23].

Compared to voltage-differencing operational amplifiers, Norton amplifiers are relatively uncommon devices in audio applications. Nevertheless, iconic vintage devices, such as VCFs in ARP synthesizers [26, 27], were designed around the LM3900; an integrated circuit housing four identical Norton amplifiers. This–now obsolete–device was favored by circuit designers for various applications due to its compactness, low cost and robust operation with a wide range of unipolar supply voltages. For instance, the LM3900 is capable of nearly full output voltage swings from approximately ground level (around 90 mV) to one diode drop below the supply voltage while maintaining stability [22]. The device even played a hidden role in shaping the sound aesthetics of video games, as the sound effects and the iconic background loop in the hit arcade game *Space Invaders* were implemented with dedicated synthesis circuits designed around the LM3900 [28]. Moreover, the LM3900 is particularly abundant in the designs of Serge Tcherepnin, who employed it widely to implement a variety of his synthesizer modules (e.g., the dual universal slope generator [3], the smooth & stepped generator [4], the bottom section of the VCM [5], a touch responsive keyboard [6], envelope generators [7], and many more). In the next section, we present a circuit analysis of the Serge TWS, an audio processor where the LM3900 was employed in an unconventional manner to perform complex waveshaping.

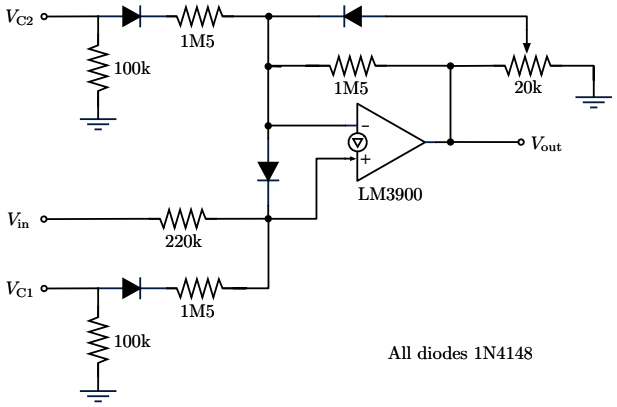


Figure 2: Circuit diagram for a single stage of the Serge TWS module. Figure adapted from [29].

3. THE SERGE TWS

Figure 2 shows a simplified schematic of a single stage of the Serge TWS [29]. The circuit takes a single input signal and applies a static waveshaping function to it. Control voltages V_{C1} and V_{C2} are then used to change the shape of this function. Figure 3(a) shows the model of the circuit used in this study. Here, we have substituted the LM3900 for the large-signal model described in Section 2. Additionally, the circuitry associated with the control voltage inputs V_{C1} and V_{C2} has been collapsed into ideal current sources I_A and I_B, respectively, both of which range from 0–3 μA. The range of these current sources was based on the standard used in Serge synthesizers, which expects DC-coupled and AC-coupled (i.e. audio) signals to range between approximately 0–5 V and ± 2.5 V, respectively. The blue diodes in Fig. 3(a) represent the BJT base–emitter junctions inside the LM3900, while the red diodes represent standard 1N4148 silicon signal diodes.

We divide the analysis of the circuits in two parts. First, we look at the input section and compute the value of currents I_p and I_f. Once these currents are known we proceed to analyze the feedback portion of the circuit and the output section. These steps are detailed in the following subsections.

[3] www.cgs.synth.net/modules/cgs114_dusg.html
[4] www.cgs.synth.net/modules/cgsssg_ssg.html
[5] www.cgs.synth.net/modules/cgs113_vcm.html
[6] www.cgs.synth.net/modules/cgs86_trk.html
[7] www.cgs.synth.net/modules/cgs76_env.html

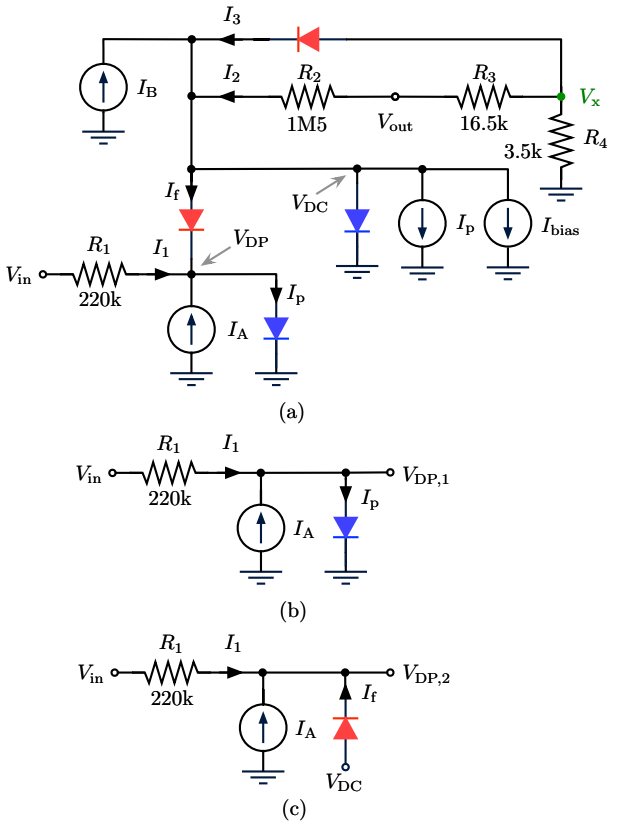


Figure 3: Circuit diagram for (a) the large signal model of a single stage of the Serge TWS and (b)–(c) the two subcircuits at the negative and positive terminals of the LM3900 amplifier, respectively. Blue and red diodes represent BJT base–emitter junction diodes and 1N4148 silicon diodes, respectively.

3.1. Input Section

We begin our analysis by computing the value of currents I_p and I_f. By applying Kirchhoff's current law (KCL) at the node labeled V_{DP} in Fig. 3(a) we can establish that

$$I_1 = I_p - I_f - I_A. \tag{1}$$

Next, we make the assumption that when either of the two diodes in the input section conducts, the contribution of the other one to the total value of I_1 will be close to zero. Therefore, we can analyze the two subcircuits shown in Figs. 3(b) and 3(c) independently. This approach will allow us to explicitly calculate I_p and I_f separately at each input sample without introducing discontinuities or significant errors in the model. As discussed in the previous section, we assume that the internal biasing in the Norton amplifier together with the application of negative feedback ensures that the diode inside the negative terminal is always forward-biased. This means that we can treat it as a fixed voltage node that we define to be clamped at $V_{DC} = 516\,\text{mV}$ based on SPICE simulations.

For the subcircuit in Fig. 3(b), we apply Kirchhoff's voltage law (KVL) and KCL to establish that

$$V_{in} = R_1 I_1 + V_{DP,1} \tag{2}$$
$$I_1 = I_P - I_A, \tag{3}$$

where I_p can be written using Shockley's diode equation as

$$I_p = I_{s,1}\left(\exp\left(\frac{V_{DP,1}}{\eta_1 V_T}\right) - 1\right), \tag{4}$$

where parameters $I_{s,1}$, η_1 and V_T represent the reverse bias saturation current, ideality factor and thermal voltage (at room temperature) of an ideal base–emitter p–n junction, respectively. All of the parameters required to implement the proposed model are given in Table 1. The tabulated semiconductor parameters were obtained from the LM3900 datasheet [22] and SPICE component models. By combining (2) and (3) with (4), we arrive at the implicit relationship

$$V_{DP,1} = V_{in} + R_1 I_A - R_1 I_{s,1}\left(\exp\left(\frac{V_{DP,1}}{\eta_1 V_T}\right) - 1\right), \tag{5}$$

which has the explicit solution

$$V_{DP,1} = V_{in} + R_1 I_A + R_1 I_{s,1} - \eta_1 V_T \mathrm{W}\left(\frac{R_1 I_{s,1}}{\eta_1 V_T}\exp\left(\frac{V_{in} + R_1 I_A + R_1 I_{s,1}}{\eta_1 V_T}\right)\right), \tag{6}$$

where $W(\,)$ is the Lambert-W function. The use of the Lambert-W function to solve the implicit current-voltage relationship of diodes was first proposed in [30], and extended in [31] and [32].

This same procedure can be followed for the subcircuit in Fig. 3(c), which gives us the explicit formulation

$$V_{DP,2} = V_{in} + R_1 I_A - R_1 I_{s,2} + \eta_2 V_T \mathrm{W}\left(\frac{R_1 I_{s,2}}{\eta_2 V_T}\exp\left(\frac{V_{DC} + R_1 I_{s,2} - V_{in} - R_1 I_A}{\eta_2 V_T}\right)\right), \tag{7}$$

where $I_{s,2}$ and η_2 are the reverse bias saturation current and ideality factor of the 1N4148 silicon diode, respectively. Once the value of $V_{DP,2}$ is known, the current I_f can be evaluated as

$$I_f = I_{s,2}\left(\exp\left(\frac{V_{DC} - V_{DP,2}}{\eta_2 V_T}\right) - 1\right). \tag{8}$$

Figure 4 shows the value of currents I_p and I_f computed using (4) and (8), respectively. Both currents are plotted against measurements obtained from a SPICE simulation (gray lines) of the large-signal model in Fig. 3(a). These results indicate our previous assumptions do not alter the overall general behavior of the model.

Table 1: *Component/parameter values.*

Name	Value	Name	Value
R_1	220 kΩ	$I_{s,2}$	2.52 nA
R_2	1.5 MΩ	η_1	1
R_3	16.5 kΩ	η_2	1.752
R_4	3.5 kΩ	V_T	25.864 mV
R_5	1 kΩ	V_{DC}	516 mV
$I_{s,1}$	10^{-14} A	I_{bias}	30 nA

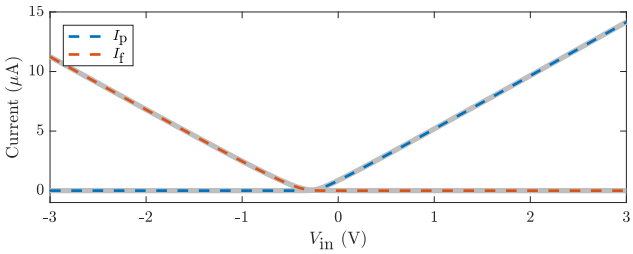


Figure 4: Value of I_p and I_f as a function of V_{in} computed using the proposed model and plotted against SPICE measurements (light gray lines).

3.2. Feedback Section

Having computed currents I_p and I_f we can analyze the feedback section of the circuit and derive a closed-form expression for V_{out}. As before, we apply KVL and KCL at the node labeled V_{DC} to derive the relationships

$$V_{out} = R_2 I_2 + V_{DC} \tag{9}$$

$$I_2 = I_f + I_p + I_{bias} - I_B - I_3, \tag{10}$$

where $I_{bias} = 30\,\text{nA}$ [22] and

$$I_3 = I_{s,2}\left(\exp\left(\frac{V_x - V_{DC}}{\eta_2 V_T}\right) - 1\right). \tag{11}$$

Here, V_x (highlighted in green in Fig. 3(a)) represents the voltage at the wiper node of the potentiometer in Fig. 2. Combining (11) with (9) and (10) gives us

$$V_{out} = V_{DC} + R_2 I_G - R_2 I_{s,2}\left(\exp\left(\frac{V_x - V_{DC}}{\eta_2 V_T}\right) - 1\right), \tag{12}$$

where the substitution $I_G = (I_f + I_p + I_{bias} - I_B)$ has been used for clarity. Similarly, applying KVL and KCL at V_x we arrive at the expression

$$V_{out} = R_3 I_{s,2}\left(\exp\left(\frac{V_x - V_{DC}}{\eta_2 V_T}\right) - 1\right) + G V_x, \tag{13}$$

where $G = (R_3/R_4 + 1)$. If we then equate (12) and (13), and solve for V_x, we arrive at the implicit expression for the wiper voltage

$$V_x = \frac{V_{DC} + R_2 I_G}{G} - \frac{R_2 + R_3}{G}\left(\exp\left(\frac{V_x - V_{DC}}{\eta_2 V_T}\right) - 1\right), \tag{14}$$

which can be solved using the Lambert-W function as

$$V_x = \frac{V_{DC} + R_2 I_G + (R_2 + R_3) I_{s,2}}{G} - \eta_2 V_T \mathrm{W}\left(\frac{(R_2 + R_3) I_{s,2}}{G \eta_2 V_T} \times \exp\left(\frac{-V_{DC}}{\eta_2 V_T}\right) \exp\left(\frac{V_{DC} + R_2 I_G + (R_2 + R_3) I_{s,2}}{G \eta_2 V_T}\right)\right). \tag{15}$$

This expression can be used to compute the value of V_x which can then be used to compute the value of V_{out} by evaluating either (12) or (13).

3.3. Output Clipping

As explained in Section 2, the LM3900 operates on a single power supply and is unable to generate voltages below approximately 90 mV. Therefore, this behavior must be accounted for in the proposed model. For the sake of simplicity, we propose an ad hoc approach that involves emulating the clipping behavior with a piecewise nonlinear function. We introduce a new voltage variable $\widetilde{V}_{out}$ which represents the value of the output voltage after clipping. The expression for the proposed clipper can be written as

$$\widetilde{V}_{out} = \begin{cases} V_{clip} & V_{out} \leq V_{clip} \\ V_{clip}\sqrt{1 + (\frac{V_{out}}{V_{clip}} - 1)^2} & \text{otherwise} \end{cases} \tag{16}$$

where $V_{clip} = 90\,\text{mV}$.

3.4. Model Summary

Having derived all the necessary expressions, in this section we provide a summary of the steps required to emulate the circuit in the digital domain. Since the circuit is static, we can compute the output directly by assuming a discrete-domain input signal $V_{in}[n]$, where n is the sample index. The steps required to compute $\widetilde{V}_{out}[n]$ are:

1. Evaluate voltages $V_{DP,1}[n]$ and $V_{DP,2}[n]$ using (6) and (7), respectively.
2. Compute currents $I_p[n]$ and $I_f[n]$ using (4) and (8).
3. Evaluate voltage $V_x[n]$ using (15).
4. Evaluate $V_{out}[n]$ using either (12) or (13).
5. Apply the clipping function (16).

Figures 5(a) and 5(b) show the input–output relationship of the circuit for different values of I_A evaluated using the proposed model and with SPICE, respectively. This comparison indicates a good match between the proposed model and its corresponding SPICE simulation, with a maximum difference of approximately 22 mV, as shown in Fig. 5(c). As shown in these figures, the system exhibits a highly nonlinear behavior which resembles that of a soft clipper cascaded with a full-wave rectifier. Adjusting the value of I_A changes the x-axis symmetry of the circuit.

Similarly, the plots in Figures (6)(a) and (6)(b) show the effect of increasing control current I_B from 0 to 3 μA. This parameter appears to "open" or "widen" the shape of the nonlinearity. The clipping behavior of the LM3900 is evident in these plots. Once again, the proposed model shows a good match with its corresponding SPICE simulation, with a maximum difference of approximately 70 mV (cf. Fig. (6)(c)). This increased difference can be attributed to the ad hoc modeling of the clipping stage.

Finally, the curves in Figures 7(a) and 7(b) shows the measured input–output relationship of a real Serge TWS built according to the schematic given in Fig. 2. The behavior of the circuit was measured for different values of control voltages V_{C1} and V_{C2}. When compared with Figs. 5(b) and 6(b), these results further demonstrate the proposed model preserves the salient characteristics of the circuit.

3.5. AC Coupling

The plots in Figures 5 and 6 show that the output of the Serge TWS will exhibit a static DC offset. The original circuit solved this by providing an additional AC-coupled output [29]. This is

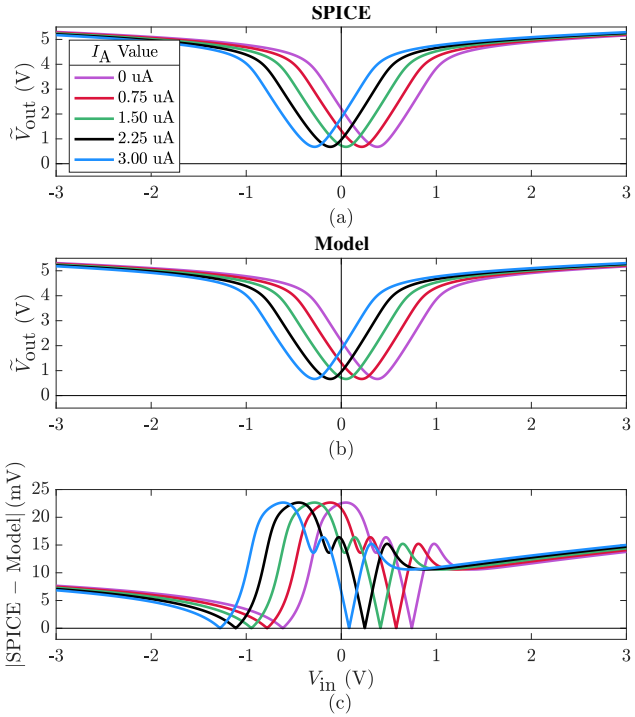


Figure 5: Input–output relationship of a single stage in the Serge TWS for values of I_A between 0–3 μA ($I_B = 0$ A) simulated using (a) SPICE and (b) the proposed model, and (c) the absolute value of the difference between both sets of curves.

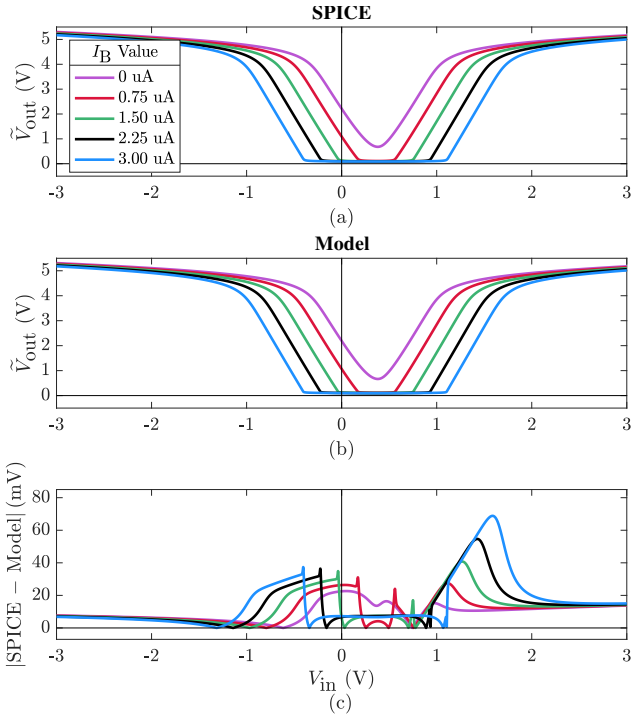


Figure 6: Input–output relationship of the system for values of I_B between 0–3 μA ($I_A = 0$ A) simulated using (a) the digital model and (b) SPICE, and (c) the absolute value of their difference.

quite typical in Serge modules as they were designed to process not only audio signals but also control voltages, which must be DC-coupled. In the digital domain, an AC-coupled version of the output can be computed, for instance, by using the first-order DC blocker proposed by Pekonen and Välimäki in [33]. The z-domain transfer function of this filter is defined as

$$H_{DC}(z) = \frac{1+p}{2}\,\frac{1-z^{-1}}{1-pz^{-1}}, \tag{17}$$

where $p = \tan(\pi/4 - \pi f_c/F_s)$, F_s is the sampling rate of the system and f_c is the cut-off frequency of the filter, set at 2 Hz in this case.

4. RESULTS

In this section we examine the time- and frequency-domain behavior of the circuit when driven by sawtooth waveforms, as recommended in the original user manual [5]. Figure 8 shows the output of a single stage of the Serge TWS when driven by an 80-Hz sawtooth waveform with peak amplitude of 1 V for different values of I_A when $I_B = 0$. The resulting waveforms have been stacked on top of each other to help visualize the evolution of the output signal as a function of I_A. To minimize the effects of aliasing, the original input waveform (shown in blue at the top of the plot) was synthesized using the first-order differentiated parabolic waveform (DPW) algorithm at a sampling rate $F_s = 352.8$ kHz (i.e. 8-times oversampling w.r.t. standard audio rate) [13]. This sample rate is used throughout the rest of this study. From this figure we can observe that the circuit does indeed transform the input waveform into something that resembles a sine wave. The best results are obtained when $I_A \approx 1.5\,\mu$A, as the circuit exhibits near-perfect even symmetry (cf. Fig. 5).

This case is presented in greater detail in Figs. 9(c)–(d) which

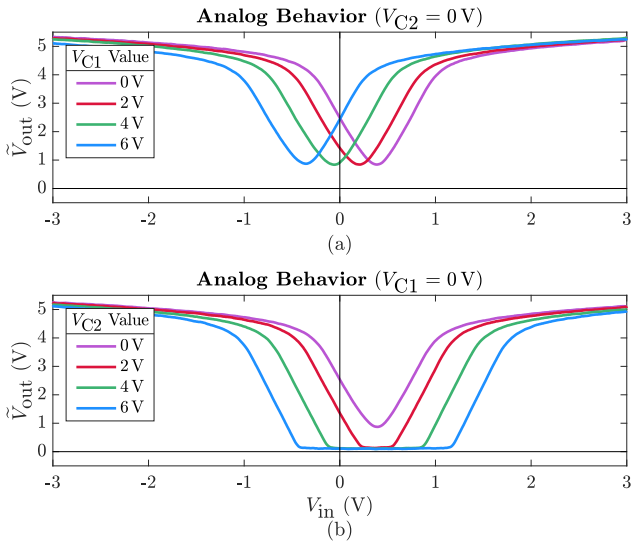


Figure 7: Measured analog input–output behavior of the circuit in Fig. 2 for different values of V_{C1} and V_{C2}.

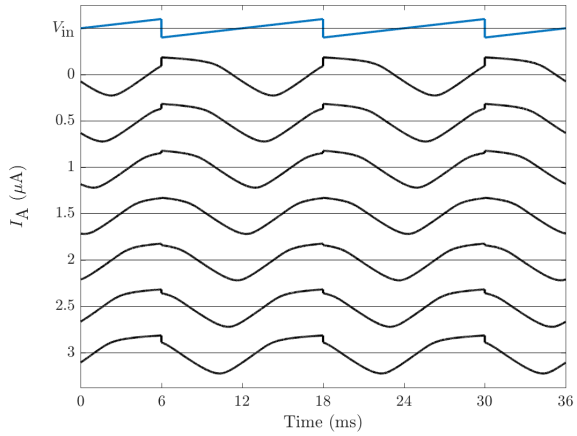


Figure 8: Output of the proposed model when driven by an 80-Hz sawtooth waveform for different values of I_{A} ($I_{\text{B}} = 0$).

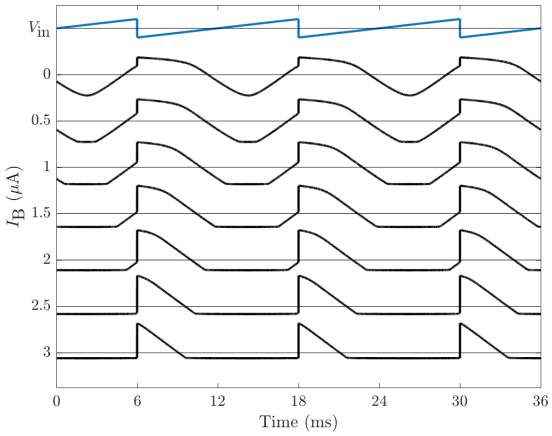


Figure 10: Output of the proposed model when driven by an 80-Hz sawtooth waveform for different values of I_{B} ($I_{\text{A}} = 0$).

show the waveform and magnitude spectrum of a 200-Hz sawtooth waveform with peak amplitude of 1 V processed by the model for $I_{\text{A}} = 1.5\,\mu$A. These results show that, as originally advertised, the circuit can indeed approximate a sinusoidal waveform when driven by a 1-V sawtooth signal. Although the resulting waveform shows a strong presence of the second harmonic, nearly all other partials have been significantly attenuated. However, this behavior is heavily dependent on the level of the input signal. Figures 9(e)–(f) show the result of driving the proposed model with a 2.5-V sawtooth waveform. In this case the resulting waveform no longer resembles a sine wave, as it exhibits considerably high harmonic content. As a reference, Figs. 9(a)–(b) present the waveform and magnitude spectrum of the 200-Hz sawtooth input.

Next, we consider the effect of control current I_{B} on the output. Figure 10 shows the output of the Serge TWS when driven by a 80-Hz sawtooth waveform with peak amplitude of 1 V for different values of I_{B} when $I_{\text{A}} = 0$. As shown in these plots, increasing the value of I_{B} increases the amount of clipping introduced by the circuit. These results go in accordance with the the input–output relationship of the model (cf. Fig 6). Lastly, Fig. 11 shows the recorded analog response of the circuit when driven by an 80-Hz sawtooth signal under different settings. These results further validate the accuracy of the proposed model, as they match the waveforms depicted in Figs. 8–9. The measured waveforms were normalized during the recording process.

We observe the frequency domain behavior of the system by considering the spectrograms in Figs. 12 and 13. The first spectrogram shows the effect of varying control current I_{A} linearly for a static 500-Hz sawtooth input (peak amplitude of 1 V). We can once again observe the region of values of I_{A} for which the waveshaper approximates a sinusoidal output. Overall, this behavior contrasts

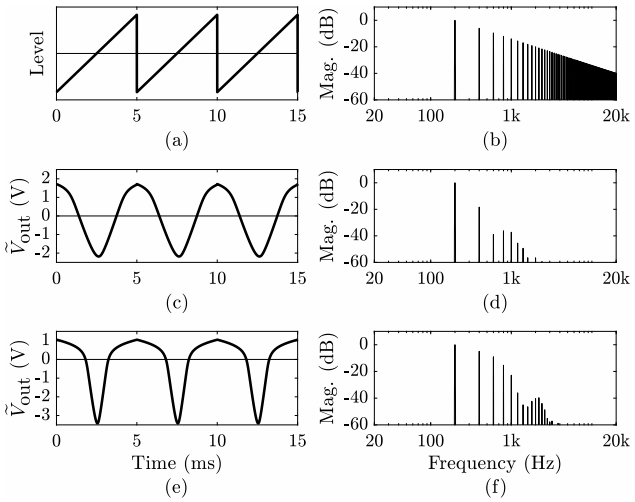


Figure 9: Waveform and magnitude spectrum of (a)–(b) a 200-Hz sawtooth, (c)–(d) a 1-V and (e)–(f) a 2.5-V sawtooth processed by the proposed model. Parameters $I_{\text{A}} = 1.5\,\mu$A and $I_{\text{B}} = 0$ A.

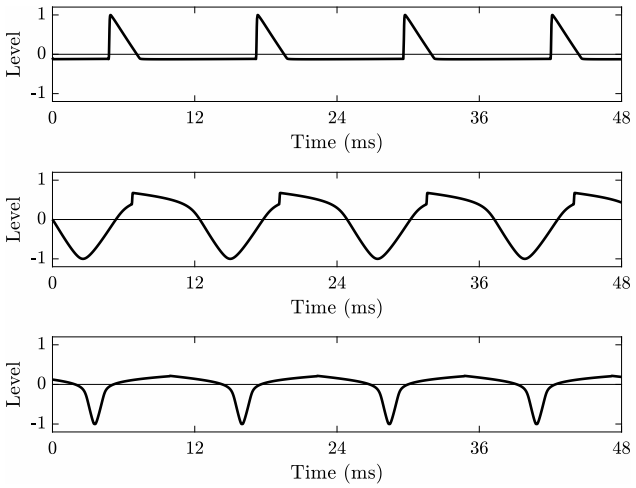


Figure 11: Measured analog time-domain behavior of a single stage in the Serge TWS when driven by an 80-Hz analog sawtooth waveform for different values of V_{C1} and V_{C2} (cf. Fig. 2).

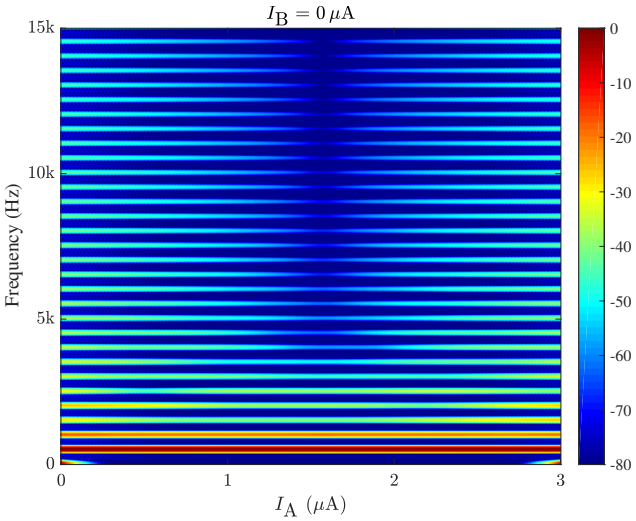


Figure 12: Magnitude response of a single stage in the Serge TWS when driven by a 1-V 500-Hz sawtooth waveform for values of I_A between 0–3 μA and $I_B = 0$ A.

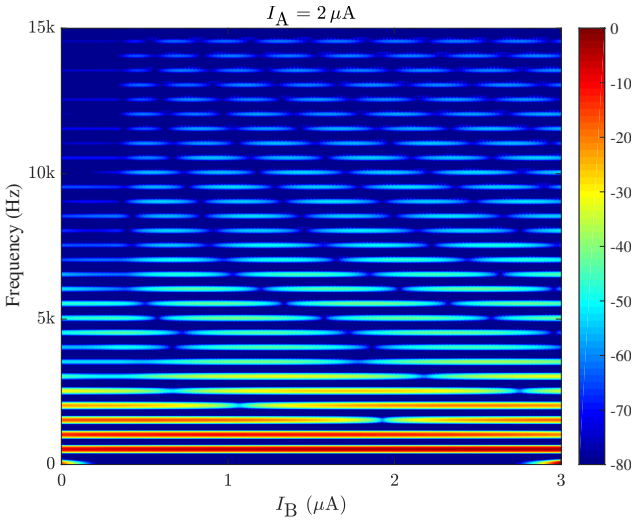


Figure 13: Magnitude response of a single stage in the Serge TWS when driven by a 1-V 500-Hz sawtooth waveform for $I_A = 2\,\mu$A and values of I_B between 0–3 μA.

that of other Serge circuits, such as the middle section of the VCM which is designed to expand the frequency content of sinusoidal waveforms [4]. The second spectrogram shows the effect of modulating I_B for a static value of $I_A = 2\,\mu$A. This value was chosen as it displayed interesting and complex harmonic patterns.

Finally, we briefly consider what happens when the three identical waveshapers in the Serge TWS are connected in series. This form of usage of the circuit is so popular that some re-issues of the module (e.g. the Random*Source Serge Triple+ Waveshaper[8]) even feature integrated switches to link the stages internally. Figure 14 shows the output waveforms that result from processing an 80-Hz sawtooth waveform (peak amplitude 2.5 V) using three stages in cascade. Control parameters I_A and I_B where kept constant between stages. The DC blocker (17) was used in between each stage. As shown in these plots, the cascaded configuration no longer operates as originally intended. Nevertheless, it can be used to produce the complex waveforms that characterize West Coast synthesis.

Overall, the sonic possibilities offered by the Serge TWS are quite vast. By manipulating all free input parameters, i.e. input level and control currents, different timbral effects can be achieved. When all three stages are cascaded, the number of combinations increases even further, as the parameters of each stage can be modulated independently. Sound articulation and timbral variety are then achieved by modulating the control currents in real-time. It should also be noted that the use of the circuit is not restricted to sawtooth signals. It can be used to process virtually any input waveform regardless of its harmonic nature. This makes the Serge TWS an extremely powerful and versatile synthesis tool.

5. CONCLUSIONS

In this work we examined the underlying structure of the Serge TWS module. We introduced Norton amplifiers and discussed the use of a simplified large-signal model for their emulation in the digital domain. A digital model of a single waveshaping stage in the module was proposed. The model was validated against a SPICE simulation of the same circuit. Results from driving the proposed model with multiple sawtooth waveforms show the Serge TWS can be used to transform sawtooth signals into sinusoidal waveforms, but can also be used to generate highly complex signals with interesting harmonic patterns. This study provides an insight into Serge Tcherepnin's approach to synthesis and opens the door for further study of his iconic circuits. Supplementary materials for this paper can be found in the accompanying website `http://research.spa.aalto.fi/publications/papers/dafx18-serge-tws`.

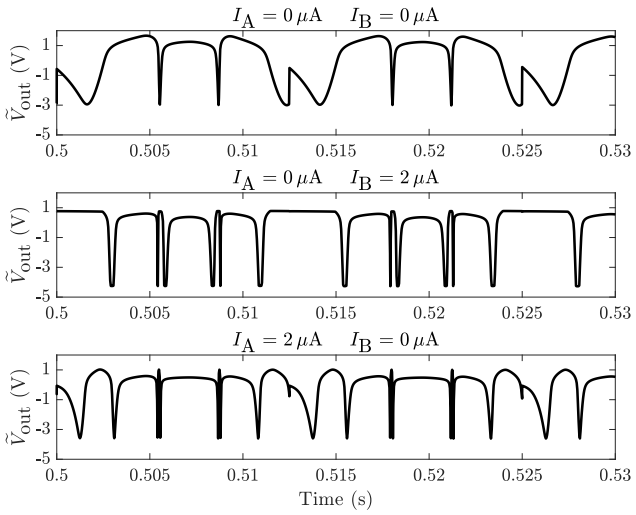


Figure 14: Results of processing an 80-Hz sawtooth waveform using three waveshapers arranged in series. The values of I_A and I_B used for these simulations are indicated on top of each subplot.

[8] `http://randomsource.net/serge_euro`

6. ACKNOWLEDGMENTS

The authors would like to thank Ken Stone and Serge Tcherepnin for the valuable correspondence during the early stages of this project. The main part of this work was conducted during Geoffrey Gormond's visit to Aalto University in summer 2017.

7. REFERENCES

[1] M. Vail, "Serge Modular Systems: Maximum Analog Horsepower," in *Vintage Synthesizers*, M. Vail, Ed., pp. 147–152. Miller Freeman Books, San Francisco, CA, USA, 2000.

[2] L. Mizzell, "Serge Modular Music Systems – Historical Bits and Pieces," Available online: `http://www.serge-fans.com/history.htm` (accessed 11 April 2018).

[3] J. Parker and S. D'Angelo, "A digital model of the Buchla lowpass-gate," in *Proc. Int. Conf. Digital Audio Effects (DAFx-13)*, Maynooth, Ireland, Sept. 2013.

[4] F. Esqueda, H. Pöntynen, J. D. Parker, and S. Bilbao, "Virtual analog models of the Lockhart and Serge wavefolders," *Appl. Sci.*, vol. 7, no. 12, Dec. 2017.

[5] R. Gold, D. Johansen, and M. LaPalma, "An Introduction to the Serge Modular Music System," Available online: `http://serge.synth.net/documents/` (accessed 11 April 2018).

[6] D. Rossum, "Making digital filters sound analog," in *Proc. Int. Comput. Music Conf.*, San Jose, CA, USA, Oct. 1992, pp. 30–33.

[7] A. Huovilainen, "Non-linear digital implementation of the Moog ladder filter," in *Proc. Int. Conf. Digital Audio Effects (DAFx-04)*, Naples, Italy, Oct. 2004, pp. 61–64.

[8] F. Fontana and M. Civolani, "Modeling of the EMS VCS3 voltage-controlled filter as a nonlinear filter network," *IEEE Trans. Audio, Speech, Language Process.*, vol. 18, no. 4, pp. 760–772, Apr. 2010.

[9] S. D'Angelo and V. Välimäki, "Generalized Moog ladder filter: Part II–explicit nonlinear model through a novel delay-free loop implementation method," *IEEE/ACM Trans. Audio, Speech, Language Process.*, vol. 22, no. 12, pp. 1873–1883, Dec. 2014.

[10] M. Rest, J. Parker, and K. J. Werner, "WDF modeling of a Korg MS-50 based non-linear diode bridge VCF," in *Proc. Int. Conf. Digital Audio Effects (DAFx-17)*, Edinburgh, UK, Sept. 2017, pp. 61–164.

[11] T. Stilson and J. Smith, "Alias-free digital synthesis of classic analog waveforms," in *Proc. Int. Comput. Music Conf.*, Hong Kong, Aug. 1996, pp. 332–335.

[12] E. Brandt, "Hard sync without aliasing," in *Proc. Int. Comput. Music Conf.*, Havana, Cuba, Sept. 2001, pp. 365–368.

[13] V. Välimäki, J. Nam, J. O. Smith, and J. S. Abel, "Alias-suppressed oscillators based on differentiated polynomial waveforms," *IEEE Trans. Audio Speech Lang. Process.*, vol. 18, no. 4, pp. 786–798, May 2010.

[14] F. Esqueda, H. Pöntynen, V. Välimäki, and J. D. Parker, "Virtual analog Buchla 259 wavefolder," in *Proc. Int. Conf. Digital Audio Effects (DAFx-17)*, Edinburgh, UK, Sept. 2017, pp. 61–164.

[15] J. Parker, "A simple digital model of the diode-based ring modulator," in *Proc. Int. Conf. Digital Audio Effects (DAFx-11)*, Paris, France, Sept. 2011, pp. 163–166.

[16] R. A. Schaefer, "Electronic musical tone production by nonlinear waveshaping," *J. Audio Eng. Soc.*, vol. 18, no. 4, pp. 413–417, 1970.

[17] D. Arfib, "Digital synthesis of complex spectra by means of multiplication of nonlinear distorted sine waves," *J. Audio Eng. Soc.*, vol. 27, no. 4, pp. 757–768, 1979.

[18] M. Le Brun, "Digital waveshaping synthesis," *J. Audio Eng. Soc.*, vol. 27, pp. 250–266, 1979.

[19] J. M. Chowning, "The synthesis of complex audio spectra by means of frequency modulation," *J. Audio Eng. Soc.*, vol. 21, no. 7, pp. 526–534, 1973.

[20] M. Ishibashi, "Electronic musical instrument," Patent No. 4,658,691, 21 April 1987.

[21] M. F. Hribšek and D. V. Tošić, "Symbolic analysis and design of current-differencing-amplifier filters," *Scientific Technical Review*, vol. 57, no. 2, pp. 19–23, 2007.

[22] National Semiconductor Corporation, *The LM3900: A New Current-Differencing Quad of ± Input Amplifiers*, Application Note 72. Sept. 1972.

[23] D. J. Dailey, *Electronics for Guitarists*, Springer, New York City, NY, USA, 2nd edition, 2012.

[24] R. Marston, "Understanding and using 'Norton' op-amps ICs – part 1," *Nuts and Volts*, vol. 23, no. 7, pp. 51–54, July 2002.

[25] R. Marston, "Understanding and using 'Norton' op-amps ICs – part 2," *Nuts and Volts*, vol. 23, no. 8, pp. 49–53, Aug. 2002.

[26] ARP Instruments, Inc., "4072 Voltage-Controlled Low-pass Filter," Available online: `http://www.yusynth.net/Modular/EN/ARPVCF/index.html` (accessed 15 April 2018).

[27] ARP Instruments, Inc., "Dynamic filter," 1977.

[28] M. Mera, R. Sadoff, and B. Winters, *The Routledge Companion to Screen Music and Sound*, Routledge, Abingdon, UK, 1st edition, 2017.

[29] K. Stone, "Triple wave shaper for music synthesizers," Available online: `https://www.cgs.synth.net/modules/cgs85_tws.html` (accessed 21 March 2018).

[30] T. Banwell and A. Jayakumar, "Exact analytical solution for current flow through diode with series resistance," *Electron. Lett.*, vol. 36, no. 4, pp. 291–292, Feb. 2000.

[31] R. C. D. de Paiva, S. D'Angelo, J. Pakarinen, and V. Välimäki, "Emulation of operational amplifiers and diodes in audio distortion circuits," *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 59, no. 10, pp. 688–692, Oct. 2012.

[32] K. J. Werner, V. Nangia, A. Bernardini, J. O. Smith III, and A. Sarti, "An improved and generalized diode clipper model for wave digital filters," in *Proc. 139th Conv. Audio Eng. Soc.*, New York, USA, Oct.–Nov. 2015.

[33] J. Pekonen and V. Välimäki, "Filter-based alias reduction for digital classical waveform synthesis," in *Proc. Int. Conf. Acoustics, Speech and Signal Processing(ICASSP)*, Las Vegas, NV, USA, Mar.–Apr. 2008, pp. 133–136.

END-TO-END EQUALIZATION WITH CONVOLUTIONAL NEURAL NETWORKS

Marco A. Martínez Ramírez, Joshua D. Reiss

Centre for Digital Music,
Queen Mary University of London
London, United Kingdom
m.a.martinezramirez, joshua.reiss@qmul.ac.uk

ABSTRACT

This work aims to implement a novel deep learning architecture to perform audio processing in the context of matched equalization. Most existing methods for automatic and matched equalization show effective performance and their goal is to find a respective transfer function given a frequency response. Nevertheless, these procedures require a prior knowledge of the type of filters to be modeled. In addition, fixed filter bank architectures are required in automatic mixing contexts. Based on end-to-end convolutional neural networks, we introduce a general purpose architecture for equalization matching. Thus, by using an end-to-end learning approach, the model approximates the equalization target as a content-based transformation without directly finding the transfer function. The network learns how to process the audio directly in order to match the equalized target audio. We train the network through unsupervised and supervised learning procedures. We analyze what the model is actually learning and how the given task is accomplished. We show the model performing matched equalization for *shelving*, *peaking*, *lowpass* and *highpass* IIR and FIR equalizers.

1. INTRODUCTION

Equalization (EQ) is an audio effect widely used in the production and consumption of music. It consists of the modification of frequency content through positive or negative gains which change the harmonic and timbral characteristics of the audio. This is performed for different purposes, such as a corrective/technical filter to reduce masking or leakage within a mixing task, to modify the frequency response of a speaker system, or as an artistic or creative tool when recording a specific audio source.

An equalizer is normally implemented via a filter bank whose coefficients are obtained from the designed cut-off frequency f_0 and quality factor Q. In general, EQ is performed through an arbitrary boost or cut at a given f_0 and Q, and it can be applied in the time-domain and frequency-domain [1]. The filters can be classified into different classes such as *lowpass, highpass, peaking, and shelving*.

Taking into account that multiplying the spectrum of signals is the same as convolving their time-domain representation [2], filtering can be described by (1).

$$y(t) = x(t) * h(t) \rightarrow Y(k) = X(k) \cdot H(k) \tag{1}$$

Where h is the time-domain representation of the filter and x and y are the input and filtered signals respectively. H, X, and Y are the respective frequency-domain representations. In this manner, EQ can be achieved with time-domain convolutions, where the transfer function of the filter bank can be expressed through various signals in the time-domain and the equalized audio signal is obtained through the respective convolutions. Therefore, we investigate EQ as a time-domain convolution transformation, where the inherent content of the input and filtered signals can lead a convolutional neural network (CNN) to match a target frequency response.

Given an arbitrary EQ configuration, our task is to train a deep neural network to learn the specific transformation. In this way, an optimal filter bank decomposition and its latent representation are learned from the input data, and these are transformed and decoded to obtain an audio signal that matches the target. Thus, we explore whether the model can be used for EQ matching using an end-to-end architecture, where raw audio is both the input and the output of the system.

We train a model that matches an EQ objective without explicitly obtaining the parameters of the filters ($gain$, f_0 and Q). We show that a procedure based on convolutional and fully connected layers, via time-domain convolutions and latent-space modifications, can lead us to perform EQ matching or modeling. We analyze what the model is actually learning and use a relevant loss function in the time and frequency domains in order to achieve the equalizer task.

The rest of the paper is organized as follows. In Section 2 we summarize the relevant literature related to equalization matching and end-to-end learning. We formulate our problem in Section 3 and in Section 4 we present the methods. Sections 5, 6 and 7 present the obtained results, their analysis and conclusion respectively.

2. BACKGROUND

2.1. EQ Matching and Automatic Equalization

Several methods have been implemented in order to obtain the parameters of the filters or to match a specific frequency response. [3] provides a review of the different state-of-the-art approaches. These methods apply numerical optimization to find a transfer function that corresponds a given complex or magnitude frequency response. Most common techniques are based on the equation error method [4], the Yule-Walker algorithm [5], the Steiglitz-McBride method [6] and the frequency warped method [7].

Within an automatic mixing framework, [8, 9] explored multitrack EQ as a cross-adaptive audio effect, where the processing

of an individual track depends on the content of all the tracks involved, then, the gains of a five filter, first order, filter bank are obtained based on a perceptual loudness weighing.

Given the raw multitrack recording an the final mixture, [10] used least-squares optimization to estimate the gains and f_0 of FIR filters. [11] proposed a pitch tracking system to perform automatic EQ within a mastering task, where the selected pitches are considered as center frequencies for a set of second order peaking filters. [12] used least squares fitting to equalize an audio signal by using IIR filters with arbitrary frequency responses. A cross-adaptive EQ was implemented in [13], where center and cut-off frequencies of peaking and shelving filters were obtained through the minimization of spectral masking and source separation. Similarly, based on unmasking, [14] obtained the center frequencies and gains of peaking filters and [15] attains the gains of a six-band equalizer based on second-order IIR filters.

Based on an perceptual task, [16] proposed a method where the model is trained manually by the users and through nearest neighbor techniques the equalizer gains are obtained in order to match the training data. In a similar approach, [17, 18] investigated a model that associates the gain of each frequency band with the user's training data.

In order to obtain optimal results, most automatic EQ implementations rely on fixed architectures of filter banks or require prior knowledge of the type of filters to be modeled. Therefore, we explore a general architecture capable of performing equalization matching given an arbitrary frequency response.

2.2. End-to-end learning

End-to-end learning corresponds to the integration of an entire problem as a single indivisible task that must be learned from *end-to-end*. The desired output is obtained from the input by learning directly from the data [19]. Deep learning architectures using this principle have experienced significant growth, since by learning directly from raw audio signals, the amount of required prior knowledge is reduced and the engineering effort is minimized [20].

Most audio applications are in the fields of music information retrieval, music recommendation, and music generation. [20, 21] explored CNNs to solve automatic tagging tasks. The networks autonomously learn features related to the frequency and phase of the raw waveforms, although architectures based on spectrograms still yielded better results. In [22] an end-to-end neural network is investigated for the transcription of polyphonic piano music. In the context of end-to-end supervised source separation, [23] proposed an adaptive autoencoder neural network capable of learning a latent representation from the raw waveform.

Likewise, [24, 25] proposed models that generate audio sample by sample without the need handcrafted features and [26] obtained a model capable of performing singing voice synthesis based on *Wavenet* [27] autoencoders.

End-to-end learning has not been implemented for audio effect processing, though recent work demonstrated the usefulness of deep learning applied to intelligent music production systems. [28, 29] explored deep neural networks (DNN) to perform source separation in order to remix the obtained stems and [30] used autoencoders to achieve automatic dynamic range compression for mastering applications. Furthermore, most implementations rely on the magnitude of different frequency representations (spectrogram, melspectogram, etc.), thus omitting the phase information. This is sometimes not ideal, since the task under study could also be based on phase transformations, and therefore would not be learned by the models.

3. PROBLEM FORMULATION

For a specific EQ configuration or arbitrary combination of filters, consider x and y the raw and equalized audio signals respectively. We train a CNN autoencoder which operates as a filter bank and produces a latent representation $\boldsymbol{Z}$ of the given task. One CNN layer can be described by:

$$\boldsymbol{X}_k = \sum_{i=0}^{N-1} \boldsymbol{X}_{k-1}(n-i) \cdot \boldsymbol{W}_k(i) \tag{2}$$

Where $\boldsymbol{X}_k$ represents the feature map of the k_{th} layer, N represents the size of the input feature map $\boldsymbol{X}_{k-1}$ or input frame x in the case of the first layer, and $\boldsymbol{W}_k$ is the kernel matrix with K filters. The latent representation $\boldsymbol{Z}$ is obtained after a designated number of convolutional and subsampling layers.

Thus, in order to obtain a $\hat{y}$ that matches the EQ target y, we implement a deep neural network to modify $\boldsymbol{Z}$ based on the EQ task. Finally, the decoder implements the deconvolution operation and reconstructs the time-domain signal by inverting the operations of the encoder. We train the whole network within an end-to-end learning framework and we minimize a suitable metric between the target and the output of the network.

Based on an EQ matching task, we expect the network to learn the relevant filters $\boldsymbol{W}_k$, latent representation $\boldsymbol{Z}$ and further manipulation. We attempt to find a general architecture that can serve as a matching equalizer based on an arbitrary time-invariant EQ target.

4. METHODS

4.1. Model

In order to implement the network, we followed a similar procedure as [23], although based entirely on the time-domain. The model can be divided into three parts: adaptive front-end, synthesis back-end and latent-space DNN. The model is depicted in Fig. 1.

4.1.1. Adaptive front-end

The adaptive front-end consist of a convolutional encoder. It contains two CNN layers, one pooling layer and one residual connection for the back-end. The front-end performs time-domain convolutions with the raw waveform in order to map it into a latent-space. It also generates a residual connection which facilitates the reconstruction of the audio signal by the back-end. This differs

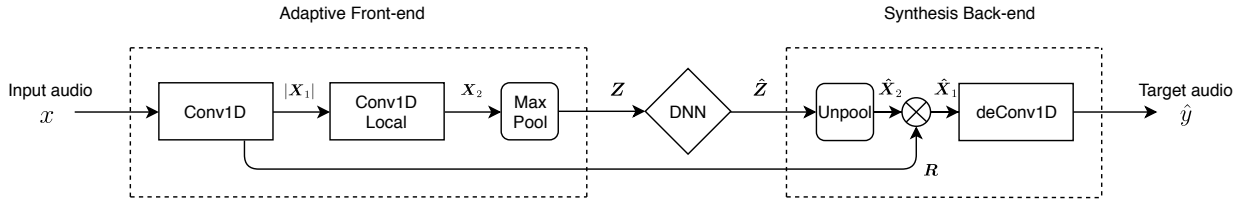


Figure 1: *Block diagram of the proposed model; adaptive front-end, synthesis back-end and latent-space DNN.*

from traditional autoencoders, where the complete input data is encoded into a latent-space, which causes each layer in the decoder to solely generate the complete desired output [31]. Furthermore, a full encoding approach such as [25, 27] will require very deep models, large data sets and difficult training procedures.

The input layer has 128 one-dimensional filters of size 64. Based on (2), the operation performed by the first layer can be described by (3).

$$\boldsymbol{X}_1 = x * \boldsymbol{W}_1 \tag{3}$$

$$\boldsymbol{R} = \boldsymbol{X}_1 \tag{4}$$

Where $\boldsymbol{R}$ is the matrix of the residual connection, $\boldsymbol{X}_1$ is the feature map or frequency decomposition matrix after the input signal x is convolved with the kernel matrix $\boldsymbol{W}_1$. The first layer is followed by the *absolute value* as non-linear activation function.

The second layer has 128 one-dimensional filters of size 128 and each filter is locally connected. This means we follow a filter bank architecture by having unshared weights in the second layer since each filter is only applied to its corresponding row in $|\boldsymbol{X}_1|$. The filters in this layer are larger due to convolving $|\boldsymbol{X}_1|$ with suitable averaging filters $\boldsymbol{W}_2$ could lead the model to learn smoother representations [23], such as envelopes. This layer is followed by the *softplus* non-linearity.

$$\boldsymbol{X}_2 = \mathit{softplus}(|\boldsymbol{X}_1| * \boldsymbol{W}_2) \tag{5}$$

Where $\boldsymbol{X}_2$ is the second feature map obtained after the local convolution with $\boldsymbol{W}_2$, the kernel matrix of the second layer.

The latent-space representation $\boldsymbol{Z}$ is achieved by the *max-pooling* operation. This pooling function consists of a moving window of size 16 applied over $\boldsymbol{X}_2$ and the maximum value within that window correspond to the output. Also, the positions in time of the maximum values are stored and used by the decoder.

4.1.2. Synthesis back-end

In order to invert the operations performed by the front-end, the decoder consists of one CNN layer and one unpooling layer. Since the *max-pooling* function is non-invertible, the inverse can be approximated by recording the locations of the maximum values in each pooling window [32] and only upsampling $\boldsymbol{Z}$ at these time indices. Thus the discrete approximation $\hat{\boldsymbol{X}}_2$ is obtained.

The approximation $\hat{\boldsymbol{X}}_1$ of matrix $\boldsymbol{X}_1$ is obtained through the element-wise multiplication of the residual $\boldsymbol{R}$ and $\hat{\boldsymbol{X}}_2$.

$$\hat{\boldsymbol{X}}_\mathbf{1} = \boldsymbol{R} \cdot \hat{\boldsymbol{X}}_2 \tag{6}$$

Depending on whether $\boldsymbol{Z}$ has been modified or not, (6) can be seen as a sampling or transformation of $\boldsymbol{X}_\mathbf{1}$.

The final layer corresponds to the deconvolution operation, which can be implemented by transposing the first layer transform. This layer is not trainable since its kernels are transposed versions of $\boldsymbol{W}_1$. In this way, the synthesis layer reconstructs the audio signal in the same manner the front-end decomposed it.

$$\hat{y}(t) = \hat{\boldsymbol{X}}_\mathbf{1} * \boldsymbol{W}_1^T \tag{7}$$

All convolutions are along the time dimension and all strides are of unit value. This means, during convolution, we move the filters one sample at a time.

4.1.3. Latent-space deep neural network

The latent-space DNN contains two layers, which are based on locally connected and fully connected dense layers respectively. Thus, following the filter bank architecture, the first layer applies a different dense layer to each row of the matrix $\boldsymbol{Z}$. Each of the locally connected dense layers has 64 hidden units and is followed by the *softplus* activation function. The second layer consists of a fully connected neural network of 64 hidden units, which is applied in each row of the output matrix from the first layer. It is also followed by the *softplus* activation function.

The output of the max pooling operation $\boldsymbol{Z}$ corresponds to an optimal latent representation of the input audio given the EQ task. The DNN is trained to modify this matrix, thus, a new latent representation $\hat{\boldsymbol{Z}}$ is fed into the synthesis back-end in order to reconstruct an audio signal that matches the target task.

4.2. Training

The training of the model is performed in two steps. The first step is to train both the adaptive front-end and the synthesis back-end for an unsupervised learning task. This can be considered as a pre-training of the autoencoder since the model showed better results than when only trained with the second training step. The second

step consists of an end-to-end supervised learning task based on a given EQ target.

During the pretraining only the weights $\boldsymbol{W}_1$ and $\boldsymbol{W}_2$ are optimized and both the raw audio x and equalized audio y are used as input and target functions. This means the model is being prepared to reconstruct the input and target data in order to have a better fitting when training for the EQ task. Once the convolutional autoencoder is trained, the latent-space DNN is incorporated in the model. Hence, the second training procedure consists in using as objectives of the model x and y as input and target respectively. During the end-to-end learning, all the weights of the convolutional and dense layers are updated. This is done independently for each EQ task.

The loss function to be minimized is based in time and frequency and described by (8).

$$loss = kl(Y, \hat{Y}) + mse(Y, \hat{Y}) + mae(y, \hat{y}) \tag{8}$$

Where *kl* is the normalized Kullback-Leibler divergence, the mean squared error is *mse*, and *mae* is the mean absolute error. Y and $\hat{Y}$ are the frequency magnitude of the target and output respectively, and y and $\hat{y}$ their respective waveforms. We use a 1024-point Fourier transform (FFT) in order to obtain Y and $\hat{Y}$, which we extract on the GPU using *Kapre* [33].

We selected a more specialized loss function since by introducing spectral terms in a frequency related task, such as EQ, fewer training iterations were required. In both training procedures the input and target audio is windowed by a *hanning* function into frames of 1024 samples with hop size of 64 samples. The batch size consisted of the total number of frames per audio sample and 100 iterations were carried out in each training step. *Adam* is used as optimizer.

4.3. Dataset

The raw audio x is obtained from the *Salamander Grand Piano V3* dataset[1], which consists of a *Yamaha C5* grand piano sampled in minor thirds from the lowest *A* note and with 16 velocity layers for each note. The dataset is augmented by pitch shifting each note until all the available semitones of the piano are obtained. This gives us a total of 1440 samples. The piano notes are downsampled to 16 kHz and trimmed to 4 seconds. The test and validation subsets correspond to 10% of the dataset and contain a musical note (*B*) not present in the training subset.

The EQ targets y are obtained by applying the filters described in Table 1.

Table 1: *Filter parameters of the EQ targets.*

EQ	filter type	order	gain (dB)	f_0 (Hz)	Q
$shelving$	IIR	2	10	500	0.707
$peaking$	IIR	2	10	500	0.707
$lowpass$	FIR	50	0	500	.
$highpass$	FIR	50	0	500	.

[1] CC BY

5. RESULTS

The unsupervised and supervised learning steps were performed for each type of EQ target. Then, the models were tested with samples from the test dataset.

Fig. 2 shows various visualizations from the front-end and back-end of the autoencoder after the unsupervised training procedure. Fig. 2a displays the waveform and frequency magnitude of a test frame x of 1024 samples and its respective reconstruction $\hat{x}$. The weights of the first convolutional layer $\boldsymbol{W}_1$ can be seen in Fig. 2b, where the first 32 filters are shown.

Consequently, in order to obtain $\hat{x}$, different plots from the front-end, latent-space and back-end are shown in Figs. 2c-2e. The results of (3) can be seen in Fig. 2c where the first 32 rows of $\boldsymbol{X}_1$ are displayed. Fig. 2d presents their latent-space representation $\boldsymbol{Z}$, which is obtained through the second convolutional and subsampling layers. Fig. 2e shows $\hat{\boldsymbol{X}}_1$, which is the result of (6) and the input to the deconvolution layer, the prior step to obtain the output frame $\hat{x}$.

Following the pretraining of the autoencoder, the model is trained through an end-to-end supervised learning method. For each EQ task, Fig. 3 shows the results of selected samples from the test dataset. For a specific frame of 1024 samples, the input, target and output waveforms as well as their FFT magnitudes are displayed. The power spectrogram of the respective 4-second samples is also shown. Finally, together with the input and the target, the complete reconstructed output waveform of a shelving EQ task is presented in Fig. 4.

The performance of the models, and their respective losses (8) in time and frequency can be seen in Table 2.

Table 2: *Evaluation of the models with the test datasets. Loss values for each EQ task.*

EQ	*kl*	*mse*	*mae*	*loss*
$shelving$	0.021845	0.007764	0.002474	0.032083
$peaking$	0.022038	0.007847	0.002521	0.032406
$lowpass$	0.025365	0.005345	0.002710	0.033420
$highpass$	0.021463	0.000951	0.001293	0.023708

6. ANALYSIS

6.1. Adaptive front-end and back-end

From the results of the encoder and decoder, a comparison between the input and output waveforms, as well as their FFT magnitude (see Fig. 2a), it can be seen the model manages to reconstruct the input frame almost perfectly. There are minor differences between the magnitudes of the lower and higher frequencies, but it is worth mentioning that the network achieves this by optimizing only two convolutional layers.

During the first training step, the model learns the $\boldsymbol{W}_1$ and $\boldsymbol{W}_2$ weight matrices with 128 filters each. These filters correspond to the optimal weights of the autoencoder for the decomposition and reconstruction of the training data. As expected, from

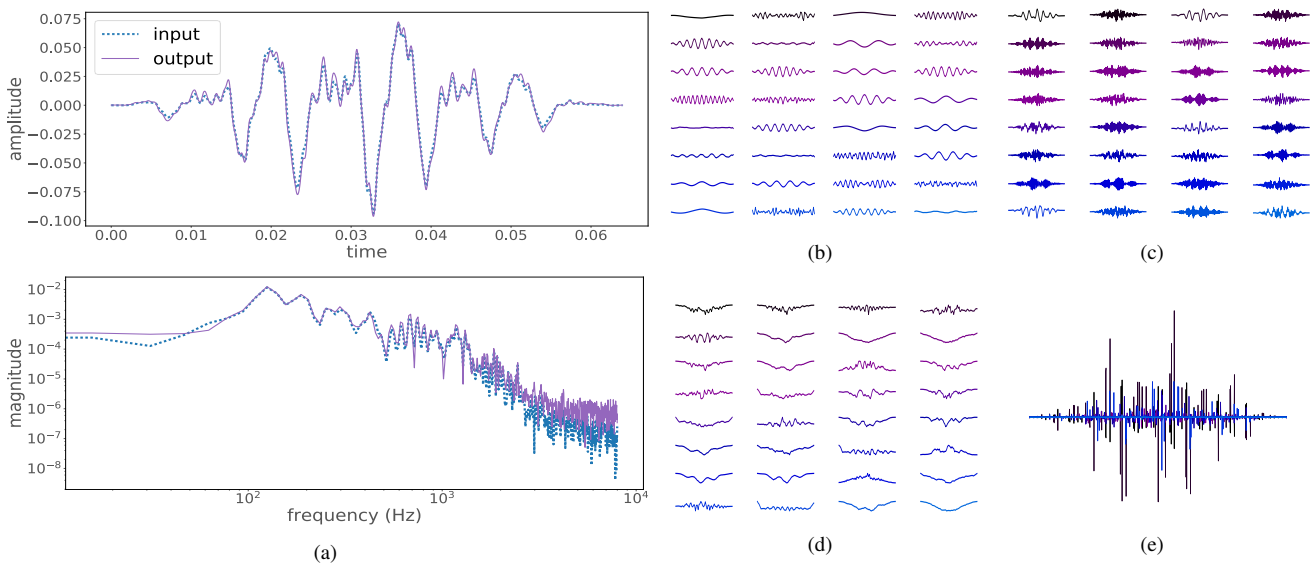


Figure 2: *Various plots from the front-end and back-end with the test dataset after the unsupervised learning step. 2a) Input (x) and output ($\hat{x}$) frames of 1024 samples and their respective FFT magnitude. 2b) First 32 filters ($\boldsymbol{W}_1$) of the first convolutional layer. 2c) First 32 rows of $\boldsymbol{X}_1$, resulting matrix of the convolution between the kernels $\boldsymbol{W}_1$ and the input frame x. 2d) Latent-space representation that is being encoded by the front-end. First 32 rows of $\boldsymbol{Z}$. 2e) Result of the element-wise multiplication between the residual $\boldsymbol{R}$ and the output of the unpooling layer. This is the input to the deconvolution layer prior to obtaining the output frame $\hat{x}(t)$. Vertical axes in 2b)-2e) are unitless and horizontal axes correspond to time.*

the $\boldsymbol{W}_1$ kernels shown in Fig. 2b, it can be observed the filters represent sinusoids and distributions of different frequencies. Also, upon examination of all the weights, we find some redundancy between the filters. This can be improved by adding kernel or activity regularizations, such as the L_1 or L_2 norm regularizes. In addition, some learned weights follow the *hanning* window shape, which makes sense given that all the input frames were windowed.

From the feature map matrix $\boldsymbol{X}_1$ (see Fig. 2c), the filters $\boldsymbol{W}_1$ are actively acting as a filter bank or frequency selectors, since $\boldsymbol{X}_1$ correspond to the decomposition of the input data into different frequencies. Since this is also the residual matrix $\boldsymbol{R}$, the resulting features consist of the required frequencies from the input data in order to be reconstructed by the back-end and encoded by rest of the front-end.

The second convolutional layer is acting as a smoothing layer, since $\boldsymbol{X}_2$ correspond to positive and negatives envelopes from $\boldsymbol{X}_1$. This is due to the learned averaging filters and the absolute and softplus activation functions. The subsampled version $\boldsymbol{Z}$ is presented in Fig. 2d, where different types envelopes are evident. Therefore, the autoencoder is learning a latent-space representation based on the envelopes of selected frequencies.

Taking into account that the result of the unpooling layer $\hat{\boldsymbol{X}}_2$ corresponds to the values of $\boldsymbol{Z}$ at the time positions registered by the max-pooling layer and padded with zeros between each maximum value. The element-multiplication of $\hat{\boldsymbol{X}}_2$ with $\boldsymbol{R}$ generates a discrete version of the latter, which indicates the amplitudes and positions in time that the deconvolution layer should use to reconstruct the input signal (see Fig. 2e). Thus, convolving $\hat{\boldsymbol{X}}_1$ with $\boldsymbol{W}_1^T$ generates the output frame presented in Fig. 2a.

The front-end and back-end manage to reconstruct the test piano notes with a loss value (8) of 0.104. Adding a simple latent-space neural network or increasing the number of filters in the convolutional layers would improve the results significantly. Also, since the training was performed with a hop size of 64 samples, an ideal unit sample hop size would decrease the loss value, although the training time will increase notably. Given that the unsupervised learning task only acts as pretraining step, and that the autoencoder has a relative small number of trainable parameters (24832) we consider these results to be satisfactory.

6.2. EQ task

Table 2 shows that the model performed well on each EQ task. To provide a reference, the mean loss value between the inputs and targets of the *shelving* testing samples is 1.21. The *kl* is fairly uniform across the four types of equalizers, with a minor increase for the *lowpass* EQ. The same can be said about the *mse* and *mae* with the exception of a significant decrease for the *highpass* EQ. Therefore, loss function values were minimal and the model is capable of matching the most common types of EQ, whether these are based on FIR or IIR filters.

The model achieved the best results during the *highpass* task, which could be an indication of the frequency distribution among the training data. Since only piano notes where used, and most spectral energy of acoustic pianos is within 250 Hz - 1 kHz with higher frequencies responsible for the perceived timbral quality of the notes [34]. Thus, having a 500 Hz cut-off frequency could sig-

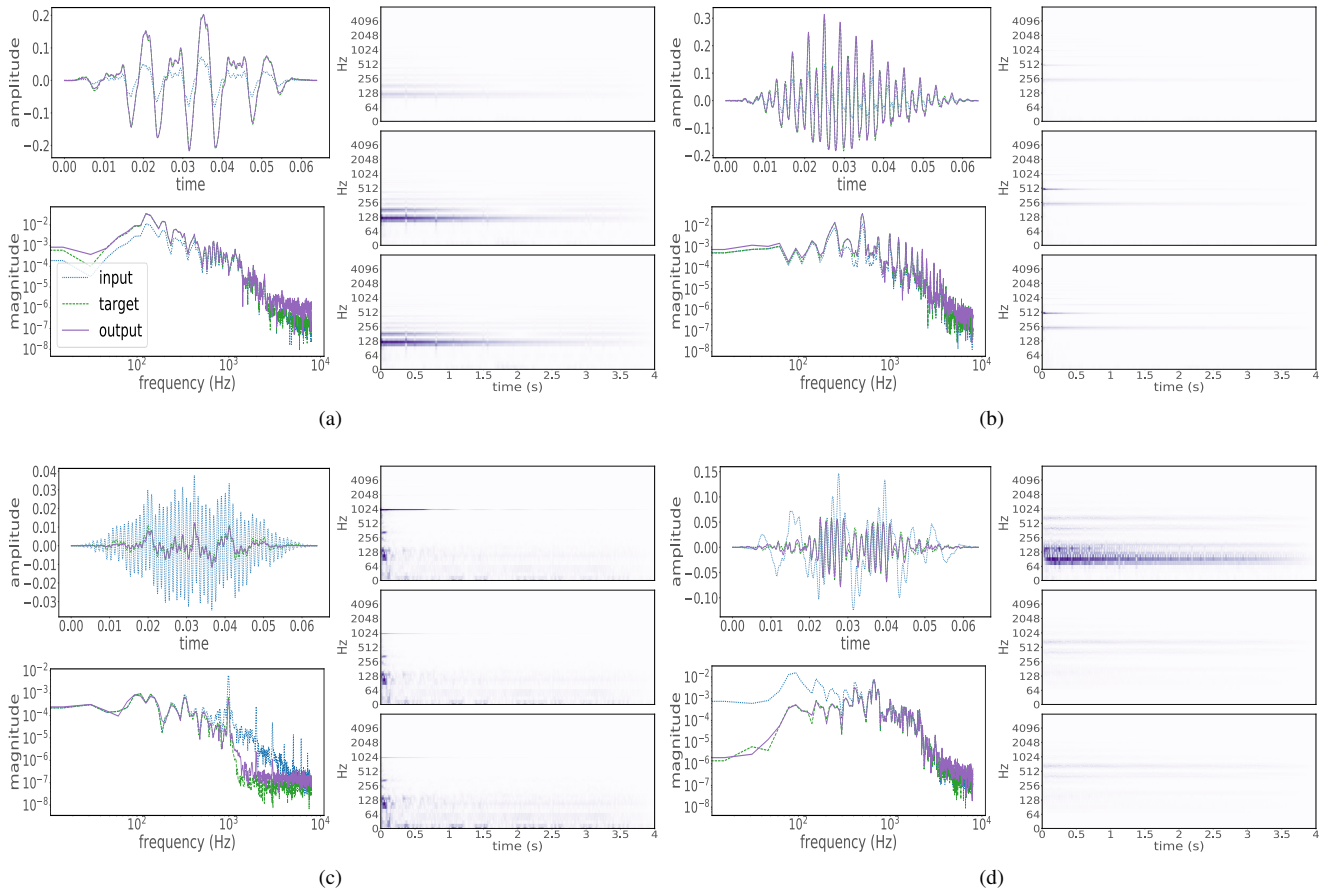


Figure 3: *Results with the test dataset for the following EQ tasks: 3a) shelving, 3b) peaking, 3c) lowpass and 3d) highpass. In 3a)-3d), the input, target and output frames of 1024 samples are shown in waveforms and their respective FFT magnitudes. In addition, for each EQ task and from top to bottom: input, target and output power spectrograms of the 4-second test samples are displayed. Color intensity represents higher energy.*

nify that the model effectively filters out the lower-end of the piano notes by efficiently learning the filters for this task. The slightly worse performance for the *lowpass* task could be further explored by adding kernel regularizations on the CNN layers.

Fig. 3 confirms the correct EQ matching for the different types of equalizers. The spectral and waveform comparison between input, target and output shows how accurate the model is at reconstructing an audio signal that matches the EQ task. For individual frames and complete piano notes, the different types of EQ are evident from the FFT magnitude and power spectogram respectively.

For the *shelving* EQ in Fig. 3a, the effect of the equalizer can be seen in the target and output spectral plots. The power spectrogram shows how the spectral energy was boosted for frequencies lower than 500 Hz. From the FFT magnitude it can be noticed a minor deviation in the lower-end of the target, where there is a boost increment around 20 Hz. This could indicate a weak generalization around these frequencies, which could be improved by using a loss function with higher resolution in the lower-end [7].

The *peaking* equalizer can be seen in Fig. 3b. The selective boost at 500 Hz is notorious both in the FFT magnitude and in the power spectrogram. There is a minor boost in the lower-end which is a consequence of the reasons discussed above. Overall the results indicate a significant fitting for the *peaking* EQ task. Accordingly, the model is able to match EQ tasks based on *peaking* and *shelving* IIR filters.

Likewise, the *lowpass* and *highpass* EQ targets were correctly accomplished. Fig. 3c-3d show the cut of frequencies higher than 500 Hz for the *lowpass* and the opposite for the *highpass*. As discussed, it can be seen the model performs the best for the *highpass* EQ task, obtaining a highly accurate matching between target and output in both time and frequency domains.

The model was trained in a frame-by-frame basis and the input frames were windowed. So the model learned the windowing procedure and the output frames followed the *hanning* shape. Therefore, in order to reconstruct the complete audio signal (see Fig. 4), no further windowing was needed. The overlapping procedure was carried out by applying a gain in order to ensure a Constant

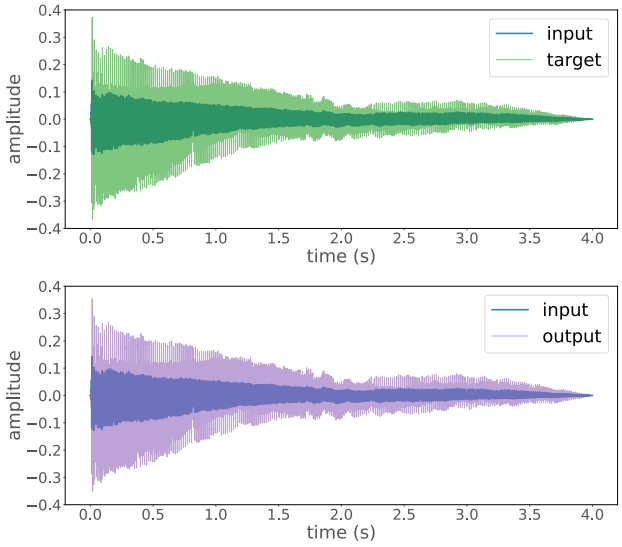


Figure 4: *For a test sample of the shelving EQ task, complete waveform reconstruction of the output and comparison with the input and target. See Fig. 3a for the power spectogram of these waveforms.*

Overlap-Add [35], which is specific to the type of window and hop size.

7. CONCLUSION

In this work, we proposed a novel deep learning architecture capable of performing an audio processing task such as EQ matching. To achieve this, based on the universal approximation capabilities of neural networks, we explored a convolutional adaptive front-end and back-end together with a latent-space deep neural network. Thus, we introduced a general purpose architecture for EQ matching able to model different types of equalizers and filters.

We showed the model matching *shelving*, *peaking*, *lowpass* and *highpass* IIR and FIR equalizers. For each EQ task the model was trained via unsupervised and supervised learning procedures. The latter corresponded to an end-to-end learning approach, which presents and advantage towards common methods of automatic EQ since no prior knowledge of the type of filters nor fixed filter bank architecture is required. Accordingly, the proposed model approximated the target as a content-based transformation without using or obtaining filter parameters. Therefore, the model learned an optimal filter bank decomposition and latent representation from the training data, and correspondingly, how to modify it in order to obtain an audio signal that matches the EQ task.

Possible applications for this architecture are within the fields of automatic mixing and audio effect modeling. For example, style-learning of a specific sound engineer could be explored, where the model is trained with several tracks equalized by the engineer and finds a generalization from the engineer's EQ practices. Also, automatic EQ for a specific instrument across one or several genres could be analyzed and implemented by the model.

Our implementation can serve as a baseline model for deep learning architectures in the context of audio processing. Linear transformations within a mixing task could be easily achieved. As future work, the exploration of recurrent or recursive neural networks or adaptive activation functions can improve the capabilities of the network to model much more complex audio effects. In this case, transformations involving temporal dependencies such as compression or different modulation effects, as well as complicated distortion effects, could be implemented.

A further exploration of the latent-space DNN, or deeper convolutional layers within the encoder and decoder could improve the results of the model. As well as regularizers and loss functions based on frequency wrappers. Also, since training on piano semitones provides only a sparse sampling of the frequency dimension, the generalization capability of the model should be extended for much more complex audio signals, such as noise, human voice or non-musical sounds. Therefore, a further exploration with a less homogeneous dataset together with an analysis of the type of filters learned by the model could benefit the design of a general architecture for modeling audio effects.

Finally, it is worth noting the immense benefit that generative music could obtain from deep learning architectures for intelligent music production. Our implementation could be used in the field of deep neural networks applied to generative music and automatic mixing production systems.

Acknowledgments

This work has been possible thanks to the computational resources by "Fusing Audio and Semantic Technologies for Intelligent Music Production and Consumption" (FAST IMPACt) EPSRC Grant EP/L019981/1.

8. REFERENCES

[1] Vincent Verfaille, U. Zölzer, and Daniel Arfib, "Adaptive digital audio effects (A-DAFx): A new class of sound transformations," *IEEE Transactions on Audio, Speech and Language Processing*, vol. 14, no. 5, pp. 1817–1831, 2006.

[2] Pedro D. Pestana, *Automatic mixing systems using adaptive digital audio effects*, Ph.D. thesis, Universidade Católica Portuguesa, 2013.

[3] Vesa Välimäki and Joshua D. Reiss, "All about audio equalization: Solutions and frontiers," *Applied Sciences*, vol. 6, no. 5, pp. 129, 2016.

[4] Julius Orion Smith, *Introduction to digital filters: with audio applications*, vol. 2, Julius Smith, 2007.

[5] Benjamin Friedlander and Boaz Porat, "The modified yule-walker method of arma spectral estimation," *IEEE Transactions on Aerospace and Electronic Systems*, , no. 2, pp. 158–173, 1984.

[6] Leland B Jackson, "Frequency-domain steiglitz-mcbride method for least-squares iir filter design, arma modeling, and periodogram smoothing," *IEEE Signal Processing Letters*, vol. 15, pp. 49–52, 2008.

[7] Aki Härmä et al., "Frequency-warped signal processing for audio applications," *Journal of the Audio Engineering Society*, vol. 48, no. 11, pp. 1011–1031, 2000.

[8] Enrique Perez-Gonzalez and Joshua D. Reiss, "Automatic equalization of multichannel audio using cross-adaptive methods," in *127th Audio Engineering Society Convention*, 2009.

[9] Enrique Perez-Gonzalez and Joshua D. Reiss, "Automatic mixing," *DAFX: Digital Audio Effects, Second Edition*, pp. 523–549, 2011.

[10] Daniele Barchiesi and Joshua D. Reiss, "Reverse engineering of a mix," *Journal of the Audio Engineering Society*, vol. 58, no. 7/8, pp. 563–576, 2010.

[11] Stylianos I. Mimilakis et al., "Automated tonal balance enhancement for audio mastering applications," in *134th Audio Engineering Society Convention*, 2013.

[12] Zheng Ma et al., "Implementation of an intelligent equalization tool using yule-walker for music mixing and mastering," in *Audio Engineering Society Convention 134*. Audio Engineering Society, 2013.

[13] Daniel Matz, Estefanía Cano, and Jakob Abeßer, "New sonorities for early jazz recordings using sound source separation and automatic mixing tools.," in *16th International Society for Music Information Retrieval Conference (ISMIR)*, 2015.

[14] Sina Hafezi and Joshua D. Reiss, "Autonomous multitrack equalization based on masking reduction," *Journal of the Audio Engineering Society*, vol. 63, no. 5, pp. 312–323, 2015.

[15] David Ronan et al., "Automatic minimisation of masking in multitrack audio using subgroups," *IEEE Transactions on Audio, Speech, and Language processing*, 2018.

[16] Dale Reed, "A perceptual assistant to do sound equalization," in *5th International Conference on Intelligent User Interfaces*. ACM, 2000, pp. 212–218.

[17] Andrew T Sabin and Bryan Pardo, "A method for rapid personalization of audio equalization parameters," in *17th ACM International Conference on Multimedia*, 2009.

[18] Bryan Pardo, David Little, and Darren Gergle, "Building a personalized audio equalizer interface with transfer learning and active learning," in *2nd International ACM Workshop on Music Information Retrieval with User-Centered and Multimodal Strategies*, 2012.

[19] Urs Muller et al., "Off-road obstacle avoidance through end-to-end learning," in *Advances in neural information processing systems*, 2006.

[20] Sander Dieleman and Benjamin Schrauwen, "End-to-end learning for music audio," in *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2014.

[21] Jordi Pons et al., "End-to-end learning for music audio tagging at scale," in *31st Conference on Neural Information Processing Systems (NIPS)*, 2017.

[22] Siddharth Sigtia, Emmanouil Benetos, and Simon Dixon, "An end-to-end neural network for polyphonic piano music transcription," *IEEE/ACM Transactions on Audio, Speech and Language Processing*, vol. 24, no. 5, pp. 927–939, 2016.

[23] Shrikant Venkataramani, Jonah Casebeer, and Paris Smaragdis, "Adaptive front-ends for end-to-end source separation," in *31st Conference on Neural Information Processing Systems (NIPS)*, 2017.

[24] Soroush Mehri et al., "Samplernn: An unconditional end-to-end neural audio generation model," in *5th International Conference on Learning Representations (ICLR)*, 2017.

[25] Jesse Engel et al., "Neural audio synthesis of musical notes with wavenet autoencoders," in *34th International Conference on Machine Learning*, 2017.

[26] Merlijn Blaauw and Jordi Bonada, "A neural parametric singing synthesizer," in *Interspeech 2017*.

[27] Aaron van den Oord et al., "Wavenet: A generative model for raw audio," *CoRR abs/1609.03499*, 2016.

[28] Gerard Roma et al., "Music remixing and upmixing using source separation," in *2nd AES Workshop on Intelligent Music Production*, 2016.

[29] Stylianos I. Mimilakis et al., "New sonorities for jazz recordings: Separation and mixing using deep neural networks," in *2nd AES Workshop on Intelligent Music Production*, 2016.

[30] Stylianos I. Mimilakis et al., "Deep neural networks for dynamic range compression in mastering applications," in *140th Audio Engineering Society Convention*, 2016.

[31] Kaiming He et al., "Deep residual learning for image recognition," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2016.

[32] Matthew D Zeiler and Rob Fergus, "Visualizing and understanding convolutional networks," in *European Conference on Computer Vision*, 2014.

[33] Keunwoo Choi, Deokjin Joo, and Juho Kim, "Kapre: On-gpu audio preprocessing layers for a quick implementation of deep neural network models with keras," in *34th International Conference on Machine Learning*, 2017.

[34] David M Koenig, *Spectral analysis of musical sounds with emphasis on the piano*, OUP Oxford, 2014.

[35] Jérôme Antoni and Johan Schoukens, "A comprehensive study of the bias and variance of frequency-response-function measurements: Optimal window selection and overlapping strategies," *Automatica*, vol. 43, no. 10, pp. 1723–1736, 2007.

CONTACT SENSOR PROCESSING FOR ACOUSTIC INSTRUMENT RECORDING USING A MODAL ARCHITECTURE

Mark Rau, Jonathan S. Abel, and Julius O. Smith III

Center for Computer Research in Music and Acoustics,
Stanford University
Stanford, USA
[mrau|abel|jos]@ccrma.stanford.edu

ABSTRACT

This paper proposes a method to filter the output of instrument contact sensors to approximate the response of a well placed microphone. A modal approach is proposed in which mode frequencies and damping ratios are fit to the frequency response of the contact sensor, and the mode gains are then determined for both the contact sensor and the microphone. The mode frequencies and damping ratios are presumed to be associated with the resonances of the instrument. Accordingly, the corresponding contact sensor and microphone mode gains will account for the instrument radiation. The ratios between the contact sensor and microphone gains are then used to create a parallel bank of second-order biquad filters to filter the contact sensor signal to estimate the microphone signal.

1. INTRODUCTION

Acoustic string instruments often lack the radiated sound power to compete with louder instruments such as drums or piano in a live or recording scenario. The most natural way to amplify their sound is using a well placed microphone, but this can be problematic as feedback and "bleed" sound from other instruments are common. To overcome these problems, pickups or contact sensors are used as they more directly capture the instrument's vibrations. Electromagnetic pickups are used with electric guitars, but they capture the strings' vibration and do not capture an authentic sound image of the instrument's body vibrations. Contact sensors such as piezoelectric or electret film sensors are more commonly used with acoustic instruments as they primarily capture the vibrations of the instrument, not purely of the strings.

In this paper, we focus on the upright bass as a test case. When used in a live jazz context, the upright bass almost always requires amplification. The most common method of achieving amplification is by using a contact sensor, typically piezoelectric, and routing the output to an amplifier. The resulting output bares little resemblance to the acoustic sound radiated by the instrument, and typically has a "rubbery" characteristic. In addition to the live scenario, it is often necessary to record upright bass in the same room as other instruments which are much louder, such as a piano or drum set. The sound of these instruments bleeds into the microphones meant for the upright bass, making it difficult to isolate the instrument or apply post-processing. It would be advantageous if the upright bass could be recorded using a contact sensor to achieve an isolated recording, but this is not often done as the acoustic response is desired.

Acoustic instrument contact sensors can be equalized, often in an attempt to make them sound more similar to the instrument's acoustically radiated sound. Commercially available acoustic instrument equalizers are limited in use and require trial and error to achieve a desirable sound. If an instrument's body is approximated as linear and time-invariant system, a transfer function between various point of measurement can be defined which will allow digital signal processing (DSP) techniques to force a signal captured at one location to sound more similar to a signal captured at a different location.

Such DSP equalization has been studied previously by Karjalainen et al. [1, 2, 3]. This work focused on the case of an acoustic guitar with an electret film pickup, and aimed to find a transfer function which was the spectral ratio of microphone and contact sensor transfer functions:

$$Q(\omega) = \frac{P(\omega)}{X(\omega)}\,, \tag{1}$$

where $Q(\omega)$ is an equalizer transfer function, $P(\omega)$ is the acoustic radiation transfer function measured with a microphone, and $X(\omega)$ is the transfer function through a contact sensor. They found transfer functions by first using an impact hammer to excite an impulse, and second by playing musical information through both sensors and deconvolving the contact sensor signal from the microphone signal. They constructed filters based on both of these methods using FIR and IIR structures. It was concluded that the deconvolution method paired with an FIR filter of order 500 or higher with an additional digital resonator tuned to the mode of the guitar's top plate produced the most desirable sound.

Rather than using a spectral ratio based approach, we propose a modal architecture which can be constructed where the mode frequencies and damping ratios are fit to the contact sensor frequency response, and the mode gains are taken as a ratio between the gains fit to the contact sensor and microphone frequency responses. A parallel bank of second-order biquad filters can be used to realize the filter in real time. A modal architecture is chosen because it is modular and has the potential to be altered in real time. This provides the option to choose from or mix between different microphone responses by tuning only the relative mode gains. This can be extended to the case of producing multiple simultaneous simulated microphone responses, which can be efficiently computed because the same set of mode filter outputs can be used to form each microphone's output according to its set of gains.

Much prior work has been done on modeling instrument transfer functions using a modal architecture [4, 5, 6, 7]. This work is typically done in the context of sound synthesis, but is equally valid for the proposed sensor equalization application. The mode parameters can be fit using traditional mode fitting techniques such as the Complex Exponential or Peak Picking methods [8, 9, 10]. The modal fits can be improved using a constrained optimization

algorithm to reduce the error between the experimental and reconstructed frequency response functions [5, 11]. We follow an approach similar to these prior methods, calculating initial mode parameter guesses and using a constrained optimization to improve the reconstructed model.

This paper is organized as follows. Section 2 introduces the process for acquiring instrument impulse response data. Section 3 describes the modal parameter fitting and optimization, and Section 4 describes the steps needed to realize the model as a digital filter. Section 5 presents preliminary results, and Section 6 is a conclusion and discussion of potential improvements and further areas of study.

2. MEASUREMENTS

The proposed method relies on impulse response measurements which serve as the basis for a modal model. An upright bass was used as a case study for measurements and fitting. The upright bass was suspended from the ceiling with the endpin rested on foam for stability. Paper was woven between the strings to prevent them from ringing. An anechoic chamber was not available so the measurements were taken in a medium sized room with ample absorption.

Two commercially available contact sensors were attached to the bass for recording. A piezoelectric sensor was placed under the treble foot of the bridge, and a dynamic contact microphone was placed on the top plate, below the bridge. Five studio microphones were placed in various positions around the bass. The positions were chosen such that they may be typical starting positions for a studio recording of the upright bass. While multiple microphones and contact sensors were used to record the measurements, only one contact sensor and microphone pair is analyzed in this paper. The contact sensor and microphone placements can be seen in Figure 1, with the contact sensor and microphone pair of interest labeled.

A force sensing impact hammer was used to excite an impulse through the instrument. The hammer was struck on the bass side of the bridge, perpendicular to the curvature of the bridge at that location. The bass side of the bridge was chosen as the impact location because it is closest to the lowest string which provides the greatest amount of energy transfer. The hammer was remotely dropped multiple times, while the sensors and microphones recorded the impulse responses at their respective locations.

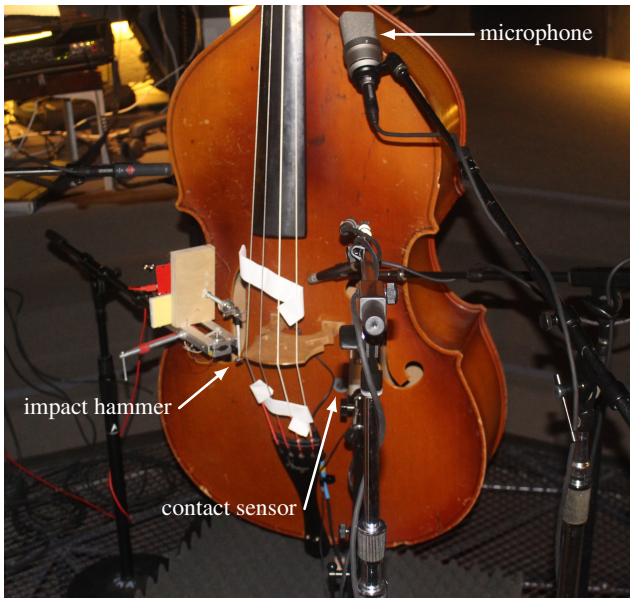


Figure 1: *Measurement setup.*

3. MODE FITTING

3.1. Modal Structure

Modal analysis can be used to investigate the vibrational characteristics of physical structures such as musical instruments [12]. The measured vibrational characteristics of a structure can be described by its frequency response function (FRF) which is a measurement function used to identify the resonant frequencies, damping ratios, and mode shapes of a physical structure. The frequency response function between points p and q of a modal structure can be written as

$$H_{pq}(s) = \sum_{r=1}^{N} \frac{\psi_{pr}\psi_{qr}}{(s^2 + 2\Omega_r\zeta_r s + \Omega_r^2)}\ , \qquad (2)$$

where r is the mode number up to a maximum number of modes, N. The undamped natural frequency Ω_r is defined as $\Omega_r = \sqrt{\sigma_r^2 + \omega_r^2}$, where σ_r is the damping factor and ω_r is the damped natural frequency. The damping ratio ζ_r is defined as $\zeta_r = -\frac{\sigma_r}{\Omega_r}$. The mode shape coefficients at points p and q are ψ_{pr} and ψ_{qr} [13].

3.2. Measurement Preprocessing

Due to the non-anechoic nature of the room and the low amount of energy transferred to the instrument from the impact hammer, the impulse response measurements required preprocessing to allow reliable transfer function fits.

Roughly 100 impulse measurements were taken. Measurements containing double hits from the hammer were discarded. Each impulse was windowed using an exponential window to improve the signal-to-noise ratio [14]. Frequency response functions were calculated for each pair of hammer excitation and sensor signals. The frequency response function is calculated for each measurement set using Welch's method and they are averaged in the frequency domain to reduce random error [13].

3.3. Initial Mode Fitting

An initial pass is made on the mode fitting which uses the Complex Exponential method [9]. The Complex Exponential method computes the time domain impulse response corresponding to the given frequency response function, and a set of complex damped sinusoids is fit using Prony's method. This is a nonlinear process which finds a solution iteratively.

The initial mode fitting process is performed over 9 different frequency bands ranging from 0 to 6 kHz, and the number of modes to fit was determined by eye. The Complex Exponential mode fitting returns estimates of Ω_r, $\hat{\zeta}_r$, and Ψ_r, the product of the complex mode shapes at the impact and measurement locations. The damping ratios $\hat{\zeta}_r$ represent damping ratios fit to the windowed impulse response measurements. Since an exponential decay window is used, it introduces additional damping which will

be corrected for at a later point. The returned undamped natural frequencies and damping coefficients were reasonably fit, but the mode shapes were not as reliable so they were recomputed using the least squares method.

3.4. Choice of Modes

The frequency response function was computed for each sensor location, yielding multiple sets of mode parameters. Theoretically, each set of mode parameters should contain the same undamped natural frequencies, and damping coefficients, varying only by mode shape. However, if a measurement sensor is at or near a relative node location, it is unlikely that an undamped natural frequency will be fit to the frequency response function. Likewise, if a mode is present at a sensor location, it still may be missed due to the measurement noise or the windowing process. Even if a mode is present in multiple sensor measurements, there will likely be numerical differences between mode fittings.

A method was developed to create a set of mode parameters which is common between multiple frequency response functions. A set of common mode parameters is created based on common undamped natural frequencies, worrying about the damping ratios at a later point. Let S_C be a set of undamped natural frequencies measured through a contact sensor, and let $S_{M_1}, ..., S_{M_N}$ be sets of undamped natural frequencies measured through N microphones at various locations around the instrument. To get the set of all undamped natural frequencies present, a union of sorts is taken.

To account for numerical differences between undamped natural frequencies that are common between both sensor sets, a tolerance δ is set, within which there is deemed to be only one unique mode. The undamped natural frequencies in S_C are taken as the true undamped natural frequencies, as only direct measurements from the contact sensor will be used in the final processing. The modes from S_{M_i} which have undamped natural frequencies within δ percent of the undamped natural frequencies in S_C are discarded. This can be summarized as

$$\hat{S}_{M_i} = S_{M_i} \setminus ((1 \pm \delta) S_C) \ , \tag{3}$$

where $\setminus$ represents the set difference, and $\hat{S}_{M_i}$ is the set of undamped natural frequencies only present in S_{M_i} within the set tolerance δ. The set of undamped natural frequencies found in all sensors of interest can then be represented as

$$S_F = S_C \cup \left(\hat{S}_{M_1} \cup ... \cup \hat{S}_{M_N} \right) \ , \tag{4}$$

where $\cup$ represents the set union.

The initial guesses for the damping ratios and mode shapes correspond to the undamped natural frequencies in S_F.

This method for choosing the mode shapes is general to any number of microphone frequency response functions, but for the rest of the paper, a setup consisting of one contact sensor and one microphone is assumed.

3.5. Optimized Mode Fitting

To further refine the modal fitting, a constrained optimization scheme is formed to minimize the error between the measured and reconstructed frequency response function pairs. The optimization problem is posed as

$$\underset{\Omega_r,\, \hat{\zeta}_r,\, \Psi_r}{\text{minimize}} \quad \varepsilon(\hat{H}_C,\, H_C,\, \hat{H}_M,\, H_M) \ , \tag{5}$$

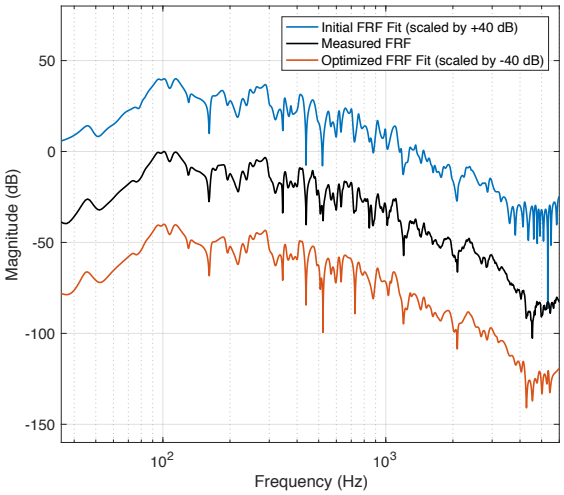


Figure 2: *Contact sensor frequency response function (FRF) and fits with $N = 88$ modes.*

where H_C and $\hat{H}_C$ are the measured and reconstructed frequency response functions for the contact sensor, H_M and $\hat{H}_M$ are the measured and reconstructed frequency response functions for the microphone, and $\varepsilon(\hat{H}_C, H_C, \hat{H}_M, H_M)$ is an error measure to be minimized. The initial mode fits calculated using the Complex Exponentials method are used as initial guesses for the optimization. The optimization constrains the values of Ω_r and $\hat{\zeta}_r$ to be within $\pm$ 50 % of the initial guess values.

During each iteration of the optimization, there is a guess for the values of Ω_r and $\hat{\zeta}_r$. These parameters are held constant for both contact sensor and microphone frequency response function reconstructions. Least squares is used to calculate the mode shapes Ψ_r^C and Ψ_r^M for the contact sensor and microphone modes respectively. The frequency response functions are reconstructed and the following error function is used:

$$\varepsilon(\hat{H}_C,\, H_C,\, \hat{H}_M,\, H_M) = ||H_C - \hat{H}_C||_1 + ||H_M - \hat{H}_M||_1 \ , \tag{6}$$

where $\hat{H}_C$ and $\hat{H}_M$ are the reconstructed frequency response functions using the same sets of undamped natural frequencies Ω_r and damping ratios $\hat{\zeta}_r$, but with their own sets of mode shapes Ψ_r^C and Ψ_r^M, and $|| \cdot ||_1$ is the L1-norm.

Example frequency response functions are shown for a dynamic contact sensor (Figure 2) and a cardioid studio microphone placed roughly 30 cm away from the the instruments top plate near the upper bout (Figure 3). The window exponential decay constant was set to $\beta = 0.07\ s^{-1}$, and the natural frequency tolerance was set to $\delta = 2$ %. The examples show the measured frequency response function as well as the frequency response functions recreated from the initial and optimized mode fits.

4. REALIZATION AS PARALLEL BANK OF SECOND-ORDER BIQUAD FILTERS

The goal of this study is to scale the contact sensor response such that it will better approximate that of the microphone. A choice was made to perform the mode fitting in the continuous domain to

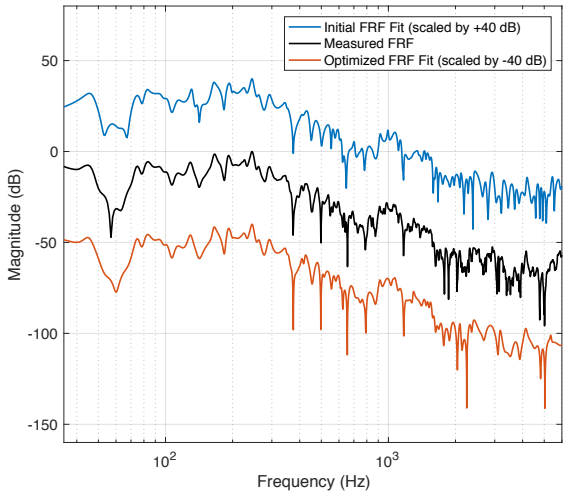


Figure 3: *Microphone frequency response function (FRF) and fits with $N = 88$ modes.*

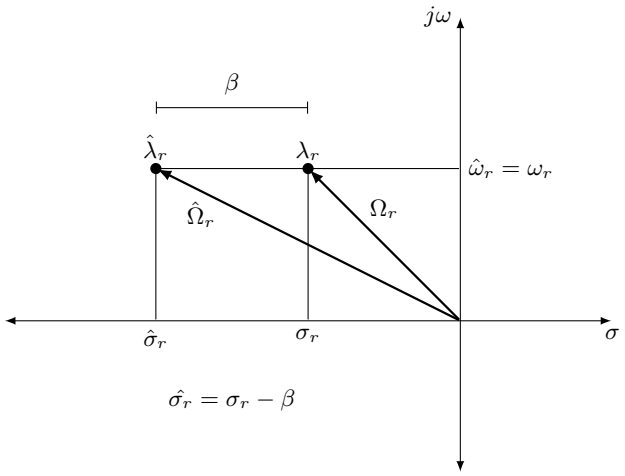


Figure 4: *Effect of the exponential decay window in the complex plane. β is the exponential decay constant of the window. λ_r, ω_r, σ_r, and Ω_r are the eigenvalue, damped natural frequency, damping factor, and undamped natural frequency for mode r. $\hat{\lambda}_r$, $\hat{\omega}_r$, $\hat{\sigma}_r$, and $\hat{\Omega}_r$ have the same meaning except for the windowed signal.*

maintain the physical parametric structure, and to later convert to the discrete domain to facilitate the DSP equalization. This equalization can be realized by using the obtained modal parameters to create a parallel bank of second-order biquad filters which can be described by their undamped natural frequencies, damping coefficients, and mode shapes or gains. The undamped natural frequencies Ω_r obtained using the previously described method can be used, but the damping ratios $\hat{\zeta}_r$ and mode shapes Ψ_r^C and Ψ_r^M need to be adjusted.

4.1. Mode Shape Scaling

It is assumed that the microphone and contact sensor measurements will contain the same set of undamped natural frequencies and damping coefficients, and will differ only by their relative mode shapes. In order to impose the microphone response on the contact sensor, a scaling needs to be performed between the mode shapes. This can be obtained by taking the ratio of the mode shapes

$$G_r = \frac{\Psi_r^M}{\Psi_r^C} \,, \tag{7}$$

which gives the scaling gain between the mode shapes G_r.

4.2. Damping Ratio Correction

The use of the exponential decay window adds additional damping to the measured frequency response which needs to be compensated for when creating the modal scaling filter. The exponential decay window is defined as

$$w_e(t) = e^{-\beta t} \,, \tag{8}$$

where β is the exponential decay constant. Figure 4 shows how the additional damping caused by the window results in a windowed damping ratio $\hat{\sigma}_r$, which is more negative than the true damping ratio σ_r, by the amount of the exponential decay constant used for the window, β.

A common correction approximation for the extra damping caused by the exponential decay window is given by

$$\zeta_r' = \hat{\zeta}_r - \frac{\beta}{\hat{\Omega}_r} \,, \tag{9}$$

where ζ_r' is an approximation to the true damping ratio $\hat{\zeta}_r$ is the damping ratio after the windowing effects, and $\hat{\Omega}_r$ is the undamped natural frequency of the windowed data [14]. The exact expression for the true damping ratio ζ_r is given in the Appendix.

4.3. Analog to Digital: Bilinear Transform

Substituting the corrected damping ratios ζ_r, and the gain between mode shapes G_r into (2) gives

$$Q(s) = \sum_{r=1}^{N} \frac{G_r}{(s^2 + 2\Omega_r \zeta_r s + \Omega_r^2)} \,, \tag{10}$$

which is the transfer function for the s-domain filter needed to scale the contact sensor.

The s-domain transfer function is converted to the discrete domain using the bilinear transform:

$$s = c_r \left(\frac{1 - z^{-1}}{1 + z^{-1}} \right) \,. \tag{11}$$

The natural frequencies are kept constant under the frequency warping caused by the bilinear transform by setting

$$c_r = \frac{\Omega_r}{\tan\left(\frac{\Omega_r}{2 f_s}\right)} \,, \tag{12}$$

where f_s is the sample rate.

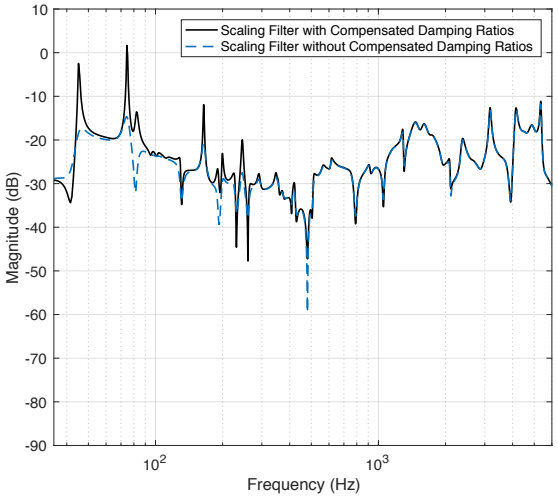


Figure 5: *Modal scaling filter frequency response with $N = 85$ modes.*

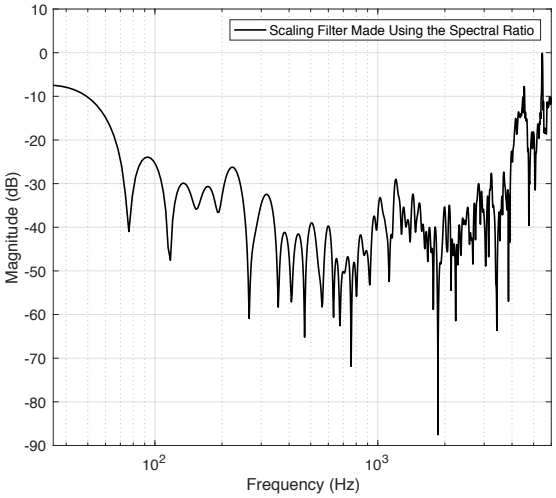


Figure 6: *Spectral ratio scaling filter frequency response implemented using a 1200 tap FIR filter.*

The resulting discrete transfer function is given by

$$Q_r(z) = \frac{b_0 + b_2 z^{-2}}{1 + a_1 z^{-1} + a_2 z^{-2}} \tag{13}$$

where

$$b_0 = b_2 = \frac{G_r}{\Omega_r^2 + c_r^2 + 2c_r\Omega_r\zeta_r}$$
$$a_1 = \frac{2\Omega_r^2 - 2c_r^2}{\Omega_r^2 + c_r^2 + 2c_r\Omega_r\zeta_r}$$
$$a_2 = \frac{\Omega_r^2 + c_r^2 - 2c_r\Omega_r\zeta_r}{\Omega_r^2 + c_r^2 + 2c_r\Omega_r\zeta_r}\,.$$

The modal scaling filter frequency response corresponding to the contact sensor and microphone from Figures 2 and 3 is shown in Figure 5. The frequency response is shown with and without the damping ratio correction.

5. RESULTS AND DISCUSSION

The modal architecture yields a parallel bank of second-order biquad filters which can be used to filter the output of an instrument through a contact sensor, resulting in a signal which should sound similar to that measured through a microphone.

As a comparison to the modal scaling filter, Figure 6 shows the equalization filter using the spectral ratio method of Karjalainen et al., for a 1200 tap FIR filter. The two filters are difficult to compare due to the low spectral resolution of the FIR filter, but some general comparisons can be made. Both filters exhibit a similar overall contour, having a higher magnitude in the low and high frequencies, with a lower magnitude in the mid frequency range of roughly 300-1000 Hz. However, while the general contours of the modal and spectral ratio equalization filters are similar, there are clear differences. Since the spectral ratio filter is implemented as a relatively short FIR filter, there is a low amount of mode resolution, making it impossible to accurately model resonant modes with low damping ratios. While the modal model is able to accurately capture highly resonant modes, it may be incorrectly modeling some modes resulting in discrepancies between the filters.

Figure 7 shows spectrograms of a hammer impulse measured through a contact sensor, a microphone, as well as the contact sensor signal filtered with the modal model. Figure 8 shows the output of the measured upright bass being played. The contact sensor, microphone, and filtered contact microphone sensor signals are shown. Audio examples of the filtered upright bass being played can be found online[1]. Qualitative observations suggest that the contact sensor filtered with the modal architecture is more acoustic sounding and similar to the microphone signal. The filtered contact sensor signal and microphone signal do not sound exactly the same, but this is to be expected as the sensor is only picking up the vibrations present at its location, so it cannot be expected to contain information about the other sounds produced by the instrument or performer.

The proposed modal architecture poses several advantages over the spectral ratio method of Karjalainen et al.. The mode gains can be altered in real time, allowing for on-line tuning of the equalization. This could be used to adjust individual modes which are problematic in a particular playing situation, say if a mode of the instrument is at the same frequency as a room mode of the performance space. If multiple microphone frequency response functions were modeled, this structure allows for simple switching between or interpolating between microphone responses. The major drawback of the modal architecture is the sensitivity of the mode parameter fitting.

The modal fitting is sensitive to the window's exponential decay constant, the set frequency tolerance, as well as the number of modes to be fit. As the window's exponential decay constant is decreased, the signal-to-noise ratio is improved, but the risk of missing modes in the fitting is increased. While decreasing the undamped natural frequency tolerance, the chance of fitting the same mode twice is minimized, but the chance of missing closely spaced modes is increased. Hence, the number of modes to fit is related to the window's exponential decay constant as well as the undamped

[1] https://ccrma.stanford.edu/~mrau/DAFX2018/

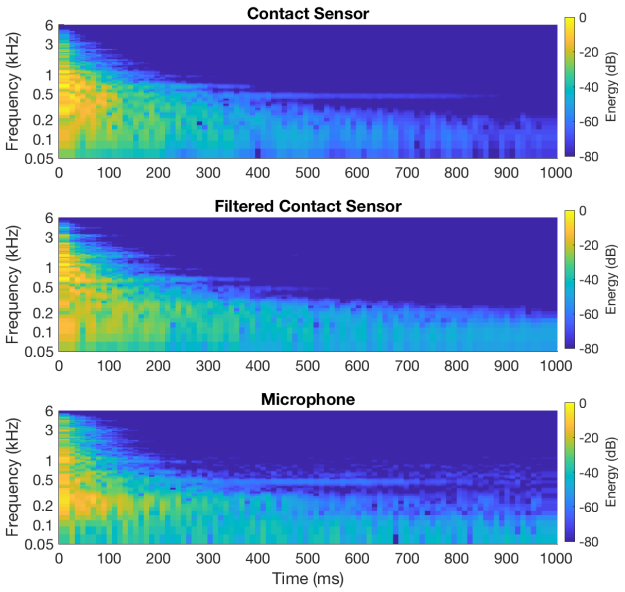


Figure 7: *Spectrogram of an impulse recording.*

natural frequency tolerance. Some trial and error is required to obtain the desired results.

The resulting filtered contact sensor sounds more acoustic, and similar to the the microphone signal; however, it is not perfect. There are likely multiple factors contributing to the differences. The measurements have a low signal-to-noise ratio and were recorded in a non-ideal location making the mode fitting challenging and sensitive to the windowing and parameter initialization. Notably, not all sounds present in the microphone signal will appear in the contact sensor signal. The contact sensor could be placed at a vibrational node of the instrument and will predominantly pick up vibrations in one direction. In this case, using multiple well placed contact sensors would overcome the problem. As well, any sounds such as finger motions on the strings are unlikely to be picked up by the contact sensor. Since these vibrations do not appear in the contact sensor signal, it will not be possible to recreate their presence in the microphone signal by filtering the contact sensor alone.

6. CONCLUSIONS

A modal analysis is developed to design filters to make instrument contact sensors sound more like microphones. An upright bass was used as a case study and impulse response measurements of the instrument were recorded through multiple contact sensors and microphones. The modal parameters are initially fit using the Complex Exponentials method, and are then improved upon using a constrained optimization scheme. The modal parameters are used to form a parallel bank of second-order biquad filters which can be used to equalize a contact sensor signal such that it sounds more similar to a microphone at a specific location.

Avenues for future study include further optimizing the modal architecture as well as expanding to and testing with multiple sensors at various locations. If multiple contact sensors are used, the chance that all sensors will be located at vibrational nodes is small, so there can be more confidence that all modes will be captured. If multiple microphones are used, the ability to interpolate between them to achieve a desirable microphone placement for the output signal is gained.

7. REFERENCES

[1] M. Karjalainen, V. Välimäki, H. Penttinen, and H. Saastamoinen, "DSP equalization of electret film pickup for the acoustic guitar," *Journal of the Audio Engineering Society*, vol. 48, no. 12, pp. 1183–1193, 2000.

[2] M. Karjalainen, H. Penttinen, and V. Välimäki, "Acoustic sound from the electric guitar using DSP techniques," in *Acoustics, Speech, and Signal Processing, 2000. ICASSP'00. Proceedings. 2000 IEEE International Conference on*. IEEE, 2000, vol. 2, pp. II773–II776.

[3] M. Karjalainen, H. Penttinen, and V. Välimäki, "More acoustic sounding timbre from guitar pickups," *in 2nd Workshop on Digital Audio Effects (DAFx-99), Trondheim, Norway*, vol. 10, pp. 1–4, Dec. 9–11, 1999.

[4] M. Karjalainen and J. O. Smith, "Body modeling techniques for string instrument synthesis," in *International Computer Music Conference (ICMC)*, 1996, pp. 232–239.

[5] E. Maestre, G. P. Scavone, and J. O. Smith, "Digital modeling of bridge driving-point admittances from measurements on violin-family instruments," in *Proc. of the Stockholm Music Acoustics Conference*, 2013, pp. 101–108.

[6] E. Maestre, G. P. Scavone, and J. O. Smith, "Digital modeling of string instrument bridge reflectance and body radiativity for sound synthesis by digital waveguides," in *Applications of Signal Processing to Audio and Acoustics (WASPAA), 2015 IEEE Workshop on*. IEEE, 2015, pp. 1–5.

[7] J. O. Smith, *Physical Audio Signal Processing*, W3K Publishing, 2004, online book: `http://-ccrma.stanford.edu/~jos/pasp/`.

[8] P. Antsalo, A. Mäkivirta, V. Välimäki, T. Peltonen, and M. Karjalainen, "Estimation of modal decay parameters from noisy response measurements," in *Proc. Audio Eng. Soc. (AES) Conv., Amsterdam, The Netherlands*, May 12–15, 2001, vol. 110, pp. 867–878.

[9] D. Brown, R. Allemang, R. Zimmerman, and M. Mergeay, "Parameter estimation techniques for modal analysis," Tech. Rep., SAE Technical paper, 1979.

[10] D. Ewins, *Modal Testing: Theory and Practice*, vol. 15, Research studies press Letchworth, 1984.

[11] E. Maestre, J. S. Abel, J. O. Smith, and G. P. Scavone, "Constrained pole optimization for modal reverberation," *in 20th Int. Conf. Digital Audio Effects (DAFx-17), Edinburgh, UK*, pp. 381–388, Sep. 5–9, 2017.

[12] N. Fletcher and T. Rossing, *The Physics of Musical Instruments*, Springer-Verlag, 1991.

[13] A. Brandt, *Noise and Vibration Analysis: Signal Analysis and Experimental Procedures*, John Wiley & Sons, 2011.

[14] W. Fladung and R. Rost, "Application and correction of the exponential window for frequency response functions," *Mechanical systems and signal processing*, vol. 11, no. 1, pp. 23–36, 1997.

[15] H. Penttinen and M. Tikander, "Sound quality differences between electret film (EMFIT) and piezoelectric under-saddle guitar pickups," in *Proc. Audio Eng. Soc. (AES) Conv., Paris, France*, May 20–23, 2006, vol. 120.

[16] B. Peeters, J. Lau, J. Lanslot, and H. Van der Auweraer, "Automatic modal analysis-Myth or reality?," *Sound and Vibration*, vol. 42, no. 3, pp. 17–21, 2008.

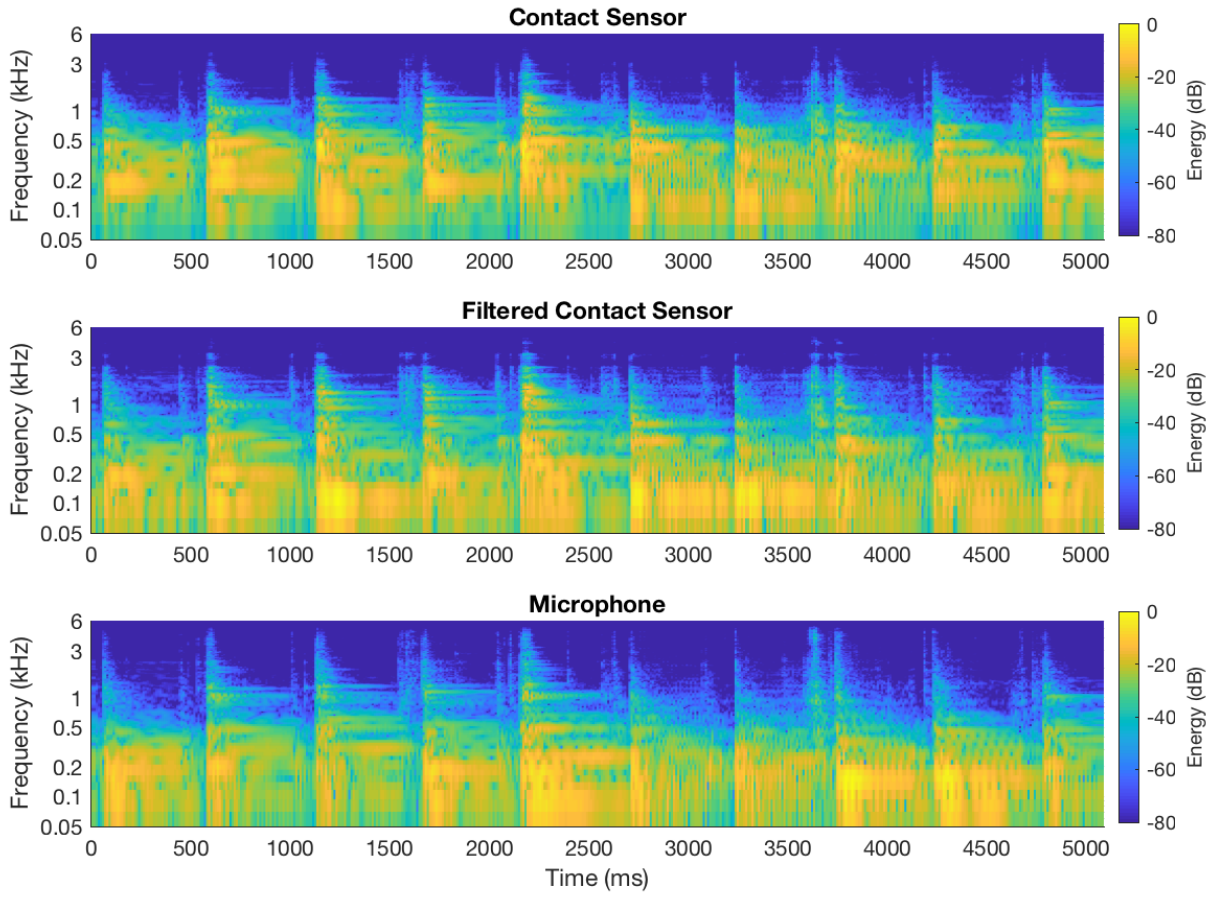


Figure 8: *Spectrogram of the bass being played.*

APPENDIX: DAMPING RATIO COMPENSATION

The exact representation of the original damping coefficient before windowing using the exponential decay window can be found by solving the equation:

$$\zeta_r = \frac{\sqrt{1-\zeta_r^2}}{\sqrt{1-\hat{\zeta}_r^2}}\hat{\zeta}_r - \frac{\beta\sqrt{1-\zeta_r^2}}{\omega_r}\,, \tag{14}$$

which yields the two solutions:

$$\zeta_r \to \pm\frac{\sqrt{\beta^4\hat{\zeta_r}^4 - 2\beta^4\hat{\zeta_r}^2 + \beta^4 + 3\beta^2\omega_r^2\hat{\zeta_r}^4 - 4\beta^2\omega_r^2\hat{\zeta_r}^2 + \beta^2\omega_r^2 - 2\beta\omega_r^3\hat{\zeta_r}\sqrt{1-\hat{\zeta_r}^2} + 2\beta\omega_r^3\hat{\zeta_r}^3\sqrt{1-\hat{\zeta_r}^2} + \omega_r^4\hat{\zeta_r}^2}}{\sqrt{\beta^4\hat{\zeta_r}^4 - 2\beta^4\hat{\zeta_r}^2 + \beta^4 + 4\beta^2\omega_r^2\hat{\zeta_r}^4 - 6\beta^2\omega_r^2\hat{\zeta_r}^2 + 2\beta^2\omega_r^2 + \omega_r^4}}\,. \tag{15}$$

Two solutions are found, but the damping ratio must be positive for a damped system, so the positive solution must be used.

TU-NOTE VIOLIN SAMPLE LIBRARY – A DATABASE OF VIOLIN SOUNDS WITH SEGMENTATION GROUND TRUTH

Henrik von Coler

Audio Communication Group
TU Berlin
Germany
voncoler@tu-berlin.de

ABSTRACT

The presented sample library of violin sounds is designed as a tool for the research, development and testing of sound analysis/synthesis algorithms. The library features single sounds which cover the entire frequency range of the instrument in four dynamic levels, two-note sequences for the study of note transitions and vibrato, as well as solo pieces for performance analysis. All parts come with a hand-labeled segmentation ground truth which mark attack, release and transition/transient segments. Additional relevant information on the samples' properties is provided for single sounds and two-note sequences. Recordings took place in an anechoic chamber with a professional violinist and a recording engineer, using two microphone positions. This document describes the content and the recording setup in detail, alongside basic statistical properties of the data.

1. INTRODUCTION

Sample libraries for the use in music production are manifold. Ever since digital recording and storage technology made it possible, they have been created for most known instruments. Commercial products like the *Vienna Symphonic Library*[1] or *The EastWest Quantum Leap*[2] offer high quality samples with many additional techniques for expressive sample based synthesis. For several reasons, these libraries are not best suited for the use in research on sound analysis and synthesis. Many relevant details are subject to business secrets and thus not documented. Copyright issues may prevent a free use as desired in a scientific application. These libraries also lack annotation and metadata which is essential for research applications, if used for machine learning or sound analysis / synthesis tasks.

The audio research community has released several databases with single instrument sounds in the past, usually closely related to a specific aspect. Libraries like the *RWC* [1] or the *MUMS* [2] aim at genre or instrument classification and timbre analysis [3]. Databases for onset and transient detection which include hand labeled onset segments have been presented by Bello et al. [4] and von Coler et al. [5].

The presented library of violin sounds is designed as a tool for the research, development and testing of sound analysis/synthesis algorithms or machine learning tasks. The contained data is structured to enable the training of sinusoidal modeling systems which distinguish between stationary and transient segments. By design, the library allows the analysis of several performance aspects, such as different articulation styles, glissando [6] and vibrato. It features recordings of a violin in an anechoic chamber and consists of three parts:

1. single sounds
2. two-note sequences
3. solo (scales and compositions/excerpts)

For single sounds and two-note sequences, hand-labeled segmentation files are delivered with the data set. These files focus on the distinction between steady state and transient or transitional segments. The prepared audio files and the segmentation files are uploaded to a static repository with a DOI [7][3]. A *Creative Commons BY-ND 4.0* license ensures the unaltered distribution of the library.

The purpose of this paper is a more thorough introduction of the library. Section 2 will explain the composition of the content, followed by details on the recording setup and procedure in Section 3. The segmentation data will be introduced in Section 4. Section 5 presents selected statistical properties of the sample library. Final remarks are included in Section 6.

2. CONTENT DESCRIPTION

2.1. Single Sounds

Similar to libraries for sample based instruments, the single sounds capture the dynamic and frequency range of the violin, using sustained sounds. The violinist was instructed to play the sounds as long as possible, using just one bow, without any expression. Steady state segments, respectively the sustain parts, of these notes are thus as played as steady as possible. This task showed to be highly demanding and unusual, even for an experienced concert violinist.

On all of the four strings, the number of semitones listed in Table 1 was captured, each starting with the open string. This leads to a total of 84 positions. All positions are captured in four dynamic levels which were specified as **pp - mp - mf - ff** resulting in a total amount of 336 single sounds. According to Meyer [8], the dynamic interval interval of a violin covers a range from $58 \ldots 99\,\mathrm{dB}$.

[1] www.vsl.co.at/

[2] http://www.soundsonline.com/symphonic-orchestra

[3] https://depositonce.tu-berlin.de//handle/11303/7527

Table 1: Number of positions on each string

String	Positions
G	18
D	18
A	18
E	30

Each item was recorded in several takes, until recording engineer, the author and the violinist agreed on success. Although all sounds were explicitly captured in both up- and down-stroke techniques, these modes have not been considered individually in the data set and thus appear randomly.

2.2. Two-Note Sequences

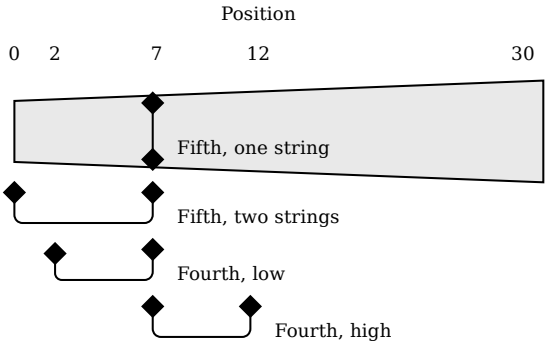


Figure 1: Violin board with positions for two-note sequences

For the study of basic articulation styles, a set of two-note sequences was recorded at different intervals, listed in Table 2. The respective positions on the board are visualized in Figure 1. All combinations were recorded at two dynamic levels **mp** and **ff**. Three different articulation styles (*detached, legato, glissando*) were used and some combinations were captured with additional vibrato. These combinations lead to a grand total of 344 two-note items.

5 semitones on one string were captured in 8 pairs with 24 versions (2 dynamic levels, 2 directions, with and without vibrato, 3 articulation styles): $2 \cdot 2 \cdot 3 = 24$.

Repeated tones were captured in 4 pairs with 6 versions (2 dynamic levels, legato and detached, the latter with and without vibrato): $2^2 + 2 = 6$

7 semitones on one string were captured in pairs with 20 versions (2 dynamic levels, two directions, detached only without vibrato, legato and glissando with and without vibrato): $2 \cdot 2 + 2^4 = 20$

7 semitones on two strings were captured in 3 pairs with 16 versions (2 dynamic levels, two directions, with and without vibrato and two articulation styles [legato, detached]):$2^4 = 16$

Table 2: All two-note pairs

5 semitones, one string

Two-note	Note 1			Note 2		
item no.	ISO	Pos.	String	ISO	Pos.	String
01-24	D4	7	G	A3	2	1
25-48	A4	7	D	E4	2	2
49-72	E5	7	A	B4	2	3
73-96	B5	7	E	F#5	2	4
97-120	D4	7	G	G4	12	1
121-144	A4	7	D	D5	12	2
145-168	E5	7	A	A5	12	3
169-192	B	7	E	E6	13	4

Repeated tones

Two-note	Note 1			Note 2		
item no.	ISO	Pos.	String	ISO	Pos.	String
193-198	D4	7	G	D4	7	G
199-204	A4	7	D	A4	7	D
205-210	E5	7	A	E5	7	A
211-216	B5	7	E	B5	7	E

7 semitones, one string

Two-note	Note 1			Note 2		
item no.	ISO	Pos.	String	ISO	Pos.	String
217-236	D4	7	G	G3	0	G
237-256	A4	7	D	D4	0	D
257-276	E5	7	A	A4	0	A
277-296	B5	7	E	E5	0	E

7 semitones, two strings

Two-note	Note 1			Note 2		
item no.	ISO	Pos.	String	ISO	Pos.	String
297-312	D4	7	G	A4	7	D
313-328	A4	7	D	E5	7	A
329-344	E5	7	A	B5	7	E

2.3. Solo: Scales and Compositions

Two scales – an ascending major scale and a descending minor scale – were each played in three interpretation styles, as listed in Table 3. The first style was plain, without any expressive gestures, followed by two expressive interpretations. Six solo pieces and excerpts, listed in Table 4 which mostly contain cantabile legato passages were recorded. All compositions were proposed by the violinist, ensuring familiarity with the material.

Table 3: Scales in the solo part

Item	Type	Interpretation
01	major, ascending	plain
02	major, ascending	expressive 1
03	major, ascending	expressive 2
04	minor, descending	plain
05	minor, descending	expressive 1
06	minor, descending	expressive 2

Table 4: Solo recordings

Item	Composition	Composer
07	Sonata in A major for Violin and Piano	César Franck
08	Violin Concerto in E minor, Op. 64, 2nd movement	Felix Mendelssohn
09	Méditation (Thaïs)	Jules Massenet
10	Chaconne in g minor	Tomaso Antonio Vitali
11	Violin Concerto in E minor, Op. 64, 3rd movement	Felix Mendelssohn
12	Violin Sonata no.5, Op.24, 12s movement	Ludwig van Beethoven

3. RECORDING SETUP

The recordings took place in the anechoic chamber at SIM[4], Berlin. Above a cutoff frequency of 100 Hz the room shows an attenuation coefficient of $\mu > 0.99$, hence the recordings are free of reverberation in the relevant frequency range. The recordings were conducted within two days, taking one day for the single sounds and the second day for two-note sequences and solo pieces. All material was captured with a sample-rate of of 96 kHz and a depth of 24 Bit.

Microphones

The following microphones were used:

- 1x DPA 4099 cardiod clip microphone
- 1x Brüel & Kjær 4006 omnidirectional small diaphragm microphone with free-field equalization, henceforth BuK

The DPA microphone was mounted as shown in Figure 2, above the lower end of the f-hole in 2 cm distance. Due to its fixed position, movements of the musician do not influence the recording. The B&K microphone was mounted in 1.5 m distance above the instrument, at an elevation angle of approximately 45°, as shown in Figure 3.

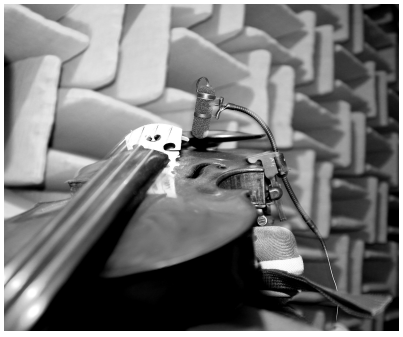


Figure 2: Position of the DPA microphone

[4] http://www.sim.spk-berlin.de/refelxionsarmer_raum_544.html

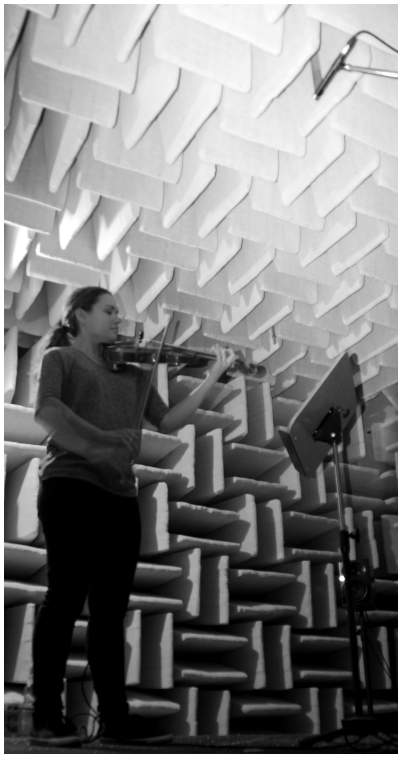
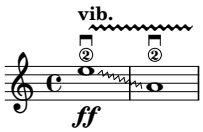


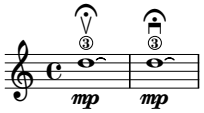
Figure 3: Position of the B&K microphone

Instructions

For each of the single-sound, two-note and scale items, a minimal score snippet was generated using *LilyPond* [9]. Examples for items' instructions are shown in Fig. 4. The resulting 63 page score was then used to guide the recordings. Although the isolated tasks may seem simple and unambiguous, this procedure ensured smooth recording sessions.



(a) Two-note example with vibrato and glissando

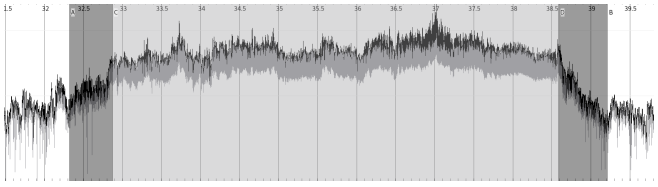


(b) Single-sound example with upbow and downbow

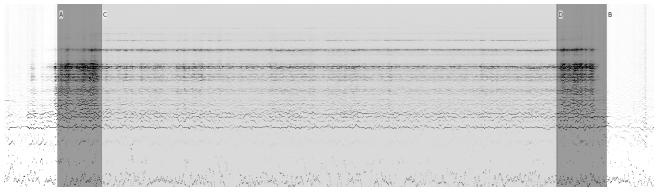
Figure 4: Instruction scores for two-note a and single-sound b

4. SEGMENTATION

The segmentation of a monophonic musical performance into notes, and even more into a note's subsegments is not trivial [10, 11]. During the labeling process, the best of the takes for each item was selected from the raw recordings and the manual segmentation scheme proposed by by von Coler et al. [5] was applied using *Sonic Visualiser* [12].



(a) Energy trajectory



(b) Peak frequency spectrogram

Figure 5: Sonic Visualiser setup for annotation of single sound 333

4.1. Single Sounds

Each single sound is divided into three segments, which are defined by four location markers in the segmentation files[5], as shown in Table 5. The first time instant (A) marks the beginning of the attack segment, the second instant (C) marks the end of the attack segment, respectively the beginning of the sustain part. The end of the sustain, which is also the beginning of the release segment, is labeled with the (D). The label (B) marks the end of the release portion and the complete sound. The left column holds the related time instants in seconds.

Table 5: Example for a single-sound segmentation file (SampLib_DPA_01.txt)

```
0.000000   A
0.940646   C
7.373000   D
8.730500   B
```

The definition of the attack segment is ambiguous in literature [13] and shall thus be specified for this context: Attack here refers to the actual *attack-transient*, the very first part of a sound with a significant inharmonic content and rapid fluctuations. In other contexts, the attack may be regarded the segment of rise in energy to the local maximum. Often, there is still a significant increase in energy after the attack-transient is finished. As the attack-transient is characterized by unsteady, evolving partials and low relative partial amplitudes, the manual segmentation process is performed using a temporal and a spectral representation. Figure 5 shows a typical Sonic Visualiser setup for the annotation of a single sound. The noisiness of the signal during attack and release can be seen in the spectral representation. How attack transient and rising slope may differ, is illustrated in Fig. 6. The gray area represents the labeled attack segment, which is finished before the end of the rising slope is reached.

Less ambiguous, the release part is labeled as the segment from the end of the excitation until the complete disappearance

[5]The segmentation files are part of the repository [7]

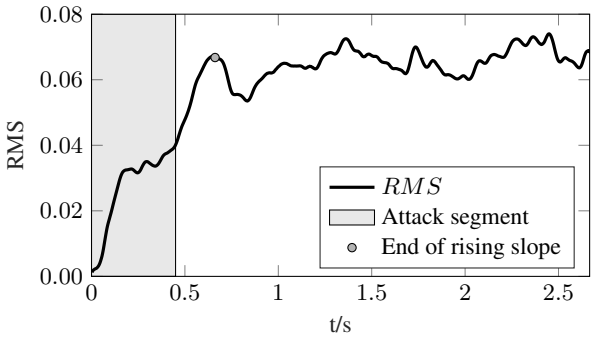


Figure 6: RMS trajectory of a note beginning with attack segment (gray) and end of the rising slope (single sound no. 19)

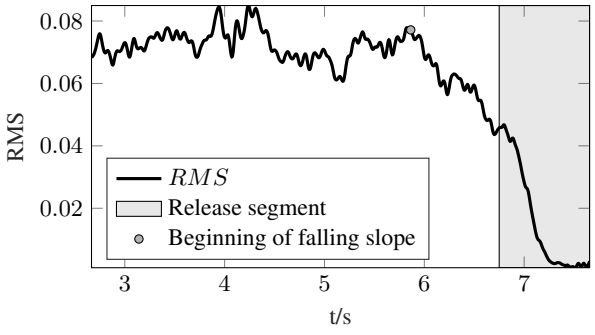


Figure 7: RMS trajectory of a note end with release segment (gray) and beginning of the falling slope (SampLib_19)

of the tone. As shown in Fig. 7, there is often a significant decrease in signal energy before the actual release starts. For items with low dynamics, the release is also covering the very last part of the excitation.

The *ease of annotation* varies between dynamic levels, as well as between the fundamental frequency of the items. Notes played at fortissimo show clear attack and decay segments with a steady sustain part, whereas pianissimo tones have less prominent boundary segments and a parabolic amplitude envelope. The higher SNR in fortissimo notes allows a better annotation of the transients. Tones with a high fundamental frequency have less prominent partials, whereas the bow noise is emphasized. They are thus more difficult to label, since attack transient are less clear in the spectrogram. The segmentation of high pitched notes at low velocities is hence most complicated.

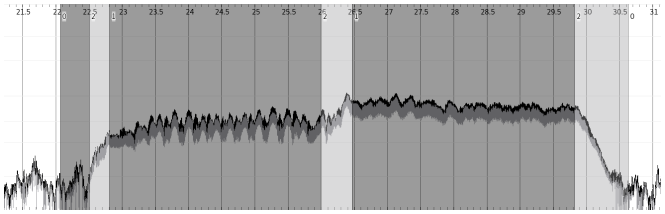
4.2. Two-Note Sequences

The two-note sequences contain the the segments note, rest and transition with the labels listed in Table 6. Stationary sustain parts are labeled as notes, whereas the transition class includes attack and release segments, as well as note transitions, such as glissando.

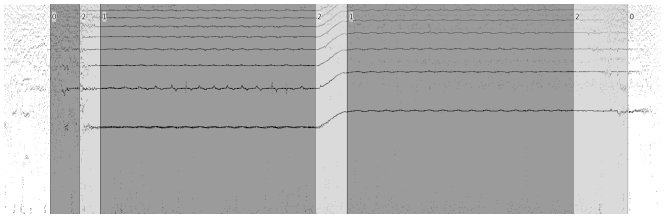
All two-note sequences follow the same sequence of segments (0-2-1-2-1-2). Figure 8 shows a labeling project in Sonic Visualiser for a two-note item with glissando. The transition segment is placed according to the slope of the glissando transition.

Table 6: Segments in the two-note labeling scheme

Label	Segment
0	rest
2	transition
1	note



(a) Energy trajectory



(b) Peak frequency spectrogram

Figure 8: Sonic Visualiser setup for annotation of two-note item 22

4.3. Solo

Solo items have been annotated using the guidelines proposed by von Coler et al. [5]. Due to the choice of the compositions, only few parts violated the restriction to pure monophony. Solo item 10, for example, starts with a chord, which is labeled as a single transitional segment.

5. STATISTICS

This section reports selected descriptive statistical properties of the sample library which are potentially useful when considering the use of the data.

5.1. Single Sounds

Fig. 9 shows the RMS for all single sounds, in box plots for each dynamic level. The median for the dynamic levels is logarithmically spaced.

Table 7: Segment length statistics for the single-sounds

	$\bar{l}/s$	μ/s
Attack	0.247	0.206
Sustain	5.296	1.118
Release	0.705	0.802

Statistics for the segment lengths of the single sounds are presented in Table 7 and Figure 10, respectively. With a mean of $5.296\,\mathrm{s}$, the sustain segments are the longest, followed by release segments with a mean of $0.705\,\mathrm{s}$. Attack segments have a mean length of $0.247\,\mathrm{s}$. Extreme outliers in the mean attack length are caused by high pitched notes with low dynamics.

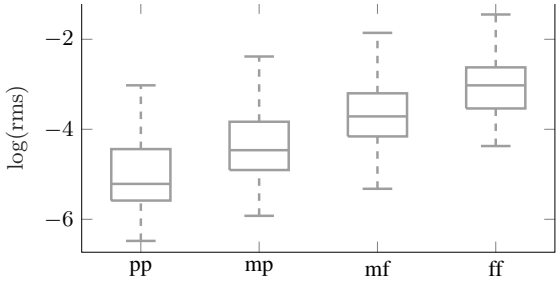


Figure 9: Boxplot of RMS for the sustain from the BuK microphone

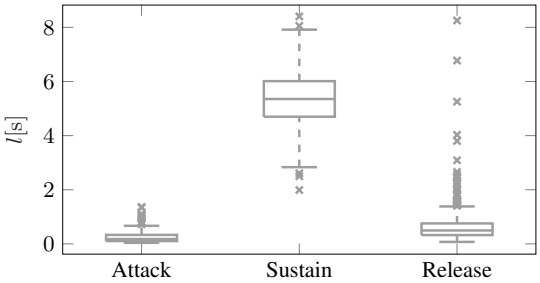


Figure 10: Box plots of segment lengths for all single sounds

5.2. Two-Note

The two-note sequences allow a comparison of different articulation styles. Figure 11 shows the lengths for detached, legato and glissando transitions in a box plot. With a median duration of $0.72\,\mathrm{s}$, glissando transitions tend to be longer than legato ($0.38\,\mathrm{s}$) and detached ($0.37\,\mathrm{s}$) transitions.

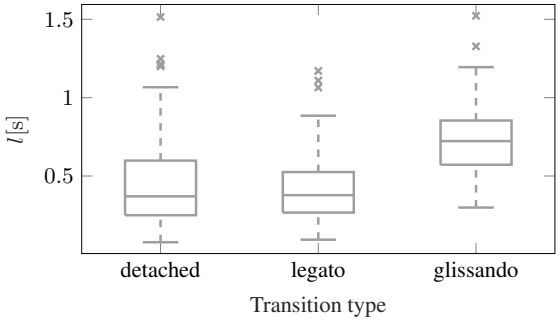


Figure 11: Box plot of transition lengths for all two-note sequences

5.3. Solo

Table 8: Note statistics for items in the solo category

Solo item	Number of notes	$\bar{l}/s$	μ/s
1	8	0.698	0.745
2	8	0.721	0.768
3	8	0.728	0.776
4	8	0.707	0.753
5	8	0.724	0.771
6	8	0.774	0.848
7	104	0.695	0.661
8	75	1.074	0.899
9	89	0.911	0.923
10	63	0.735	0.690
11	76	0.689	0.707
12	56	0.615	0.740

For the solo category, the basic statistics on the note occurrences and lengths are listed in Table 8. All scales (items 1 - 6) contain 8 notes, compositions (items 7-12) have a mean of 77 notes per item. With a mean note length of $0.614\,906\,\mathrm{s}$, item 12 has the shortest, and with $1.074\,361\,\mathrm{s}$, item 8 has the longest notes.

6. CONCLUSION

The presented sample library is already in application within sinusoidal modeling projects and for the analysis of expressive musical content. Overall recording quality proves to be well suited for most tasks in sound analysis. Since the segmentation ground truth follows strict rules and has undergone repeated reviews, it may be considered consistent.

7. ACKNOWLEDGMENTS

The author would like to thank the violin player, Michiko Feuerlein, and the sound engineer, Philipp Pawlowski, for their work during the recordings, as well as the SIM Berlin for the support. Further acknowledgment is addressed to Moritz Götz, Jonas Margraf, Paul Schuladen and Benjamin Wiemann for the contributions to the annotation.

8. REFERENCES

[1] Masataka Goto et al. “Development of the RWC music database”. In: *Proceedings of the 18th International Congress on Acoustics (ICA 2004)*. Vol. 1. 2004, pp. 553–556.

[2] Tuomas Eerola and Rafael Ferrer. “Instrument library (MUMS) revised”. In: *Music Perception: An Interdisciplinary Journal* 25.3 (2008), pp. 253–255.

[3] Gregory J Sandell. “A Library of Orchestral Instrument Spectra”. In: *Proceedings of the International Computer Music Conference*. 1991, pp. 98–98.

[4] J.P. Bello et al. “A Tutorial on Onset Detection in Music Signals”. In: *IEEE Transactions on Speech and Audio Processing* 13.5 (2005), pp. 1035–1047.

[5] Henrik von Coler and Alexander Lerch. “CMMSD: A Data Set for Note-Level Segmentation of Monophonic Music”. In: *Proceedings of the AES 53rd International Conference on Semantic Audio*. London, England, 2014.

[6] Henrik von Coler, Moritz Götz, and Steffen Lepa. “Parametric Synthesis of Glissando Note Transitions - A user Study in a Real-Time Application”. In: *Proc. of the 21st Int. Conference on Digital Audio Effects (DAFx-18)*. Aveiro, Portugal, 2018.

[7] Henrik von Coler, Jonas Margraf, and Paul Schuladen. *TU-Note Violin Sample Library*. TU-Berlin, 2018. DOI: 10.14279/depositonce-6747.

[8] Jürgen Meyer. “Musikalische Akustik”. In: *Handbuch der Audiotechnik*. Ed. by Stefan Weinzierl. VDI-Buch. Springer Berlin Heidelberg, 2008, pp. 123–180.

[9] Han-Wen Nienhuys and Jan Nieuwenhuizen. “LilyPond, a system for automated music engraving”. In: *Proceedings of the XIV Colloquium on Musical Informatics (XIV CIM 2003)*. Vol. 1. 2003, pp. 167–171.

[10] E. Gómez et al. “Melodic Characterization of Monophonic Recordings for Expressive Tempo Transformations”. In: *Proceedings of the Stockholm Music and Acoustics Conference*. 2003.

[11] Norman H. Adams, Mark A. Bartsch, and Gregory H. Wakefield. “Note Segmentation and Quantization for Music Information Retrieval”. In: *IEEE Transactions on Speech and Audio Processing* 14.1 (2006), pp. 131–141.

[12] Chris Cannam, Christian Landone, and Mark Sandler. “Sonic visualiser: An open source application for viewing, analysing, and annotating music audio files”. In: *Proceedings of the 18th ACM international conference on Multimedia*. ACM. 2010, pp. 1467–1468.

[13] Xavier Rodet and Florent Jaillet. “Detection and Modeling of Fast Attack Transients”. In: *Proceedings of the International Computer Music Conference*. 2001, pp. 30–33.

PARAMETRIC MULTI-CHANNEL SEPARATION AND RE-PANNING OF HARMONIC SOURCES

M. W. Hansen †, *J. M. Hjerrild* †, *M. G. Christensen* †

†Audio Analysis Lab, CREATE
Aalborg University
Aalborg, Denmark
{mwh,jmhh,mgc}@create.aau.dk

J. Kjeldskov *

*Department of Computer Science
Aalborg University
Aalborg, Denmark
jesper@cs.aau.dk

ABSTRACT

In this paper, a method for separating stereophonic mixtures into their harmonic constituents is proposed. The method is based on a harmonic signal model. An observed mixture is decomposed by first estimating the panning parameters of the sources, and then estimating the fundamental frequencies and the amplitudes of the harmonic components. The number of sources and their panning parameters are estimated using an approach based on clustering of narrowband interaural level and time differences. The panning parameter distribution is modelled as a Gaussian mixture and the generalized variance is used for selecting the number of sources. The fundamental frequencies of the sources are estimated using an iterative approach. To enforce spectral smoothness when estimating the fundamental frequencies, a codebook of magnitude amplitudes is used to limit the amount of energy assigned to each harmonic. The source models are used to form Wiener filters which are used to reconstruct the sources. The proposed method can be used for source re-panning (demonstration given), remixing, and multi-channel upmixing, e.g. for hi-fi systems with multiple loudspeakers.

1. INTRODUCTION

Music signals often contain a mixture of multiple instrument recordings. To process such a mixture, e.g., with the goal of modifying the sources independently, it may be beneficial to extract the individual sources in the mixture. This task is known as source separation, and it has applications in areas such as music information retrieval [1], sound scene modification [2], and enhancement [3].

The problem of separating sources in a music mixture is in general very difficult, because of the presence of overlap in both time and frequency. In such cases, the source separation problem is in many cases ill-posed, and the single-channel source separation problem is very difficult to solve, and would rely heavily on prior information about the sources. When multiple channels of data are available, it is possible to exploit information about the mixing process. A method for separating two sources from a single-channel mixture was proposed in [4], based on a sparse non-negative decomposition algorithm, whereas in [5] a method based on single-channel non-negative matrix factorization (NMF) was proposed for polyphony music transcription. In [6], a method based on non-negative matrix factorization (NMF) for stereophonic source separation is presented, while in [7] a framework for incorporating prior knowledge in source separation is presented. Separation of moving sources is considered in [8] using a method based on multi-channel NMF. Time-variation is allowed through the use of spatial covariance matrices (SCMs) which are generated based on estimated directions of arrival (DOAs). Separation of sources from multi-channel reverberant mixtures, although in a semi-blind fashion, with known mixing filters, was considered in [9]. Re-panning of stereophonic sources was proposed in [10] for a known number of sources without delay panning.

Parametric signal models, where the sinusoidal components of a signal are modelled as a sum of sinusoids, can also be used for source separation. A method for source separation and auditory scene analysis based on a multi-pitch and periodicity analysis method is presented in [11], while sinusoidal modelling was used for separating harmonic sources using a classification method to group extracted sinusoids in [12]. Spectral overlap often occur in music signals, and this should be taken into account when estimating the parameters of the sources. A source separation method based on pitch, amplitude modulation and spatial cues for separation of harmonic instruments from stereo music recordings is proposed in [13]. In [14], a method for reconstruction of completely overlapped notes is presented, where the spectral envelope of each source is learnt in segments without overlap, and then used to extract the sources. A separation approach based on optimal filtering is presented in [15], where a linearly constrainted minimum variance (LCMV) filter is constructed based on a priori knowledge in the form of score information. Furthermore, the balance between overlapping harmonics is adjusted using a priori knowledge about the magnitude of each harmonic.

In this paper, we present a method for extracting harmonic sources from stereophonic mixtures of music recordings, such as those made artificially in a studio. First, the panning parameters and activations of the sources are estimated using a method based on clustering of narrowband interaural level and time differences (ILDs and ITDs) (see [16] for further details). Usually, in source separation algorithms, the number of sources is assumed known a priori (see, e.g., [6]), however, here the number of sources does not need to be known. Equipped with the estimated panning parameters, the fundamental frequencies of the harmonic sources are estimated, along with the number of harmonics, and the harmonic amplitudes, using an iterative approach. To enforce spectral smoothness, a codebook of magnitude amplitudes trained on recordings of harmonic sources is used (see [17] for further details). The source models are used to form a Wiener filter for extraction of each source from the mixture. It should be noted that the proposed method is also capable of separating sources from monophonic, i.e., single-channel mixtures. After the sources have been extracted, they are combined with new panning parameters, and the residual, i.e., the parts of the mixture not captured by the harmonic model of the sources.

* Supported by the Technical Faculty of IT and Design, Aalborg University.

2. SIGNAL MODEL

An observed multichannel mixture is modelled as a sum of M harmonic sources s_m, $m = 1, \ldots, M$, plus a noise term e. The signal in the kth channel at time n is

$$x_k(n) = \sum_{m=1}^{M} g_{k,m} s_m(n - \tau_{k,m}) + e_k(n), \tag{1}$$

where $g_{k,m}$ and $\tau_{k,m}$ are the amplitude and delay panning parameters, respectively. An example of an amplitude panning law, which could used to calculate the gains applied to each channel of a stereophonic mixture is [18]

$$g_{k,m} = \begin{cases} \cos \phi_m, & \text{for } k = 1 \\ \sin \phi_m, & \text{for } k = 2 \end{cases}, \tag{2}$$

where $\phi_m \in [0, \text{pi}/2]$. The mth source s_m is modelled as a sum of L_m harmonic components, i.e,

$$s_m(n) = \sum_{l=1}^{L_m} \alpha_{m,l} e^{j\omega_{0,m} l n}, \tag{3}$$

where $\omega_{0,m}$ is the fundamental frequency of the mth source, L_m is the model order, and $\alpha_{m,l} = A_{m,l} e^{j\phi_{m,l}}$ is the complex amplitude of the lth harmonic, where $A_{m,l}$ is the real amplitude and $\phi_{m,l}$ its phase. A complex signal model is used because it may result in simplified expressions, and a lower computational complexity. The signal model may be used with real signals by applying the Hilbert transform. It should be noted that although we focus on the stereophonic case ($k = 2$), we here present a general multi-channel signal model, which can be used in scenarios where $k > 2$ using a different panning law. Furthermore, according to the source model (3), an instrument recording may contain multiple sources, e.g., when a chord is played on a guitar, where the signal generated by each string is considered to be a source. Furthermore, we define a submixture as a sum of sources that share panning parameters. The kth channel of an observed mixture is processed in segments each containing N consecutive samples, i.e,

$$\mathbf{x}_k = [x_k(0) \;\; x_k(1) \;\; \cdots \;\; x_k(N-1)]^T, \tag{4}$$

which can be used to write the signal model in vector form as

$$\mathbf{x}_k = \sum_{m=1}^{M} \mathbf{Z}_m \mathbf{G}_{k,m} \boldsymbol{\alpha}_m + \mathbf{e}_k, \tag{5}$$

where $\mathbf{Z}_m$ is a Vandermonde matrix, with the harmonic components of source m with fundamental frequency $\omega_{0,m}$ in the columns, i.e.,

$$\mathbf{Z}_m = \begin{bmatrix} 1 & \cdots & 1 \\ e^{j\omega_{0,m}} & \cdots & e^{j\omega_{0,m} L_m} \\ \vdots & \ddots & \vdots \\ e^{j\omega_{0,m}(N-1)} & \cdots & e^{j\omega_{0,m} L_m (N-1)} \end{bmatrix},$$

and $\mathbf{G}_{k,m}$ is a diagonal matrix containing the panning parameters in (2) and $\tau_{k,m}$ for channel k of source m, i.e.,

$$\mathbf{G}_{k,m} = \begin{bmatrix} g_{k,m} e^{-j\omega_{0,m} f_s \tau_{k,m}} \cdots & 0 \\ \vdots & \ddots \\ 0 & \cdots g_{k,m} e^{-jL_m \omega_{0,m} f_s \tau_{k,m}} \end{bmatrix}.$$

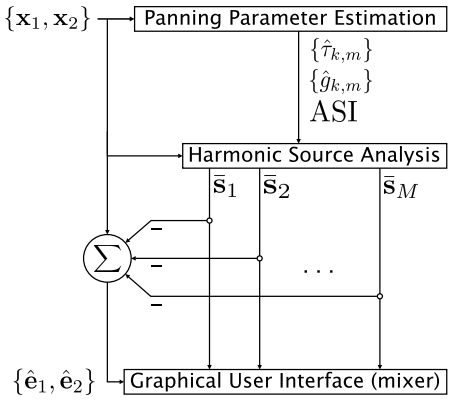


Figure 1: Overview of the proposed method.

When only amplitude panning is applied, $\tau_{k,m} = 0 \;\forall\; \{k, m\}$, and when only delay panning is used, $g_{k,m} = 1 \;\forall\; \{k, m\}$. Also, we assume that the panning parameters are constant throughout a segment of the signal. The vector of complex amplitudes for source m is given by

$$\boldsymbol{\alpha}_m = [\alpha_{m,1} \;\; \cdots \;\; \alpha_{m,L_m}]^T, \tag{6}$$

and the noise vector is

$$\mathbf{e}_k = [e_k(0) \;\; e_k(1) \;\; \cdots \;\; e_k(N-1)]^T. \tag{7}$$

Since we model the sinusoidal source components, the noise term contains the non-periodicities that are not captured by the harmonic model. In the next section, we present the proposed method for estimating the panning parameters $g_{k,m}$ and $\tau_{k,m}$, along with the number of unique panning parameters, which corresponds to the number of submixtures.

3. PROPOSED METHOD

The proposed method consist of several sub-systems, as shown in Figure 1. In the initial step of the proposed method, the panning parameters of the sources in the mixture are estimated, along with an active source indication (ASI) of when the corresponding sources are active. This knowledge is exploited in the harmonic source analysis, where the parameters of each source $\mathbf{s}_m$ in the mixture are estimated, i.e., its fundamental frequency $\omega_{0,m}$, the number of harmonics L_m, and the amplitude vector $\boldsymbol{\alpha}_m$. The harmonic models of the sources are used to form Wiener filters, which are used to extract the sources from the mixture. The resulting frames are combined using overlap-add, and a graphical user interface (GUI) is used to re-pan the sources.

3.1. Panning Parameter Estimation and Activity Detection

As shown in Figure 1, the panning parameters of the sources in the observed multi-channel mixture are required as input for the proposed harmonic signal analysis sub-system. The source panning parameters are estimated along with the number of unique panning parameters using the method presented in [16]. The method is a blind source panning estimation algorithm based on clustering of narrowband interaural level and time differences (ILDs, ITDs). For an unknown number of sources, the parameter distribution

across all segments of the mixture is modelled as a Gaussian mixture. The generalized variance and degree of membership of the Gaussian components across segments are used as a basis for the selection of clusters amongst candidates. In the time-frequency domain we define a vector $\mathbf{y}$ for each frame containing the relative amplitude panning parameters and relative channel delays, i.e.,

$$\mathbf{y}=\Big[\hat{g}(\omega),\hat{\tau}(\omega)\Big]^T=\left[\arctan\left(\left|\frac{X_1(\omega)}{X_2(\omega)}\right|\right),\frac{1}{\omega}\angle\frac{X_2(\omega)}{X_1(\omega)}\right]^T, \quad (8)$$

where $\hat{\tau}(\omega)=\hat{\tau}_1(\omega)-\hat{\tau}_2(\omega)$, $X_k(\omega)$ is the discrete Fourier transform of the kth channel of a segment of the mixture, and $\angle$ denotes phase. Eq. (8) is constrained on the assumption of W-disjoint orthogonality [19] and on the so-called narrowband assumption that requires the maximum frequency ω_{max} and maximum delay τ_{max} to be strictly within the range $|\omega_{\text{max}}\tau_{\text{max}}|<\pi$. From (8) we collect P observations $\mathcal{Y}=\{\mathbf{y}^{(1)},\ldots,\mathbf{y}^{(P)}\}$ with identical probability distributions, each being mutually independent. The log-likelihood function of the P observations is

$$\ln p(\mathcal{Y}|\boldsymbol{\theta})=\sum_{p=1}^{P}\ln\sum_{m=1}^{M}\gamma_m p(\mathbf{y}^{(p)}|\boldsymbol{\theta}_m), \quad (9)$$

where $\boldsymbol{\theta}_m$ is the unknown and deterministic parameter vector of the mth source. For the purpose of estimating panning parameters, the distribution of $\mathbf{y}$ from Eq. (8) is modelled as a Gaussian mixture of M sources, with diagonal covariance matrices, i.e.,

$$p(\mathbf{y}|\boldsymbol{\theta})=\sum_{m=1}^{M}\gamma_m\frac{\exp\big\{-\frac{1}{2}(\mathbf{y}-\mu_m)^T\mathbf{C}_m^{-1}(\mathbf{y}-\mu_m)\big\}}{\sqrt{(2\pi)^d\det(\mathbf{C}_m)}}, \quad (10)$$

where $\boldsymbol{\theta}\triangleq\{\gamma_1,\ldots,\gamma_M,\hat{\boldsymbol{\mu}}_1,\ldots,\hat{\boldsymbol{\mu}}_M,\mathbf{C}_1,\ldots,\mathbf{C}_M\}$ is the complete set of parameters, where the set $\{\gamma_m,\hat{\boldsymbol{\mu}}_m,\mathbf{C}_m\}$ denotes the mixing probability, mean and covariance of the mth Gaussian. In general, $\gamma_m\geq 0,\quad\sum_{m=1}^{M}\gamma_m=1,$ for $\quad m=1,\ldots,M$. The maximum likelihood (ML) estimate of the parameter vector is

$$\hat{\boldsymbol{\theta}}_{\text{ML}}=\arg\max_{\boldsymbol{\theta}}\ \ln p(\mathcal{Y}|\boldsymbol{\theta}) \quad (11)$$

for a value of M such that the GMM is overfitted, see [16]. The ML GMM parameter estimates in $\hat{\boldsymbol{\theta}}_{\text{ML}}$ are obtained using an EM-algorithm. Several GMM EM-methods have been proposed for estimating the number of sources, using a penalty term such as the Bayesian information criterion (BIC) or the minimum description length (MDL) [20]. However, the problem is complicated for audio recordings for two reasons: no unique definition of a "true cluster" necessarily exists, and the assumption of normality does not exactly hold, see, e.g., [21]. Therefore, each of the underlying GMM components does not necessarily correspond to a source cluster.

In the present method clusters are selected among Gaussian component candidates by fitting a GMM to the observed data with a large number of components. From the overfitted GMM clusters are defined as having lowest generalized variance δ and as being well separated from other candidates as described in the following. The cluster indices are columns of $\zeta_{\omega s}$ which have low generalized variance δ, and are well separated from all GMM components, and $\hat{\boldsymbol{\theta}}_s$ is arranged such that $\delta_1<\delta_2<\cdots<\delta_S$, where $\delta=\det(\hat{\mathbf{C}})$ and $s=\{1,2,\cdots,S\}$. The a posteriori probability $\zeta_{\omega s}$ that $\mathbf{y}_\omega$ belongs to mixture component s is

$$\zeta_{\omega s}=\frac{\hat{\gamma}_s\mathcal{N}(\mathbf{y}_\omega|\hat{\boldsymbol{\mu}}_s,\hat{\mathbf{C}}_s)}{\sum_{j=1}^{S}\hat{\gamma}_j\mathcal{N}(\mathbf{y}_\omega|\hat{\boldsymbol{\mu}}_j,\hat{\mathbf{C}}_j)}. \quad (12)$$

The sth column does not represent a cluster if $0<\zeta_{\omega\epsilon}<1\wedge 0<\zeta_{\omega s}<1$, where $\epsilon=\{1,2,\cdots,s-1\}\,\forall\,\omega$. After ranking, the $\hat{M}$ clusters are in the first columns of $\zeta_{\omega s}$, as observed in [16]. This leads to an estimate of the M unique panning parameters and the statistics $\hat{\boldsymbol{\theta}}_{\hat{M}}$ from which the vector $\hat{\boldsymbol{\mu}}_m$ is the panning parameters of the mth source, across all segments.

We compute an active source indication (ASI) for each frame of the observed mixture. Specifically, the input signal is processed in frames of length 60 ms, with a hop size of 15 ms. In each frame all possible combinations of the obtained $\hat{\boldsymbol{\theta}}_{\hat{M}}$ statistics are fitted to the observed data $\mathbf{y}$ resulting in a new GMM likelihood. The maximum likelihood combination is chosen for each frame. The obtained ASI is a binary indication of activity of each panning parameter in each frame of the mixture, and is used as input to the harmonic analysis sub system.

3.2. Harmonic Signal Analysis

In this section the method used to analyse the harmonic sources in a stereophonic mixture is presented. The goal is to estimate the fundamental frequencies of the harmonic components in the mixture, along with the number of harmonics for each source, and the complex amplitudes, provided with information about the source panning parameters, and source activity indication, as described in the previous section. The proposed method is based on the maximum likelihood principle, and the log-likelihood of the kth channel of an observed signal is parametrized by $\boldsymbol{\psi}_k=[\boldsymbol{\psi}_{k,1}\ \cdots\ \boldsymbol{\psi}_{k,M}]^T$, where $\boldsymbol{\psi}_{k,m}=[\omega_{0,m}\ g_{k,m}\ \tau_{k,m}\ \boldsymbol{\alpha}_m^T]^T$, for $m=1,\ldots,M$. We assume that the deterministic part of the signal is stationary, and that the noise is independent and identically distributed over n and k. Furthermore, we assume that the noise is white Gaussian with different variance in each channel, σ_k^2. Defining the error as $\mathbf{e}_k=\mathbf{x}_k-\sum_{m=1}^{M}\mathbf{Z}_m\mathbf{G}_{k,m}\boldsymbol{\alpha}_m$, the likelihood of the kth channel of the observed signal is defined as

$$p\left(\mathbf{x}_k;\boldsymbol{\psi}_k\right)=\frac{1}{\left(\pi\sigma_k^2\right)^N}e^{-\frac{1}{\sigma_k^2}\|\mathbf{e}_k\|_2^2}, \quad (13)$$

which across channels becomes

$$p\left(\{\mathbf{x}_k\};\{\boldsymbol{\psi}_k\}\right)=\prod_{k=1}^{K}\frac{1}{\left(\pi\sigma_k^2\right)^N}e^{-\frac{1}{\sigma_k^2}\|\mathbf{e}_k\|_2^2}. \quad (14)$$

The log-likelihood of a single channel of the observed signal is

$$\ln p\left(\mathbf{x}_k;\boldsymbol{\psi}_k\right)=-N\ln\pi-N\ln\sigma_k^2-\frac{\|\mathbf{e}_k\|_2^2}{\sigma_k^2} \quad (15)$$

while the log-likelihood for all channels of the observed signal is

$$\ln p\left(\{\mathbf{x}_k\};\{\boldsymbol{\psi}_k\}\right)=-KN\ln\pi-N\sum_{k=1}^{K}\ln\sigma_k^2-\sum_{k=1}^{K}\frac{\|\mathbf{e}_k\|_2^2}{\sigma_k^2}. \quad (16)$$

The fundamental frequencies, complex amplitudes, and noise variance for each channel are estimated by maximizing (16). Since the problem of estimating the parameters of all the sources at once is impractical in terms of computational complexity, the parameters are estimated iteratively using an EM algorithm. For each iteration of the method, the log-likelihood of the observed segment of the mixture is increased. The observed signal is modelled as a sum of M sources, where the kth channel of source m is modelled as

$$\mathbf{x}_{k,m}=\mathbf{Z}_m\mathbf{G}_{k,m}\boldsymbol{\alpha}_m+\mathbf{e}_{k,m}, \quad (17)$$

where $\mathbf{G}_{k,m}$ is now formed using the estimates $\{\hat{g}_{k,m}, \hat{\tau}_{k,m}\}$ for each source, and where the noise term $\mathbf{e}_k$ is decomposed into M sources, i.e.,

$$\mathbf{e}_{k,m} = \beta_m \mathbf{e}_k, \tag{18}$$

where $\beta_m \geq 0$ is chosen such that $\sum_{m=1}^{M} \beta_m = 1$. Here, β_m is chosen such that the entire error term is assigned to a single component in each iteration, i.e., $\beta_{p=m} = 1$ and $\beta_{p\neq m} = 0$, and $p = \mod(i-1, M) + 1$, with i being the EM iteration index [22, 23]. Assuming white Gaussian noise (see [24, 25]), in the E-step, the kth channel of the mth source in iteration i is modelled according to (17) based on parameters estimated in the previous iteration, i.e.,

$$\hat{\mathbf{x}}_{k,m}^{(i)} = \mathbf{Z}_m^{(i)} \mathbf{G}_{k,m} \hat{\boldsymbol{\alpha}}_m^{(i)} + \beta_m \left(\mathbf{x}_k - \sum_{m=1}^{M} \mathbf{Z}_m^{(i)} \mathbf{G}_{k,m} \tilde{\boldsymbol{\alpha}}_m^{(i)} \right), \tag{19}$$

where $\widetilde{\boldsymbol{\alpha}}_m = [\widetilde{A}_{1,m} e^{j\angle \widehat{\alpha}_{1,m}} \ \cdots \ \widetilde{A}_{L_m,m} e^{j\angle \widehat{\alpha}_{L_m,m}}]^T$ is formed using a scaled codebook entry $\tilde{\mathbf{A}}_m$ from a codebook $\mathcal{C}$ of magnitude amplitude vectors trained on individual notes played on a variety of instruments, and combined with the phases resulting from the least squares estimate of the complex amplitude vector, given $\hat{\omega}_m^{(i+1)}$ as [26] (see [17] for more information)

$$\hat{\boldsymbol{\alpha}}_m^{(i+1)} = \left[\sum_{k=1}^{K} \frac{\mathbf{G}_{k,m}^H \mathbf{Z}_m^H \mathbf{Z}_m \mathbf{G}_{k,m}}{\hat{\sigma}_k^{2(i+1)}} \right]^{-1} \sum_{k=1}^{K} \frac{\mathbf{G}_{k,m}^H \mathbf{Z}_m^H \hat{\mathbf{x}}_{k,m}^{(i)}}{\hat{\sigma}_k^{2(i+1)}}. \tag{20}$$

In the M-step, the fundamental frequency of the mth source is estimated using the NLS method, based on the estimate of each source from the previous iteration, i.e.,

$$\hat{\omega}_m^{(i+1)} = \arg\min_{\omega_m} \sum_{k=1}^{K} \ln \left\| \hat{\mathbf{x}}_{k,m}^{(i)} - \mathbf{Z}_m \mathbf{G}_{k,m} \tilde{\boldsymbol{\alpha}}_m^{(i+1)} \right\|_2^2, \tag{21}$$

The estimate of the variance σ_k^2 in iteration $i+1$ is

$$\hat{\sigma}_k^{2(i+1)} = \frac{1}{N} \left\| \hat{\mathbf{x}}_{k,m}^{(i)} - \mathbf{Z}_m \mathbf{G}_{k,m} \tilde{\boldsymbol{\alpha}}_m^{(i+1)} \right\|_2^2. \tag{22}$$

The complex amplitude vector and the noise variance are estimated in an iterative fashion, because they depend on each other. It is not necessary to iterate between (20) and (22) if the noise variance for both channels are equal. The E- and M-steps are repeated until a convergence criterion is met. The method is guaranteed to converge to a local minimum in each step, and increases the likelihood of the observed data at each step. Initialization of the EM algorithm is not simple, and can result in poor performance, if it is not done carefully. We here use the harmonic matching pursuit (HMP) [27, 24], which is based on a residual for channel k in iteration m at time n, defined as

$$r_k^{(m)}(n) = r_k^{(m-1)}(n) - \sum_{l=1}^{L_m} g_{k,m} \alpha_{m,l} e^{j\omega_{0,m} l (n - \tau_{k,m})}. \tag{23}$$

The model parameters are estimated iteratively for each modelled harmonic source m. The method is initialized using the observed signal, i.e., $r_k^{(0)}(n) = x_k(n)$. As previously mentioned, the fundamental frequencies of the M sources are estimated jointly with the model order. The maximum a posteriori (MAP) model selection criterion [28, 24] is used as a model selection rule, i.e.,

$$\hat{\mathcal{M}}_m = \arg\min_{\mathcal{M}_m} \sum_{k=1}^{K} -\ln p\left(\mathbf{x}_k; \hat{\psi}_m, \mathcal{M}_m\right) + \frac{1}{2} \ln |\hat{\mathbf{H}}_m|,$$

where $\hat{\mathcal{M}}_m$ is the model of the mth source, and $|\cdot|$ denotes the determinant of a matrix. The determinant of the Hessian, $\hat{\mathbf{H}}_m$, can be approximated using the Fisher information matrix, and a normalization matrix is introduced (see [28]) such that

$$\ln |\hat{\mathbf{H}}_m| = \ln |\mathbf{K}^{-2}| + \ln |\mathbf{K} \hat{\mathbf{H}}_m \mathbf{K}|, \tag{24}$$

where the last term, which is of order $\mathcal{O}(1)$, is ignored, and the first term is used as a penalty term (see [17] for more details). We can now state the joint pitch and model order estimator used to compute initial estimates for sources $m = 1, \ldots, M$, i.e.,

$$\left\{ \hat{\omega}_{0,m}, \hat{L}_m \right\} = \arg\min_{\boldsymbol{\alpha}_m, \{\omega_{0,m}, L_m\}} \frac{\ln |\mathbf{K}^{-2}|}{2} + N \sum_{k=1}^{K} \ln \left\| \boldsymbol{\beta}_{k,m} \right\|_2^2, \tag{25}$$

where

$$\boldsymbol{\beta}_{k,m} = \mathbf{r}_k^{(m-1)} - \mathbf{Z}_m \mathbf{G}_{k,m} \tilde{\boldsymbol{\alpha}}_m, \tag{26}$$

and $\mathbf{r}_k^{(m)} = [r_k^m(0) \ r_k^m(1) \ \cdots \ r_k^m(N-1)]^T$. Since the cost function is multi-modal, it is minimized with respect to $\omega_{0,m}$ using a grid search (grid size selection is discussed in [29]). The fundamental frequencies and amplitudes of the M sources are obtained by iterating between the expectation and maximization steps, i.e., (19), and (20)-(22), respectively, until convergence.

3.3. Source Reconstruction and Re-Panning

The harmonic sources in an observed stereophonic mixture are implicitly modelled in the iterative parameter estimation process, i.e., the estimate of the mth source is

$$\hat{\mathbf{s}}_m(n) = \mathbf{Z}_m(n) \hat{\boldsymbol{\alpha}}_m, \tag{27}$$

for $n = 1, \ldots, N$. Since the number of entries in the amplitude codebook $\mathcal{C}$ is relatively small, the signals $\hat{\mathbf{s}}_m$, for $m = 1, \ldots, M$, may sound a bit rough when listened to directly. Instead, we propose to use the estimated parameters to form a frequency-domain Wiener filter to extract each source from a segment of the observed mixture, i.e.,

$$\bar{S}_m(\omega) = \frac{\|\hat{S}_m(\omega)\|^2}{\|\hat{S}_m(\omega)\|^2 + \|\hat{V}(\omega)\|} X(\omega), \tag{28}$$

where $\bar{S}_m(\omega)$ is the frequency-domain filter output at a certain frequency bin corresponding to ω, $\hat{S}_m(\omega)$ is the DFT of the source estimate $\hat{\mathbf{s}}_m$, $\hat{V}(\omega)$ is the DFT of the estimates of the interfering sources and the noise, i.e., $\mathbf{v} = \mathbf{x} - \hat{\mathbf{s}}_m$, $X(\omega)$ is the DFT of a single-channel version of the mixture. Each time-domain segment of each the M sources is generated as the inverse DFT of the filtered output above. The segments are combined using overlap-add.

4. EXPERIMENTS

The experimental evaluation of the proposed method for panning parameter estimation, source separation and re-panning consists of multiple experiments. To evaluate the performance of the proposed method for source separation, a multitrack recording from the MedleyDB database of music recordings [30] is used, i.e., Aimee Norwich - Flying. A segment containing 24 seconds (start: 105.5 s, end: 129.5 s) of audio from three instrument recordings

Table 1: Description of the data used in the experiments.

File name (.wav)	Instrument	ϕ (degrees)	τ (samples)
Flying_RAW_14_01	Trombone	30	0
Flying_RAW_03_02	Bass	5	0
Flying_RAW_15_02	Clarinet	-30	0

Table 2: Pannning parameter estimates.

Track	$\hat{\phi}$ (degrees)	$\hat{\tau}$ (samples)
Trombone	29.99	0.00
Bass	4.99	0.01
Clarinet	-29.97	0.00

are amplitude panned to synthetically generate a stereophonic mixture. Descriptions of the tracks used in the mixture and their panning parameters are presented in Table 1.

The estimation of the number submixtures and their panning parameters are evaluated on the observed stereo mixture with $f_s = 44.1$ kHz. The input signal is processed in samples of length $N = 2640$ samples (60 ms), with a hop size of $H = 662$ samples (15 ms). The GMM is overfitted with $M = 10$ and from the overfitted GMM components, an estimate of the source clusters are obtained. To lower the computational complexity and remove part of the noise floor from the spectrum, we select the frequency bins in the measurement vector (8) according to an indicator function $b(\omega)$ defined for all ω, i.e.,

$$b(\omega) = \begin{cases} 1, & |X_1(\omega)||X_2(\omega)| > |\mathbf{X}_1|^T|\mathbf{X}_2|/N \\ 0, & \text{otherwise} \end{cases} . \quad (29)$$

The estimated source clusters are shown in Figure 3. The source panning clusters are visualized, as overlayed on the data and $\mathbf{y}$, and the contours of the initial overfitted GMM components. Both amplitude panning angle and delay were estimated correctly and the results are shown in table 2. We observe that the panning parameters are almost equal to the true parameters. The number of sources has been estimated to the true value of $M = 3$. Next, we can evaluate the ASI estimation shown in Figure 2. The Figure shows the ASI overlayed on the unmixed sources. A black vertical line indicates activity in the given frame at the estimated panning angle, while no line means no activity. We observe that the overall trend is that the binary ASI resembles the activity of the sources, both in silent periods and when the sources contain significant energy.

The fundamental frequency estimates of the harmonic sources are obtained using the estimated panning parameters and the ASI. The mixture is downsampled to $f_s = 8$ kHz, and processed in segments of length $N = 480$ samples (60 ms), with a hop size of $H = 120$ samples (15 ms). The fundamental frequencies are estimated using a grid with 1 Hz spacing, from $f_{0,\text{min}} = 50$ Hz to $f_{0,\text{min}} = 1000$ Hz. As explained in Section 3.2, a codebook of magnitude amplitudes is used when estimating the complex amplitudes of the sources. The codebook is trained using anechoic instrument recordings from the IOWA database[1], and the signals

[1] Available at `http://theremin.music.uiowa.edu`.

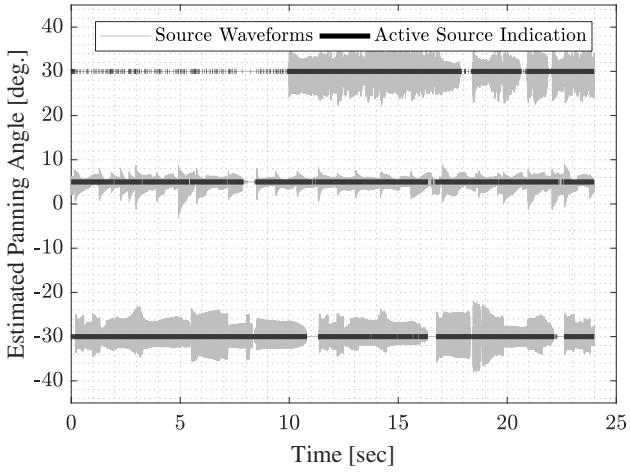


Figure 2: Active source indication (ASI) shown as black lines. For each frame of 15 ms there is an indicator. The ASI is overlayed on the original source signals which do not relate to the panning axis.

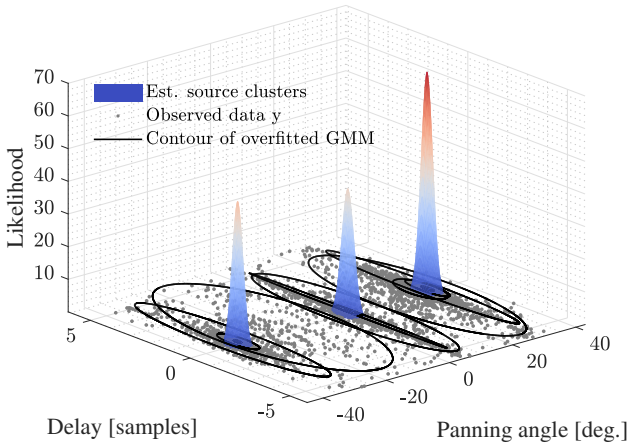


Figure 3: Proposed GMM estimation of source panning clusters.

used for training are listed in Table 3. See [17] for further details. The fundamental frequency estimates of the sources are shown in Figure 4, along with the ground truth which was obtained using the `joint_anls()` function from the Multi-Pitch Estimation Toolbox [24] on the individual instrument recordings from the dataset resulting in single-pitch estimates. No smoothing has been applied to the parameter estimates. The separation of the sources from the mixture is done using Wiener filtering, as described in Section 3.3. A spectrogram of a monophonic version of the observed mixture, obtained as an average of the stereo channels, is shown in Figure 6 along with the residual, which is obtained by subtracting the estimated sources from the mixture. We observe that most of the harmonic components in the mixture have been removed. The spectrograms of the unmixed and reconstructed bass tracks are shown in Figure 7. The reconstructed bass track contains most of the harmonic content in the unmixed source, however, some of the higher harmonics are missing. In Figure 8 the spectrograms of the unmixed and reconstructed trombone tracks are presented. The reconstructed trombone signal again contains most of the harmonic

Table 3: Data used for generating the amplitude codebook (v: played with vibrato).

Instrument	Instr. type	Note ranges	Duration (s)
Alto flute	Woodwind	G3-B3, C4-B4	68.3
Alto sax	Woodwind	Db3-B3, C4-B4	118.9
Alto sax (v)	Woodwind	Db3-B3, C4-B4	129.2
Bass flute	Woodwind	C3-B3, C4-B4	113.3
Bassoon	Woodwind	C3-B3, C4-B4	55.7
Bb clarinet	Woodwind	D3-B3, C4-B4	111.4
Eb clarinet	Woodwind	G3-B3,C4-B4	47.5
French horn	Brass	C2-B2, C4-B4	68.0
Oboe	Woodwind	Bb3-B3, C4-B4	46.6
Soprano sax	Woodwind	Ab3-B3, C4-B4	64.3
Soprano sax (v)	Woodwind	Ab3-B3, C4-B4	69.2
Tenor trombone	Brass	C3-B3, C4-B4	106.2
Trumpet	Brass	E3-B3, C4-B4	170.3
Trumpet (v)	Brass	E3-B3, C4-B4	182.9

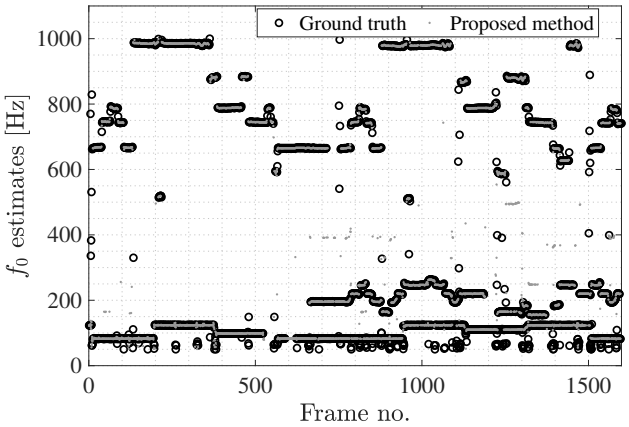


Figure 4: Fundamental frequency estimates of the sources in the mixture.

content, however, some segments in the beginning of the signal contain energy which was not present in the unmixed source; this is due to errors in the ASI. The spectrograms of the unmixed and reconstrcuted clarinet tracks are shown in Figure 9. Comparing the spectrograms of the unmixed and reconstructed tracks, it can be seen that the main harmonic components of the source have been captured in the reconstruction. A graphical user interface (GUI) is written in MATLAB in which the sources can be re-panned, using either the original panning parameters, or using new parameters[2]. Figure 5 shows a screenshot of the mixing GUI. An informal listening test suggests that including the residual ensures that information not captured by the harmonic model, such as breathing noises and other non-stationarities greatly improves the perceived quality of the reconstructed mixture.

[2]An audiovisual demonstration of the re-panning is available at `https://youtu.be/0HHoMVyOGcU`

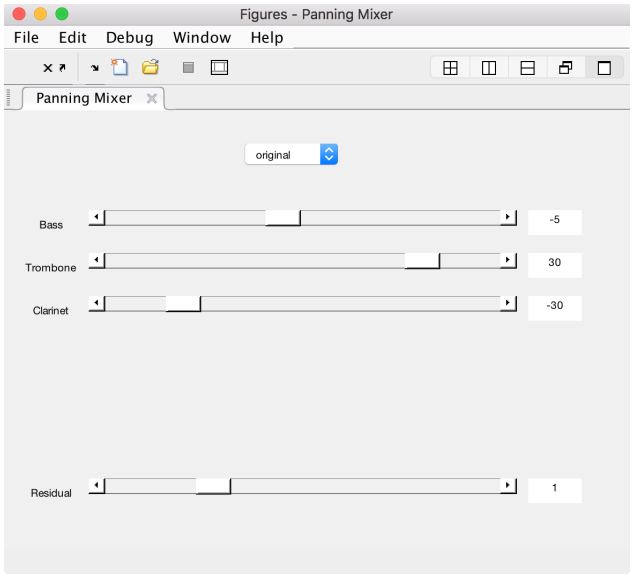


Figure 5: Screenshot of the GUI for mixture reconstruction.

5. DISCUSSION

In this paper, a method for separating an observed stereophonic mixture into its harmonic components, is presented. The method does not require knowledge of the number of sources in the mixture. The sources are extracted using a multi-channel harmonic signal model, where the panning parameters and the number of active sources in each frame of the mixture are estimated in an initial step. The fundamental frequencies, amplitudes and number of harmonics are estimated using an iterative approach. To enforce spectral smoothness, the magnitude amplitudes of the harmonics are mapped to entries in a codebook, which has been trained on individual notes played on a variation of instruments. The harmonic components are extracted by modelling the sources using the harmonic model and the estimated parameters. When the harmonic sources have been extracted, they are processed individually, i.e, the panning parameters of the sources are altered. The reconstruction of the mixture includes the residual, which contains the parts of the signal that are not captured by the harmonic signal model. When the residual is added to the mixture of extracted harmonic components, the resulting mixture is more pleasing to listen to. Extensions to this work could be the inclusion of inharmonicity in the signal model, to allow more precise modelling of string instrument signals, such as guitar, bass and piano recordings. Temporal smoothness could also be imposed in the parameter estimation steps. Furthermore, the signal model presented here is anechoic, i.e., the performance of the proposed method will degrade in the presense of reverberation effects. One option is to use a method for dereverberation, such as one of the methods presented in [31].

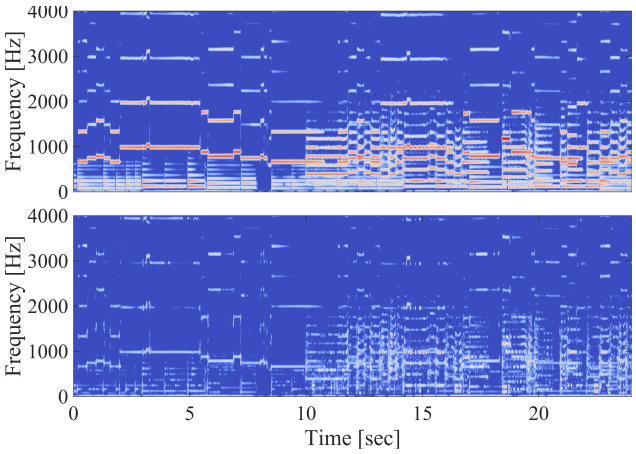


Figure 6: Spectrogram of the observed mixture (top) and the residual after subtraction of the harmonic sources (bottom).

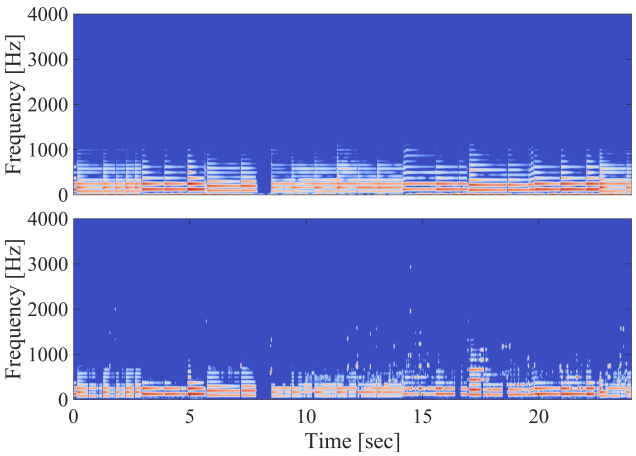


Figure 7: Spectrogram of the unmixed bass track (top) and the reconstructed bass track (bottom).

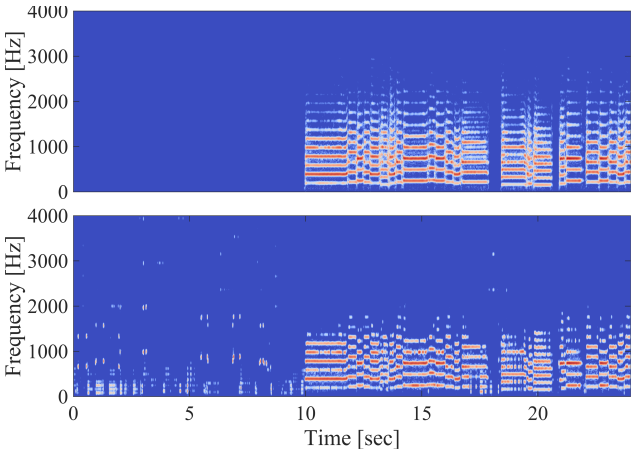


Figure 8: Spectrogram of the unmixed trombone track (top) and the reconstructed trombone track (bottom).

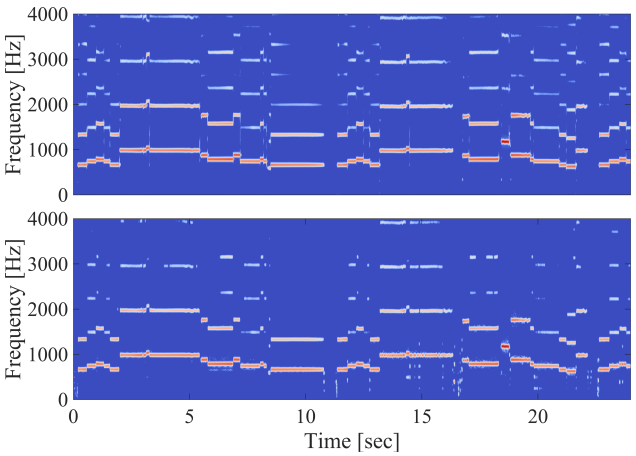


Figure 9: Spectrogram of the unmixed clarinet track (top) and the reconstructed clarinet track (bottom).

6. REFERENCES

[1] A. Klapuri and M. Davy, Eds., *Signal Processing Methods for Music Transcription*, Springer, New York, 2006.

[2] K. Kowalczyk, O. Thiergart, M. Taseska, G. Del Gado, V. Pulkki, and E. Habets, "Parametric spatial sound processing: A flexible and efficient solution to sound scene acquisition, modification, and reproduction," *IEEE Signal Process. Mag.*, vol. 32, no. 2, pp. 31–42, 2015.

[3] S. Gannot, E. Vincent, S. Markovich-Golan, and A. Ozerov, "A consolidated perspective on multimicrophone speech enhancement and source separation," *IEEE/ACM Trans. Audio, Speech, Language Process.*, vol. 25, no. 4, pp. 692–730, April 2017.

[4] L. Benaroya, L. M. Donagh, F. Bimbot, and R. Gribonval, "Non negative sparse representation for Wiener based source separation with a single sensor," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, April 2003, vol. 6, pp. VI–613–16 vol.6.

[5] P. Smaragdis and J. C. Brown, "Non-negative matrix factorization for polyphonic music transcription," in *Proc. IEEE Workshop Appl. of Signal Process. to Aud. and Acoust.*, 2003, pp. 177–180.

[6] A. Ozerov and C. Févotte, "Multichannel nonnegative matrix factorization in convolutive mixtures for audio source separation," *IEEE Trans. Audio, Speech, Language Process.*, vol. 18, no. 3, pp. 550–563, March 2010.

[7] A. Ozerov, E. Vincent, and F. Bimbot, "A general flexible framework for the handling of prior information in audio source separation," *IEEE Trans. Audio, Speech, Language Process.*, vol. 20, no. 4, pp. 1118–1133, May 2012.

[8] J. Nikunen, A. Diment, and T. Virtanen, "Separation of moving sound sources using multichannel NMF and acoustic tracking," *IEEE/ACM Trans. Audio, Speech, Language Process.*, vol. 26, no. 2, pp. 281–295, Feb 2018.

[9] S. Leglaive, R. Badeau, and G. Richard, "Separating time-frequency sources from time-domain convolutive mixtures using non-negative matrix factorization," in *Proc. IEEE Workshop Appl. of Signal Process. to Aud. and Acoust.*, 2017.

[10] C. Avendano, "Frequency-domain source identification and manipulation in stereo mixes for enhancement, suppression and re-panning applications," in *Proc. IEEE Workshop Appl. of Signal Process. to Aud. and Acoust.*, Oct 2003, pp. 55–58.

[11] M. Karjalainen and T. Tolonen, "Multi-pitch and periodicity analysis model for sound separation and auditory scene analysis," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 1999.

[12] T. Virtanen and A. Klapuri, "Separation of harmonic sound sources using sinusoidal modeling," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2000, vol. 2, pp. II765–II768 vol.2.

[13] J. Woodruff and B. Pardo, "Using pitch, amplitude modulation, and spatial cues for separation of harmonic instruments from stereo music recordings," *EURASIP J. on Applied Signal Processing*, vol. 2007, no. 1, pp. 086369, Dec 2006.

[14] J. Han and B. Pardo, "Reconstructing completely overlapped notes from musical mixtures," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, May 2011.

[15] A. Ben-Shalom and S. Dubnov, "Optimal filtering of an instrument sound in a mixed recording given approximate pitch prior," *Proc. Int. Computer Music Conf. (ICMC)*, 2004.

[16] J. M. Hjerrild and M. G. Christensen, "Estimation of source panning parameters and segmentation of stereophonic mixtures," *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2018.

[17] M. W. Hansen, J. R. Jensen, and M. G. Christensen, "Estimation of multiple pitches in stereophonic mixtures using a codebook-based approach," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2017.

[18] V. Pulkki, *Spatial sound generation and perception by amplitude panning techniques (PhD thesis)*, Helsinki University of Technology, 2001.

[19] S. Rickard and Ö. Yilmaz, "On the approximate W-disjoint orthogonality of speech," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, May 2002, vol. 1, pp. I–529–I–532.

[20] M. A. T. Figueiredo and A. K. Jain, "Unsupervised learning of finite mixture models," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, pp. 381–396, 2002.

[21] C. Hennig, M. Meila, F. Murtagh, and R. Rocci, *Handbook of Cluster Analysis*, Chapman and Hall/CRC, 2015.

[22] D. Chazan, Y. Stettiner, and D. Malah, "Optimal multi-pitch estimation using the em algorithm for co-channel speech separation," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, April 1993, vol. 2, pp. 728–731 vol.2.

[23] J. A. Fessler and A. O. Hero, "Space-alternating generalized expectation-maximization algorithm," *IEEE Trans. Signal Process.*, vol. 42, no. 10, pp. 2664–2677, Oct 1994.

[24] M. G. Christensen and A. Jakobsson, *Multi-Pitch Estimation*, Synthesis lectures on speech and audio processing. Morgan & Claypool Publishers, 2009.

[25] M. Feder and E. Weinstein, "Parameter estimation of superimposed signals using the EM algorithm," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. 36, no. 4, pp. 477–489, Apr 1988.

[26] P. Stoica, H. Li, and J. Li, "Amplitude estimation of sinusoidal signals: survey, new results, and an application," *IEEE Trans. Signal Process.*, vol. 48, no. 2, pp. 338–352, Feb 2000.

[27] R. Gribonval and E. Bacry, "Harmonic decomposition of audio signals with matching pursuit," *IEEE Trans. Signal Process.*, vol. 51, no. 1, pp. 101–111, Jan 2003.

[28] P. Stoica and Y. Selen, "Model-order selection: a review of information criterion rules," *Signal Process. Mag.*, vol. 21, no. 4, pp. 36–47, July 2004.

[29] J. K. Nielsen, T. L. Jensen, J. R. Jensen, M. G. Christensen, and S. H. Jensen, "Fast fundamental frequency estimation: Making a statistically efficient estimator computationally efficient," *Signal Process.*, vol. 135, pp. 188 – 197, 2017.

[30] R. M. Bittner, J. Salamon, M. Tierney, M. Mauch, C. Cannam, and J. P. Bello, "Medleydb: A multitrack dataset for annotation-intensive mir research.," in *Proc. Int. Conf. Music Information Retrieval*, 2014, vol. 14, pp. 155–160.

[31] P. A. Naylor and N. D. Gaubitch, *Speech Dereverberation*, Signals and Communication Technology. Springer, 2010.

FAST PARTIAL TRACKING OF AUDIO WITH REAL-TIME CAPABILITY THROUGH LINEAR PROGRAMMING

Julian Neri

SPCL* , CIRMMT†
McGill University, Montréal, Canada
julian.neri@mail.mcgill.ca

Philippe Depalle

SPCL* , CIRMMT†
McGill University, Montréal, Canada
philippe.depalle@mcgill.ca

ABSTRACT

This paper proposes a new partial tracking method, based on linear programming, that can run in real-time, is simple to implement, and performs well in difficult tracking situations by considering spurious peaks, crossing partials, and a non-stationary short-term sinusoidal model. Complex constant parameters of a generalized short-term signal model are explicitly estimated to inform peak matching decisions. Peak matching is formulated as a variation of the linear assignment problem. Combinatorially optimal peak-to-peak assignments are found in polynomial time using the Hungarian algorithm. Results show that the proposed method creates high-quality representations of monophonic and polyphonic sounds.

1. INTRODUCTION

The sinusoidal model proves beneficial for its capacity to represent non-stationary sounds. The sinusoidal model represents a sound signal as a sum of P time-varying sinusoids, called *partials*, with instantaneous log-amplitude $a_p(t)$, phase $\phi_p(t)$, and frequency $f_p(t)$,

$$s(t) = \sum_{p=1}^{P} \exp\big(a_p(t) + i\phi_p(t)\big) \tag{1}$$

$$\phi_p(t) = \phi_p(0) + 2\pi \int_0^t f_p(u)du \tag{2}$$

Decomposing a sound signal into a set of partials, or *partial tracking*, is useful for a variety of applications, including sound synthesis [1], sound source separation [2] [3], audio coding [4], audio effects [5] [6], and automatic music transcription [7] [8].

Partial tracking consists of two operations that are performed either sequentially or jointly. First, instantaneous sinusoidal model parameters are estimated from a short-term analysis of the sound signal. Second, the instantaneous parameters are linked according to their expected temporal progressions, forming partial trajectories. The parameter estimates are interpolated between each short-term analysis frame so that $a_p(t)$ and $\phi_p(t)$ can be evaluated at the sampling rate.

Despite practical applications of partial tracking and its wide use in the field, aspects of the process complicate the potential for a flawless outcome. A complex sound often has hundreds of partials, plus a stochastic component, sculpting its time-varying spectral envelope [9]. Sinusoidal model parameters must be estimated accurately from short-term estimates to ensure appropriate tracking decisions. Polyphonic sounds further complicate the analysis because the frequency trajectories of two partials might cross [10]. Peak matching poses a large combinatorial problem that must be repeated for many, typically thousands, of time frames. Thus, there are not only difficulties associated with the quality of tracking, but also with speed and tractability [11]. Many partial tracking methods have been proposed over the last several decades, as summarized in Section 1.1.

This paper presents a new partial tracking method, based on linear programming, that improves the state of the art of sinusoidal modeling. The proposed method can operate in real-time, is simpler to implement than the McAulay and Quatieri (MCQ) method [12], and creates sinusoidal model representations comparable to the leading hidden Markov model (HMM)-based methods [10] [11]. For parameter estimation, the method considers a generalized non-stationary short-term sinusoidal model. The peak matching procedure is formulated as a variation of the linear assignment problem [13], a fundamental combinatorial optimization problem, allowing for an optimal peak-to-peak assignment solution in polynomial time.

This paper is organized as follows. Section 2 overviews the assignment problem. Section 3 establishes the new method of partial tracking, first by describing short-term analysis additive model parameter estimation, then by deriving the assignment problem costs. Section 4 details the results from experiments that demonstrate the ability of the new partial tracker. Section 5 concludes the paper and proposes future research on the applications of the assignment problem in audio.

1.1. Overview of Previous Work

McAulay and Quatieri (MCQ) [12] developed the first partial tracking algorithm for sinusoidal modeling of speech. The MCQ method connects peaks that have minimum frequency difference between consecutive analysis frames. The MCQ method uses a non-optimal greedy algorithm, does not consider spurious peaks, and assumes a stationary short-term signal model. Modifications of the MCQ method include using a reassigned bandwidth enhanced model [14] and considering an intermediate "sleep" state for every trajectory [15]. A linear prediction coding-based method was proposed in [16] [17] that determines the most probable match using the trajectory's previous samples and can interpolate missing data. A non-causal strategy was proposed in [18] that builds each trajectory starting from a reliable two-point connection then growing it in every direction by appending smaller pieces to it. The adaptive method from [19] uses B-splines to estimate the parameters of the additive model. Adaptive oscillators were used to track partials in [20], and a Kalman filtering approach was described in [21]. The

* Sound Processing and Control Laboratory
† Centre for Interdisciplinary Research in Music Media and Technology

hidden Markov model (HMM) partial tracker [10] optimizes the combination of trajectories within an analysis window, considers spurious peaks, and performs well in several difficult tracking situations. The HMM tracker was improved in [11] for non-stationary and noisy signals by formulating a new peak matching criterion that incorporates explicitly measured frequency slope information.

The assignment problem [22], presented in Section 2, is a fundamental combinatorial optimization problem that describes a variety of real-world problems. Variations of the assignment problem, especially the multidimensional assignment problem [23], have been used to describe the problem of multi-target tracking [24], jointly estimating the number of targets and their trajectories from sensor measurements. Although the assignment problem has been successfully applied to such problems for over a half century, to the extent of our knowledge, it has not been applied to tracking problems in audio.

2. THE ASSIGNMENT PROBLEM

2.1. Problem statement

The assignment problem is a fundamental combinatorial optimization problem in the field of operations research [13].

The problem involves assigning R members of one set, *agents*, to another, *tasks*. Any agent can perform any task. An agent-task assignment incurs a *cost* that may vary depending on the assignment. The goal is to assign an agent to perform one task, and assign a task to one agent, such that the sum of individual costs is minimized.

The assignment problem is formally expressed as a linear programming problem with the following mathematical model:

$$\text{minimize} \quad \sum_{i=1}^{R}\sum_{j=1}^{R} C_{ij}X_{ij} \tag{3a}$$

$$\text{subject to} \quad \sum_{i=1}^{R} X_{ij} = 1 \qquad j = 1,\ldots,R \tag{3b}$$

$$\sum_{j=1}^{R} X_{ij} = 1 \qquad i = 1,\ldots,R \tag{3c}$$

where C_{ij} is the cost of assigning agent i to task j, and X_{ij} is a binary variable that equals 1 if agent i is assigned to task j and 0 otherwise. The first constraint (3b) ensures that every agent is assigned to one task, while the second constraint (3c) ensures that every task is assigned to one agent.

In terms of graph theory, this is equivalent to finding the minimum cost assignment in a weighted bipartite graph [22]. Figure 1 represents the assignment of agents to tasks as a graph and as an annotated cost matrix.

Linear programming problems can be solved by the simplex algorithm [25], however, more efficient algorithms have been developed that take advantage of the assignment problem's specific structure.

2.2. Hungarian algorithm

The Hungarian algorithm is a combinatorial algorithm that can solve the assignment problem in polynomial time [26]. The algorithm takes as an input the cost matrix $\mathbf{C}$ and outputs the optimal assignments matrix $\mathbf{X}$. If the number of agents does not equal the number of tasks, dummy variables are appended to the cost matrix to make it square. The Hungarian algorithm consists of the following three steps.

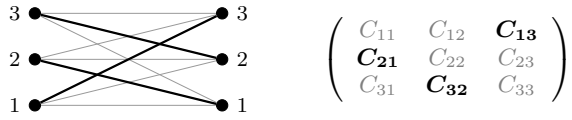


Figure 1: Assignments represented as bold lines in a bipartite graph (left) and as bold elements in a cost matrix (right).

1. For each row, subtract the row's minimum value from every value in that row. For each column, subtract the column's minimum value from every value in that column.

2. Cover the zeros in the resulting matrix with the minimum number of vertical and horizontal lines. If R lines are required, an optimal assignment of zeros exists and the algorithm stops. If less than R lines are required, proceed to Step 3.

3. Find the minimum value in the matrix that is not covered by the lines from Step 2. Subtract the value from every uncovered element and add the value to every covered element. Return to Step 2.

This popular algorithm's implementation is freely available online (commonly as a single function) in several software languages [27].

2.3. Variations of the Assignment Problem

Variations of the assignment problem use different objectives, constraints, or dimensions. A survey of such variations is in [13]. For example, a *one-to-many* assignment problem has a looser constraint that allows an agent to perform more than one task. A variation that is particularly applicable to multi-target tracking is the *multidimensional* assignment problem.

2.4. Multidimensional Assignment Problem

Multidimensional assignment problems consist of assigning the members of three or more sets [28]. A type of multidimensional problem that has been applied to multi-target tracking is the axial three-dimensional assignment problem. This type of problem involves assigning members over three sets, where each assignment incurs a cost C_{hij}, such that the total cost is minimized.

Multidimensional assignment problems are NP-hard [23]. The simplest way to solve a multidimensional assignment problem is to enumerate every possible combination of assignments then choose the one with the lowest cost [28], however, this solves the problem in factorial time. Research has led to algorithms that either solve or approximately solve the problem with improved tractability. For example, [29] details a branch and bound algorithm that approximately solves the axial three-dimensional case. Alternatively, [23] shows that an axial three-dimensional problem can be solved in polynomial time if the cost C_{hij} can be split into the sum of two sub-costs, $C_{hij} = C_{hi} + C_{ij}$.

3. PROPOSED METHOD

3.1. Overview

The proposed partial tracking method sequentially performs two processes. First, short-term sinusoidal model parameters are estimated for each peak j in frame k. Second, the short-term parameter estimates are connected over consecutive analysis frames, $k-1$ and k, by solving an assignment problem, forming trajectories.

This paper considers a trajectory to be a time-sequence of spectral peaks with short-term sinusoidal model parameters, defined in Section 3.2, that satisfy continuity constraints at the midpoint of consecutive analysis frames. Accordingly, *useful* assignments satisfy those continuity constraints while *spurious* assignments do not. Section 3.3 defines a cost for both assignment types. The assignment type with the lowest cost is the most probable. The optimal combination of assignments is found using the Hungarian algorithm.

3.2. Short-Term Additive Parameter Estimation

Parameters are estimated over short-term analysis frame k at time $t_k = kH/f_s$, where H is the hop size and f_s is the sampling frequency. The frame's time index n ranges from $-N/2$ to $N/2$, where $N+1$ is the frame's duration. The center of the frame is at $n = 0$ and aligned with t_k.

The short-term signal model over frame k is a sum of R_k generalized sinusoids

$$s(n) = \sum_{j}^{R_k} \exp\left(\sum_{i=0}^{Q} \alpha_{ij} n^i\right) \tag{4}$$

where α_{ij} are the complex constants of sinusoid j and Q is the order of the polynomial [30]. The instantaneous log-amplitude and phase of sinusoid j are

$$a_j^k(n) = \Re\left(\sum_{i=0}^{Q} \alpha_{ij} n^i\right) \tag{5}$$

$$\phi_j^k(n) = \Im\left(\sum_{i=0}^{Q} \alpha_{ij} n^i\right) \tag{6}$$

Since the sinusoid's normalized angular frequency is the time derivative of the phase,

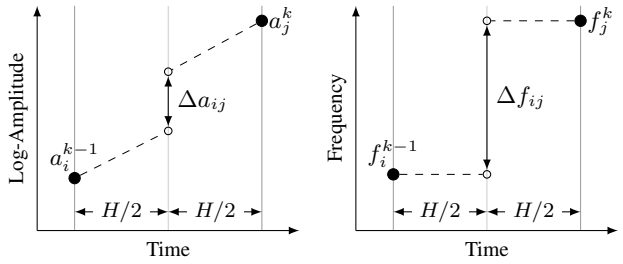
$$f_j^k(n) = \frac{f_s}{2\pi} \Im\left(\sum_{i=0}^{Q} \alpha_{ij} i n^{i-1}\right) \tag{7}$$

There are several options for estimating α_{ij}. A comparison of sinusoidal model parameter estimators is in [31]. Using the distribution derivative method (DDM) [30] allows for the estimation of α_{ij} up to an arbitrary polynomial order Q.

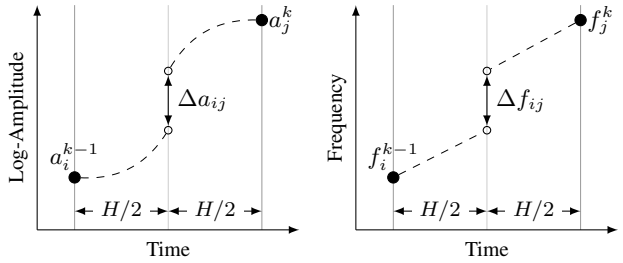
3.3. Costs of Useful and Spurious Assignments

An assignment cost is quantified by a multivariate Gaussian, similarly to the "matching criterion" defined in [10]. The cost of assigning peak i in frame $k-1$ to peak j in frame k is

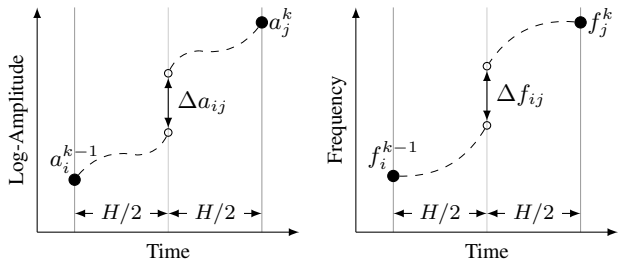
$$A_{ij} = 1 - \exp\left(-\frac{\Delta f_{ij}^2}{2\sigma_f^2} - \frac{\Delta a_{ij}^2}{2\sigma_a^2}\right) \tag{8}$$



(a) $Q = 1$.



(b) $Q = 2$.



(c) $Q = 3$.

Figure 2: Illustration of equation (10) (left) and (11) (right) for different settings of polynomial order Q.

for useful assignments and

$$B_{ij} = 1 - (1-\delta)A_{ij} \tag{9}$$

for spurious assignments, where

$$\Delta a_{ij} = a_i^{k-1}(H/2) - a_j^k(-H/2) \tag{10}$$

$$\Delta f_{ij} = f_i^{k-1}(H/2) - f_j^k(-H/2) \tag{11}$$

Figure 2 illustrates that equations (10) and (11) evaluate the midpoint continuity over peak i and j by extending their short-term sinusoidal model amplitude and frequency trajectories.

Standard deviations σ_f and σ_a are defined by the formulas

$$\sigma_f^2 = \zeta_f^2 / \big(2\ln(\delta - 2) - 2\ln(\delta - 1)\big) \tag{12}$$

$$\sigma_a^2 = \zeta_a^2 / \big(2\ln(\delta - 2) - 2\ln(\delta - 1)\big) \tag{13}$$

The parameter δ changes the relative preference towards spurious or useful assignments: smaller values promote spurious ones and larger values promote useful ones. Parameters ζ_f and ζ_a are values of Δf and Δa, respectively, that mark the point of transition between a useful or spurious assignment. Figure 3 shows how the cost functions change with respect to these parameters.

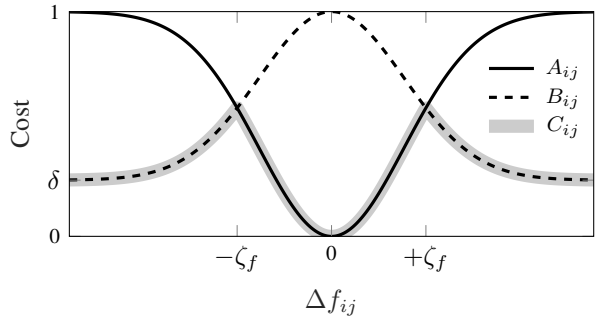


Figure 3: Illustration of how the parameters ζ_f and δ change the useful cost A_{ij}, spurious cost B_{ij}, and cost C_{ij} defined in equation (14).

3.4. Cost Matrix

This is a *multi-criteria* assignment problem [13] because it consists of minimizing an objective function that has two decision criteria. There is not only a cost A_{ij} of connecting two peaks as a useful trajectory, but also a cost B_{ij} of not connecting them. We can recognize this decision model's multiple criteria simply by constructing a single cost matrix whose values are

$$C_{ij} = \min\{A_{ij}, B_{ij}\} \tag{14}$$

3.5. Solving the Assignment Problem

The optimal assignments matrix $\mathbf{X}$ is retrieved by inputting the cost matrix $\mathbf{C}$ into the Hungarian algorithm. Following equation (14), an assignment $X_{ij} = 1$ is useful if A_{ij} is less than B_{ij}.

3.6. Partial Labeling

A trajectory is an unbroken (continuous) path from a peak in some frame to a peak in a future frame. Therefore, a useful assignment that continues an existing trajectory from the previous observation is labeled with that trajectory's index. On the other hand, if a useful assignment does not continue a path but rather starts one, it is labeled with a new index. Figure 4 illustrates the labeling of useful assignments over time.

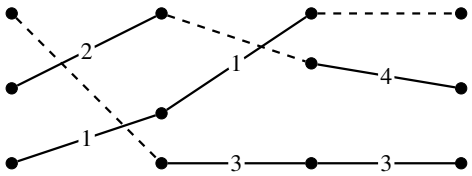


Figure 4: Illustration of labeling useful assignments (solid lines) over a sequence of frames. Dashed lines show spurious assignments.

3.7. Computation Cost & Implementation

True real-time operation is possible not only because the peak matching method has a low computational cost, but also because peak assignment and labeling only depends on the current frame k and previous frame $k-1$. Solving the assignment problem is a polynomial time operation, $O(R^3)$, where R is the largest of the two number of peaks R_{k-1} and R_k. The proposed method can run in real-time in many practical situations, depending on R, the hop size H, and the speed of the parameter estimator.

Implementing this partial tracker is simpler than other ones, including the MCQ method. Peak matching only consists of defining a cost matrix with equation (14) and running the Hungarian algorithm.

3.8. Recasting Previous Partial Tracking Methods

The McAulay and Quatieri (MCQ) method matches peaks over consecutive frames based on a minimum frequency difference criterion. In terms of an assignment problem, the cost is simply

$$C_{ij} = |f_i - f_j| \tag{15}$$

Rather than use the MCQ method's non-optimal greedy algorithm, optimal assignments can be found using the Hungarian algorithm. This approach avoids all the heuristics associated with the MCQ method. The MCQ method recast as an assignment problem is a simple case of the proposed method with $Q = 1$ that does not consider amplitude information or spurious assignments.

The hidden Markov model (HMM)-based method considers the peak connections over two frames as a hidden state. State transition probabilities are the product of matching criteria. Each matching criterion θ_{hij} quantifies how well peaks h, i, and j, over frames $k-2$, $k-1$, and k (two states), satisfy parameter slope continuity constraints,

$$\theta_{hij} = \exp\left(-\frac{(\Delta f_{hi} - \Delta f_{ij})^2}{\sigma_f^2} - \frac{(\Delta a_{hi} - \Delta a_{ij})^2}{\sigma_a^2} \right) \tag{16}$$

where $\Delta f_{ij} = f_i - f_j$ and $\Delta a_{ij} = a_i - a_j$.

The HMM-based method can be recast as a multidimensional assignment problem. The peak connections that admit the largest product of matching criteria (state transition probability) are the same ones that admit the smallest sum of assignment costs.

More specifically, the recast HMM method involves a three-dimensional assignment problem because the cost depends on peak parameters (members) over three frames (sets). Recall from Section 2.4 that such a problem is NP-hard. Making the HMM method tractable involves constraining the number of potential states.

Alternatively, if the matching criterion can be expressed as a product of two sub-criteria, $\theta_{hij} = \theta_{hi}\theta_{ij}$, then a polynomial time solution is possible through an assignment problem with cost $C_{hij} = C_{hi} + C_{ij}$. In [11] frequency slope ψ is explicitly estimated and the matching criterion is

$$\theta_{hij} = \exp\left(-\frac{\Delta f_{hi}^2}{\sigma_f^2} - \frac{\Delta f_{ij}^2}{\sigma_f^2} - \frac{(\Delta a_{hi} - \Delta a_{ij})^2}{\sigma_a^2} \right) \tag{17}$$

where $\Delta f_{ij} = (f_i + \psi_i H/2f_s) - (f_j - \psi_j H/2f_s)$. While the frequency slope calculation depends on only two frames, the calculation of amplitude slope depends on parameters over three frames, so the problem is still NP-hard.

The cost function developed in Section 3.3 is expressed in terms of parameters across only two frames, k and $k-1$, by explicitly estimating both the amplitude and the frequency slope, enabling a linear assignment problem formulation and polynomial time solution.

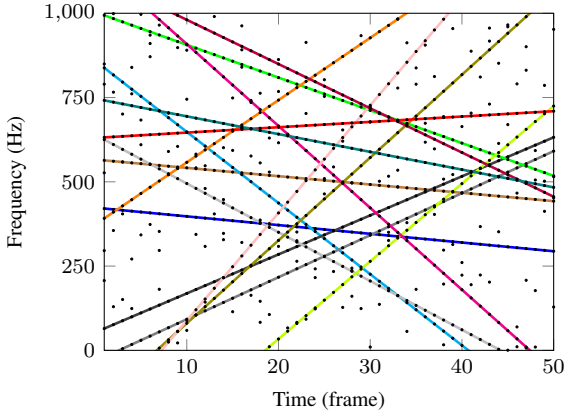


Figure 5: Detected partials (lines) from simulated data (dots) that resemble overlapping chirp sinusoids plus noise.

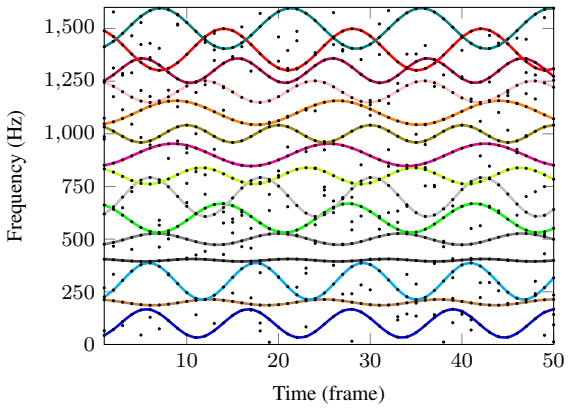
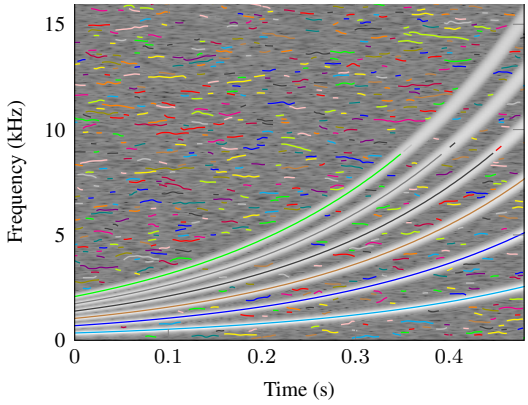
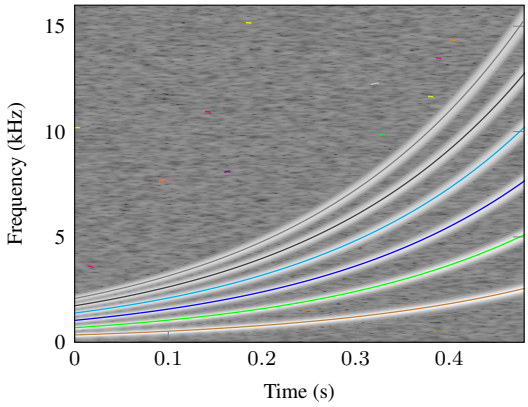


Figure 6: Detected partials (lines) from simulated data (dots) that resemble overlapping frequency modulated sinusoids plus noise.



(a) $Q = 1$. 12 dB R-SNR.



(b) $Q = 2$. 52 dB R-SNR.

Figure 7: Detected partials from a synthesized audio signal consisting of harmonically-related logarithmic chirp sinusoids plus noise for different settings of polynomial order Q.

4. RESULTS

4.1. Simulated Data

In these examples, peak parameters are simulated (set "by hand"). Circumventing the short-term analysis highlights the ability of the proposed peak matching method. Useful peaks are simulated by sampling parameter values from analytic expressions of partial trajectories, while spurious peaks are simulated by setting parameter values randomly. Each peak's amplitude-related values are set to zero, which further complicates tracking. Figures 5 and 6 show that the tracker perfectly classifies useful and spurious peaks and resolves overlapping partials that have similar frequency slopes.

4.2. Audio Signals

In the following examples, peak parameters are estimated from a short-term analysis of an audio signal, $s(n)$, using the distribution derivative method (DDM) [30]. The first group of examples involve audio signals that are synthesized from partial trajectories with constant amplitude and corrupted with -40 dB of white noise. The second group of examples involve real speech and musical audio signals. The signal is reconstructed as $\hat{s}(n)$ using the synthesis method described in [12]. The reconstruction signal-to-noise ratio (R-SNR) is used to help quantify the results, given by

$$\text{R-SNR} = 10\log_{10}\left(\frac{\sum_{n=0}^{N-1} s(n)^2}{\sum_{n=0}^{N-1}\left(s(n)-\hat{s}(n)\right)^2}\right) \qquad (18)$$

Figure 7 shows the results of tracking a synthetic harmonic signal whose fundamental frequency quickly increases on a logarithmic scale. If frequency slope is not estimated, as shown in Figure 7a, then ζ_f must be large enough to ensure useful peaks are connected over large frequency differences. This results in many false detections of useful assignments from spurious data. Figure 7b shows how the results improve dramatically when frequency slope is estimated.

Figure 8 shows the results of tracking a harmonic signal with strong vibrato (± 5 semitones). A further challenge is posed at 0.5 seconds where the fundamental frequency smoothly steps up by 5 semitones, resulting in close partials with steep slopes.

Figure 9 shows the results of tracking synthetic polyphonic audio that resembles a violin sound with unnaturally strong and fast vibrato superimposed with a trombone sound performing a fast upward glissando.

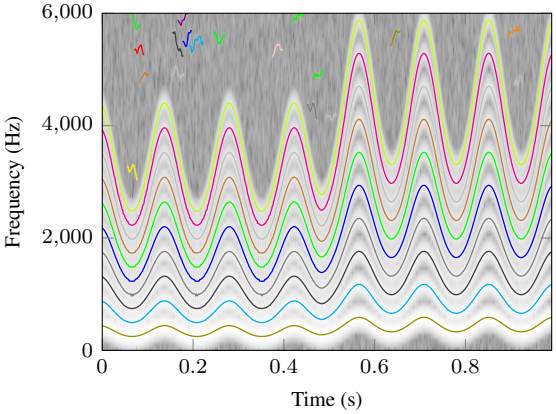


Figure 8: Detected partials from synthesized harmonic audio with vibrato plus noise. The fundamental frequency smoothly steps up by 5 semitones, from 330 Hz to 440 Hz. 26 dB R-SNR.

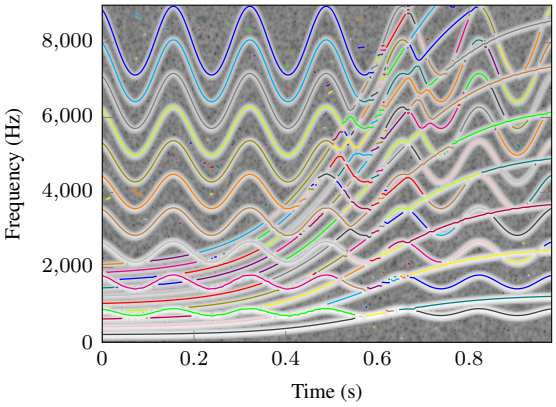
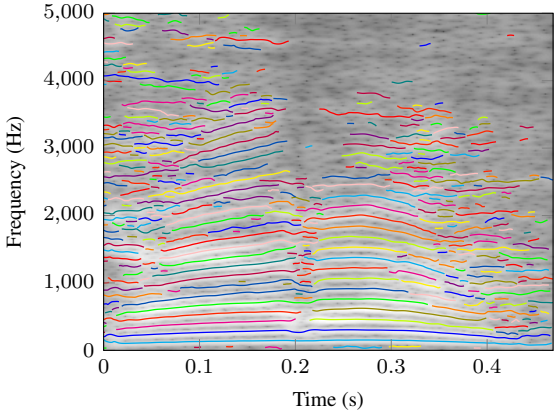


Figure 9: Detected partials from synthesized polyphonic audio plus noise. 19 dB R-SNR.

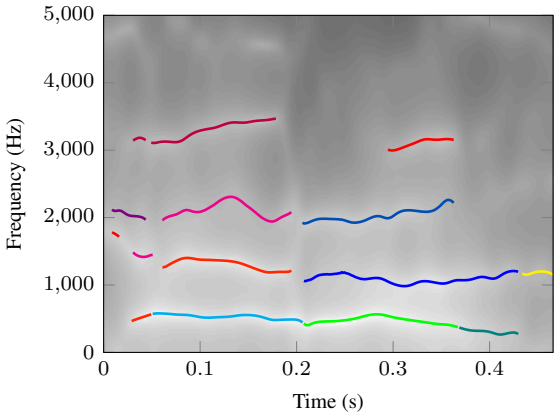
Formant tracking is another time-frequency tracking process that is especially suitable for vocal sounds. The proposed method can be used without modification for formant tracking applications. Figure 10a shows the partials detected from a real male voice sound while Figure 10b shows the results of tracking the formants of the same sound. Linear predictive coding (LPC) with 24 coefficients was used instead of DDM to estimate each formant's short-term amplitude and frequency, corresponding to an order $Q = 1$ polynomial.

Finally, the results of tracking a tango excerpt by Piazolla are shown in Figure 11. This multi-instrumental composition admitted dense short-term spectra with several frames having greater than 150 peaks. For this 14-second long signal over 12,000 partials were detected in a total computational time of 13 seconds on a 2.8 GHz quad-core processor: parameter estimation took 9 seconds and tracking took 4 seconds. The reconstructed sound is perceptually close to the original with a 15 dB R-SNR.

All test signals and reconstructed sounds are available for listening at http://www.music.mcgill.ca/~julian/dafx18.



(a) Detected partials. 22 dB R-SNR.



(b) Detected formants.

Figure 10: Male voice signal tracking results (/kara/).

5. CONCLUSIONS AND FUTURE WORK

This paper developed a new partial tracking method that matches sinusoidal model parameters over consecutive analysis frames by solving a linear assignment problem with the Hungarian algorithm. Results show that the proposed method easily handles exceptionally difficult partial tracking scenarios, involving strongly modulated partials embedded in noise and crossing partials that are common in polyphonic recordings. Moreover, the proposed tracker can operate in real-time and is simple to implement. Other popular methods were recast under the assignment problem framework, revealing them as specific cases of the proposed method. Future work may examine the results of tracking without slope information by solving a multidimensional assignment problem. More generally, other audio applications may be advantageously described as assignment problems.

6. REFERENCES

[1] M. Caetano, G. Kafentzis, A. Mouchtaris, and Y. Stylianou, "Full-band quasi-harmonic analysis and synthesis of musi-

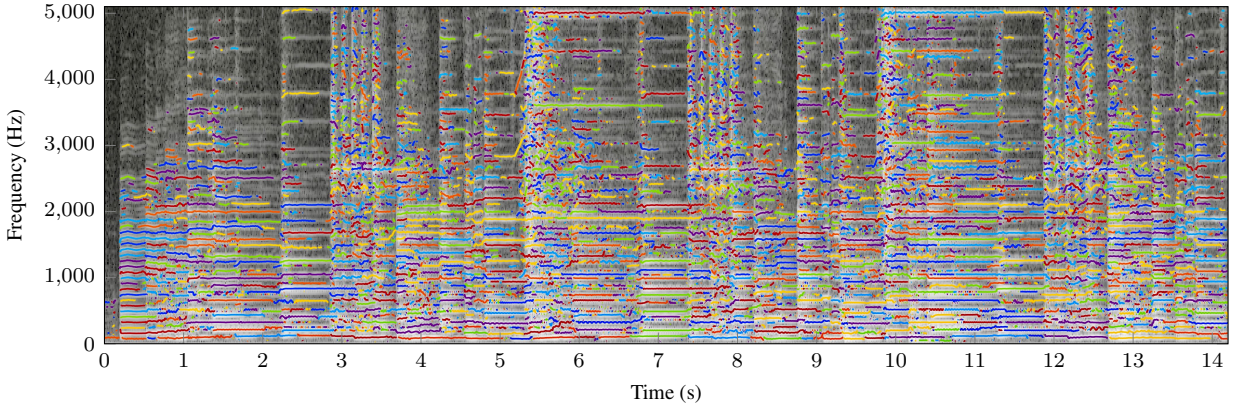


Figure 11: 12,000 detected partials from a polyphonic song sampled at 44.1 kHz. 15 dB R-SNR.

cal instrument sounds with adaptive sinusoids," *Applied Sciences*, vol. 6, no. 5, May 2016.

[2] T. Virtanen and A. Klapuri, "Separation of harmonic sound sources using sinusoidal modeling," in *IEEE Proc. Int. Conf. Acoust. Speech Signal Process. (ICASSP)*, Istanbul, Turkey, Jun. 2000, vol. 2, pp. 765–768.

[3] E. Creager, N. D. Stein, R. Badeau, and P. Depalle, "Non-negative tensor factorization with frequency modulation cues for blind audio source separation," in *Proc. of the 17th ISMIR Conf.*, New York City, USA, Aug. 2016.

[4] O. Derrien, R. Badeau, and Gaël Richard, "Parametric audio coding with exponentially damped sinusoids," *IEEE Trans. Audio, Speech, Language Process.*, vol. 21, no. 7, pp. 1489–1501, Jul. 2012.

[5] M. Raspaud, S. Marchand, and L. Girin, "A generalized polynomial and sinusoidal model for partial tracking and time stretching," in *Proc. of the 8th Int. Conf. on Digital Audio Effects (DAFx-05)*, Madrid, Spain, Sep. 2005.

[6] S. Kazazis, P. Depalle, and S. McAdams, "Sound morphing by audio descriptors and parameter interpolation," in *Proc. of the 19th Int. Conf. on Digital Audio Effects (DAFx-16)*, Brno, Czech Republic, Sep. 2016.

[7] M. Christensen and A. Jakobsson, "Multi-pitch estimation," *Synthesis Lectures on Speech and Audio Processing*, vol. 5, no. 1, Morgan and Claypool, pp. 1–160, 2009.

[8] J. Burred, A. Röbel, and T. Sikora, "Polyphonic musical instrument recognition based on a dynamic model of the spectral envelope," in *IEEE Proc. Int. Conf. Acoust. Speech Signal Process. (ICASSP)*, Taipei, Taiwan, Apr. 2009, pp. 173–176.

[9] N. Fletcher and T. Rossing, *The Physics of Musical Instruments*, Springer New York, 2nd edition, 1998.

[10] P. Depalle, G. Garcia, and X. Rodet, "Tracking of partials for additive sound synthesis using hidden Markov models," in *IEEE Proc. Int. Conf. Acoust. Speech Signal Process. (ICASSP)*, Apr. 1993, vol. 1, pp. 225–228.

[11] C. Kereliuk and P. Depalle, "Improved hidden Markov model partial tracking through time-frequency analysis," in *Proc. of the 11th Int. Conf. on Digital Audio Effects (DAFx-08)*, Espoo, Finland, Sep. 2008.

[12] R. McAulay and T. Quatieri, "Speech analysis/synthesis based on a sinusoidal representation," *IEEE Trans. Acoustics, Speech, Signal Process.*, vol. 34, no. 4, pp. 744–754, 1986.

[13] D. Pentico, "Assignment problems: A golden anniversary survey," *European Journal of Operational Research*, vol. 176, no. 2, pp. 774–793, Jan. 2007.

[14] K. Fitz and L. Haken, "Bandwidth enhanced sinusoidal modeling in Lemur," in *Proc. Int. Computer Music Conf. (ICMC)*, Banff, Canada, 1995, pp. 154–157.

[15] X. Serra and J. O. Smith, "Spectral modeling synthesis: a sound analysis/synthesis system based on a deterministic plus stochastic decomposition," *Computer Music Journal*, vol. 14, no. 4, pp. 12–24, 1990.

[16] M. Lagrange, S. Marchand, and J. Rault, "Tracking partials for the sinusoidal modeling of polyphonic sounds," in *IEEE Proc. Int. Conf. Acoust. Speech Signal Process. (ICASSP)*, Philadelphia, USA, 2005, vol. 3, pp. 229–232.

[17] M. Lagrange, S. Marchand, and J. Rault, "Enhancing the tracking of partials for the sinusoidal modeling of polyphonic sounds," *IEEE Trans. Acoustics, Speech, Signal Process.*, vol. 15, no. 5, pp. 1625–1634, Jul. 2007.

[18] M. Bartkowiak and T. Żernicki, "A non-time-progressive partial tracking algorithm for sinusoidal modeling," in *AES 131st International Conference*, New York, USA, Oct. 2011.

[19] A. Röbel, "Adaptive additive modeling with continuous parameter trajectories," *IEEE Trans. Audio, Speech, Language Process.*, vol. 14, no. 4, pp. 1440–1453, Jul. 2006.

[20] M. Marolt, "A connectionist approach to automatic transcription of polyphonic piano music," *IEEE Transactions on Multimedia*, vol. 6, no. 3, pp. 439–449, Jun. 2004.

[21] H. Satar-Boroujeni and B. Shafai, "A robust algorithm for partial tracking of music signals," in *Proc. of the 8th Int. Conf. on Digital Audio Effects (DAFx-05)*, Madrid, Spain, Sep. 2005, pp. 202–207.

[22] R. Burkard, M. Dell'Amico, and S. Martello, *Assignment Problems, Revised Reprint*, Society for Industrial and Applied Mathematics, 2012.

[23] K. Gilbert and R. Hofstra, "Multidimensional assignment problems," *Decision Sciences*, vol. 19, no. 2, pp. 306–321, Jun. 1988.

[24] A. Poore and S. Gadaleta, "Some assignment problems arising from multiple target tracking," *Mathematical and Computer Modelling*, vol. 43, pp. 1074–1091, May 2006.

[25] D. Luenberger and Y. Ye, *Linear and Nonlinear Programming*, Springer, 4th edition, 2016.

[26] H. Kuhn, "The Hungarian method for the assignment problem," in *Naval Research Logistic Quarterly*, Mar. 1955, vol. 2, pp. 83–97.

[27] Y. Cao, "Hungarian Algorithm for Linear Assignment Problems (V2.3)," MATLAB Central File Exchange, 2011.

[28] W.P. Pierskalla, "The multidimensional assignment problem," *Operations Research*, vol. 16, no. 2, pp. 422–431, Apr. 1968.

[29] E. Balas and M. Saltzman, "An algorithm for the three-index assignment problem," *Operations Research*, vol. 39, no. 1, pp. 150–161, Feb. 1991.

[30] M. Betser, "Sinusoidal polynomial parameter estimation using the distribution derivative," *IEEE Transactions on Signal Processing*, vol. 57, no. 12, pp. 4633–4645, Dec. 2009.

[31] B. Hamilton and P. Depalle, "Comparisons of parameter estimation methods for an exponential polynomial sound signal model," in *AES 45th International Conference*, Helsinki, Finland, Mar. 2012.

MODAL ANALYSIS OF ROOM IMPULSE RESPONSES USING SUBBAND ESPRIT

Corey Kereliuk *
Reverberate.ca
St. John's, NL, Canada
info@reverberate.ca

Woody Herman
Eventide Inc.
Little Ferry, NJ
WHerman@eventide.com

Russell Wedelich
Eventide Inc.
Little Ferry, NJ
RWedelich@eventide.com

*Daniel J. Gillespie**
Newfangled Audio
New York, NY
DGillespie@eventide.com

ABSTRACT

This paper describes a modification of the ESPRIT algorithm which can be used to determine the parameters (frequency, decay time, initial magnitude and initial phase) of a modal reverberator that best match a provided room impulse response. By applying perceptual criteria we are able to match room impulse responses using a variable number of modes, with an emphasis on high quality for lower mode counts; this allows the synthesis algorithm to scale to different computational environments. A hybrid FIR/modal reverb architecture is also presented which allows for the efficient modeling of room impulse responses that contain sparse early reflections and dense late reverb. MUSHRA tests comparing the analysis/synthesis using various mode numbers for our algorithms, and for another state of the art algorithm, are included as well.

1. INTRODUCTION

Artificial reverberation is a now ubiquitous effect that is often used to add a sense of space and color to a live performance or recording. The acoustics of a reverberant space depend on several factors including a building's architecture, wall materials, furniture, and so on. These factors affect the intensity and directionality of echoes arriving at a listener over time. Artificial reverberation algorithms aim to model these echoes, either directly or indirectly, and often with different goals in mind as explained below.

Digital signal processing algorithms for artificial reverberation have a long history. A comprehensive examination of this history is given by the review article of Välimäki et al. [1]. A brief taxonomy of reverb algorithms includes:

- purely algorithmic and parametric approaches, e.g., Schroeder's allpass chains [2], feedback delay networks [3][4], sparse FIR filters [5], and modal filter banks [6] [7] [8],
- convolutional reverbs [9], and
- physical modelling [10].

The wide-variety of techniques for artificial reverberation is a testament to the importance of this effect. We may also conjecture that the development of different reverb algorithms has been led by different design goals. To illustrate, convolutional reverbs are capable of very accurate modelling[1] but are relatively inflexible. On the other hand, feedback delay networks are computationally efficient and easily modulated. The latter properties are important considerations when designing a reverb effect meant to act as an instrument in its own right [11].

An important concern of ours is the musicality/playability of the reverb, especially with respect to real-time manipulation of perceptually relevant qualities. At the same time, we desire a model that can accurately simulate real spaces[2]. These requirements led us to eschew the traditional convolution-based reverb in favor of a fully parametric approach. In particular, we have chosen to adopt a modal reverb architecture [6] because the mapping of modes to perceptually important parameters (room size, decay time), is relatively straightforward, and because the parameters of a modal filter bank can be stably modulated at audio-rate. Recent work has also demonstrated a variety of interesting techniques that can be used with modal filter banks for pitch processing, time-scaling, and distortion [12].

1.1. Previous work

Although modal architectures for reverb processing are relatively recent [6], similar techniques have been used in other contexts for quite some time. See for example: Laroche's model of heavily damped percussive sounds [13]; The source-filter piano model of Meillier et al [14]; Bank's instrument body model [8]; Paatero et al.'s modelling of loudspeaker responses [15]; and, Sirdey et al's modal analysis of impact sounds [16];

Within the realm of reverb effects several works address the estimation of modal parameters, including: the frequency zooming-ARMA model of Karjalainen et al. [7][17]; Abel et al.'s modal reverberator [6]; Maestre et al.'s pole optimization algorithm [18]; the Gabor ESPRIT model of Sirdey et al. [19]; Schoenle et al.'s model of room responses [20]; and Hashemgeloogerdi et al.'s work on subband Kautz-filter modelling [21].

1.2. Contributions

A particular problem with modal modeling of reverb is the high density of modes exemplary of real room responses. After a short duration, and above the Schroeder frequency, both the echo and modal density become so dense as to make estimation of explicit modes very difficult [22]. Even if we had access to these parameters, running a modal filter bank with more than a few thousand modes would unreasonably tax a typical CPU.

In order to confront the problem of modal estimation for very dense impulse responses we have chosen to use a high-resolution,

* For Eventide Inc.

[1] For a fixed source-listener positioning.

[2] In the same sense as a convolutional reverb.

parametric estimator: the ESPRIT algorithm [23] [24]. Due to its parametric nature, ESPRIT, does not suffer from the same resolution limitations encountered with Fourier transform-based estimators, e.g., [6]. Other works applying ESPRIT to estimate modal parameters include [25] [26] [19].

A difficulty with ESPRIT is that it becomes computationally intractable for very dense and very long responses, like those typically encountered for real rooms. For this reason, we have chosen to use a subband approach, which has several critical benefits as discussed in section 4.

In order to prune mode counts down to a realizable number for synthesis with a modal filter bank, our work presents an approach to reduce the model order using the K-means algorithm.

We also discuss a technique for managing early reflections, which are not always easy to model using a small number of modes.

1.3. Outline

The remainder of this paper is laid out as follows. Section 2 describes the synthesis model of the modal reverberator. Section 3 gives an overview and derivation of the ESPRIT algorithm. Section 4 gives an explanation of the subband modifications we've made to make ESPRIT tractable for such a large problem. Section 5 is a brief word on estimating the model order. Section 6 introduces our algorithm for fitting the initial magnitude and phase parameters of the modal reverberator. Section 7 shows how we reduce the number of modes while maintaining perceptual accuracy, while Section 8 describes an extension to handle early reflections. Section 9 describes 3 experiments we ran comparing this method to a ground truth, another algorithm, and with and without the special early reflection handling. Finally Section 10 shares conclusions and Section 11 contains references.

2. THE MODAL MODEL

A starting point for this work is the assumption that a measured room response, $h[n]$, can be perfectly modeled using a linear digital filter with a rational z-transform

$$H(z) = \frac{B(z)}{A(z)} = \frac{\sum_{k=0}^{N} b_k z^{-k}}{\sum_{k=0}^{M} a_k z^{-k}} \tag{1}$$

the poles of which correspond to roots of the polynomial $A(z)$. Using long division, followed by partial fraction expansion, we can re-write $H(z)$ as [27]:

$$H(z) = \underbrace{\sum_{k=0}^{N-M} B_k z^{-k}}_{H_{FIR}(z)} + \underbrace{\sum_{k=1}^{M} \frac{A_k}{1 - z_k z^{-1}}}_{H_{Modal}(z)} \tag{2}$$

which represents an FIR filter in parallel with a bank of 1-pole filters that define the resonant modes of the system. In the special case $N < M$, the FIR part disappears and $H(z) = H_{Modal}(z)$. We will assume this is the case for the time-being, and revisit the estimation of $H_{FIR}(z)$ in section 8.

Taking the inverse z-transform of $H(z) = H_{Modal}(z)$ gives

$$h[n] = \sum_{k=1}^{M} h_k[n] = \sum_{k=1}^{M} A_k z_k^n \tag{3}$$

assuming the impulse response is stable and causal. When the poles occur in complex conjugate pairs, the time-domain view of the modal filter bank represents an exponentially damped sinusoidal (EDS) model. The complex amplitudes $A_k = e^{\alpha_k + j\phi_k}$ define the initial magnitude and phase of each damped sinusoid $z_k^n = e^{(d_k + jw_k)n}$.

Given this model two goals remain: i) estimate the model order, M; ii) estimate the model parameters: initial magnitude, initial phase, frequency, and damping. The model order should be as small as possible, while still maintaining perceptual transparency of the impulse response.

3. ESPRIT

The ESPRIT algorithm can be used to find the frequency and damping parameters for the EDS model in equation (3). The seminal ESPRIT reference is [23], however it focuses on direction-of-arrival estimation for antenna arrays. A more recent reference that focuses specifically on audio signal processing is [24]. The ESPRIT algorithm is briefly described below.

First, we collect L samples of the impulse response $h[n]$ into a vector $\mathbf{h}$. We can then re-write the EDS model from (3) using vector matrix notation as follows

$$\mathbf{h} = \mathbf{E}\mathbf{a} \tag{4}$$

where $\mathbf{E}_{nk}$ and $\mathbf{a}_k$ correspond to z_k^n and A_k, respectively. Using the delay property:

$$z_k^{n+R} = z_k^R z_k^n \tag{5}$$

we can write the EDS model for the Hankel matrix $\mathbf{H}_{nk} = h[n+k]$ (consisting of delayed copies of $\mathbf{h}$) as

$$\mathbf{H} = \mathbf{E}\mathbf{A}\mathbf{E}^T \tag{6}$$

where $\mathbf{A}_{kk} = A_k$ is a diagonal matrix containing the complex amplitudes. The superscripts T and H indicate the matrix transpose and Hermitian transpose, respectively. The columns of $\mathbf{H}$ lie in the M-dimensional *signal space*, spanned by the modal vectors, i.e., the columns of $\mathbf{E}$. Although these are unknown, we can find another set of vectors that span the signal space via a singular value decomposition (SVD) of $\mathbf{H}$

$$\mathbf{H} = \mathbf{U}\mathbf{\Lambda}\mathbf{V}^T \tag{7}$$

The column vectors of $\mathbf{U}$ are, in general, different from the signal vectors, however, they are related by an unknown linear transform $\mathbf{T}$ (a rotation and scaling)

$$\mathbf{E} = \mathbf{U}\mathbf{T} \tag{8}$$

The *rotational invariance* property of complex exponentials can now be invoked to determine the modal frequencies and dampings. Mathematically, the rotational invariance property states that

$$\mathbf{E}_{\uparrow} = \mathbf{E}_{\downarrow}\mathbf{D} \tag{9}$$

where $\mathbf{E}_{\uparrow}$ signifies deleting the first row of $\mathbf{E}$, $\mathbf{E}_{\downarrow}$ signifies deleting the last row of $\mathbf{E}$, and $\mathbf{D} = \text{diag}(z_0, z_1, \ldots, z_M)$. Substituting (8) into (9) and performing some algebra gives

$$(\mathbf{U}\mathbf{T})_{\uparrow} = (\mathbf{U}\mathbf{T})_{\downarrow}\mathbf{D} \tag{10}$$

$$\mathbf{U}_{\uparrow}\mathbf{T} = \mathbf{U}_{\downarrow}\mathbf{T}\mathbf{D} \tag{11}$$

$$\mathbf{U}_{\uparrow} = \mathbf{U}_{\downarrow}\underbrace{\mathbf{T}\mathbf{D}\mathbf{T}^{-1}}_{\mathbf{\Phi}} \tag{12}$$

$$\mathbf{\Phi} = (\mathbf{U}_{\downarrow}^H\mathbf{U}_{\downarrow})^{-1}\mathbf{U}_{\downarrow}^H\mathbf{U}_{\uparrow} \tag{13}$$

The matrix $\mathbf{\Phi}$ is computed using the Moore-Penrose pseudo inverse, since the matrix $\mathbf{U}$ is not typically square. The eigenvalues of $\mathbf{\Phi}$ are the complex modes $(z_1, z_2, \ldots, z_M)$, which can be recovered from an eigenvalue decomposition (EVD). Summarizing, the steps in the ESPRIT algorithm are:

1. Compute the signal space $\mathbf{U}$ [equation (7)]
2. Compute $\mathbf{\Phi}$ using the pseudo inverse [equation (13)]
3. Compute the complex modes from an EVD of $\mathbf{\Phi}$

4. SUBBAND PROCESSING

As alluded to previously, it is difficult to apply ESPRIT on long signals with high model orders because its complexity scales like $\mathcal{O}(LM(M + log(L)))$ [24].

One way to make ESPRIT tractable is to apply a divide and conquer approach. This can be done by passing the input through a filter bank to divide the input into a set of narrow subbands. There are four main benefits to this approach:

1. Since each subband has a narrow passband, we can safely assume that each subband contains a small number of significant modes. This in turn reduces the ESPRIT model order, M;
2. Using a suitable filter bank, we can downsample each subband without significant aliasing, which greatly reduces the amount of data, L, we need to consider when computing the SVD of the Hankel matrix;
3. Downsampling increases the distance between closely spaced modes, making them potentially easier to identify [7];
4. When using complex filters we can reduce the ESPRIT model order by a factor of 2 when analyzing real signals. During synthesis the complex conjugate modes can be restored to create a real impulse response.

Taken together, these aspects make it possible to apply ESPRIT to long IRs with potentially tens of thousands of modes. This approach was demonstrated in [19] using the Gabor transform and a similar idea was presented earlier by Laroche (using Prony's method instead of ESPRIT) [13].

We have experimented with three different filter bank architectures: the Gabor transform [19], the alias-free pyramidal filter bank described in [28], and the Audio FFT filter bank described in [29]. We currently use the Audio FFT filter bank in our analysis algorithm because it can be used to specify an arbitrary set of non-uniformly spaced subbands.

The r^{th} subband is produced by filtering the input with a causal N-tap FIR filter $g_r[n]$:

$$y_r[n] = h[n]*g_r[n] = \begin{cases} \sum_{k=1}^{M} \alpha_k \sum_{l=0}^{n} g_r[l] z_k^{n-l}, & \text{if } n < N-1 \\ \sum_{k=1}^{M} \hat{\alpha}_{kr} z_k^n, & \text{if } n \geq N-1 \end{cases} \tag{14}$$

where

$$\hat{\alpha}_{kr} = \alpha_k s_{kr} \tag{15}$$

$$s_{kr} = \sum_{l=0}^{N-1} g_r[l] z_k^{-l} \quad \text{(constant w.r.t. } n\text{)} \tag{16}$$

The first $N-1$ samples of the output $y_r[n]$ represent a start-up transient, which does not exhibit an EDS behavior. After the start-up transient dies out, however, each subband once again follows an EDS model, with the addition of a scaling factor s_{kr} that can be subsumed into the magnitude and phase for the current subband. For this reason, we ignore the first $N-1$ samples from each filter bank channel when applying ESPRIT on subbands. In our experience, this operation reduces the bias in the ESPRIT frequency and damping estimates. On the other hand, modes that have decay times comparable to the subband filter lengths cannot be accurately estimated.

For modes with center frequencies lying in the stopband of the r^{th} channel filter s_{kr}[3] should be negligibly small, allowing us to effectively ignore these modes in the current subband.

The Audio FFT filter bank's channel filters have been designed using the *window* method. It was demonstrated by [30] how the window method can be used to design perfect non-uniform reconstruction filter banks. We first choose R brickwall filters such that the sum of channel responses is unity

$$\sum_{r=1}^{R} G_r(e^{j\omega}) = 1 \tag{17}$$

where G_r is the frequency response of the r^{th} subband. This requirement is easily met by partitioning the frequency domain into a set of non-overlapping bands. Taking the inverse DTFT shows that

$$\sum_{r=1}^{R} G_r(e^{j\omega}) = 1 \iff \sum_{r=1}^{R} g_r[n] = \delta[n] \tag{18}$$

This set of filters is perfect reconstruction since we can recover the input signal $x[n]$ by adding together the subband responses, i.e.,

$$\sum_{r=1}^{R} y_r[n] = \sum_{r=1}^{R} x[n] * g_r[n] \tag{19}$$

$$= x[n] * \left(\sum_{r=1}^{R} g_r[n] \right) = x[n] * \delta[n] = x[n]. \tag{20}$$

However, due to the brickwall response of the channel filters each impulse response, g_r, is an IIR filter. Using the window method each channel IR is truncated via multiplication with a short window, creating an FIR filter. Using an N-tap window, $w[n]$, the r^{th} channel IR becomes $\hat{g}_r[n] = w[n]g_r[n]$. This set of filters is *still* a perfect reconstruction, if $w[0]$ is normalized to 1 since

$$\sum_{r=1}^{R} w[n]g_r[n] = w[n] \sum_{r=1}^{R} g_r[n] = w[n]\delta[n] = w[0]\delta[n] \tag{21}$$

Time-domain multiplication by $w[n]$ results in a convolution between the ideal channel filter and the window in the frequency-domain: $G_r(e^{j\omega}) * W(e^{j\omega})$. This results in a frequency-domain spreading of the filters, causing the filter responses to overlap in frequency. Figure 1 illustrates an example of this type of filter bank. The region marked as *partition* indicates the original boundaries of the ideal brickwall filter, and the region marked as *passband* shows the widened filter response due to the windowing. This particular filter bank was designed using a Chebychev window as suggested in [29].

[3] We recognize s_{kr} as the z-transform of the r^{th} subband filter evaluated at the k^{th} pole location.

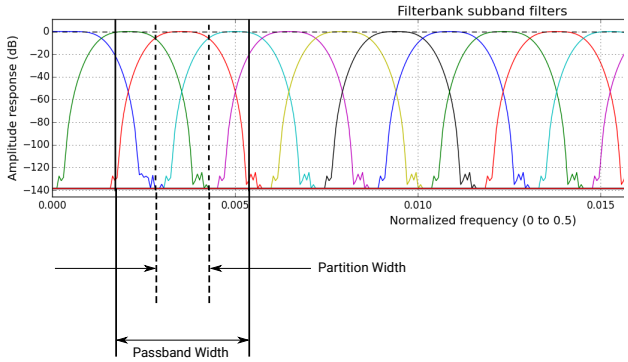


Figure 1: Filter bank design

When performing ESPRIT on subbands we can leverage the design of our filter bank in order to automatically prune out irrelevant modes. We first estimate how many modes are in each subband's *passband* as described in section 5 below. We then run ESPRIT using this model order. After ESPRIT returns we can safely discard any modes with center frequencies outside of the *partition*. We can do this because the partition perfectly divides the frequency spectrum into non-overlapping bands. Modes that do not lie in the current partition must belong to a neighboring partition (and therefore they should be estimated in the subband they lie closest to).

5. ORDER ESTIMATION

An inherent difficulty with parametric estimators lies in the specification of the model order—in our case the number of modes to estimate in each subband. There exist a few techniques that attempt to automatically estimate the model order based on information theoretic criteria, namely [31] and [32]. We have implemented these techniques, but found they did not perform particularly well for high model orders, e.g., more than 20 or so modes. Therefore, we have resorted to a simple model order selection algorithm based on peak picking from the discrete Fourier spectrum. We multiply the number of peaks detected by a *relaxation factor* greater than or equal to 1, recognizing the fact that some modes may not lead to a distinct peak in the sampled spectrum, or may be replicated (e.g., in the cases of two-stage and non-exponential decay). In practice, overestimating the model order does not usually pose a problem, because modes selected from the *noise subspace* generally have very small magnitudes.

6. MAGNITUDE AND PHASE ESTIMATION

After the modes z_k^n in each subband have been estimated using ESPRIT we must estimate the the complex amplitudes A_k. This can be done by minimizing the approximation error

$$\arg\min_{\mathbf{a}} ||\mathbf{h} - \mathbf{E}\mathbf{a}||_2^2 \tag{22}$$

A closed form solution to equation (22) is

$$\mathbf{a} = (\mathbf{E}^H\mathbf{E})^{-1}\mathbf{E}^H\mathbf{h} \tag{23}$$

However, this requires the inversion of a matrix with M^2 entries, which becomes very slow once the number of modes M exceeds a few thousand or so. We have experimented with conjugate gradient decent (which does not require a matrix inversion) to iteratively solve equation (22). This works well, but is still fairly slow once the number of modes exceeds several thousand.

Owing to Parseval's theorem, equation (22) can also be tackled in the frequency domain:

$$\arg\min_{\mathbf{a}} ||\mathbf{h} - \mathbf{E}\mathbf{a}||_2^2 = \arg\min_{\mathbf{a}} ||\check{\mathbf{h}} - \check{\mathbf{E}}\mathbf{a}||_2^2 \tag{24}$$

where $\check{\mathbf{h}}$ and $\check{\mathbf{E}}$ are the discrete Fourier transforms of $\mathbf{h}$ and the columns of $\mathbf{E}$, respectively. Note that each column of $\check{\mathbf{E}}$ can be computed analytically using the geometric series

$$\check{\mathbf{E}}_k[l] = \sum_{n=0}^{N-1} z_k^n e^{-j2\pi nl/N} \tag{25}$$

$$= \frac{1 - z_k^N}{1 - z_k e^{-j2\pi nl/N}} \tag{26}$$

In order to speed up the magnitude and phase estimation we once again resort to a divide and conquer approach. In particular, given a spectral filter $\mathbf{F}_k$ we can estimate the complex amplitudes of a subset of modes

$$\arg\min_{\mathbf{a_i}, i\in I_k} ||\mathbf{F}_k\check{\mathbf{h}} - \mathbf{F}_k\check{\mathbf{E}}\mathbf{a}||_2^2 \tag{27}$$

Modes that have minimal overlap with the filter $\mathbf{F}_k$ can be effectively ignored by removing columns from $\check{\mathbf{E}}$. Furthermore, we only need to minimize the norm in equation (27) over frequencies that fall in the passband of $\mathbf{F}_k$.

Using the DTFT we can calculate the 3dB bandwidth of the m^{th} mode to be

$$b_m = \arccos(2 - 0.5 * (e^{d_m} + e^{-d_m}))N/(2\pi) \tag{28}$$

where d_m is the damping factor and N is the DFT length. For the k^{th} subband we estimate the magnitude and phase of any modes for which the range $[\omega_m - b_m/2, \omega_m + b_m/2]$ intersects with the passband of the k^{th} spectral filter.

This procedure is applied repeatedly using a set of spectral filters $\{\mathbf{F}_k\}$ designed to completely cover the audible spectrum. This algorithm is much faster than any of the above techniques, and can be performed in parallel on architectures with multiple cores.

7. MODEL COMPRESSION

As mentioned in the introduction, in order to limit the CPU usage of a real-time modal reverberator we must restrict the total number of modes to no more than a few thousand. Subband ESPRIT routinely estimates upwards of 5000-10000 modes for real and dense IRs, meaning we require a strategy to reduce the overall number of modes used, ideally without sacrificing sound quality.

Luckily, it is possible to heavily compress our model by taking advantage of limitations in the human auditory perception system. In particular, it has been found that dramatically lower modal densities (compared to physically reality) can be used to generate perceptually accurate late reverberation. Therefore, we have developed a number of ad-hoc strategies to reduce the size of our modal filter bank

Following [18] we first partition the frequency spectrum into uniform bands on a Bark scale. We then divide our fixed modal

budget evenly across these bands. If some bands have fewer modes than they were allocated, the extra modes are reallocated among the remaining bands until no modes remain.

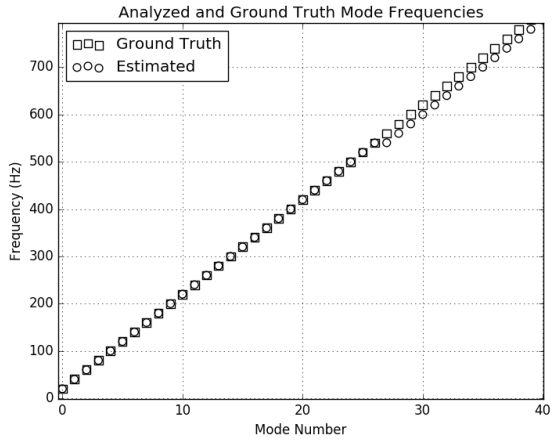
After the allocation step we have 2 numbers for each band i) M_a: the actual number of modes in each band (estimated using ESPRIT); and ii) M_d the desired number of modes in each band. While we could simply prune the extra modes $M_a - M_d$, this would change the distribution of modal frequencies in each band. Instead, we use the K-means algorithm to find a new set of M_d modes whose average distance from the estimated modes is minimized. An advantage of K-means is that is has the ability to 'average' the contributions of several modes by picking a new modal location that represents the center-of-gravity in a local neighborhood.

Empirically, we have found that the decay time estimates from ESPRIT exhibit a high degree of variance for real impulse responses. This in turn has a negative affect on the results of the K-means algorithm for small values of K (i.e., heavy model compression). In order to counteract this effect we smooth the decay time estimates from ESPRIT prior to running K-means. First, we apply a median filter to the decay times (after sorting them by frequency), which helps to eliminate outliers. Our median filter window starts with a length of 1 (at the boundaries) and grows until it reaches its maximum length (which is an algorithmic parameter in the range of 10 to several 100 modes). We have also experimented with weighted median filtering but no real benefit was noted. The median filtered decay times are then smoothed using a FIR lowpass filter to reduce the variance between nearby frequencies. It has been found that these three aspects: i) median filtering; ii) decay time smoothing; and, iii) K-means clustering are crucial for synthesizing *perceptually* good sounding IRs using a very small number of modes.

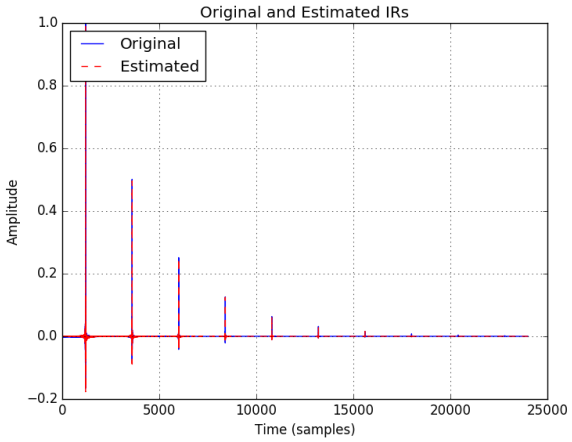
Once the model size is reduced the magnitude and phase of each mode (as discussed in section 6) must be re-estimated. In actuality, we always run the magnitude and phase estimation last, and hence only once.

We have applied a few additional ad-hoc strategies that should be noted. Immediately after running ESPRIT on each subband:

1. we discard any modes with a very low amplitude (estimated using least squares)
2. we discard any underdamped modes (which occur very rarely, and are unstable)



(a) Zoomed-in view of found modes.



(b) Synthesized and Estimated Irs.

Figure 2: Generated and found modes of a modally generated IR with 1000 modes.

8. HANDLING EARLY REFLECTIONS

Recall that our factorization of the rational transfer function in equation (2) included a parallel FIR path, $H_{FIR}(z)$. We can think $H_{FIR}(z)$ as modelling the early reflection portion of the reverb response. Fixing $H_{Modal}(z)$, the least squares solution for the FIR filter is $h_{FIR}[n] = h[n] - h_{Modal}[n]$ for $n \in [0, N)$. It is also possible to estimate the modal response from a delayed copy of the measured IR, i.e., $h[n - N_d]$. In this case

$$h_{FIR}[n] = \begin{cases} h[n] & \text{for} \quad n \in [0, N_d - 1] \\ h[n] - h_{Modal}[n] & \text{for} \quad n \in [N_d, N) \end{cases} \tag{29}$$

This later approach allows us to control the overlap between the responses which can lead to improved numerical performance as discussed in [33].

Before we can estimate the FIR part, however, we require some way to determine the tap-length of the FIR filter, N. In some cases, we have found that excluding the FIR part completely is a viable option, in which case we take $N = 0$. However, when an impulse response has prominent early reflections the modal synthesis algorithm may require an unreasonably large number of modes, M, to produce a good reconstruction on its own. We believe these unreasonably high mode counts originate from the EDS model's inability to efficiently model time sparsity[4]. A significant number of modes is required to build up the constructive/destructive interference pattern needed to model the sparsity between distinct echoes. In these situations we have implemented the early reflections using an FIR filter whose length, N, is estimated using Abel and Huang's echo density estimator [34].

[4]Indeed, the density of a Dirac comb in the time-domain is inversely proportional to its density in the frequency-domain.

9. EXPERIMENTS AND RESULTS

To validate the methods presented in this paper we conducted experiments with synthetic impulse responses having known modes, and performed several MUSHRA style listening tests. The recordings used in all of the listening tests are available online[5].

9.1. Synthetic Modal Impulse Response

In order to verify that the subband ESPIRIT analysis algorithm correctly identifies the true modes of an impulse response, we ran it on synthetically generated modal impulse responses with known sets of modes. The synthetic impulse responses were generated by adjusting the distribution of modes over frequency, the number of modes, and the decay times and magnitudes of the modes.

Figure 2 shows a plot of the known mode frequencies, obtained by spacing 1000 modes with a decay time of 0.5 seconds and magnitude of 1.0 linearly across the frequency spectrum up to 20kHz, as well as the modal frequencies detected by our subband ESPRIT analysis. Figure 2 aslo shows a plot of the original impulse response, and the one generated with modes found by subband ESPRIT. We can make the following observations: subband ESPRIT does indeed find the true modes of the impulse response, and it also finds a variety of spurious, or non-existent, modes. Note that even though the algorithm has added an extra mode at mode number 26, the subsequent mode frequencies are still correct. The addition of these spurious modes is, in part, due to the purposeful over-estimation of mode counts in the algorithm as previously discussed.

We can calculate the error between the known and estimated modal parameters by pairing the known and detected modes that are closest in frequency.

$$l_k = \arg\min_j(||f[k] - f_{\text{est}}[j]||_2) \tag{30}$$

$$e_f[k] = f[k] - f_{\text{est}}[l_k] \tag{31}$$

$$e_d[k] = d[k] - d_{\text{est}}[l_k] \tag{32}$$

Where l_k is the index of the detected mode that is closest to the k^{th} known mode in frequency. The mean and standard deviation of the error in the decay time estimates, e_d, are 0.000858 and 0.008301 *seconds* respectively. Similarly, we can calculate the mean and standard deviation of the errors between the known and found mode frequencies, e_f. These are $0.002329Hz$ and $0.015249Hz$, respectively. As a result of the close match between the modal parameters, the impulse response synthesized using the found modes is nearly identical to the one synthesized using the known modes. Comparing the two IRs, we find that the Mean Squared Error (MSE) between the two is -120.8147dB.

9.2. Monophonic Real Room Impulse Response

In order to compare our system with another state-of-the-art method, we chose to process the same impulse response presented in Maestre et al. [18] using mode counts of 400, 800, and 1800. In an effort to make the comparison fair, we did not include an FIR model of the early reflections in our model. We used the web-MUSHRA software [35] to administer a standard listening test in which users were asked to rate the quality of the modeled IRs with respect to a reference. A total of 12 users participated and there was no time-limit for the task. Figure 3 shows a box plot of the collected data, from which we may draw several conclusions. Firstly, the two algorithms perform quite similarly to one another. At low mode counts, our model was ranked slightly higher on average, whereas at high mode counts, Maestre et al.'s model was ranked higher. In our view, both models seem to impart very subtle artifacts to the impulse responses, however, test participants seemed to judge these artifacts differently depending on the model order.

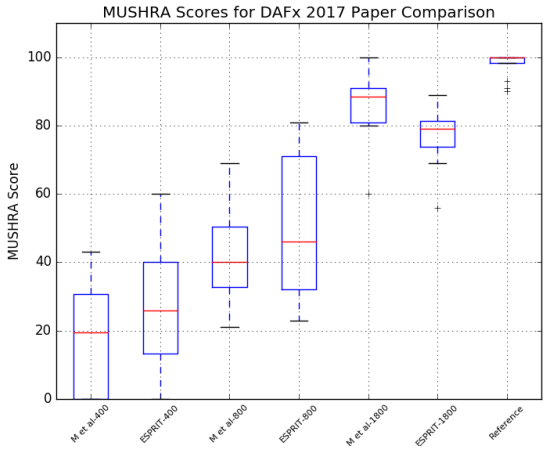


Figure 3: Listening test results I

Because the artifacts present in the synthesized impulse responses might manifest themselves differently when convolved with a source, we chose to perform a second test comparing our results with the results of Maestre et al. Using the same mode counts as before, listeners were asked to compare impulse responses which had been convolved with a source. The results of Maestre et al., and the dry source material, were obtained from their supplemental website[6]. Figure 4 shows the results of this test based on 9 users. Comparing Figures 3 and 4, two important observations stand out. Firstly, the scores of each individual response are, on average, higher than in the previous test and second, while our model was rated higher for a mode count of 800 in the previous test, the models of Maestre et al. were rated higher in this test.

9.3. Early Reflection Improvement with Parallel Synthesis Model

Figure 5 shows MUSHRA listening test results where 12 expert listeners rated the quality of differing subband ESPIRIT syntheses (and a hidden reference) with respect to a reference IR. The syntheses vary by model type, either pure modal or the FIR+modal model from Section 8, and the number of modes. In this experiment the reference is an IR from the Hall algorithm on a Lexicon PCM 90 digital reverb unit. This particular IR has a rather long early reflection field measuring 482 ms, measured using the Abel and Huang echo density estimator from [34]. The RT60 of this IR was also comparably long, measuring around 3 secs.

Listeners overwhelmingly favored the parallel FIR+modal model over the pure modal model. This trend held at very high

[5] http://dgillespie.github.io/Corey/

[6] https://ccrma.stanford.edu/ esteban/modrev/dafx2017/

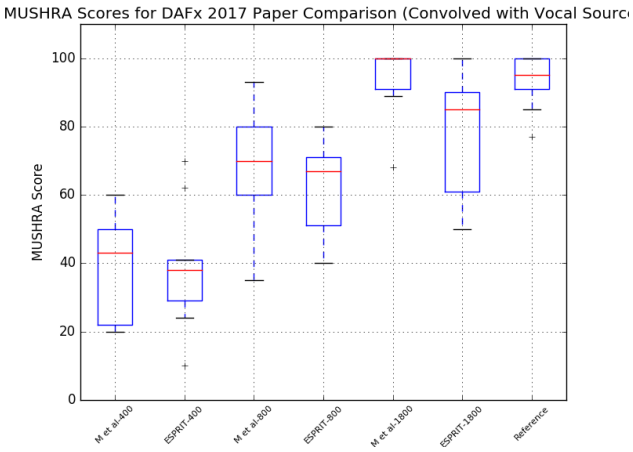


Figure 4: Listening test results II

mode counts for the pure modal model (12000). Even in this case, the parallel FIR+modal model using only 1500 modes rates significantly more similar to the reference. 12000 modes is taxing even on modern CPUs, while 1500 modes plus a fast convolution remains reasonably attainable.

We conclude that our hypothesis from Section 8 holds: it's difficult to guarantee accurate synthesis of significant early reflections in any efficient manner using a pure modal approach. Given that the early field is important in the perception and accurate synthesis of any given IR, the parallel model described in Section 8 can alleviate this particular issue. However, the parallel model is not without its drawbacks. The parallel model presents its own challenges for realtime audio effects like morphing, decay scaling, and size scaling because now the two parallel synthesis models must be parameterized in two differing domains and modulated in tandem to achieve perceptually pleasing and relevant results.

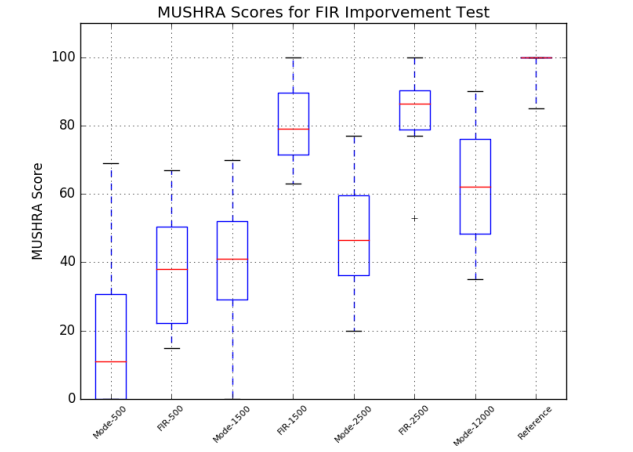


Figure 5: Listening test results III

10. CONCLUSION AND FUTURE WORK

In this paper we presented an end-to-end system for the modal analysis of real room impulse responses. Using the high-resolution ESPRIT estimator, we were are able to very accurately identify the frequency and damping parameters of impulse responses. Furthermore, we presented a number of strategies to i) make ESPRIT tractable on real-recordings; and, ii) yield models that can operate with fixed modal budgets. While our use of a subbband approach is not new, we have described several important considerations for practitioners of this method. This includes: trimming of the start-up transient, our approach to filter bank design, and our strategy for handling out-of-band modes. In order to reduce mode counts in the final model we presented a novel model compression algorithm based around K-means.

As mentioned previously, one interesting result of our listening tests was that, when convolved with a source, the results of Maestre et al. performed better than the method presented here. We have shown that the subband ESPRIT analysis method will find the correct modes of a system, given the correct model order, so it is likely that this error is introduced in our method of pruning excess modes. Because K-means is a clustering algortihm based on averages, the resulting set of modes after pruning may no longer be modes that were present in the original signal, but rather a new set of modes representing the average of several modes. Future work will surely focus on finding the best possible pruning method for reducing mode counts. This could include exploration of psychoacoustic-based methods, such as in [6]. In addition, it is worth noting that the rating of the generated impulse responses increased when they were convolved with a source. Presumably this is because some of the artifacts are masked. It would be of great value to know what artifacts are masked more heavily and vice versa. One could see the advantage in tuning the algorithm to be more accepting of artifacts that are more easily masked when used with source material.

11. REFERENCES

[1] Vesa Valimaki, Julian D Parker, Lauri Savioja, Julius O Smith, and Jonathan S Abel, "Fifty years of artificial reverberation," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 20, no. 5, pp. 1421–1448, 2012.

[2] Manfred R Schroeder, "Natural sounding artificial reverberation," *Journal of the Audio Engineering Society*, vol. 10, no. 3, pp. 219–223, 1962.

[3] John Stautner and Miller Puckette, "Designing multi-channel reverberators," *Computer Music Journal*, vol. 6, no. 1, pp. 52–65, 1982.

[4] Jean-Marc Jot and Antoine Chaigne, "Digital delay networks for designing artificial reverberators," in *Audio Engineering Society Convention 90*. Audio Engineering Society, 1991.

[5] Bo Holm-Rasmussena, Heidi-Maria Lehtonenb, and Vesa Välimäkib, "A new reverberator based on variable sparsity convolution," *Proc. of the 16th Int. Conference on Digital Audio Effects (DAFx-13)*, vol. 5, no. 6, pp. 7–8, 2013.

[6] Jonathan S Abel, Sean Coffin, and Kyle Spratt, "A modal architecture for artificial reverberation with application to room acoustics modeling," in *Audio Engineering Society Convention 137*. Audio Engineering Society, 2014.

[7] Matti Karjalainen, Paulo AA Esquef, Poju Antsalo, Aki Mäkivirta, and Vesa Välimäki, "Frequency-zooming arma modeling of resonant and reverberant systems," *Journal of the Audio Engineering Society*, vol. 50, no. 12, pp. 1012–1029, 2002.

[8] Balázs Bank, "Direct design of parallel second-order filters for instrument body modeling.," in *ICMC*, 2007.

[9] William G Gardner, "Efficient convolution without input/output delay," in *Audio Engineering Society Convention 97*. Audio Engineering Society, 1994.

[10] Craig J Webb and Stefan Bilbao, "Virtual room acoustics: A comparison of techniques for computing 3d-fdtd schemes using cuda," in *Audio Engineering Society Convention 130*. Audio Engineering Society, 2011.

[11] Tom Erbe, *Building the Erbe-Verb: Extending the Feedback Delay Network Reverb for Modular Synthesizer Use*, Ann Arbor, MI: Michigan Publishing, University of Michigan Library, 2015.

[12] Jonathan S Abel and Kurt James Werner, "Distortion and pitch processing using a modal reverberator architecture," in *Proceedings of the 18th International Conference on Digital Audio Effects (DAFx-15)*, 2015.

[13] Jean Laroche, "A new analysis/synthesis system of musical signals using prony's method-application to heavily damped percussive sounds," in *Acoustics, Speech, and Signal Processing, 1989. ICASSP-89., 1989 International Conference on*. IEEE, 1989, pp. 2053–2056.

[14] Jean Laroche and J-L Meillier, "Multichannel excitation/filter modeling of percussive sounds with application to the piano," *IEEE Transactions on Speech and Audio Processing*, vol. 2, no. 2, pp. 329–344, 1994.

[15] Tuomas Paatero and Matti Karjalainen, "Kautz filters and generalized frequency resolution: Theory and audio applications," *Journal of the Audio Engineering Society*, vol. 51, no. 1/2, pp. 27–44, 2003.

[16] Adrien Sirdey, Olivier Derrien, Richard Kronland-Martinet, and Mitsuko Aramaki, "Modal analysis of impact sounds with esprit in gabor transforms," in *14th International Conference on Digital Audio Effects (DAFx-11)*, 2011, pp. 1–6.

[17] Tuomas Paatero and Matti Karjalainen, "New digital filter techniques for room response modeling," in *Audio Engineering Society Conference: 21st International Conference: Architectural Acoustics and Sound Reinforcement*. Audio Engineering Society, 2002.

[18] Esteban Maestre, Jonathan S Abel, Julius O Smith, and Gary P Scavone, "Constrained pole optimization for modal reverberation," in *Proc. of the 5th International Conference on Digital Audio Effects (DAFx)*, 2017.

[19] Adrien Sirdey, Olivier Derrien, Richard Kronland-Martinet, and Mitsuko Aramaki, "Esprit in gabor frames," in *Audio Engineering Society Conference: 45th International Conference: Applications of Time-Frequency Processing in Audio*. Audio Engineering Society, 2012.

[20] M Schoenle, N Fliege, and U Zölzer, "Parametric approximation of room impulse responses by multirate systems," in *Acoustics, Speech, and Signal Processing, 1993. ICASSP-93., 1993 IEEE International Conference on*. IEEE, 1993, vol. 1, pp. 153–156.

[21] Sahar Hashemgeloogerdi and Mark F Bocko, "Precise modeling of reverberant room responses using wavelet decomposition and orthonormal basis functions," *Journal of the Audio Engineering Society*, vol. 66, no. 1/2, pp. 21–33, 2018.

[22] Jean-Marc Jot, Laurent Cerveau, and Olivier Warusfel, "Analysis and synthesis of room reverberation based on a statistical time-frequency model," in *Audio Engineering Society Convention 103*. Audio Engineering Society, 1997.

[23] Richard Roy and Thomas Kailath, "Esprit-estimation of signal parameters via rotational invariance techniques," *IEEE Transactions on acoustics, speech, and signal processing*, vol. 37, no. 7, pp. 984–995, 1989.

[24] Roland Badeau, Rémy Boyer, and Bertrand David, "Eds parametric modeling and tracking of audio signals," in *Proc. of the 5th International Conference on Digital Audio Effects (DAFx)*, 2002, pp. 139–144.

[25] Jean Laroche, "The use of the matrix pencil method for the spectrum analysis of musical signals," *The Journal of the Acoustical Society of America*, vol. 94, no. 4, pp. 1958–1965, 1993.

[26] Mathieu Lagrange and Bertrand Scherrer, "Two-step modal identification for increased resolution analysis of percussive sounds," in *Proc. Int. Conf. Digital Audio Effects (DAFx)*, 2008.

[27] Alan V Oppenheim, *Discrete-time signal processing*, Pearson Education India, 1999.

[28] NJ Fliege and U Zolzer, "Multi-complementary filter bank," in *Acoustics, Speech, and Signal Processing, 1993. ICASSP-93., 1993 IEEE International Conference on*. IEEE, 1993, vol. 3, pp. 193–196.

[29] Julius O Smith, "Audio fft filter banks," in *Proceedings of 12th International Conference on Digital Audio Effects (DAFx-09), Como*, 2009.

[30] Michael Goodwin, "Nonuniform filterbank design for audio signal modeling," in *Signals, Systems and Computers, 1996. Conference Record of the Thirtieth Asilomar Conference on*. IEEE, 1996, pp. 1229–1233.

[31] Mati Wax and Thomas Kailath, "Detection of signals by information theoretic criteria," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 33, no. 2, pp. 387–392, 1985.

[32] Roland Badeau, Bertrand David, and Gaël Richard, "Selecting the modeling order for the esprit high resolution method: an alternative approach," in *Acoustics, Speech, and Signal Processing, 2004. Proceedings.(ICASSP'04). IEEE International Conference on*. IEEE, 2004, vol. 2.

[33] Balázs Bank and Julius O Smith III, "A delayed parallel filter structure with an fir part having improved numerical properties," in *Audio Engineering Society Convention 136*. Audio Engineering Society, 2014.

[34] Jonathan S Abel and Patty Huang, "A simple, robust measure of reverberation echo density," in *Audio Engineering Society Convention 121*. Audio Engineering Society, 2006.

[35] Michael Schoeffler, Sarah Bartoschek, Fabian-Robert Stöter, Marlene Roess, Susanne Westphal, Bernd Edler, and Jürgen Herre, "webmushra: A comprehensive framework for web-based listening tests," *Journal of Open Research Software*, vol. 6, no. 1, 2018.

FAST MUSIC – AN EFFICIENT IMPLEMENTATION OF THE MUSIC ALGORITHM FOR FREQUENCY ESTIMATION OF APPROXIMATELY PERIODIC SIGNALS

Orchisama Das, Jonathan S. Abel, Julius O. Smith III

Center for Computer Research in Music and Acoustics,
Stanford University
Stanford, USA
[orchi|abel|jos]@ccrma.stanford.edu

ABSTRACT

Noise subspace methods are popular for estimating the parameters of complex sinusoids in the presence of uncorrelated noise and have applications in musical instrument modeling and microphone array processing. One such algorithm, MUSIC (Multiple Signal Classification) has been popular for its ability to resolve closely spaced sinusoids. However, the computational efficiency of MUSIC is relatively low, since it requires an explicit eigenvalue decomposition of an autocorrelation matrix, followed by a linear search over a large space. In this paper, we discuss methods for and the benefits of converting the Toeplitz structure of the autocorrelation matrix to *circulant* form, so that eigenvalue decomposition can be replaced by a Fast Fourier Transform (FFT) of one row of the matrix. This transformation requires modeling the signal as at least approximately periodic over some duration. For these periodic signals, the pseudospectrum calculation becomes trivial and the accuracy of the frequency estimates only depends on how well periodicity detection works. We derive a closed-form expression for the pseudospectrum, yielding large savings in computation time. We test our algorithm to resolve closely spaced piano partials.

1. INTRODUCTION

Sinusoidal parameter estimation is a classical problem with applications in radar, sonar, music, and speech, among others. When the frequencies of sinusoids are well resolved, looking for spectral peaks is adequate. It is shown in [1] that the maximum likelihood (ML) frequency estimate for a single sinusoid in Gaussian white noise is given by the frequency of the magnitude peak in the periodogram. The ML approach is extended and Cramer-Rao bounds are derived for multiple sinusoids in noise in a follow-on paper by the same authors [2]. Some other estimators are covered in [3].

For closely spaced sinusoidal frequencies, however, other approaches have been developed. Noise subspace methods are a class of sinusoidal parameter estimators that utilize the fact that the noise subspace of the measured signal is orthogonal to the signal subspace. Pisarenko Harmonic Decomposition [4] makes use of the eigenvector associated with the minimum eigenvalue of the estimated autocorrelation matrix to find frequencies. However, it has been found to exhibit relatively poor accuracy [3]. Schmidt [5] improved over Pisarenko with the MUSIC (MUltiple SIgnal Classification) algorithm which could estimate the frequencies of multiple closely spaced signals more accurately in the presence of noise. In MUSIC, a pseudospectrum is generated by projecting a complex sinusoid onto all of the noise subspace eigenvectors, defining peaks where this projection magnitude is minimum. This method is shown to be asymptotically unbiased. An enhancement to MUSIC, root-MUSIC, was proposed in [6]. It uses the properties of the signal-space eigenvectors to define a rational spectrum with poles and zeros. It is said to have better resolution than MUSIC at low SNRs. Similarly, another popular algorithm, ESPRIT [7], was invented by Roy et al. which makes use of the underlying rotational invariance of the signal subspace. The generalized eigenvalues of the matrix pencil formed by an auto-covariance matrix and a cross-covariance matrix gives the unknown frequencies. ESPRIT performs better than MUSIC, especially when the signal is sampled nonuniformly. More recently, another enhancement to MUSIC, gold-MUSIC [8] has been proposed which uses two stages for coarse and fine search, respectively.

One of the disadvantages of the MUSIC algorithm is its computational complexity. Typical eigenvalue decomposition algorithms are of the order $O(N^3)$ [9]. In this paper, we use the fact that, for periodic signals, the autocorrelation matrix is *circulant* when it spans an integer multiple of the signal's period. In this case, looking for the eigenvalues with largest magnitude is equivalent to looking for peaks in the power spectrum. We know in the circulant case that all noise eigenvectors are DFT sinusoids [10], and hence we can derive a closed-form solution when we project our search space onto the noise subspace, thereby reducing further the calculations required to find the pseudospectrum. Replacing eigenvalue decomposition in MUSIC with efficient Fourier transform based methods has been previously studied in [11] where the eigenvectors are derived to be some linear combinations of the data vectors, while maintaining the orthonormality constraint. In this paper, we take a different approach and show that for periodic signals, the MUSIC pseudospectrum can be exactly calculated using a sum of *aliased sinc* functions and its accuracy only depends on the accuracy with which the periodicity of the autocorrelation function is detected. We also propose speeding up MUSIC for non-periodic signals by initializing QR factorization for eigenvalue decomposition with the DFT matrix.

We test our algorithm to resolve closely spaced partials of the A3 note played on a piano. It is a well known fact that the strings corresponding to a particular piano key are slightly mistuned. Coupled motion of piano strings has been studied in detail by Weinreich in [12, 13]. There is a slight difference in frequency of the individual strings, giving rise to closely spaced peaks in the spectra. We show that FAST MUSIC can resolve two closely spaced peaks much faster than MUSIC.

The rest of this paper is organized as follows : Section 2 gives an outline of the MUSIC algorithm, Section 3 derives the FAST MUSIC algorithm, Section 4 describes the experimental results on a) an artificially synthesized signal containing two sinusoids with additive white noise and b) a partial of the A3 note played on the

piano which contains beating frequencies. We conclude the paper in Section 5 and delineate the scope for future work. Estimation of real-valued sine wave frequencies with MUSIC has been studied in [14]. For all derivations in this paper, we work with real signals, because we are interested in audio applications. This saves some time in computing the pseudospectrum since we know it will be symmetric. Our derivations can be easily extended to complex signals.

2. MUSIC

2.1. Model

We wish to estimate the parameters of a signal composed of additive sinusoids from noisy observations. Let $y(n)$ be the noisy signal, composed of a deterministic part, $x(n)$, made of r real sinusoids and random noise, $w(n)$. We assume that $w(n) \sim N(0, \sigma^2)$, and that $w(n)$ and $x(n)$ are uncorrelated. The sinusoidal phases ϕ_i's are assumed to be i.i.d. and uniformly distributed $\phi_i \sim U(-\pi, \pi)$.

$$\begin{aligned} y(n) &= \sum_{i=1}^{r} A_i \cos(\omega_i n + \phi_i) + w(n) \\ y(n) &= s(n) + w(n) \end{aligned}$$

In vector notation, the signal $\mathbf{y} \in \mathbb{R}^M$ can be characterized by the $M \times M$ autocorrelation matrix $K_y = \mathbb{E}(\mathbf{y}\mathbf{y}^T)$. For a zero-mean signal, the autocorrelation matrix coincides with the covariance matrix. Since this matrix is Toeplitz and symmetric positive-definite, its eigenvalues are real and nonnegative (and positive when $\sigma > 0$). We can perform an eigenvalue decomposition on this matrix to get a diagonal matrix Λ consisting of the eigenvalues, and an eigenvector matrix Q. The $2r$ eigenvectors corresponding to the $2r$ largest eigenvalues, Q_s, contain signal plus noise information, whereas the remaining *M-2r* eigenvectors, Q_w, only represent the noise subspace. Thus, we have the following relationships:

$$\begin{aligned} K_y &= K_s + K_w = K_s + \sigma^2 I \\ K_y &= Q \Lambda Q^H \\ K_y &= \begin{bmatrix} Q_s & Q_w \end{bmatrix} \begin{bmatrix} \Lambda' & 0 \\ 0 & \sigma^2 I_{M-2r} \end{bmatrix} \begin{bmatrix} Q_s^H \\ Q_w^H \end{bmatrix} \end{aligned} \tag{1}$$

2.2. Pseudospectrum Estimation

Let a vector of M harmonic frequencies be denoted as $\mathbf{b}(\omega) = [1, e^{j\omega}, e^{2j\omega} \cdots e^{(M-1)j\omega}]^T$. We project this vector onto Q_w, i.e., the subspace occupied by the noise (where there is no signal component). MUSIC defines the following pseudospectrum as a function of a set of ω's:

$$\begin{aligned} P(\omega) &= \frac{1}{\mathbf{b}(\omega)^H Q_w Q_w^H \mathbf{b}(\omega)} \\ P(\omega) &= \frac{1}{||Q_w^H \mathbf{b}(\omega)||^2} \end{aligned} \tag{2}$$

For a particular value of ω that is actually present in the signal, the sum of projections of **b** onto the eigenvectors spanning the noise subspace will be zero. This is because the subspace occupied by the signal is orthogonal to that occupied by noise since they are uncorrelated. Thus, we see that $P(\omega)$ will take on a very high value in such cases (theoretically infinite). In conclusion, we can find peaks in the function $P(\omega)$ and those will correspond to our estimated frequencies. Since the search space can consist of any number of densely packed frequencies, very closely spaced peaks can show up in the pseudospectrum. However, as the search-space grows, so does computational complexity.

3. FAST MUSIC

3.1. Deriving the autocorrelation matrix

In vector form, $\mathbf{y} \in \mathbb{R}^M$ can be written as:

$$\begin{aligned} \mathbf{y} &= S\mathbf{a} + \mathbf{w} \\ \mathbf{w} &\sim N(0, \sigma^2 I) \end{aligned} \tag{3}$$

$$\begin{bmatrix} y(n) \\ y(n-1) \\ \vdots \\ y(n-M+1) \end{bmatrix} = \begin{bmatrix} 1 & 0 & \cdots \\ \cos(\omega_1) & \sin(\omega_1) & \cdots \\ \vdots & \vdots & \cdots \\ \cos[(M-1)\omega_1] & \sin[(M-1)\omega_1] & \cdots \end{bmatrix} \times \begin{bmatrix} A_1 \cos(\omega_1 n + \phi_1) \\ A_1 \sin(\omega_1 n + \phi_2) \\ \vdots \\ A_r \sin(\omega_r n + \phi_r) \end{bmatrix} + \begin{bmatrix} w(n) \\ w(n-1) \\ \vdots \\ w(n-M+1) \end{bmatrix} \tag{4}$$

Since **y** is zero-mean, its covariance matrix is

$$K_y = \mathbb{E}(\mathbf{y}\mathbf{y}^T) = S K_a S^T + \sigma^2 I. \tag{5}$$

We now want to get K_y in terms of K_a. We have assumed $\phi_i \sim U(-\pi, \pi)$ (uniformly identically distributed random phase). We observe that every term of K_a is of the form $K_a(i,j) = \mathbb{E}[A_i \cos(\omega_i n + \phi_i) A_j \cos(\omega_j n + \phi_j)]$, or $\mathbb{E}[A_i \sin(\omega_i n + \phi_i) A_j \sin(\omega_j n + \phi_j)]$, or $\mathbb{E}[A_i \sin(\omega_i n + \phi_i) A_j \cos(\omega_j n + \phi_j)]$, or $\mathbb{E}[A_i \cos(\omega_i n + \phi_i) A_j \sin(\omega_j n + \phi_j)]$. All of these terms are zero, except the first two when $i = j$, i.e. $\mathbb{E}[A_i^2 \cos(\omega_i n + \phi_i)^2] = \mathbb{E}[A_i^2 \sin(\omega_i n + \phi_i)^2] = \frac{A_i^2}{2}$, which makes it a diagonal matrix:

$$K_a = \begin{bmatrix} \frac{A_1^2}{2} & 0 & \cdots & 0 & 0 \\ 0 & \frac{A_1^2}{2} & \cdots & 0 & 0 \\ \vdots & \vdots & \cdots & \vdots & \vdots \\ 0 & 0 & \cdots & 0 & \frac{A_r^2}{2} \end{bmatrix} \tag{6}$$

The autocorrelation matrix of the observed signal is given in (7). This is an $M \times M$ real, symmetric Toeplitz matrix.

3.2. For periodic signals

Under conditions to be specified, it is possible to replace the eigenvalue decomposition required in MUSIC by a Fast Fourier Transform (FFT). In this subsection, we derive the order of the autocorrelation matrix for which it is circulant instead of only Toeplitz. We also derive a closed-form expression for finding the pseudospectrum.

$$SK_aS^T + \sigma^2 I = \begin{bmatrix} \sum_i \frac{A_i^2}{2} + \sigma^2 & \sum_i \frac{A_i^2}{2}\cos\omega_i & \sum_i \frac{A_i^2}{2}\cos 2\omega_i & \cdots & \sum_i \frac{A_i^2}{2}\cos(M-1)\omega_i \\ \sum_i \frac{A_i^2}{2}\cos\omega_i & \sum_i \frac{A_i^2}{2} + \sigma^2 & \sum_i \frac{A_i^2}{2}\cos\omega_i & \cdots & \sum_i \frac{A_i^2}{2}\cos(M-2)\omega_i \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \sum_i \frac{A_i^2}{2}\cos(M-1)\omega_i & \sum_i \frac{A_i^2}{2}\cos(M-2)\omega_i & \cdots & \cdots & \sum_i \frac{A_i^2}{2} + \sigma^2 \end{bmatrix} \tag{7}$$

3.2.1. Circulancy of the autocorrelation matrix

We have seen that the autocorrelation matrix K_y is symmetric Toeplitz. However it is to be noted that for $k = 1, 2, ...$, if M is an integer such that $M = 2\pi n/\omega_i, n \in \mathbb{Z}^+$, then

$$\sum_{i=1}^{r} \frac{A_i^2}{2}\cos(M-k)\omega_i = \sum_{i=1}^{r} \frac{A_i^2}{2}\cos k\omega_i \tag{8}$$

If we choose M carefully, then the autocorrelation matrix may be written as :

$$\begin{bmatrix} \sum_i \frac{A_i^2}{2} + \sigma^2 & \sum_i \frac{A_i^2}{2}\cos\omega_i & \cdots & \sum_i \frac{A_i^2}{2}\cos\omega_i \\ \sum_i \frac{A_i^2}{2}\cos\omega_i & \sum_i \frac{A_i^2}{2} + \sigma^2 & \cdots & \sum_i \frac{A_i^2}{2}\cos 2\omega_i \\ \sum_i \frac{A_i^2}{2}\cos 2\omega_i & \sum_i \frac{A_i^2}{2}\cos\omega_i & \vdots & \sum_i \frac{A_i^2}{2}\cos 3\omega_i \\ \vdots & \vdots & \ddots & \vdots \\ \sum_i \frac{A_i^2}{2}\cos\omega_i & \sum_i \frac{A_i^2}{2}\cos 2\omega_i & \cdots & \sum_i \frac{A_i^2}{2} + \sigma^2 \end{bmatrix} \tag{9}$$

This matrix is circulant! Hence, its eigenvectors are given by the DFT sinusoids and its eigenvalues are the DFT coefficients of the first row [10]. The eigenvalues can be computed using an FFT algorithm when M is a power of 2 or highly composite. The relationship between eigenvalues of Toeplitz matrices and those of asymptotically equivalent circulant matrices have been studied in [15].

For example, if the signal consists of 3 sinusoids with frequencies $\frac{\pi}{2}$, $\frac{\pi}{4}$ and $\frac{\pi}{5}$ then the minimum order of M which will make the autocorrelation matrix circulant is given by $2\times LCM(2, 4, 5) = 40$. However, if any of the denominators is irrational, then the LCM does not exist, and hence no value of M will make the autocorrelation matrix circulant. Of course, in reality we do not know the frequencies and cannot determine M this way. However, we can instead detect when the autocorrelation corresponds to a signal that is periodic. If we can find the periodicity of the estimated autocorrelation function and set M to be that period, then the resulting autocorrelation matrix will be circulant. Since no signal is truly precisely periodic, this procedure can be viewed as introducing an approximation based on assuming the signal is periodic. Such a periodic/harmonic approximation is common when the underlying signal source is known to be a quasi periodic oscillator such in voiced speech, bowed strings, woodwinds, flutes, brasses, organs, and so on.

In this paper, we use the Average Magnitude Difference Function [16] to detect the period. We find all local minima in the AMDF and pick the period as the lowest minimum index which is smaller than its adjacent neighbors. We set M to be an integer multiple of the detected period. This gives us more data points for the FFT, thus increasing accuracy. It also comes at a higher cost, but the FFT is still orders of magnitude faster than eigenvalue decomposition, hence the trade-off is justified.

3.2.2. Searching over a large range of frequencies

Suppose we want to calculate the pseudospectrum for $N \geq M$ distinct frequencies $\omega_k = 2\pi\frac{k}{N}$ for $k = -\frac{N}{2}, \ldots, \frac{N}{2} - 1$ covering the range $[-\pi, \pi)$, i.e, the search space has N points. Each search space vector is

$$\mathbf{b}(k) = [1, e^{\frac{2\pi jk}{N}}, e^{\frac{4\pi jk}{N}} \ldots, e^{\frac{2\pi(M-1)jk}{N}}]^T. \tag{10}$$

The noise subspace consists of $M - 2r$ vectors. Instead of projecting on to the noise subspace, Q_w, we can make the computation easier by using the signal subspace, Q_s instead, which only has $2r$ vectors. This is because the noise subspace and the signal subspace are orthogonal complements and hence the following holds

$$Q_sQ_s^H + Q_wQ_w^H = I \tag{11}$$

The projection onto the noise subspace can be simplified as

$$\begin{aligned} ||Q_w^H\mathbf{b}||^2 &= \mathbf{b}^HQ_wQ_w^H\mathbf{b} \\ &= \mathbf{b}^H(I - Q_sQ_s^H)\mathbf{b} \\ &= \mathbf{b}^H\mathbf{b} - \mathbf{b}^HQ_sQ_s^H\mathbf{b} \\ &= ||\mathbf{b}||^2 - ||Q_s^H\mathbf{b}||^2 \end{aligned} \tag{12}$$

Since $\mathbf{b}(k)$ is a vector of length M consisting of complex exponentials of unit magnitude, $||\mathbf{b}(k)||^2 = M$. The matrix $Q_s \in \mathbb{C}^{M\times 2r}$ is composed of columns of signal eigenvectors, such that each column is denoted as

$$\mathbf{q}_s = \frac{1}{\sqrt{M}}[1, e^{\frac{2\pi jm_i}{M}}, e^{\frac{4\pi jm_i}{M}}, \cdots, e^{\frac{2\pi(M-1)jm_i}{M}}]^T \tag{13}$$

where m_i are the complex frequencies associated with the signal eigenvectors [10], i.e, the indices of the top $2r$ FFT magnitudes. The projection of $\mathbf{b}(k)$ onto the signal subspace can be written as :

$$\begin{aligned} Q_s^H\mathbf{b}(k) &= \frac{1}{\sqrt{M}}\begin{bmatrix} \sum_{p=0}^{M-1}\exp[2\pi jp(\frac{k}{N} - \frac{m_1}{M})] \\ \vdots \\ \sum_{p=0}^{M-1}\exp[2\pi jp(\frac{k}{N} - \frac{m_{2r}}{M})] \end{bmatrix} \\ &= \frac{1}{\sqrt{M}}\begin{bmatrix} e^{-\pi j(\frac{k}{N} - \frac{m_1}{M})(M-1)}\frac{\sin[\pi(\frac{k}{N} - \frac{m_1}{M})M]}{\sin[\pi(\frac{k}{N} - \frac{m_1}{M})]} \\ \vdots \\ e^{-\pi j(\frac{k}{N} - \frac{m_{2r}}{M})(M-1)}\frac{\sin[\pi(\frac{k}{N} - \frac{m_{2r}}{M})M]}{\sin[\pi(\frac{k}{N} - \frac{m_{2r}}{M})]} \end{bmatrix} \end{aligned} \tag{14}$$

The pseudospectrum can be approximated as:

$$\begin{aligned} P(k) &= \frac{1}{||\mathbf{b}(k)||^2 - ||Q_s^H \mathbf{b}(k)||^2} \\ &= \frac{1}{M - \frac{1}{M}\sum_{i=1}^{2r}\left[\frac{\sin[\pi(\frac{k}{N}-\frac{m_i}{M})M]}{\sin[\pi(\frac{k}{N}-\frac{m_i}{M})]}\right]^2} \\ &= \frac{1}{M - \sum_{i=1}^{2r}\left[asinc_M(\frac{k}{N}-\frac{m_i}{M})\right]^2} \end{aligned} \tag{15}$$

where $asinc$ stands for the *aliased sinc* function[1]. The pseudospectrum is independent of the data and only depends on the calculated period, M. At signal frequencies, when $\frac{k}{N} = \frac{m_i}{M}$, one *aliased sinc* term in the summation dominates and we can evaluate it using L'Hospital's rule.

$$\lim_{x\to 0} asinc_M(x) = M \tag{16}$$

Therefore, at signal frequencies, the pseudospectrum is theoretically infinite.

$$P(k) \approx \infty \quad if \quad \frac{k}{N} = \frac{m_i}{M} \tag{17}$$

For the special case of periodic signals, MUSIC is essentially equivalent to looking for the top $2r$ peaks in the power spectrum and using the positions of those peaks to form the signal space.

3.2.3. Algorithm Summary

1. Estimate the autocorrelation function (ACF) of the given signal.
2. Find the periodicity M of the ACF and take the FFT of its first M samples. This is equivalent to computing the power spectrum.
3. Sort the FFT magnitudes in descending order. The indices corresponding to the largest $2r$ magnitudes are the signal eigenvector frequencies.
4. Form search space vectors according to (10) with $k = -\frac{N}{2}$, $\ldots$, $\frac{N}{2} - 1$.
5. Calculate the pseudospectrum according to (15) and find $2r$ peaks in it.
6. Do parabolic interpolation on the peaks to get more accurate frequency estimates [17].

3.3. For non-periodic signals

Most signals in practical applications are non-periodic. In that case, these derivations do not hold exactly. However, we can still speed up the eigenvalue decomposition process. From (1), we can write the diagonal eigenvalue matrix as

$$\Lambda = Q^H K_y Q \tag{18}$$

The eigenvectors Q are usually estimated with QR factorization [9]. We can use the DFT matrix W as an initial value for QR factorization, which will ensure its convergence in fewer steps.

[1] https://ccrma.stanford.edu/~jos/sasp/Rectangular_Window.html

$$\Lambda + \epsilon = W^H K_y W \tag{19}$$

For exactly periodic signals ϵ is a null matrix. For approximately periodic signals, ϵ is a non-diagonal matrix with small entries. We can see that within some iterations W will converge to Q. The speed of convergence will depend on how close to being periodic the signal is.

4. EXPERIMENTS AND RESULTS

4.1. Synthesized signal

To compare FAST MUSIC with MUSIC, we tested a signal composed of cosines at frequencies 0.004 Hz and 0.005 Hz at $f_s = 1$ Hz and added normally distributed noise $w(n)$ at an SNR of 10dB.

$$y(n) = \cos(0.01\pi n) + 0.5\cos(0.008\pi n + \phi) + w(n) \tag{20}$$

To detect periodicity of the autocorrelation function, we need at least two periods of the signal. This signal has a periodicity of M = 1000 samples. Thus, we made the signal 2500 samples long. It is to be noted that the closer the frequencies in the signal, the larger will be its periodicity, and hence we will need more samples of data to accurately determine it.

To measure computation time, we compared various eigenvalue decomposition algorithms with Fast Fourier Transform algorithms for increasing orders of the autocorrelation matrix. The results can be seen in Figure 1. QR factorization is used to find eigenvalues and eigenvectors for MUSIC. QR factorization with Gram Schmidt orthogonalization is slow, symmetric tridiagonal QR with implicit Wilkinson shift is slightly faster whereas reduction to the Hessenberg form is fastest. More details about these algorithms can be found in [9]. The Fourier transform algorithms are orders of magnitude faster, with the DFT dominating at lower orders and self-sorting mixed radix FFT [18] and resampled split radix FFT [19] giving faster speeds at higher orders. It is to be noted that the order of the autocorrelation matrix for which circulancy is achieved is not likely to be a power of 2, and hence we cannot use the well-known radix-2 FFT algorithm. However, we can first resample the data to a power of 2 [20] and then apply a radix-2/split radix FFT algorithm or use a mixed-radix FFT algorithm on any composite order. All of these functions have been implemented in MATLAB. More efficient implementations can be done in C, where the FFT functions should overtake the DFT at much lower orders.

We also conducted 1000 Monte-Carlo simulations on the above example, with uniformly distributed random phase. We plotted the mean-squared errors vs SNR for MUSIC and FAST MUSIC, along with the Cramer-Rao bounds (CRB) as given in [21] in Figure 2. The order of the autocorrelation matrix for MUSIC is set to 200. For FAST MUSIC, the period is calculated for each simulation and found to be 1000 samples. The number of points in the search space is 2000. The poor performance of FAST MUSIC at low SNRs is due to the inaccuracy in periodicity detection. In Figure 2a, FAST MUSIC overtakes the CRB at high SNRs, where periodicity is detected accurately, hence MSE $= 0$. At high SNRs, FAST MUSIC also outperforms MUSIC. Increasing the order of the autocorrelation matrix would have improved the accuracy of MUSIC at a cost of high computational time, so we decided to work with a reasonable order of 200, while FAST MUSIC used 2000 samples in the autocorrelation function (an integer multiple of the period). As seen in Figure 2b, both methods have significant bias.

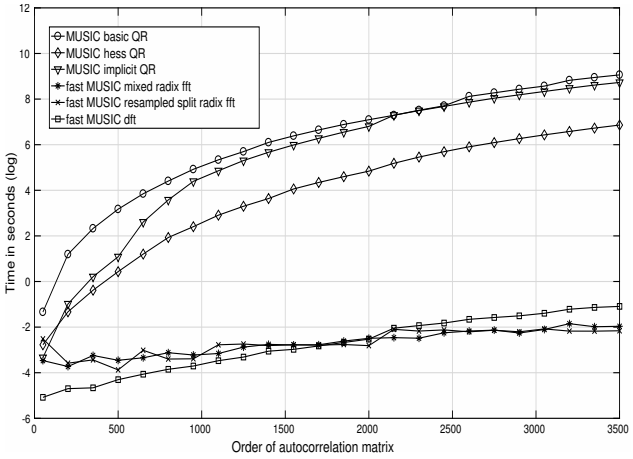
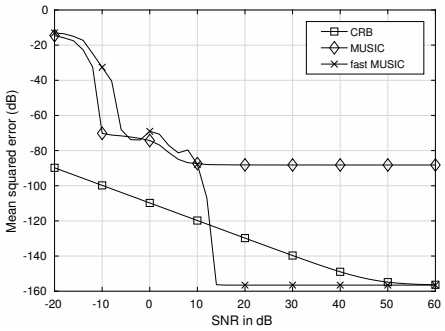
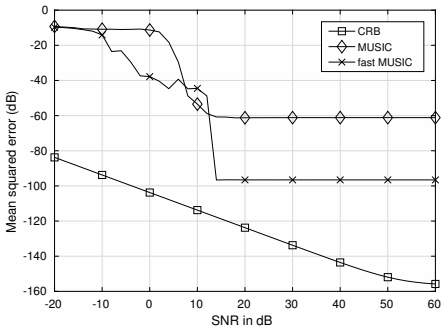


Figure 1: Computation time in log seconds versus matrix order

Bias can be reduced arbitrarily in FFT based peak finding methods [17] by increasing the amount of zero-padding, as well as by other methods [22]. We expect it to reduce similarly in FAST-MUSIC and MUSIC when the number of points in the search space is increased.



(a) Frequency = 0.004 Hz



(b) Frequency = 0.005 Hz

Figure 2: MSE vs SNR plots

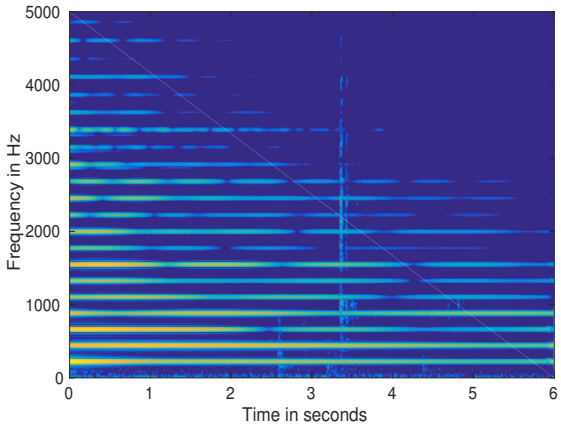


Figure 3: Spectrogram of piano note

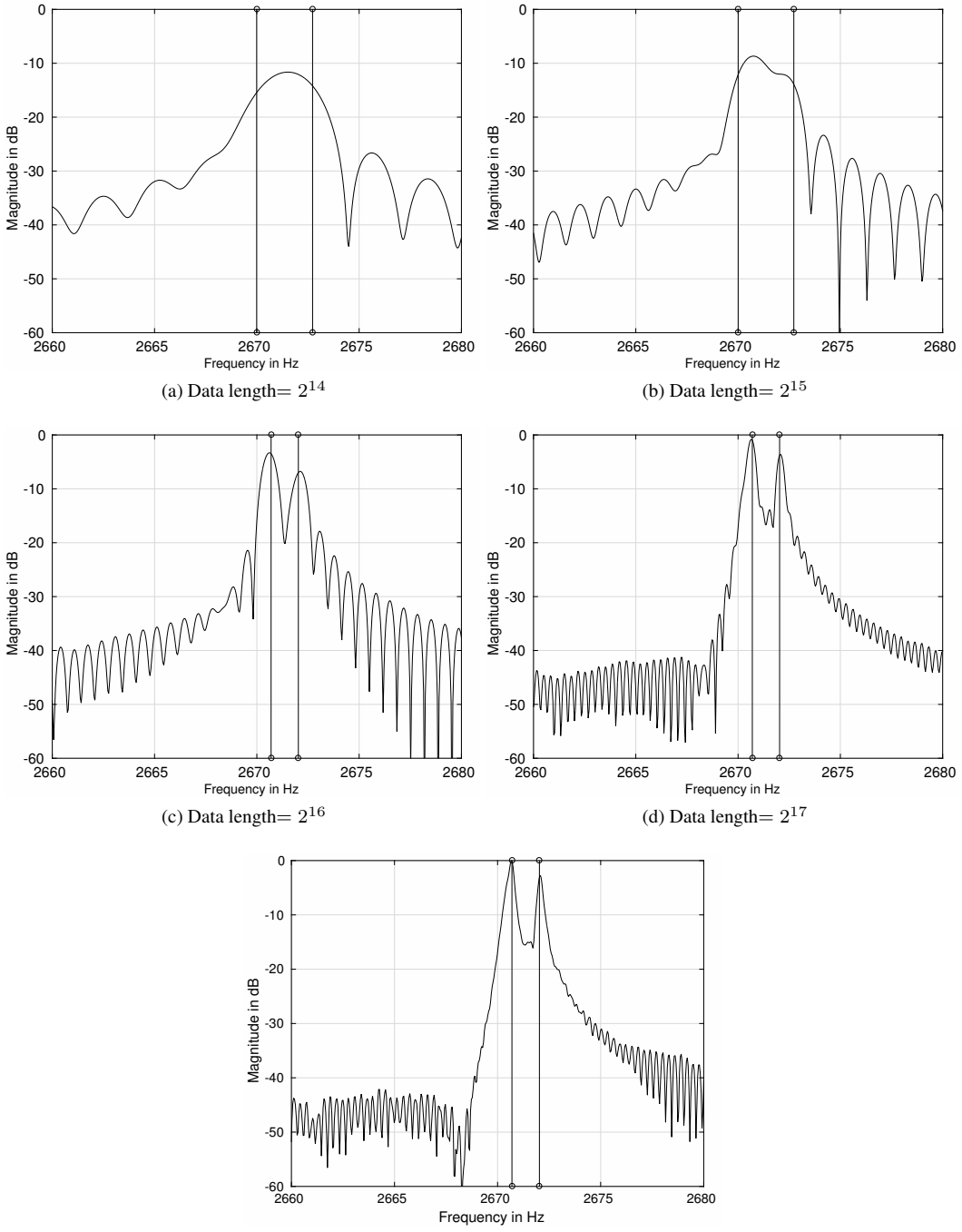
4.2. Piano data

We tested our algorithm on the A3 note played on the piano. The spectrogram of the steady state portion of the note is given in Figure 3. We can observe beating in some of the partials. We decided to work with the 11th partial, located close to 2600 Hz, where a beating of roughly 1 Hz is observed. We bandpass-filtered the signal using a 4th-order Butterworth filter with cut-off frequencies at 2400 Hz and 2900 Hz. We ran FFTs with the rectangular window and FAST MUSIC on different data lengths, as shown in Figure 4, where the vertical lines indicate the frequencies detected by FAST MUSIC. One disadvantage of using the rectangular window is high side lobe height as seen in Figure 4, but we compromise side lobe height for the narrowest main lobe width for the sake of best resolution. We see that for window size of 2^{14}, the FFT magnitude does not exhibit two separately discernible peaks at all, whereas FAST MUSIC provides two peak frequencies with some error. This is because we specify the number of sinusoids to be 2 in FAST MUSIC, whereas the FFT has no prior information about the number of peaks expected in the magnitude spectrum. For longer window sizes, both FFT and FAST MUSIC are able to resolve the two peaks with greater accuracy. One potential application is in piano tuning, where FAST MUSIC could be used to quickly resolve closely spaced peaks caused by the coupled motion of the piano strings.

5. DISCUSSION AND FUTURE WORK

In this paper, we have proposed a computationally efficient interpretation of the MUSIC algorithm for periodic signals that makes use of the peaks in the power spectrum. The autocorrelation matrix has been derived and approximated by a circulant matrix. This approximation has allowed us to replace computationally intensive eigenvalue decomposition algorithms with an FFT. We have subsequently derived a closed-form expression for searching over a range of frequencies. These modifications have yielded a significant improvement in computational speed. For non-periodic signals, we have proposed initialization of QR factorization with the DFT matrix to speed up eigenvalue decomposition.[2]

[2]The code and the simulations can be found at `https://github.com/orchidas/fast_MUSIC`



(a) Data length= 2^{14}

(b) Data length= 2^{15}

(c) Data length= 2^{16}

(d) Data length= 2^{17}

(e) Data length= 2^{18}

Figure 4: FFT magnitude plots and FAST MUSIC frequency estimates (vertical lines)

A key factor in the accuracy of FAST MUSIC is the precision in periodicity detection. If the period is off by a significant number of samples, the autocorrelation matrix is no longer circulant and FAST MUSIC falls apart. AMDF based periodicity detector is simple but time consuming, not foolproof and often yields wrong results if the number of lags in the autocorrelation function is very high. Ideally, a better method for periodicity detection should be used.

Another issue is finding the number of sinusoids present in a given signal, when not known a priori. To do so, one can look at the relative magnitude of the eigenvalues (power spectrum peak values in our case). This works well if the signal to noise ratio is

sufficiently high and the peak separation sufficient. More robust partitioning schemes have been used in [8]. Once the signal frequencies are known, the estimation of amplitudes is simple and can be done using linear least squares.

We found that the estimator mean squared errors for both FAST-MUSIC and MUSIC were dominated by bias at high SNRs. Future work should reduce or eliminate the bias so that the relative performance can be observed at high SNRs. FAST MUSIC also needs to be better evaluated with non-periodic signals. We have not tested its performance with non-periodic signals in this paper.

6. REFERENCES

[1] David C. Rife and Robert R. Boorstyn, "Single-Tone Parameter Estimation from Discrete-Time Observations," *IEEE Transactions on Information Theory*, vol. 20, no. 5, pp. 591–598, 1974.

[2] David C Rife and Robert R Boorstyn, "Multiple tone parameter estimation from discrete-time observations," *Bell Labs Technical Journal*, vol. 55, no. 9, pp. 1389–1410, 1976.

[3] Steven M Kay, "Modern spectral estimation," chapter 13. Prentice Hall, 1988.

[4] Vladilen F Pisarenko, "The retrieval of harmonics from a covariance function," *Geophysical Journal International*, vol. 33, no. 3, pp. 347–366, 1973.

[5] Ralph Schmidt, "Multiple emitter location and signal parameter estimation," *IEEE transactions on antennas and propagation*, vol. 34, no. 3, pp. 276–280, 1986.

[6] Arthur Barabell, "Improving the resolution performance of eigenstructure-based direction-finding algorithms," in *Acoustics, Speech, and Signal Processing, IEEE International Conference on ICASSP'83*. IEEE, 1983, vol. 8, pp. 336–339.

[7] Richard Roy and Thomas Kailath, "Esprit-estimation of signal parameters via rotational invariance techniques," *IEEE Transactions on acoustics, speech, and signal processing*, vol. 37, no. 7, pp. 984–995, 1989.

[8] Kaluri V Rangarao and Shridhar Venkatanarasimhan, "gold-music: A variation on music to accurately determine peaks of the spectrum," *IEEE Transactions on Antennas and Propagation*, vol. 61, no. 4, pp. 2263–2268, 2013.

[9] Peter Arbenz, "Lecture notes on solving large scale eigenvalue problems," chapter 4. ETH Zurich.

[10] Robert M Gray et al., "Toeplitz and circulant matrices: A review," *Foundations and Trends® in Communications and Information Theory*, vol. 2, no. 3, pp. 155–239, 2006.

[11] Juha T Karhunen and Jyrki Joutsensalo, "Sinusoidal frequency estimation by signal subspace approximation," *IEEE Transactions on signal processing*, vol. 40, no. 12, pp. 2961–2972, 1992.

[12] Gabriel Weinreich, "Coupled piano strings," *The Journal of the Acoustical Society of America*, vol. 62, no. 6, pp. 1474–1484, 1977.

[13] Gabriel Weinreich, "The coupled motions of piano strings," *Scientific American*, vol. 240, no. 1, pp. 118–127, 1979.

[14] Petre Stoica and Anders Eriksson, "Music estimation of real-valued sine-wave frequencies," *Signal processing*, vol. 42, no. 2, pp. 139–146, 1995.

[15] Robert Gray, "On the asymptotic eigenvalue distribution of toeplitz matrices," *IEEE Transactions on Information Theory*, vol. 18, no. 6, pp. 725–730, 1972.

[16] Myron Ross, Harry Shaffer, Andrew Cohen, Richard Freudberg, and Harold Manley, "Average magnitude difference function pitch extractor," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 22, no. 5, pp. 353–362, 1974.

[17] Mototsugu Abe and Julius O. Smith, "Design criteria for simple sinusoidal parameter estimation based on quadratic interpolation of FFT magnitude peaks," in *Audio Engineering Society Conventions and Conferences (AES'04)*, 2004, vol. 117.

[18] Clive Temperton, "Self-sorting mixed-radix fast Fourier transforms," *Journal of Computational Physics*, vol. 52, no. 1, pp. 1–23, 1983.

[19] Pierre Duhamel, "Implementation of Split-Radix FFT Algorithms for Complex, Real, and Real-Symmetric Data," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 34, pp. 285–295, 1986.

[20] Julius O. Smith, "Digital Audio Resampling Home Page," Tech. Rep., Center for Computer Research in Music and Acoustics (CCRMA), Stanford University, 2002.

[21] Petre Stoica and Arye Nehorai, "Music, maximum likelihood, and cramer-rao bound," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 37, no. 5, pp. 720–741, 1989.

[22] Kurt James Werner and François Georges Germain, "Sinusoidal parameter estimation using quadratic interpolation around power-scaled magnitude spectrum peaks," *Applied Sciences*, vol. 6, no. 10, pp. 306, 2016.

HARD REAL-TIME ONSET DETECTION OF PERCUSSIVE SOUNDS

Luca Turchet *

Center for Digital Music
Queen Mary University of London
London, United Kingdom
luca.turchet@qmul.ac.uk

ABSTRACT

To date, the most successful onset detectors are those based on frequency representation of the signal. However, for such methods the time between the physical onset and the reported one is unpredictable and may largely vary according to the type of sound being analyzed. Such variability and unpredictability of spectrum-based onset detectors may not be convenient in some real-time applications. This paper proposes a real-time method to improve the temporal accuracy of state-of-the-art onset detectors. The method is grounded on the theory of hard real-time operating systems where the result of a task must be reported at a certain deadline. It consists of the combination of a time-base technique (which has a high degree of accuracy in detecting the physical onset time but is more prone to false positives and false negatives) with a spectrum-based technique (which has a high detection accuracy but a low temporal accuracy). The developed hard real-time onset detector was tested on a dataset of single non-pitched percussive sounds using the high frequency content detector as spectral technique. Experimental validation showed that the proposed approach was effective in better retrieving the physical onset time of about 50% of the hits detected by the spectral technique, with an average improvement of about 3 ms and maximum one of about 12 ms. The results also revealed that the use of a longer deadline may capture better the variability of the spectral technique, but at the cost of a bigger latency.

1. INTRODUCTION

The research field of Music Information Retrieval (MIR) focuses on the automatic extraction of different types of information from musical signals. One of the most common application domains of such a field is that of automatic music transcription [1]. Another domain is represented by the identification of timbral aspects [2], which might be associated to different expressive intents of a musician [3] or to a particular playing technique that generated a sound [4]. The retrieval of the instant in which a pitched or unpitched musical sound begins, generally referred to as *onset detection*, is a crucial step in a MIR process. Numerous time- and spectrum-based techniques have been proposed for this purpose (see e.g., [5, 6]), some of which are based on the fusion of various methods [7].

Up to now, the majority of MIR research on onset detection has focused on offline methods based on the analysis of large datasets of audio files. Nevertheless, different techniques have also been developed for real-time contexts [8, 9, 10], especially for retrieving information from the audio signal of a single musical instrument [11, 12]. Real-time implementations of some onset detection techniques have been made available in open source libraries (e.g., *aubio*[1] [13]). Typically, the performance of an onset detector is assessed against annotated datasets. Such annotations may define onset times in line with human perception [14] or with the actual physics (which are generally referred to as *perceptual* and *physical* onset times respectively [6]).

Once an onset has been detected, it is possible to apply, to the adjacent part of the signal, algorithms capable of extracting different types of information (e.g., spectral, cepstral, or temporal features [15, 16]). For instance, such information may be used to identify the timbre of the musical event associated to the detected onset. In turn, the identified timbre may be utilized for classification tasks by means of machine learning techniques [17]. A challenging timbral classification concerns the identification of different gestures performed on a same instrument. For this purpose, it is crucial to understand the exact moment in which an onset begins. Indeed lot of the timbral information is contained in the very first part of the signal of a musical event.

However, to date, the onset detection methods available in the literature are little sensitive to the challenge of retrieving the exact initial moment of a musical event (i.e., the physical onset time). For instance, the Onset Detection Task specifications of the Music Information Retrieval Evaluation eXchange (MIREX)[2], and most of the papers in the area of onset detection, consider detected onsets as true positives if they fall within a window of 50 ms around the onset time reported in an annotated dataset. Furthermore, the vast majority of freely available datasets for MIR research are not accurate at millisecond or sub-millisecond level, which would be useful to designers of real-time MIR systems.

Currently, the most successful onset detectors are those based on frequency representation of the signal [5, 6, 18] (as shown by the results of MIREX context between 2005 and 2017[3]). Typically, detecting efficiently and effectively an onset using spectral methods requires at least 5.8 milliseconds after the occurrence of the peak of the involved onset detection function (ODF), considering a window size of 256 samples for the Short Time Fourier Transform and a sampling rate of 44.1 kHz. However, for such methods the time between the actual onset and the reported onset is unpredictable and may largely vary according to the type of sound in question. This is due to the fact that spectral methods are not based on the actual initial moment of the hit but on the identification of the ODF's peak (or its beginning), which may occur some millisec-

* This work was supported by a Marie-Curie Individual fellowship from the European Union's Horizon 2020 research and innovation programme (749561).

[1] Available at www.aubio.org

[2] http://www.music-ir.org/mirex/wiki/2017:Audio_Onset_Detection

[3] http://www.music-ir.org/mirex/wiki/MIREX_HOME

onds after the physical onset. Such variability and unpredictability of spectrum-based onset detectors may not be convenient in some real-time applications. An example of such applications is represented by those hybrid acoustic-electronic musical instruments that must react with minimal latency to a performer's action, involving a response (such as the triggering of a sound sample) that accounts for the correct classification of the timbre of the sound acoustically produced (see e.g., [4]).

This paper addresses the improvement of existing onset detectors to achieve a less variable and more predictable time accuracy in real-time contexts. Specifically, we limit our investigation to sounds of single non-pitched percussive instruments (therefore implementing a "context-dependent" method, not a "blind" one). In more detail, we do not consider instruments capable of producing radically different sounds, such as those of a full drum kit, but rather all the possible gamut of sounds resulting from hits on a same instrument (which may be produced by the player using different gestures). This research originated while developing an improved version of the smart cajón reported in [19], which belongs to the family of smart musical instruments [20]. For that application it was fundamental to retrieve with a higher degree of temporal accuracy the onsets corresponding to each hit produced on the smartified acoustic cajón, since the portion of signal subsequent to each onset was utilized for gesture classification (using audio feature extraction methods and machine learning algorithms based on the extracted features). The classified gesture was then repurposed into a triggered sound sample concurrent with the acoustic sound.

Notably, the real-time repurposing of a hit in hybrid acoustic-electronic percussive instruments such as the smart cajón, poses very strict constraints in terms of accuracy of detection and temporal reporting: the system not only must guarantee that a produced hit is always detected, but also that the onset is reported within a certain latency as well as that such latency is constant. Any success rate of onset detection different from 100% or with a too high latency is simply not an option for professional musicians, who require a perfectly responsive instrument and feel that they can truly rely on it. This imposes that the latency between their action on the instrument and the digital sound produced in response to it must be imperceivable.

Such strict requirements parallel those of hard real-time operating systems where a task must be accomplished at the end of a defined temporal window (deadline), otherwise the system performance will fail [21]. Therefore, for the terminology's sake, to distinguish our method from other real-time algorithms less sensitive to temporal accuracy we introduce the notion of *hard real-time onset detector (HRTOD)* and *soft real-time onset detector (SRTOD)*[4]. The latter are those methods that have more tolerant constraints in terms of the accurate onset time identification as well as in the variability of such time. Examples of methods belonging to the SRTOD category are the implementations reported in [11] and [12], which present a real-time drum transcription system available for the real-time programming languages Pure Data and Max/MSP. Another example is represented by the study reported in [22], where a recurrent neural network is employed for the onset detection task. Notably, our proposed method does not intend to reduce the actual latency of state-of-the art methods. Instead it aims at guaranteeing that the time of an onset is reported more accurately at the end of a set time window computed from the physical onset, in the same way as it happens for tasks in a hard real-time operating system.

The remainder of the paper is organized as follows. Section 2 describes the proposed onset detector that meets the requirements mentioned above as well as an implementation for it in Pure Data. Section 3 presents the results of the technical evaluation performed on various datasets of single percussive non-pitched instruments, while Section 4 discusses them. Section 5 concludes the paper.

2. PROPOSED HARD REAL-TIME ONSET DETECTOR

The proposed onset detection algorithm relies on the combination of time- and spectrum-based techniques. This choice was motivated by our initial experimentations, which suggested that methods based on temporal features may have a higher degree of accuracy in detecting the physical onset time. On the other hand, onset detection methods based on the spectral content may be less prone to false positives and false negatives compared to methods based on temporal features if their parameters are appropriately tuned, although they may suffer from unpredictability and variability issues in timing accuracy.

The proposed onset detector aims to take advantage of the strengths of the two approaches. Specifically, a time-based technique capable of detecting more reliably the very initial moment of a hit, but also more sensitive to false positives and false negatives, was used in parallel with a spectrum-based technique that was tuned to optimize the performance in terms of F-measure. Moreover, our goal was not only to detect an onset with minimal delay after the initial moment of contact of the exciter (e.g., hand, stick, etc.) and the resonator (e.g., skin of a drum, wood of a cajón panel), but also to ensure a high temporal resolution in tracking two subsequent hits. We set such resolution to 30 ms since this is approximatively the temporal resolution of the human hearing system to distinguish two sequential sound events [23]. Such a resolution is also adopted by the real-time onset detector proposed in [22].

The implementation of the proposed onset detector was accomplished in Pure Data, considering as input a mono live audio signal sampled at 44.1 kHz. The implementation was devised to achieve high computational efficiency, and more specifically, to run on low-latency embedded audio systems with low computational power (e.g., the Bela board [24]), which may be involved in the prototypization of smart instruments. The next three sections detail the utilized time- and spectrum-based techniques as well as the adopted fusion policy.

2.1. Time-based method

The time-based method (TBM) here proposed is inspired by the approaches to onset detection described in [5] and [8]. It must be specified that this technique only provides as output an onset timing, not the associated peak. Notably, the time-based method proposed in [25], which employs the logarithm of the input signal's energy to model human perception, was not utilized. This was due to the fact that we were interested in the physical onset not in the perceptual one. Figure 1 illustrates the various steps in the onset detection process.

We generated an ODF as follows. Firstly, we filtered the input signal with a high pass filter whose cutoff frequency was tuned on the basis of the type of percussive instrument being analyzed. This is the main difference with the time-based methods reported in [5],

[4]This terminology should not be confused with that used to discriminate onsets as hard (usually by percussive instruments, pitched and unpitched) or soft (e.g., produced by bowed string instruments).

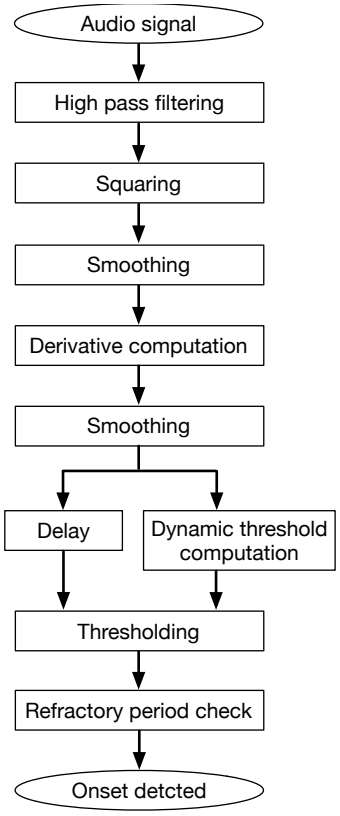


Figure 1: *Block diagram of the various steps involved in the time-based onset detector.*

which do not follow this initial step. Performing such a step allows one to drastically reduce the number of false positives while at the same time preserving (or only marginally affecting) the true positives. Secondly, we computed the energy by squaring the filtered signal. Subsequently, the energy signal underwent a smoothing process accomplished by a lowpass filter. This was followed by the calculation of the first derivative and again the application of a lowpass filter. The cutoff frequencies of the lowpass filters are configurable parameters.

Subsequently, a dynamic threshold (which is capable of compensating for pronounced amplitude changes in the signal profile) was subtracted from the signal. We utilized a threshold consisting of the weighted median and mean of a section of the signal centered around the current sample n:

$$\delta(n) = \lambda \cdot median(D[n_m]) + \alpha \cdot mean(D[n_m]) \qquad (1)$$

with $n_m \in [m-a, m+b]$ where the section $D[n_m]$ contains a samples before m and b after, and where λ and α are positive weighting factors. For the purpose of correctly calculating the median and the mean around the current sample, the pre-thresholded signal must be delayed of b samples before being subtracted from the threshold. The parameters a, b, λ and α are configurable. The real-time implementation of the median was accomplished by a Pure Data object performing the technique reported in [26].

The detection of an onset was finally accomplished by considering the first sample n of the ODF satisfying the condition:

$$n > \delta(n) \quad \& \quad n > \beta \qquad (2)$$

where β is a positive constant, which is configurable. To prevent repeated reporting of an onset (and thus producing false positive detections), an onset was only reported if no onsets had been detected in the previous 30 ms.

2.2. Spectrum-based onset detection technique

Various algorithms for onset detection available as external objects for Pure Data were assessed, all of which implemented techniques based on the spectral content. Specifically, we compared the objects i) *bonk~* [27], which is based on the analysis of the spectral growth of 11 spectral bands; ii) *bark~*, from the *timbreID* library[5], which consists of a variation of bonk~ relying on the Bark scale; iii) *aubioonset~* from the *aubio* library [13], which makes available different techniques, i.e., broadband energy rise ODF [5], high frequency content ODF (HFC) [28], complex domain ODF [29], phase-based ODF [30], spectral difference ODF [31], Kulback-Liebler ODF [32], modified Kulback-Liebler ODF [13], and spectral flux-based ODF [6]. Several combinations of parameters were used in order to find the best performances for each method.

All these spectral methods shared in common a variable delay between the actual onset time and the time in which the onset was detected. In the end *aubioonset~*, configured to implement the HFC was selected because it was empirically found to be capable of providing the best detection accuracy. This in line with Brossier's observations reported in [13]. A refractory period of 30 ms was applied after a detection to eliminate possible false positives within that window.

2.3. Fusion policy

Our strategy for combining the two onset detectors calculated in parallel consists in considering an onset as true positive if detected by HFC, and subsequently retrieving the initial moment by looking at the onset time of the corresponding onset (possibly) detected by TBM. The policy to fuse these two types of information highly depends on the deadline for reporting the onset after the physical one. In our HRTOD such a deadline is a configurable parameter, which must be greater than the duration of the window size chosen for HFC. On a separate note, we specify that while the time based method acts on a high-pass filtered version of the input signal, HFC uses the original signal.

The fusion policy is presented in the pseudocode of algorithm 1. For clarity's sake, the reader is referred to Figure 2. If HFC produces an onset and TBM has not yet, then the onset time is computed by subtracting the duration of HFC's window size from the time of the onset detected by HFC, and such an onset is reported after the difference between the deadline and the duration of HFC's window size. Any onset candidate deriving from TBM produced in the 30 ms subsequent to the reporting of HFC gets discarded.

Conversely, if TBM produces an onset and HFC has not yet, then the algorithms checks whether an onset is produced by HFC in the next amount of time corresponding to the duration of HFC's window size minus the *temporal error* that is estimated affecting TBM (i.e., the delay between the time of the physical onset and the time of the onset reported by TBM). If this happens, then such

[5] Available at `www.williambrent.com`

onset is reported after the amount of time corresponding to the deadline minus the duration of HFC's window size, and the onset time is computed by subtracting the duration of HFC's window size from the time of the onset detected by HFC. The error that affects TBM is a configurable parameter for the algorithm, whose value must be less than the duration of HFC's window size. Such an error is estimated on the basis of analyses performed on the input signal of the percussive instrument in question.

If HFC has not produced an onset in the time corresponding to the duration of HFC's window size minus the estimated error after the reporting of the onset by TBM, then the algorithm checks whether HFC has produced an onset in the next amount of time corresponding to deadline minus the duration of HFC's window size plus the estimated error. If this happens, then such onset is reported immediately and the onset time is computed by subtracting the estimated error from the time of the onset detected by TBM.

Critical to this fusion policy is the choice of the parameters governing the behavior of TBM. Indeed, if TBM produces too many false positives there is the risk of erroneous associations of onsets detected by TBM to onsets detected by HFC, as these might happen just before the actual physical onset. Conversely, if TBM produces too many false negatives, then HFC will be much less improved in terms of accuracy.

To estimate the TBM error while designing a real-time audio system, one could record the live audio produced by the system, apply the TBM configured to optimize the F-measure, and calculate the temporal distance between the time of the onset reported by TBM and the time of the physical onset (which can be determined by annotating the recorded dataset). Subsequently, the found minimum value could be used as the TBM error estimate. This guarantees that all onset times marked as improved with respect to the corresponding ones of the HFC, are effectively improved. Nevertheless, this would also limit the amount of improvement, as some onsets detected by HFC could be improved using a slightly greater TBM error estimate.

A less conservative strategy here recommended, consists in tolerating a small error on the time reporting of few onsets, such that the temporal accuracy for those onsets would be worsen only marginally, while at the same time increasing the temporal accuracy of a much greater number of HFC onsets. Specifically, our criterium adopted to determine an estimation of the TBM error is to select the minimum between the value of the first quartile and the result of the sum of 1 ms to the minimum delay found between the beginning of the sinusoid and the annotated physical onset:

$$TBM\_estimated\_error = min \begin{cases} 1^{st} quartile \\ 1 + min(error) \end{cases} \quad (3)$$

This allows one to tolerate in the worst case a maximum error of 1 ms for some of the hits (whose amount is lower or equal than the 25% of the total hits of the dataset). Therefore, the calculated onset times deriving from TBM can be effectively considered as an improvement compared to HFC in the majority of the cases.

3. EVALUATION

The temporal accuracy of the developed HRTOD was assessed on a dataset of recordings of four single percussive non-pitched instruments: conga, djembe, cajón, and bongo. In this evaluation we were not interested in assessing the detection accuracy of our HRTOD in terms of F-measure as this is fully determined by HFC (whose performance is well documented in the literature [28, 13]). Our focus was exclusively on the assessment of the actual improvement offered by HRTOD in terms of temporal accuracy compared to HFC. For this purpose, we carefully selected the parameters of TBM in order to maximize the F-measure and avoid any error in the fusion policy, likewise for HFC (see Table 1). In this investigation we were also interested in assessing whether the performance of HRTOD differed between the instruments and for two deadlines.

3.1. Procedure

In absence of accurate annotations of datasets of single percussive non-pitched instruments among those normally used by the MIR community, which could have served as a ground truth, we opted for using two freely available online libraries[6]. Such libraries were selected for the high quality recordings and the involvement of a large variety of playing styles and percussive techniques on the four investigated instruments. Those libraries contain 81 short recordings of hits on conga, 38 for djembe, 85 for cajon, and 31 for bongo.

To annotate the datasets we visually inspected the waveforms of the files and considered the first clear change in the waveform as an actual physical onset. Specifically, in this manual process we aimed at achieving an error tolerance of 0.5 ms. We did not annotate the whole database but only 100 hits per each instrument. Such annotated hits were those utilized to determine the estimated error of TBM. They were selected as follows. We recorded along with the file waveform, two additional tracks containing short sinusoidal waves beginning at the instants in which the onset were detected respectively by HFC and TBM (see Figure 2). Subsequently, for each sinusoid in the TBM track that was related to a true positive detected by HFC but happening before it, we calculated the time difference between the annotated physical onset and the beginning of the sinusoid. In this calculations one needs to add the time corresponding to b samples of which the waveform was delayed (in our case this corresponds to 0.045 ms as 2 samples were used for b).

For each instrument we randomly chose a subset of files and considered the first 100 hits satisfying the mentioned condition. For our purpose, an amount of 100 hits gives a reasonably accurate measurement in statistical sense and could be considered as the number that a designer of a real-time system would use to get the estimate of TBM error from analyzing live recordings of the system. Table 2 shows for each instrument the results of the analysis conducted on the 400 annotated hits to determine the estimate of TBM error, as well as the corresponding average and maximum error one would still get using it.

We configured HRTOD with two deadlines, at 11.6 and 18 ms, to compare its performance in the case of a short and long deadline. Indeed a longer deadline would have been able to capture those onsets detected by HFC after the short deadline is elapsed, given the HFC variability. The deadline of 11.6 ms was selected because it is equivalent to the time needed to compute analyses on 512 samples at 44.1 kHz sampling rate, therefore, the first 11.6 ms of the signal can be utilized without involving in the analysis any

[6] `http://cdn.mos.musicradar.com/audio/samples/musicradar-percussion-samples.zip` and `http://www.stayonbeat.com/wp-content/uploads/2013/07/Bongo-Loops_StayOnBeat.com_.zip`

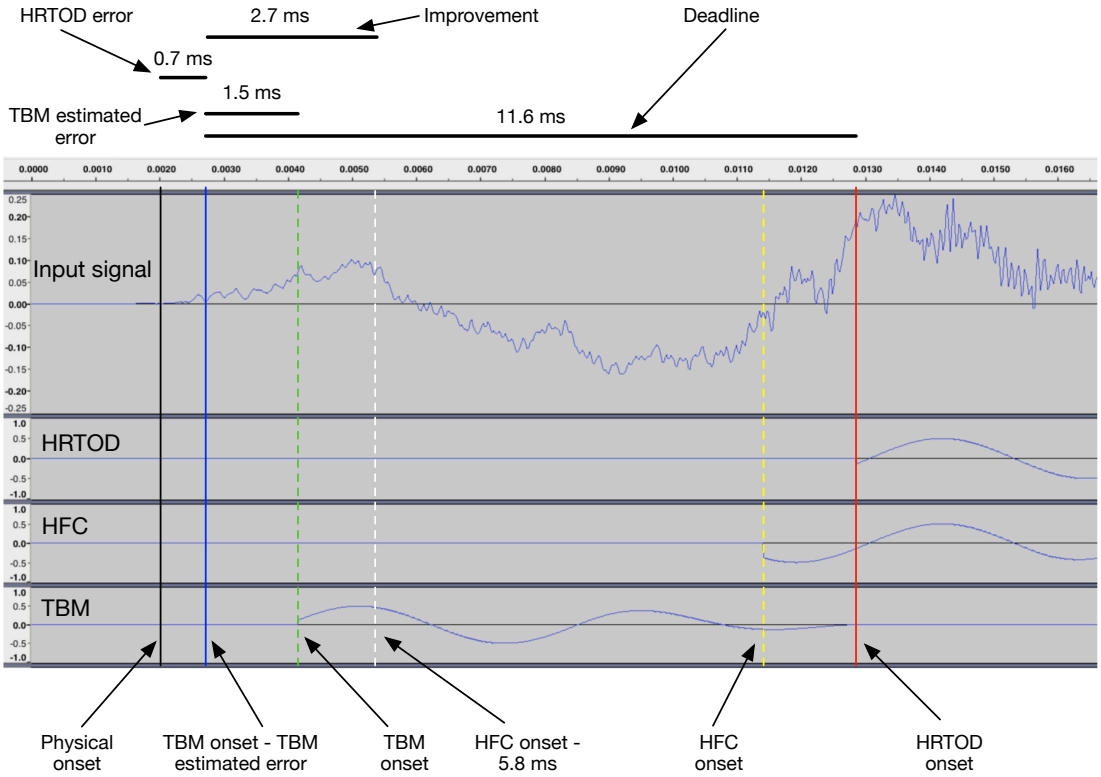


Figure 2: *Waveforms of the input signal of a hit on cajón and of three short sine waves triggered at the times of detecting the onsets using TBM, HFC, and HRTOD, with indications of the temporal events relevant to the HRTOD.*

pre-onset portion of the signal. The deadline at 18 ms was selected by considering a maximum reporting time of 20 ms for possible operations computed on such portion of the signal, which could take up to 2 ms (considering for instance real-time feature extraction, application of machine learning techniques, and repurposing of the analyzed sound). Specifically, this amount was justified by the results of the evaluation of the smart cajón prototype presented in [19]. These showed that a measured average latency of 20 ms between action and electronically generated sounds was deemed to be imperceivable by four professional cajón players. This was likely due to a masking effect in the attack of the acoustic sound that superimposes over the digital one.

3.2. Results

Table 3 presents the results of the application of the developed HRTOD to the dataset using the parameters for TBM reported in Table 2, and the two deadlines of 11.6 and 18 ms. For each instrument and for the whole dataset, we computed the number of hits detected by HFC, the number of hits affected by the temporal accuracy improvement of TBM, along with their percentage, their average improvement, and the maximum improvement. It is worth noticing that in calculating the improved performances of HRTOD compared to HFC we compared each onset time reported by HRTOD against the time reported by HFC minus 5.8 ms (this would be indeed the minimum time employed by HFC to report an onset after its actual occurrence given the 256-point window).

Table 3 also offers a comparison of the performances of HRTOD for the two deadlines by calculating their difference along the investigated metrics.

4. DISCUSSION

The first noticeable result emerging from Table 3 is that HRTOD effectively improved the temporal accuracy of HFC for all instruments and for both the investigated deadlines. The variability of HFC was drastically reduced since about 50% of the hits of the dataset were effectively improved for both the deadlines involved, with an average improvement of about 3 ms and maximum one of about 12 ms. Bongo was found to be the instrument most improved in terms of percentage of improved hits, although the average improvement was the lowest compared to the other instruments. Considering both the number of improved hits and the amount of average and maximum improvement, the cajón was found the instrument most positively affected by our HRTOD.

Furthermore, the results show that the use of a longer deadline generally improves all the considered metrics. Almost the 5% of the total hits were improved between the two deadlines, which shows the variability of HFC (and of spectral-based methods in general). Such a variability might constitute an issue in certain real-time applications. Indeed an error of more than 12 ms, as found for some hits on conga, may be critical when attempting to analyze in real-time the corresponding sound and classify it against other hits detected with no delay. The achieved average improvement due to the longer deadline was less than 0.5 ms compared

Algorithm 1: Pseudocode of the fusion policy of the involved TBM and HFC onset detection techniques in the developed HRTOD.

```
Input: Input signal, deadline, TBM_estimated_error, HFC_window_time
Output: Time of the detected onset reported when the deadline is elapsed
TBM_detected ← TBM(input_signal)
HFC_detected ← HFC(input_signal)
if HFC_detected == true && TBM_detected == false then
    HFC_onset_time ← get_time(HFC_detected)
    for the next 30 ms ignore any TBM_detected == true
    sleep(deadline - HFC_window_time)
    onset_time ← set_time(HFC_onset_time - HFC_window_time)
    return onset_time
else
    if HFC_detected == false && TBM_detected == true then
        TBM_onset_time ← get_time(TBM_detected)
        sleep(HFC_window_time - TBM_estimated_error)
        if HFC_detected == true then
            HFC_onset_time ← get_time(HFC_detected)
            sleep(deadline - HFC_window_time
            onset_time ← set_time(HFC_onset_time - HFC_window_time)
            return onset_time
        else
            sleep(deadline - HFC_window_time + TBM_estimated_error)
            if HFC_detected == true then
                onset_time ← set_time(TBM_onset_time - TBM_estimated_error)
                return onset_time
```

Table 1: *Values of parameters of TBM and HFC utilized for each instrument. Legend: HP = high-pass, LP = low-pass, f_c = cutoff frequency.*

	TBM								**HFC**		
	HP f_c (Hz)	**LP 1 f_c (Hz)**	**LP 2 f_c (Hz)**	a **(samples)**	b **(samples)**	β	λ	α	**threshold**	**window (samples)**	**hop (samples)**
Conga	4000	25	25	62	2	6e-09	0.8	0.8	0.2	256	64
Djembe	7500	25	25	62	2	7e-09	0.8	0.8	0.2	256	64
Cajón	7500	25	25	62	2	2e-09	0.8	0.8	0.2	256	64
Bongo	7500	25	25	62	2	2e-08	0.8	0.8	0.2	256	64

to the shorter one, but the maximum improvement was found to be more than 7 ms. The instrument that was mostly affected by such increment in the duration of the deadline was the cajón, while bongo was basically unaffected. This shows that for certain instruments a short deadline may be sufficient in capturing reliably the physical onset time of almost all hits.

Despite these encouraging results, it should be noticed that there are still margins for improvement as the method is affected by errors: as shown in the last two columns of Table 2, about the 75% of the hits would have needed a larger value for the TBM error estimate parameter. According to the analysis on the 400 annotated hits, the average error is below 2 ms but the maximum one could amount to about 11 ms. On a different vein, it is also worth noticing that the proposed method is context-dependent as it was built and tested by exploiting knowledge on the input signals investigated.

Although the algorithm has been conceived for real-time purposes, it can be applied to offline contexts as well. Offline algorithms have a number of advantages compared to real-time methods that might be exploited to refine the HRTOD here proposed. For instance, one could consider portions of the signal in the future, apply normalizations, use post-processing techniques, or utilize buffers larger than those here involved. A more timely accurate onset detector might have important implications not only for the design of musical instruments such as the smart ones [20], but also for automatic music transcription tasks [1], including those operating in real-time (see e.g., [11, 12]). Moreover, another application domain of the temporal accuracy improvements produced by the proposed method may be that of computational auditory scene analysis [33]. Although the sounds involved in this study belonged to the category of percussive non-pitched instruments, the method is expected to work well on several other categories of sounds (including the non musical ones as for instance footstep sounds, which have clearly discernible temporal characteristics like the sounds of percussive instruments [34]).

5. CONCLUSIONS AND FUTURE WORK

This paper proposed a real-time method to improve the temporal accuracy of state-of-the-art onset detectors. The study focused

Table 2: *Results of the analysis conducted on 100 annotated onsets for each instrument to determine the value of TBM estimated error, the expected average and maximum error of HRTOD.*

	mean±std err (ms)	**min (ms)**	**max (ms)**	**1st quartile (ms)**	**TBM estimated error (ms)**	**max error on 1st quartile (ms)**	**HRTOD mean error (ms)**	**HRTOD max error (ms)**
Conga	2.03±0.13	0.5	7	1	1	0.5	1.03	6
Djembe	1.7±0.14	0.5	7	1	1	0.5	0.7	6
Cajón	3.45±0.2	0.5	13	2	1.5	1	1.95	11.05
Bongo	2.98±0.09	1	6	2	2	1	1.98	4

Table 3: *Results of the proposed HRTOD involving the two deadlines and their differences.*

deadline (ms)	**instrument**	**# hits**	**# improved**	**% improved**	**mean improvement ± standard error (ms)**	**max improvement (ms)**
11.6	Conga	916	292	31.87	2.78±0.06	4.94
	Djembe	485	183	37.73	2.7±0.06	4.94
	Cajón	1094	532	48.62	3.7±0.05	4.94
	Bongo	965	643	66.63	2.02±0.04	4.94
	Total	**3460**	**1650**	**47.68**	**2.77±0.03**	**4.94**
18	Conga	916	325	35.48	3.33±0.11	12.2
	Djembe	485	200	41.23	3.1±0.11	10.75
	Cajón	1094	646	59.04	4.26±0.06	9.83
	Bongo	965	644	66.73	2.03±0.04	4.94
	Total	**3460**	**1815**	**52.45**	**3.17±0.04**	**12.2**
Difference	Conga	0	33	3.61	0.55	7.26
	Djembe	0	17	3.5	0.4	5.81
	Cajón	0	114	10.42	0.56	4.89
	Bongo	0	1	0.1	0.01	0
	Total	**0**	**165**	**4.77**	**0.38±0.12**	**7.26**

on percussive non-pitched sounds and for this purpose the spectral technique based on the high frequency content [28] was employed, which was reported in the literature to work the best for this type of sounds [13]. Experimental validation showed that the proposed approach was effective in better retrieving the physical onset time of about 50% of the hits in a dataset of four percussive non-pitched instruments compared to the performance of the onset detector based on high frequency content. The proposed method was inspired to hard real-time operating systems, which aim to guarantee that a task is accomplished at certain deadline. Our results revealed that the use of a longer deadline may capture better the variability of the spectral method (but at the cost of a bigger latency). Indeed, about 5% of the hits of the whole dataset could not be improved by involving a shorter deadline, although not all instruments were affected equally by a longer deadline.

The proposed method is expected to extend to sounds from other musical instruments as well as to non-musical sounds. Several directions for future work can be explored. Firstly, we plan to involve the proposed HRTOD in the development of percussive smart instruments such as the smart cajón reported in [19]. Secondly, future work will include experimenting with other types of data, in particular sounds from pitched instruments. An open question is whether the method would work for polyphonic pitched percussive instruments, where there can be one or more onsets roughly produced at the same time. Another future direction consists in exploring the performance of the proposed onset detector in noisy or multi-source environments, where for instance pitched onsets might be present. Finally, concerning context-awareness, it would be interesting to investigate whether the concepts presented in this study can be generalized to a more "blind" scenario.

The dataset involved in this study, the corresponding annotations, and the Pure Data source code are available online[7].

6. ACKNOWLEDGMENTS

Luca Turchet acknowledges supports from a Marie-Curie Individual fellowship of the European Union's Horizon 2020 research and innovation programme (grant nr. 749561).

7. REFERENCES

[1] E. Benetos, S. Dixon, D. Giannoulis, H. Kirchhoff, and A. Klapuri, "Automatic music transcription: challenges and future directions," *Journal of Intelligent Information Systems*, vol. 41, no. 3, pp. 407–434, 2013.

[2] X. Zhang and W.R Zbigniew, "Analysis of sound features for music timbre recognition," in *IEEE International Conference on Multimedia and Ubiquitous Engineering*. IEEE, 2007, pp. 3–8.

[3] M. Barthet, P. Depalle, R. Kronland-Martinet, and S. Ystad, "Acoustical correlates of timbre and expressiveness in clarinet performance," *Music Perception: An Interdisciplinary Journal*, vol. 28, no. 2, pp. 135–154, 2010.

[4] K. Jathal, "Real-time timbre classification for tabletop hand drumming," *Computer Music Journal*, vol. 41, no. 2, pp. 38–51, 2017.

[7] `https://github.com/lucaturchet`

[5] J.P. Bello, L. Daudet, S. Abdallah, C. Duxbury, M. Davies, and M.B Sandler, "A tutorial on onset detection in music signals," *IEEE Transactions on speech and audio processing*, vol. 13, no. 5, pp. 1035–1047, 2005.

[6] S. Dixon, "Onset detection revisited," in *Proceedings of the International Conference on Digital Audio Effects*, 2006, vol. 120, pp. 133–137.

[7] M. Tian, G. Fazekas, D. Black, and M.B Sandler, "Design and evaluation of onset detectors using different fusion policies," in *Proceedings of International Society for Music Information Retrieval Conference*, 2014, pp. 631–636.

[8] P. Brossier, J.P. Bello, and M.D Plumbley, "Real-time temporal segmentation of note objects in music signals," in *Proceedings of the International Computer Music Conference*, 2004.

[9] D. Stowell and M. Plumbley, "Adaptive whitening for improved real-time audio onset detection," in *Proceedings of the International Computer Music Conference*, 2007, pp. 312–319.

[10] S. Böck, F. Krebs, and M. Schedl, "Evaluating the online capabilities of onset detection methods," in *Proceedings of International Society for Music Information Retrieval Conference*, 2012, pp. 49–54.

[11] M. Miron, M.E.P. Davies, and F. Gouyon, "An open-source drum transcription system for pure data and max msp," in *IEEE International Conference on Acoustics, Speech and Signal Processing*. IEEE, 2013, pp. 221–225.

[12] M. Miron, M.E.P. Davies, and F. Gouyon, "Improving the real-time performance of a causal audio drum transcription system," in *Proceedings of the Sound and Music Computing Conference*, 2013, pp. 402–407.

[13] P. Brossier, *Automatic annotation of musical audio for interactive systems*, Ph.D. thesis, Queen Mary University of London, 2006.

[14] J. Vos and R. Rasch, "The perceptual onset of musical tones," *Perception & psychophysics*, vol. 29, no. 4, pp. 323–335, 1981.

[15] M. McKinney and J. Breebaart, "Features for audio and music classification," in *Proceedings of International Society for Music Information Retrieval Conference*, 2003, pp. 151–158.

[16] W. Brent, "Cepstral analysis tools for percussive timbre identification," in *Proceedings of the International Pure Data Convention*, 2009.

[17] W. Brent, "A timbre analysis and classification toolkit for pure data," in *Proceedings of the International Computer Music Conference*, 2010.

[18] C. Rosão, R. Ribeiro, and D.M de Matos, "Comparing onset detection methods based on spectral features," in *Proceedings of the Workshop on Open Source and Design of Communication*. ACM, 2012, pp. 71–78.

[19] L. Turchet, A. McPherson, and M. Barthet, "Co-design of a Smart Cajón," *Journal of the Audio Engineering Society*, vol. 66, no. 4, pp. 220–230, 2018.

[20] L. Turchet, A. McPherson, and C. Fischione, "Smart Instruments: Towards an Ecosystem of Interoperable Devices Connecting Performers and Audiences," in *Proceedings of the Sound and Music Computing Conference*, 2016, pp. 498–505.

[21] G.C Buttazzo, *Hard real-time computing systems: predictable scheduling algorithms and applications*, vol. 24, Springer Science & Business Media, 2011.

[22] S. Böck, A. Arzt, F. Krebs, and M. Schedl, "Online real-time onset detection with recurrent neural networks," in *Proceedings of the International Conference on Digital Audio Effects*, 2012.

[23] B.C.J Moore, *An introduction to the psychology of hearing*, Brill, 2012.

[24] A. McPherson and V. Zappi, "An environment for Submillisecond-Latency audio and sensor processing on BeagleBone black," in *Audio Engineering Society Convention 138*. 2015, Audio Engineering Society.

[25] A. Klapuri, "Sound onset detection by applying psychoacoustic knowledge," in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*. IEEE, 1999, vol. 6, pp. 3089–3092.

[26] S. Herzog, "Efficient dsp implementation of median filtering for real-time audio noise reduction," in *Proceedings of the international conference on Digital Audio Effects*, 2013, pp. 1–6.

[27] M.S. Puckette, T. Apel, and D.D Ziccarelli, "Real-time audio analysis tools for pd and msp," in *Proceedings of the International Computer Music Conference*, 1998.

[28] P. Masri, *Computer modelling of sound for transformation and synthesis of musical signals*, Ph.D. thesis, University of Bristol, Department of Electrical and Electronic Engineering, 1996.

[29] C. Duxbury, J.P. Bello, M. Davies, and M.B Sandler, "Complex domain onset detection for musical signals," in *Proceedings of the Digital Audio Effects Conference*, 2003, pp. 1–4.

[30] J.P. Bello and M.B Sandler, "Phase-based note onset detection for music signals," in *Proceedings of the IEEE International Conference on Acoustics Speech and Signal Processing*, 2003, vol. 5, pp. 441–444.

[31] J. Foote and S. Uchihashi, "The beat spectrum: A new approach to rhythm analysis," in *Proceedings of IEEE International Conference on Multimedia and Expo*. IEEE, 2001, pp. 881–884.

[32] S. Hainsworth and M. Macleod, "Onset detection in musical audio signals," in *Proceedings of the International Computer Music Conference*, 2003.

[33] D. Stowell, D. Giannoulis, E. Benetos, M. Lagrange, and M.D Plumbley, "Detection and classification of acoustic scenes and events," *IEEE Transactions on Multimedia*, vol. 17, no. 10, pp. 1733–1746, 2015.

[34] L. Turchet, "Footstep sounds synthesis: design, implementation, and evaluation of foot-floor interactions, surface materials, shoe types, and walkers' features," *Applied Acoustics*, vol. 107, pp. 46–68, 2016.

MUSIKVERB: A HARMONICALLY ADAPTIVE AUDIO REVERBERATION

João Paulo Caetano Pereira
University of Porto, Faculty of Engineering, MIEEC
Porto, Portugal
ee12035@fe.up.pt

Gilberto Bernardes
INESC TEC and University of Aveiro
Portugal
gba@inesctec.pt

Rui Penha
INESC TEC and Univeristy of Porto, Faculty of Engineering
Porto, Portugal
rui.penha@inesctec.pt

ABSTRACT

We present MusikVerb, a novel digital reverberation capable of adapting its output to the harmonic context of a live music performance. The proposed reverberation is aware of the harmonic content of an audio input signal and 'tunes' the reverberation output to its harmonic content using a spectral filtering technique. The dynamic behavior of MusikVerb avoids the sonic clutter of traditional reverberation, and most importantly, fosters creative endeavor by providing new expressive and musically-aware uses of reverberation. Despite its applicability to any input audio signal, the proposed effect has been designed primarily as a guitar pedal effect and a standalone software application.

1. INTRODUCTION

Adaptive digital audio effects (ADAFx) are a class of audio effects, whose control parameters are mapped to attributes of the audio input signal to be transformed [1]. This level of symbiotic information exchange between an input signal and control parameters of the transformation effect has attracted the attention of academia and industry over the last decade as a new strategy for music creation [2].

The mappings between audio input attributes and effect parameters are central to ADAFx [3]. In this context, we can understand the emergence of ADAFx in light of the breakthroughs in audio-content processing for audio signals description, which have been proposed by the signal processing and music information retrieval communities.

Within the academic literature several ADAFx studies and prototype applications have been proposed [1, 4, 5]. These contributions focus mostly on mapping strategies between signal attributes and effect parameters [1]. Within industry and for the specific case of the guitar, the target instrument of our study, the following three commercial ADAFx have been recently identified in [3]: 'TE-2 Tera Echo', 'MO-2 Multi Overtone' and 'DA-2 Adaptive Distortion' [6, 7, 8].

In this paper, we extend existing guitar ADAFx by proposing a harmonically adaptive audio reverberation as a guitar pedal effect and a standalone software application. To the best of our knowledge, the sole existing application that implements such an ADAFx is Zynaptiq's Adaptiverb [9], for which no technical descriptions is known to be available.

In contrast to traditional digital reverberation, which models the physical phenomena of sound waves reflecting on enclosed space surfaces [4], MusikVerb aims at controlling the tonal clarity (understood as levels of consonance/dissonance) and harmonic richness of a reverberation tail. To this end, MusikVerb transforms the output of a traditional audio reverberation by filtering its output according to a ranked list of pitch classes (i.e., the twelve notes of the chromatic scale) computed from the perceptual-inspired Tonal Interval Space space [10]. Given this ranked list of pitch classes, the user can then 'tune' the reverberated signal to the harmonic context of an audio input signal.

The remainder of this paper is organized as follows. Section 2 presents the architecture of the MusikVerb system and the information flow between its component modules. Section 3 presents the extraction of harmonic attributes from an audio input signal to create a ranked list of pitch classes according to their perceptual distance to an input audio signal. Section 4 details how a ranked pitch class list is mapped to a frequency-domain representation (i.e., spectrum). Section 5 describes an algorithm which filters an audio reverberation tail to 'fit' the harmonic context of a performance. Section 6 provides an overview of the user control parameters of MusikVerb in both hardware and software instantiations of the system. Section 7 details the creative applicability of MusikVerb as highlighted by expert musicians when interacting with the system. Finally, Section 8 states the conclusions of our work and future directions.

2. MUSIKVERB ARCHITECTURE

Fig. 1 shows the architecture of MusikVerb, which follows the threefold typical ADAFx structure: 1) extraction of audio attributes from an input signal; 2) mappings between audio attributes and effect parameters; and 3) the processing of the effect transformation [3].

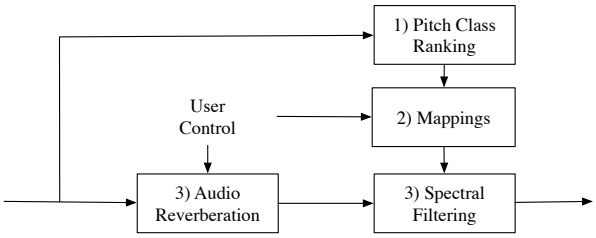


Figure 1: *MusikVerb architecture. The audio signal flux flows from left to right between the (squared) component modules.*

The harmonic content of an audio input signal is 1) analyzed to extract a ranked list of pitch classes according to a perceptual distance measure. 2) Then, a mapping between the ranked pitch class list and a frequency-domain audio representation is created to 3) draw a filtering shape to be applied to a reverberated audio input signal. While the choice of digital reverberation is critical to the sounding result of MusikVerb, the model can incorporate any algorithm of this class, while preserving its main characteristics.

3. PERCEPTUAL PITCH CLASS RANKING

We adopt the Tonal Interval Space [10] in MusikVerb to compute the perceptual distance between two given sonorities driven from both symbolic music representation and musical audio. Ultimately, these perceptual distances support the creation of a ranked list of pitch classes from an audio input signal. The choice of such a perceptually-guided space over other related tonal pitch spaces (e.g., Spiral Array [11] and Tonal Pitch Space [12]) is due to its possibility: i) to process both symbolic music representations and audio input signals without the need for a error-prone audio-to-score transcription; ii) to represent the most common pitch levels, i.e., pitch, chord, and key, in a single space; and iii) to efficiently compute the perceptual distance between tonal pitch.

The Tonal Interval Space uses the fast Fourier transform to convert a given sonority, represented as the L_1 normalized Harmonic Pitch Class Profile (HPCP) vector [13], $c(n)$, expressing the energy of the 12 pitch classes, into a Tonal Interval Vector (TIV), $T(k)$, expressing musical interval periodicities, such that:

$$T(k) = w_a(k) \sum_{n=0}^{N-1} \bar{c}(n) e^{\frac{-j2\pi kn}{N}}, \quad k \in \mathbb{Z} \quad , \tag{1}$$

where $N = 12$ is the dimension of the chroma vector. $w_a(k) = \{3, 8, 11.5, 11.5, 15, 14.5, 7.5\}$ are weights derived from empirical ratings of dyads consonance used to adjust the contribution of each interval, k, thus making the space perceptually relevant [14]. We set k to $1 \leq k \leq 6$ for $T(k)$ since the remaining coefficients are symmetric. $T(k)$ uses $\bar{c}(n)$ which is $c(n)$ normalized by the DC component $T(0) = \sum_{n=0}^{N-1} c(n)$ to allow the representation and comparison of music at different hierarchical levels of tonal pitch [10].

The resulting spatial location of TIVs, $T(k)$, ensures that tonal pitch understood as perceptually related within the Western music context correspond to small Euclidean distances. For example, at the pitch class level, it places intervals that play an important role in the tonal system (e.g., octaves, fifths, and thirds) at smaller distances. At the key level, the Tonal Interval Space represents our expectancy of proximity between the 24 major and minor keys by placing the dominant, subdominant and their relative minor keys at close distances as well as the diatonic pitch class and chord sets of a particular key in its neighborhood [10]. Mathematically, the Euclidean distance between two given TIVs, $T_i(k)$ and $T_j(k)$, is given by:

$$P_{i,j} = \sqrt{\sum_{k=1}^{M} |T_i(k) - T_j(k)|^2} \quad , \tag{2}$$

where $M = 12$ is the dimension of a TIV, $T(k)$.

By interpreting $T_i(k)$ and $T_j(k)$ in Eq. (2) as an audio input TIV and a pitch class TIV, respectively, and repeating the operation for the 12 pitch classes (i.e., 0-11), we compute the distances of an input TIV from the 12 pitch classes, which we then concatenate into a single list. Finally, the list values are reordered by increasing distance and a list with ranked pitch class indexes is created. Fig. 2 shows the various steps involved in the creation of a ranked list of pitch classes from an audio input TIV of the C major chord (i.e., the pitch class set {0,4,7}).

To control the output rate of the ranked pitch class vectors, we compute mean values per TIV bin from a user-defined number of $W_s = 4096$ sample window TIVs with 50% overlap. This adaptation parameter, A, is further detailed in Section 7 and has been shown to have a critical importance in the applicability scenarios of MusikVerb by expert musicians.

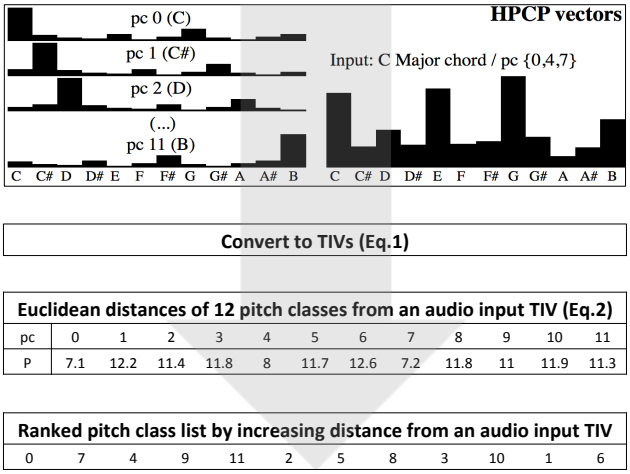


Figure 2: *Illustration of the main algorithmic steps involved in the creation of a ranked list od pitch class distances from an audio input TIV of the C major chord.*

4. MAPPINGS

The mappings module is responsible for translating the ranked pitch class distance list into a spectral representation, which is then used to control the amplitude of frequency bins in a spectral filtering algorithm.

From the 12-element ranked list of pitch classes, a set of N_{pc} user-defined pitch classes are retrieved sequentially from the first element. N_{pc} is an integer value ranging from $N_{pc} = 1$, the first element of the list, to $N_{pc} = 12$, the entire list. The greater the N_{pc} value, the more perceptually distant notes to the input audio signal are introduced. The trimmed pitch class list, $m[k]$, is then mapped to an array of $0.5 \cdot W_s$ elements, representing the entire pitch range given by Eq.(3), where f_{ref} is the tuning reference (e.g $f_{ref} = 440Hz$)

$$x[k] = f_{ref} \cdot 2^{\frac{m[k]}{12}}, \quad 0 \leqslant k < N_{pc} \quad , \tag{3}$$

where $x[k]$ is a vector containing the frequency corresponding to the first octave of the notes that should be on the output. For each pitch class in Eq. (3), a user-defined number of harmonics, N_h, is added, to regulate the harmonic richness of the re-synthesized signal. We empirically defined the number of harmonics N_h to be an integer value between 1 and 20, which we compute as:

$$y_k[n] = \prod n \cdot x[k], \quad 1 \leqslant n < N_h, \quad 0 \leqslant k < N_{pc} \quad . \tag{4}$$

After obtaining the vectors y_k, containing the frequencies that correspond to the selected N_{pc} and N_h we map them to elements of the $0.5 \cdot W_s$ window-sized filtering shape, H_f, using Eq. (5) where f_{res} corresponds to the FFT frequency resolution.

$$H_f[p] = 1, \quad p = \frac{y_k[n]}{f_{res}} \tag{5}$$

5. SPECTRAL FILTERING

MusikVerb resynthesises the input signal processed by a digital reverberation using a spectral filtering algorithm, similar to the one of the phase vocoder [15]. By multiplying the equal-sized frequency-domain representations of both the reverberated signal and the spectral filter shape resulting from Eq. 4, we then regulate the amplitude of each frequency bin.

6. USER CONTROL

MusikVerb has a dual implementation as a guitar pedal and a standalone software application. The Pure Data [16] software environment was initially adopted to prototype the effect due to its the flexibility in running as a standalone application, a VST plug-in [17] and in embedded DSP systems, such as the low-latency audio processing BELA[1] [18].

Both hardware (guitar pedal) and software (standalone application) instantiations of MusikVerb have two main groups of control parameters. The first group includes the digital reverberation parameters, such as room size, reverberation time, and spread, to cite a few. These parameters depend on the adopted digital reverberation algorithm, and thus can change accordingly. The digital reverberation adopted in the current version of our system includes several well-known digital reverberations implemented in Pure Data by Tom Erbe [19].

The second group includes the control parameters specific to MusikVerb: adaptation, harmonicity, and richness. Adaptation regulates the rate at which the ranked list of pitch classes is computed, which the user can control using a potentiometer in the guitar pedal and a slider in the software application (see Fig. 3). The harmonicity and richness parameters regulate the number of (ranked) pitch classes which are present in the output reverberated signal and the number of harmonics assigned to each note, respectively. These two latter parameters are controlled simultaneously with a single control in both hardware and software implementation of MusikVerb. In the hardware implementation, an expression pedal is scaled logarithmically to both parameters simultaneously. The choice of a logarithmic scale allows a finer degree of control over the initial range of the scale, where the effect more significantly alters a traditional digital reverberation. In the software implementation, the control of these two parameters are done via a 2-dimensional panel, whose x and y axis are assigned to each parameter (see Fig. 3).

7. APPLICATION

We have conducted several informal sessions with expert guitarists acquainted with different musical styles to infer recurrent applicability scenarios of MusikVerb and their creative potential. Three typical parameter combinations have caught the attention of the participants. These three parameter combinations explore MusikVerb in a wide range of creative applicability scenarios from a clutter-free reverberation with control over the reverberation harmonic quality to effects which are rather situated in the accompaniment systems domain.

The first two cases adopt low degrees of harmonicity and (harmonic) richness (e.g., $N_{pc} = 3$ and $N_h = 5$) and focus on the manipulation of the adaptation and reverberation time parameters.

[1] https://bela.io/

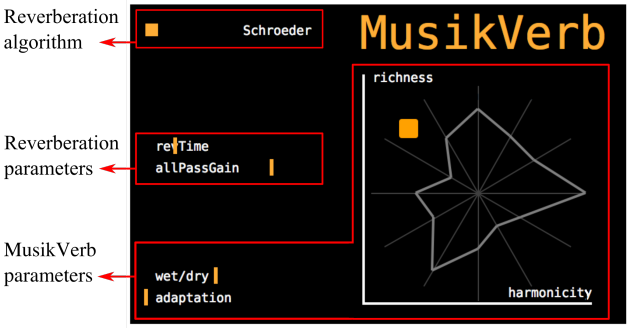


Figure 3: *MusikVerb software application interface.*

Adopting a low adaptation (e.g., $A = 6$) and a reverberation time typical of concert venues (e.g., around two seconds of decay time), MusikVerb significantly reduces the typical clutter of traditional reverberations, which result from the superposition of inharmonic frequencies around the frequency range of the source (as shown in Fig. 4. While this parametrization mode preserves most attributes of a reverberation without obscuring the source, it does not model the acoustic reflections of a room, as such an harmonically-tuned space does not exist.

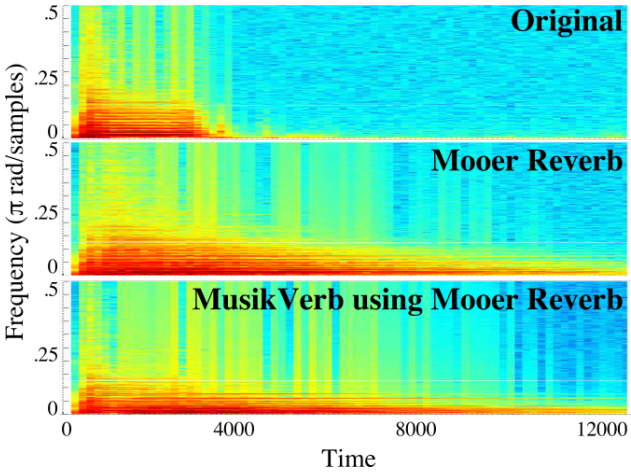


Figure 4: *Three sonogram representations of an (original) audio soundfile (top), and two processed renditions of the soundfile after being processed by Mooer reverberation (middle) and MusikVerb using the Mooer reveberation (bottom).*

The second case retains the low degrees of harmonicity and richness and opposes the first scenario by adopting high adaptation and reverberation time values (e.g., $A = 15$ and reverberation times around 5-10 seconds of decay time). This parameter combination creates an accompaniment close to drones or pedal tones which are predominant in the harmonic context of large sections of the input signal. Harmonicity in the context of this parameter combination can alter the density of pitch classes in the accompaniment which can range from a monophonic pedal tone to chords changes over time with variable number of notes. High adaptation

values impose a certain shift in time between the input signal and the (filtered) reverberation response to a level which no physical space can create or its digital reverberation models. This scenario provides ambient artists, film composers and sound designers with exciting new creative options for making evolving drones, organic pads, lush ambient and soundscapes.

Finally, the third parameter combination fixes the adaptation and reverberation time to average values across their range (e.g., $A = 10$ and a 1 second reverberation tail) and explore the dynamic manipulation of the linked harmonicity and richness parameters across the musical time. In manipulating these linked dimensions via the guitar pedal, for example, we can change the harmonic quality of the reverberation output in real-time in light of the harmonic content of the input. Manipulating the degree of harmonic proximity to the input signal, has a clear perceptual correlate with consonance (lower values) and dissonance (higher values), which can be dynamically manipulated irrespective of the performance audio content, thus promoting new strategies for creation.

The MusikVerb application, some sound examples demonstrating the three aforementioned applicability scenarios, and a demonstration video of a session with a guitarist performing with MusikVerb can be found online at: `https://bit.ly/2Jw3OoP`.

8. CONCLUSIONS AND FUTURE WORK

We presented MusikVerb, a system which promotes a novel adaptive reverberation audio effect, which results from technical and artistic contributions. The system is effective in reducing the sonic clutter, commonly introduced by traditional reverberation effects, while promoting the exploration of new creative spaces, notably those close to an automatic accompaniment system, by leveraging a constant symbiosis between engineering and creativity. MusikVerb was developed as a embedded guitar pedal system using the BELA platform and as a software standalone application in the Pure Data programming language.

To further extend MusikVerb, it would be interesting to adapt it and test it with different input sources, either instruments, ambient sounds or any other sonic input. Adapting the weights, $w_a(k)$ of the Tonal Interval Space, to privilege intervals other than octaves, fifths and thirds, can extend the creative potential of the tool beyond the perceptually-inspired syntax of the Western tonal harmony. Finally, we aim to compare our system with Zynaptiq's Adaptiverb [9] to unveil their sonic and usability differences.

9. ACKNOWLEDGMENTS

This work is supported by national funds through the FCT - Foundation for Science and Technology, I.P., under the project IF/01566/2015.

10. REFERENCES

[1] V. Verfaille and D. Arfib, "A-dafx: Adaptive digital audio effects," in *Proceedings of COST G-6 Conference on Digital Audio Effects*, 2001.

[2] O. Campbell, C. Roads, A. Cabrera, M. Wright, and Y. Visell, "Adept: A framework for adaptive digital audio effects," in *2nd AES Workshop on Intelligent Music Production (WIMP)*, 2016.

[3] J. Holfelt, G. Csapo, N. Andersson, S. Zabetian, M. Castenieto, S. Dahl, D. Overholt, and C. Erkut, "Extraction, mapping, and evaluation of expressive acoustic features for adaptive digital audio effects," in *Proceedings of the Sound & Music Computing Conference*, 2017.

[4] U. Zölzer, *DAFX: Digital Audio Effects*, John Wiley & Sons, Ltd, Sussex, UK, second edition, 2011.

[5] V. Verfaille, M. Wanderley, and P. Depalle, "Mapping strategies for gestural and adaptive control of digital audio effects," *Journal of New Music Research*, vol. 35, no. 1, pp. 71–93, 2006.

[6] B. Corporation, "Te-2–tera echo," `https://www.boss.info/global/products/te-2/`, accessed April 9, 2018.

[7] B. Corporation, "Da-2–adaptive distortion," `https://www.boss.info/global/products/da-2/`, accessed April 9, 2018.

[8] B. Corporation, "Mo-2–multi overtone," `https://www.boss.info/global/products/mo-2/`, accessed April 9, 2018.

[9] Zynaptiq, "Adaptiverb," `http://www.zynaptiq.com/adaptiverb/`, accessed March 30, 2018.

[10] G. Bernardes, D. Cocharro, M. Caetano, C. Guedes, and M. E. P. Davies, "A multi-level tonal interval space for modelling pitch relatedness and musical consonance," *Journal of New Music Research*, vol. 45, no. 4, pp. 281–294, 2016.

[11] E. Chew, "The spiral array: An algorithm for determining key boundaries," in *Music and artificial intelligence*, pp. 18–31. Springer, 2002.

[12] F. Lerdahl, "Tonal pitch space," *Music Perception: An Interdisciplinary Journal*, vol. 5, no. 3, pp. 315–349, 1988.

[13] E. Gómez, *Tonal Description of Music Audio Signals*, Ph.D. thesis, Universitat Pompeu Fabra, 2006.

[14] G. Bernardes, M. E.P. Davies, and C. Guedes, "A perceptually-motivated harmonic compatibility method for music mixing," in *Procedings of the CMMR conference*, 2017, pp. 104–115.

[15] M. Dolson, "The phase vocoder: A tutorial," *Computer Music Journal*, vol. 10, no. 4, pp. 14–27, 1986.

[16] M. Puckette, "Pure data: another integrated computer music environment," in *Proceedings of the second intercollege computer music concerts*, 1996, pp. 37–41.

[17] Enzien Audio Ltd., "Heavy audio tools," `https://enzienaudio.com`, accessed March 28, 2018.

[18] P. Brinkmann, P. Kirn, R. Lawler, C. Mccormick, M. Roth, and H.-C. Steiner, "Embedding pure data with libpd," in *Proceedings of the Pure Data Convention*, 2011, pp. 291–.

[19] T. Erbe, *Building the Erbe-Verb: Extending the Feedback Delay Network Reverb for Modular Synthesizer Use*, Ann Arbor, MI: Michigan Publishing, University of Michigan Library, 2015.

[20] J. Sterne, "pace within space: Artificial reverb and the detachable echo," *Grey Room*, vol. 60, pp. 110–131, 2015.

[21] X. Serra, *Musical Signal Processing*, chapter Musical Sound Modeling with Sinusoids plus Noise, pp. 91–122, G. D. Poli and A. Picialli and S. T. Pope and C. Roads, Lisse, Switzerland, 1996.

A VIRTUAL TUBE DELAY EFFECT

Riccardo Simionato
University of Padova
Dept. of Information Engineering
Padova, Italy
riccardo.simionato.vib@gmail.com

Juho Liski *, *Vesa Välimäki*
Aalto University, Acoustics Lab
Dept. of Signal Processing and Acoustics
Espoo, Finland
juho.liski@aalto.fi

Federico Avanzini
University of Milan
Dept. of Computer Science
Milan, Italy
federico.avanzini@di.unimi.it

ABSTRACT

A virtual tube delay effect based on the real-time simulation of acoustic wave propagation in a garden hose is presented. The paper describes the acoustic measurements conducted and the analysis of the sound propagation in long narrow tubes. The obtained impulse responses are used to design delay lines and digital filters, which simulate the propagation delay, losses, and reflections from the end of the tube which may be open, closed, or acoustically attenuated. A study on the reflection caused by a finite-length tube is described. The resulting system consists of a digital waveguide model and produces delay effects having a realistic low-pass filtering. A stereo delay effect plugin in PURE DATA[1] has been implemented and it is described here.

1. INTRODUCTION

Analog and digital delays are at the basis of several audio effects, including vibrato, flanger, chorus, echo, as well as spatial effects such as reverberation [1]. This paper investigates in particular the delay effects produced by a long narrow tube and presents a digital model of sound propagation in such a medium, including time delay, propagation losses, and end-reflections.

The first analog audio effect based on a narrow long tube was proposed in 1960 [2]. Olson and Bleazey presented a synthetic reverberator built with a tube, a loudspeaker, transducers, and a microphone delay unit in combination with a feedback system. A horn-loudspeaker coupled to a tube with three microphones located at different distances realized three different delays that, in conjunction with a positive feedback system, provided time spaced components.

In 1971, Bill Putman and Duane H. Cooper designed a garden-hose-based mechanical delay[2]. The echo-free acoustic delay device, called the Cooper Time Cube, sends audio through long coiled tubing with mic capsules, used as speakers and pickups, to create a time delay. In addition, a series of tooled aluminum blocks tune the delay to a relatively flat response.

Examples of simulated analog delay system are the Echoplex Tape Delay [3], and the Bucket Brigade Device [4]. The Echoplex is a tape delay device with fixed playback and erase heads, a movable record head, and a tape loop. A simulation using a circular buffer and pointers moving along it was presented in [3]. The bucket-brigade device instead realizes a time delay with an analog circuit. The input signal is sampled in time and passed into a series of capacitors and MOS transistor switches. The device is modeled with low-order digital infinite impulse-response (IIR) filters based on the resistance and capacitance values of the filters [4].

Other delay-based system examples are the spring [5, 6, 7] and plate reverbs [8, 9]. Spring reverberation is an electromechanical effect based on metal springs [10]. A first simulation by measuring the response of a real spring reverberation unit and by using digital waveguide methods was proposed in [5]. Two other methods involving a finite difference scheme [6], and by using delay-network reverberation techniques [7] were later presented. Instead, plate reverberation uses steel plates under tension [11], and it can be simulated with finite difference methods [8] and by using a hybrid structure consisting of a short convolution section and a feedback delay network [9].

The reverberation and coloration caused by a long tube has also been shown to be a robust cue for the distance perception of a sound source [12]. In a recent study, a digital-waveguide-mesh model of a small tubular shape has been used to simulate distance in a virtual environment [13]. The virtual tube delay effect presented in this paper can also be employed for this application.

Digital waveguide modeling for wave propagation in cylindrical and conical instruments is often used [14, 15, 16]. A technique for estimating a waveguide model of wind instrument from acoustic tube measurements was also presented in [17].

The rest of the paper is organized as follows. An overview of the performed measurements is given in Sec. 2, while their analysis is presented in Sec. 3. Sections 4 and 5 describe the approach used to design the digital propagation and reflections filters, which are then compared to the measurements in Sec. 6 in order to provide an objective evaluation of the results. Section 7 presents and discusses the implementation of a real-time plugin in the PURE DATA environment. Finally, Sec. 8 concludes this paper.

Supplementary materials including the plugin, the externals for MAC OS X and LINUX, the source C++ file, and some dry sounds are available for download at https://github.com/RiccardoVib/VIRTUAL_TUBE_DELAY-EFFECT-.

2. ACOUSTIC TUBE DELAY MEASUREMENT

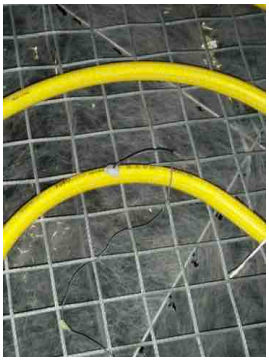
Three different tubes were used with an internal diameter of 1.2, 1.9 and 2.5 cm, respectively. The first tube was 8.8 m long and the other ones 25 m. The tube responses were measured with a logarithmic sine sweep that was played back to the tube with a full range loudspeaker. Figure 1 shows the equipment and the setup of the measurements.

The measurements were conducted in an anechoic chamber and in two modalities: closed end and open end. The goal of the first modality was to obtain a clean impulse response caused by propagation and losses without any reflections. Polyurethane and a metal plate were used to absorb and block reflections from the

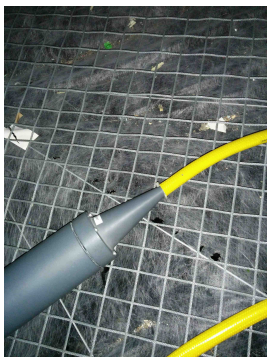
* J. Liski's work was supported by the Aalto ELEC Doctoral School.

[1] http://puredata.info.

[2] https://www.uaudio.com/blog/cooper-time-cube-power/.

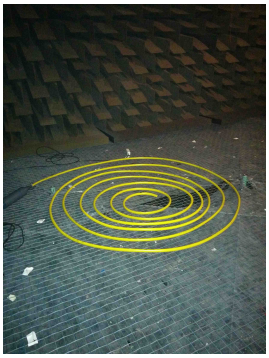


(a) Microphone inside the tube with gray moldable plastic to attach it to the hole.



(b) Loudspeaker attached to the end of the tube with a conical adaptor.

(c) Short narrow tube (length 8.8 m, inner diameter 1.2 cm).



(d) Long medium-sized tube (25 m, 1.9 cm).

Figure 1: *Measurement setup in the anechoic chamber.*

tube end. The measurements with the open end were performed by using the acoustic pulse reflectometry technique [18] and required further analysis of the reflection behavior, as the impulse responses contained clearly observable repeating reflections. The polyurethane and the metal plate were chosen based on initial experiments to minimize the reflections from the end.

Ten holes were drilled 1 m apart along the length of the tube starting 2.5 cm from the loudspeaker end of the tube. Multiple measurements were made, recording the response of one hole at a time with a miniature microphone while blocking the others with moldable plastic material in order to avoid a "flute finger-hole effect" in the recordings. In addition, in order to record the cleanest possible impulse responses, the measurements were taken from hole positions drilled up to 10 m from the loudspeaker end of the tube to ensure at least 15 m of length to the opposite end (and a round-trip travel distance of 30 m before returning to the microphone). An exception was made with the 1.2 cm diameter tube, since it was only 8.8 m long.

The impulse response of the system was computed using Farina's method, convolving the recorded signal with the time-inverted logarithmic sweep [19]. The input signal was 3 s long and with an amplitude of -41 dB, chosen after several experiments in order to find a trade-off between the signal-to-noise ratio and the harmonic distortion. The average SNR in the measurements ranged from 50 dB (for the narrowest tube) up to 40 dB (in the largest one).

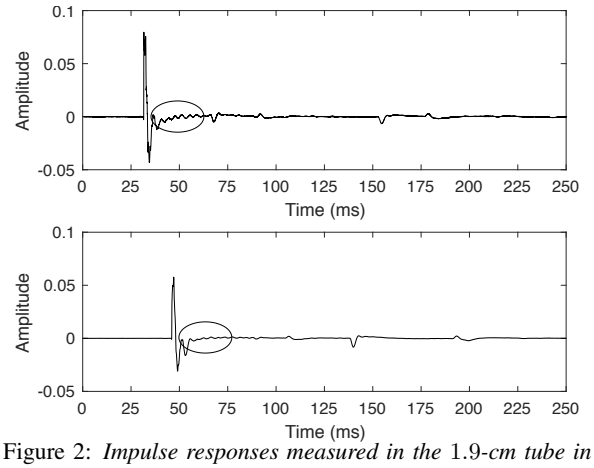


Figure 2: *Impulse responses measured in the* 1.9*-cm tube in the open end case, at the distance of* 4.25 *m (top) and* 9.25 *m (bottom).*

Finally, the measurements were performed with a sample rate of 44.1 kHz.

3. TUBE DELAY ANALYSIS

Figure 2 shows two example impulse responses collected in the open-end mode. The main spike of the impulse response followed by some ripple, identified with circles, and reflections can be seen. The ripple is due to the holes along the tube. The holes could not be filled completely, and the resulting cavities created small reflections.

Due to the finite length of the tube, the microphone recordings contain reflections from both ends. The waves propagating through the tube are reflected at the open end and, coming back, they are reflected again from the loudspeaker. Reflections appear in pairs repeated in time and progressively attenuated along the response. The location of the impulses can also be seen to differ between the two measurements in Fig. 2 due to the increased distance of the microphone from the loudspeaker.

3.1. Impulse Response Analysis

The measured responses were windowed in time to remove harmonic distortion components and unwanted reflections. A processed impulse response is presented in Fig. 3. The frequency response exhibits losses in the high end of the spectrum caused by propagation losses through the tube. There are also some losses in the low frequencies caused by the windowing. Significant attenuations of 20 dB or more appear above about 300 Hz.

As expected, spectral analysis of the windowed responses exhibits highly attenuated behavior at very high frequencies, as seen in the example in Fig. 3. This can be caused in part by the effect of non-planar wave propagation above the cutoff of planar waves. The behavior of the spectrum in the extreme high end is very noisy and, thus, unreliable.

The group delay was also computed. It showed an approximately flat response, indicating no time delay between the various sinusoidal components of the signal. This suggests that a delay line is suitable for simulating the propagation delay.

Figure 4 (left) shows the impulse responses recorded at three different holes. The time delay and the propagation loss can be

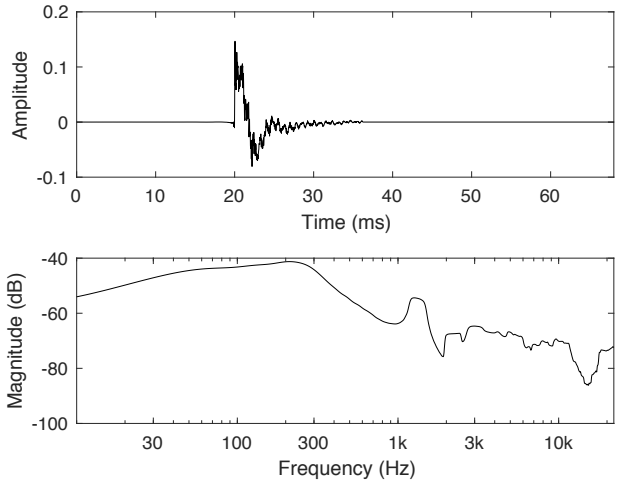


Figure 3: *Example of windowed impulse response (top) obtained from the* 2.5*-cm tube and its magnitude spectrum (bottom).*

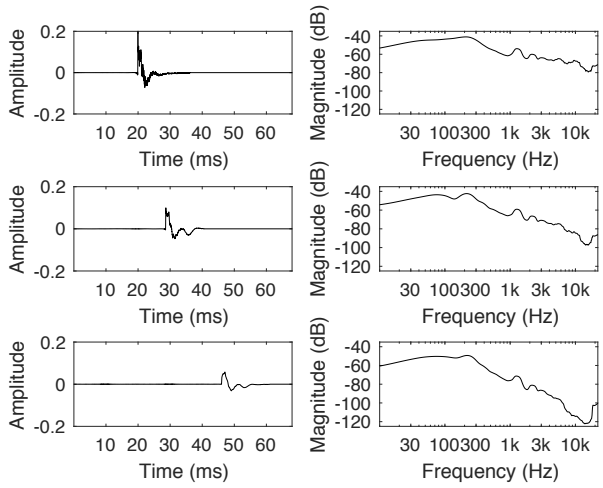


Figure 4: *Impulse response measured (left) and corresponding magnitude responses (right) in the* 1.9*-cm garden hose at the distance of* 2.5 *cm (top),* 3.25 *m (middle), and* 9.25 *m (bottom) from the loudspeaker.*

observed here. Their corresponding frequency contents are shown in Fig. 4 (right), and they reveal an increase of the attenuation with the increasing distance traveled and more significant losses at high frequencies when compared to low frequencies.

In addition, our measurements show that energy losses at high frequencies depend on the diameter of the tube. This behavior can be observed in Fig. 5, where the windowed responses captured at 4.25 m from the beginning of the tube, together with their corresponding frequency spectra, are shown for the three different tube diameters 1.2, 1.9, and 2.5 cm. The attenuation is seen to increase with decreasing diameter, showing more losses especially at high frequencies.

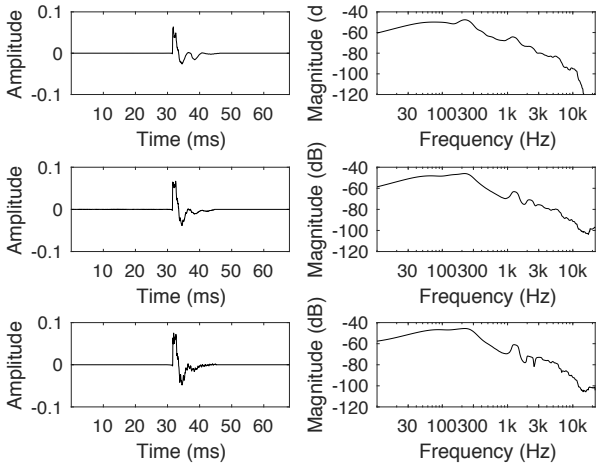


Figure 5: *Impulse response measured (left) and corresponding magnitude responses (right) at the distance of* 4.25 *m in the* 1.2*-cm (top),* 1.9*-cm (middle), and* 2.5*-cm (bottom) tube.*

3.2. Reflection Analysis

Figure 2 shows the behavior of the reflections at the closest and the farthest hole to the loudspeaker. The negative reflection and the positive one can be clearly seen. The gap between reflections depends on the position of the microphone which recorded them. The farthest hole is 9.25 m from the loudspeaker and 15.75 m from the open end, which means a longer distance for the reflections to meet the microphone.

The reflections were windowed as well. The analysis shows that energy exhibits losses in the high end of the spectrum and, instead, it is concentrated in the low frequencies. Figure 6 shows the windowed reflection result at the tube end together with its spectrum. From 300 Hz up to 1.5 kHz the spectrum exhibits a steep slope and above that extreme low energy values. The inverted pressure pulse due the open end can also be noticed.

4. VIRTUAL TUBE MODEL

The spectra of all the windowed signals were analyzed collectively. More specifically, in order to analyze the spectral changes associated with each meter traveled through the tube, the differences in the spectra of the respective signals were computed with the following equation:

$$\frac{H^i_{\text{dB}}(f) - H^j_{\text{dB}}(f)}{d_{ij}} \quad \forall i, j \,, \tag{1}$$

where $H_{\text{dB}}(f)$ is the spectrum magnitude of the signal in decibels, smoothed with a third-octave filter, and d_{ij} the distance in meters between the i-th and j-th holes, where the signals were recorded. These differences were computed for each tube. Then, the arithmetic mean of the results obtained was computed for each tube. In this way, an average behavior for a 1 m segment of each tube was obtained. The results are shown together in Fig. 7.

It can be noticed that the attenuation increases towards the high end of the spectrum and that it depends on the tube diameter.

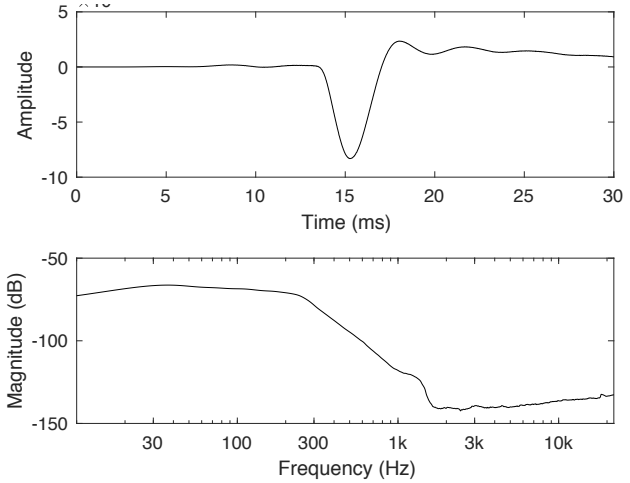


Figure 6: *A windowed reflection (top) and its magnitude spectrum (bottom) recorded with the* 1.9*-cm tube.*

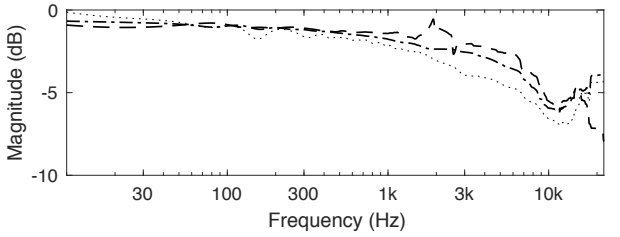


Figure 7: *Average "difference filters" for a* 1*-m segment (see Eq. (1)) of a* 1.2*-cm (dotted line), a* 1.9*-cm (dash-dot line), and a* 2.5*-cm (dashed line) tube.*

Increasing diameters result in a steeper shape, but with smaller attenuation. The responses below 300 Hz, despite some oscillations, are very similar to each other near 0 dB. Attenuation is noticeable above 300 Hz and becomes more significant around 1 kHz.

Since the first modes of the tubes are at 8054 Hz, 10598, and 16780, the results above these frequencies are unreliable. For this reason, the responses above these frequencies were not considered, and a continuous slope for the frequencies larger than 10 kHz in the design of the filters was taken.

Based on the above considerations, the spectrum can be assumed to have a low-pass shape. Increasing the tube diameter decreases the spectral slope and increases the cutoff frequency.

4.1. Reflections from the End of the Tube

In order to understand the effect of the open end on the responses, a different approach was chosen. Using the acquired information, the impulse response measured at the farthest hole from the loudspeaker was filtered with the filter approximating an appropriate power of the 1 m segment shape of Fig. 7. The filter design procedure will be described in Sec. 5. The aim here was to simulate the losses of the same distance that the reflected pulse had traveled. This simulation could be compared with the reflection, separating the reflection effect of the open end. The distance traveled by

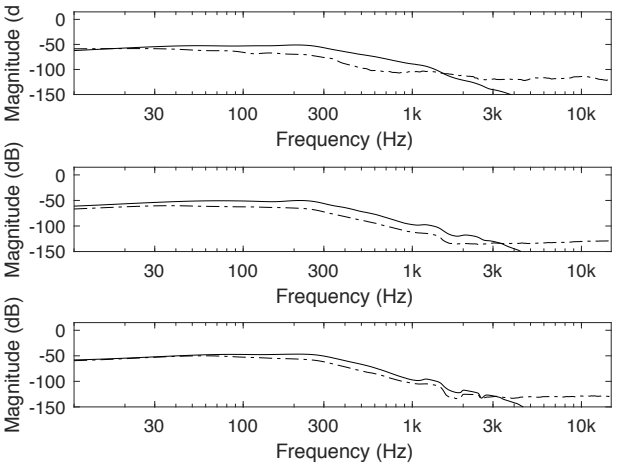


Figure 8: *Comparison between the spectrum of the reflection captured by the microphone (dash-dot line), and the simulated spectrum as it should be without the open end effect (solid line):* 1.2*-cm (top) ,* 1.9*-cm (middle) and* 2.5*-cm (bottom) diameter tubes.*

the reflection was computed and used to build the filter, accounting for the approximation error which becomes significant for long distances.

Figure 8 shows the spectrum of the reflection captured by the microphone and the simulated spectrum as it should be without the open-end effect. A slight attenuation can be seen below 100 Hz, and a stronger one up to 1 kHz. Since the impulse travels along the whole tube before reaching the open end, it has very low energy above 3 kHz and the recorded reflection is superimposed by the noise. When the impulse crosses the boundary at open end, the pressure wave hits the outside air, at atmospheric pressure, creating a compression wave heading back down the tube with some energy left.

Using the filter designed for the tube model, the effect of the reflection R due the open end was obtained:

$$R = \frac{H^i_{\text{ref}}(f)}{H^i_{\text{sim}}(f)}, \tag{2}$$

where $H_{\text{sim}}(f)$ is the spectrum of the response without the open end effect simulated with the approach described above using the same distance traveled by the corresponding windowed reflection $H_{\text{ref}}(f)$. This allows for the estimation of how the reflection affects the spectrum. Equation (2) was estimated for each measure where the reflections were isolated enough and could be windowed. Finally, the average for each tube size was computed. The shapes shown in Fig. 9 summarize the results.

The results show that the attenuation depends on the diameter of the tube, starting with a low value increasing above 100 Hz. The attenuation becomes smaller at higher frequencies because of the noise level.

5. FILTER DESIGN

This section describes the design of the filters simulating the sound propagation through the tube and the reflection effect by the open

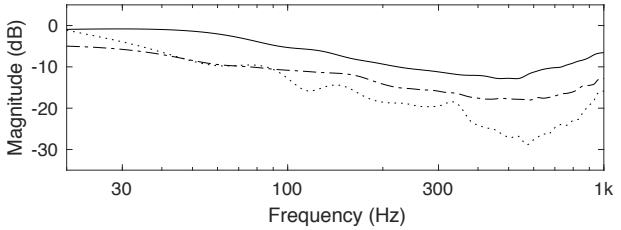


Figure 9: *Average filters estimating the open end effect (see Eq. (2)) of a* 1.2*-cm (dotted line),* 1.9*-cm (dash-dot line), and* 2.5*-cm (dashed line) tube.*

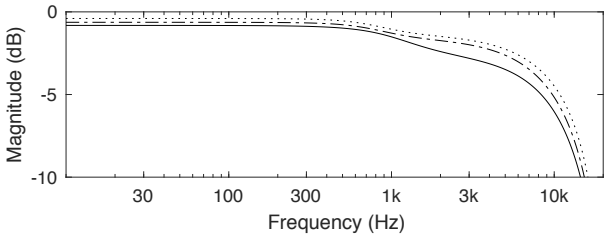


Figure 10: *Low-order approximations of the average "difference filters" for a* 1*-m segment (see Eq. (1)):* 1.2*-cm (solid line),* 1.9*-cm (dash-dot line), and* 2.5*-cm (dotted line) diameter tubes.*

end. For each of these effects, the average filters previously computed and summarized in Figs. 7 and 9 were used as target shapes to be approximated with low-order filters. Then, a unique form to interpolate between the different diameters values was found.

5.1. Propagation Filter

Given the simple shapes of these filters (see Fig. 7), attempts were made to find a low-order filter simulating their behavior. Keeping the three averages as targets, three parametric filters were computed, approximating the shape in order to minimize audible errors.

A cascade of two high-shelving filters and one low-pass filter was built, resulting in a 5^{th}-order parametric filter. The high-shelving filters were used to approximate the shape from 300 Hz to 3 kHz, while the low-pass filter was needed to cut the high end of the spectrum.

Since the three target shapes behave very similarly at low frequencies, the filters have the same behavior until 300 Hz with a slight attenuation depending on the diameter of the tube. The significant variations are in the range above 1 kHz, where different attenuations and cut-offs can be seen. The cut-off frequencies for the three target shapes are 4062, 5950, and 7015 Hz, respectively.

Figure 10 shows the different filters designed for the three diameter tubes to be compared with those in Fig. 7. With these low-order filters, a tube with arbitrary length can be simulated. Moreover, interpolating between the three filters allows to simulate different diameters sizes.

Since a cascade is an inefficient approach to produce tubes longer than 1 m, an approximation was found. Starting from the filter computed for the 1.2 cm tube, all the parameters of the three basic filters composing it were gradually varied in a linear way to achieve an approximated filter for longer lengths. A cascade of two 1^{st}-order low-pass filter replaced the simple 1^{st}-order one, resulting in a 6^{th}-order parametric filter. A good approximation up to 30 m (which is sufficient for the purpose of the audio effect) was obtained with an error smaller than 0.6 dB. In addition, with this method a better accuracy creating the tube can be achieved. Instead of 1 m as the incremental step, a finer control, like 1 cm, can be implemented. Figure 11 shows the approximation for 30 m. The designed filter follows accurately the general shape except for a critical range between 300 Hz and 1 kHz. In the case of 30 m tube, the maximum error is 0.57 dB.

After obtaining an accurate approximation of frequency attenuations due to propagation in the tube, the final filter was obtained by using a delay line that simulates the propagation delay and is

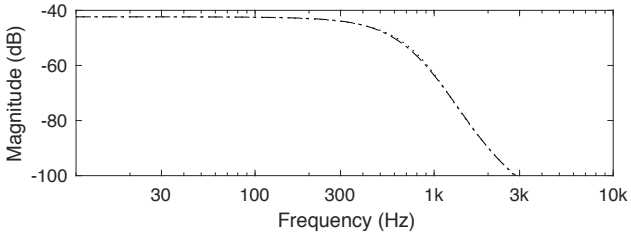


Figure 11: *Example of a parametric filter designed to approximate* 30 *m long tube: target filter (dotted line), and approximation (dash-dot line).*

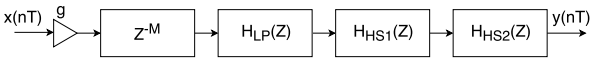


Figure 12: *Modeling the sound propagation using a delay line and three filters.*

connected in series with the previously discussed filter.

Figure 12 shows the three parametric filters in cascade and the delay line composing the system. The system can be described mathematically as follows:

$$H_{\text{tube}}(z) = gz^{-M} H_{\text{HS1}}(z) H_{\text{HS2}}(z) H_{\text{LP}}(z), \tag{3}$$

where g is a gain factor, z^{-M} is the delay line of M samples, $H_{\text{HS1}}(z)$ and $H_{\text{HS2}}(z)$ are 2^{nd}-order IIR high-shelving filters, and $H_{\text{LP}}(z)$ is a 1^{st}-order IIR low-pass filter.

The coefficients of the high-shelving and low-pass filters were computed with the usual formulas of the 1^{st}- and 2^{nd}-order filters [20]. Three different IIR filters were designed, one for each tube diameter (1.2, 1.9, 2.5 cm), giving the possibility to approximate the different behaviors by controlling the shape with the cut-off frequencies of the designed IIR digital filter.

In order to control the filter behavior as a function of the diameter of the simulated tube, the cut-off frequencies of all the filters and the gain factor g are linearly varied while the gains (dB) and the quality factors of the two high-shelving filters are kept fixed. Table 1 reports these latter values while Table 2 summarizes the filter cut-off frequencies and the gain factor for each tube diameter. Starting from these values, an interpolation was made with a granularity of 1 mm.

Table 1: *Propagation filter: gain and quality factor values for the two high-shelving filters.*

Type of filter	$G[dB]$	Q
HS1	-1	0.65
HS2	-0.9	0.5

Table 2: *Propagation filter: cut-off frequencies of low-pass and high-shelving filters and overall gain for the three tube diameters.*

Type of filter	f_{HS1} [Hz]	f_{HS2} [Hz]	f_{LP} [Hz]	g
1.2 cm	1200	1500	9500	0.85
1.9 cm	900	7000	10200	0.87
2.5 cm	900	7000	11000	0.90

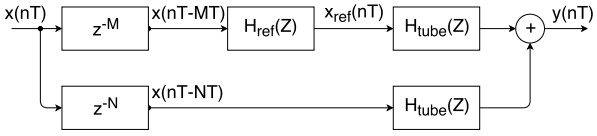


Figure 13: *Block diagram of the reflection simulation.*

5.2. Reflections

The block scheme in Fig. 13 shows the approach used to simulate the reflection. The delayed input is first filtered with the filter $H_{\text{ref}}(z)$ that approximates the losses given by the open end reflection, and the output is fed to the filter $H_{\text{tube}}(z)$ that simulates the losses caused by sound propagation in the tube. The computed reflection is finally added to the delayed sound resulting from unperturbed propagation in the tube.

The measured reflections have extremely low values in the high end of the spectrum (above 3 kHz) because of the long distance traveled. The simulation produces lower values in the high frequency region than the measured values. The extremely low values superimposed by noise produce unreliable results in this region of the spectrum. Since a steeper shape in the high frequency side due to high frequencies losses were expected, an approximation of the differences found with a continuous slope was done.

In order to approximate $H_{\text{ref}}(z)$, a cascade of a 2^{nd}-order high-shelving filter and a 1^{st}-order low-pass was chosen. Similarly to the propagation filter, by controlling the quality factors, the gains, and the cut-off frequencies, we were able to perform a linear interpolation between different diameters. An additional gain factor g_{ref} was introduced to control the scale for the different sizes. Table 3 summarizes the parameters values of the different filters.

6. COMPARISON

In this section, a comparison between the designed filters and the measurements is performed. The accuracy of the design is discussed, presenting the maximum approximation error in the frequency range of interest. Considering that the frequencies above 10 kHz are unreliable, as discussed in Sec. 4, the comparison refers the range between 20 Hz and 10 kHz.

Table 3: *Reflection filter: parameters of the low-pass and high-shelving filter and overall gain for the three tube diameters.*

Type of filter	1.2 cm	1.9 cm	2.5 cm
f_{LP} [Hz]	100	250	600
f_{HS} [Hz]	500	225	160
G_{HS} [dB]	-14	-14	-10
Q_{HS}	0.35	0.4	0.60
g_{ref}	0.4225	0.5180	0.9

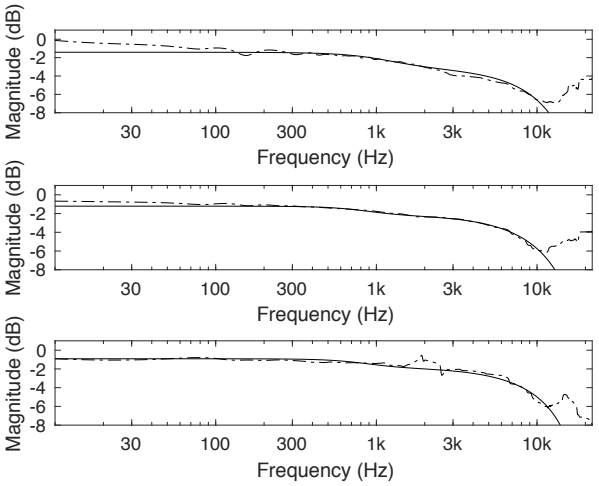


Figure 14: *Filters designed (solid line) and their corresponding targets (dash-dot line) for the* 1.2*-cm (top),* 1.9*-cm (middle) and* 2.5*-cm (bottom) tube.*

6.1. Propagation Filter

Figure 14 shows the three designed propagation filters compared with the results obtained from the measurements. The filter approximating the 1.2-cm tube has a maximum error of 0.97 dB, which is mainly due to the shelf filter having a flat magnitude response at low frequencies instead of the declining slope of the measured response as shown in the top of Fig. 14. This way, a good approximation at high frequencies is obtained, which is considered to be more important that the response below 100 Hz.

The 1.9-cm filter presents a maximum error of 0.5 dB in the lowest part of the frequency range. The fit becomes very accurate at higher frequencies as seen in Fig. 14 (middle). The error is 0.31 dB at 60 Hz and decreases close to zero at frequencies above 100 Hz.

The third filter is shown in Fig. 14 (bottom) that, with the exception of an anomaly at about 1900 Hz, also fits the target shape with good accuracy. It has a maximum error of 0.5 dB at 6184 Hz, and an error smaller than 0.3 dB in the rest of the frequency range.

6.2. Reflection Filter

Figure 15 shows the difference between the three designed reflection filters and the simulation results. In this case, the range between 20 and 500 Hz is significant for the comparison as discussed in Sec. 4.1.

The filter for the 1.2-cm diameter tube, shown in the top of

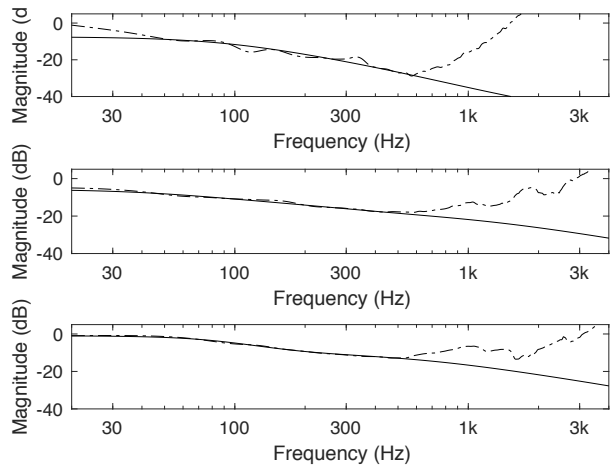


Figure 15: *Filters designed for the reflection (solid line) and their corresponding targets (dash-dot line) of the* 1.2*-cm (top),* 1.9*-cm (middle) and* 2.5*-cm (bottom) size tube.*

Fig. 14, presents the same initial behavior of the one compared in the previous section. Because of the high variability in the magnitude target, it is difficult to approximate accurately the shape, and the maximum error is 4.57 dB. The error becomes smaller than 1 dB after 60 Hz except for a deviation at 330 Hz where the error is 3.57 dB. Also in this design, a better approximation for frequencies higher than 60 Hz at the expense of the frequencies below was done.

The reflection filter for the 1.9-cm tube can be seen in the middle of Fig. 15. In the beginning of the spectrum, it has a maximum error of 1.26 dB. The error becomes smaller than 1.2 dB above 30 Hz, thus providing a good fit in the remaining range.

The third filter, as seen in Fig. 15 (bottom), is the most accurate with a maximum error of 0.52 dB at 40 Hz and close to zero above 100 Hz.

7. IMPLEMENTATION

The implementation was written in C++ as an external library for PURE DATA, an open-source real-time environment for audio processing. The stereo plugin, working at sample rate 44.1 kHz, simulates the wave propagation in a narrow tube and produces associated audio effects. It creates two virtual tubes, one for each channel. The diameter of the two tubes is always the same. The length of each tube can be set by the user and determines the desired delay in milliseconds. The speed of sound is assumed to be 345 m/s corresponding to a temperature of 23°C.

In addition, it is possible to control the volume of the delayed sound and the ratio of the dry and the wet signals in the output. The filter simulates the tube length for each 1 cm added. However, the size parameter gives the possibility to change the virtual tube diameter with a granularity of 1 mm by changing the filter parameters.

To enrich the system, the possibility of summing a reflection in the output was also implemented. This option simulates the wave reflection due the open end of the tube. A reflection, whose frequency content depends on the distance chosen for the "virtual open end," can be created for each virtual tube. This way, the length of the virtual tube becomes the sum of the length chosen for the delay effect and the length chosen in the reflection options. The sound is captured at a virtual microphone at the distance selected by combining the delayed part of the sound and the reflection coming from the end of the tube. Since the reflection captured this way is too soft to be clearly audible, a gain control was added.

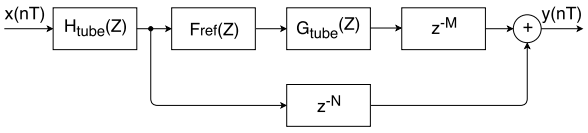


Figure 16: *Block scheme for the audio flow in the plugin.*

Including the reflection option, the system computes three filters: the filter simulating the length desired for the main delay, the filter simulating the open end, and the one simulating the residual length traveled by the sound to reach the end of the tube and come back to meet the virtual microphone. The block scheme shown in Fig. 16 summarizes the system. The residual length is represented by $G_{\text{tube}}(z)$ and is twice the length chosen in the reflection options. In order to decrease the complexity of the computation, the different coefficients of the reflection filters were pre-computed and stored.

The plugin offers the possibility to create virtual tubes up to 30 m long in default mode, and 40 m long tubes in the reflection mode. These maximums correspond to a delay of 87 ms and a reflection coming after 29 ms. Figure 17 shows a screenshot of the plugin implemented in PURE DATA.

8. CONCLUSION

A simulation of a tube delay effect was proposed in this paper. Acoustic wave propagation in garden hoses of three different diameter was measured and analyzed. Studying and elaborating the recorded tube responses, a virtual tube model was developed and a digital IIR filter controlling the length and the diameter of the virtual tube was designed with a negligible error. From the analysis of the measurements, a parametric filter was designed in which the tube diameter and length can be continuously varied. Because of the simplicity of the magnitude response shapes, a cascade of two high shelving filters and a low-pass filter was sufficient for approximating the behavior correctly. In addition, an analysis on the reflection due to the open end of the tube was conducted, and a filter approximating it was added in the model. Finally, a stereo delay effect plugin in PURE DATA was presented describing the design specifications.

9. ACKNOWLEDGMENT

This research work was conducted between August 2017 and January 2018, when Riccardo Simionato was visiting the Aalto Acoustics Lab within the framework of the Erasmus+ program.

10. REFERENCES

[1] U. Zölzer, Ed., *DAFX – Digital Audio Effects*, John Wiley & Sons, 2. edition, 2011.

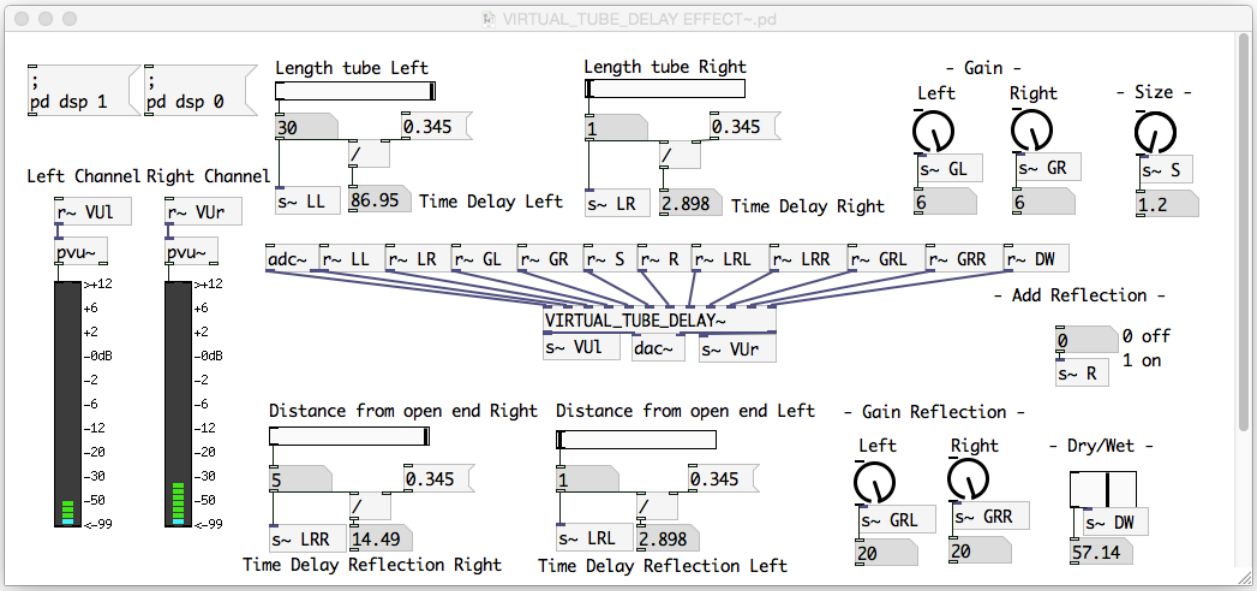


Figure 17: *The virtual tube delay effect plugin in* PURE DATA.

[2] H. F. Olson and J. C. Bleazey, "Synthetic reverberator," *J. Audio Eng. Soc.*, vol. 8, no. 1, pp. 37–41, Jan. 1960.

[3] S. Arnardottir, J. S. Abel, and J. O. Smith, "A digital model of the Echoplex tape delay," in *Proc. Audio Eng. Soc. 125th Conv.*, San Francisco, CA, USA, Oct. 2008.

[4] C. Raffel and J. Smith, "Practical modeling of bucket-brigade device circuits," in *Proc. 13th Int. Conf. Digital Audio Effects (DAFx'10)*, Graz, Austria, Sept. 2010.

[5] J. S. Abel, D. P. Berners, S. Costello, and J. O. Smith III, "Spring reverb emulation using dispersive allpass filters in a waveguide structure," in *Proc. Audio Eng. Soc. 121st Conv.*, San Francisco, CA, USA, Oct. 2006.

[6] S. Bilbao and J. Parker, "A virtual model of spring reverberation," *IEEE Audio, Speech, Language Process.*, vol. 18, no. 4, pp. 799–808, May 2010.

[7] V. Välimäki, J. Parker, and J. S. Abel, "Parametric spring reverberation effect," *J. Audio Eng. Soc.*, vol. 58, no. 7/8, pp. 547–562, Jul./Aug. 2010.

[8] S. Bilbao, "A digital plate reverberation algorithm," *J. Audio Eng. Soc.*, vol. 55, no. 3, pp. 135–144, Mar. 2007.

[9] J. S. Abel, D. P. Berners, and A. Greenblatt, "An emulation of the EMT 140 plate reverberator using a hybrid reverberator structure," in *Proc. Audio Eng. Soc. 127th Conv.*, New York, USA, Oct. 2009.

[10] J. Parker and S. Bilbao, "Spring reverberation: A physical perspective," in *Proc. 12th Int. Conf. Digital Audio Effects (DAFx'09)*, Sep. 2009, pp. 416–421.

[11] K. Arcas, "Physical modelling and measurements of plate reverberation," in *Proc. ICA*, Madrid, Spain, Sep. 2009.

[12] F. Fontana and D. Rocchesso, "Auditory distance perception in an acoustic pipe," *ACM Trans. Appl. Percpt.*, vol. 5, no. 3, 2008, Article 16.

[13] M. Geronazzo, F. Avanzini, and F. Fontana, "Auditory navigation with a tubular acoustic model for interactive distance cues and personalized head-related transfer functions," *J. Multimodal User Interfaces*, vol. 10, no. 3, pp. 273–284, Sep. 2016.

[14] V. Välimäki, *Discrete-Time Modeling of Acoustic Tubes using Fractional Delay Filters*, Ph.D. thesis, Helsinki University of Technology, Espoo, Finland, 1995.

[15] D. P. Berners, *Acoustics and Signal Processing Techniques for Physical Modeling of Brass Instruments*, Ph.D. thesis, Stanford University, Stanford, CA, USA, 1999.

[16] J. O. Smith, "Principles of digital waveguide models of musical instruments," in *Applications of Digital Signal Processing to Audio and Acoustics*, Kahrs M. and Brandenburg K., Eds., pp. 417–466. Springer, 2002.

[17] T. Smyth and J. Abel, "Estimating waveguide model elements from acoustic tube measurements," *Acta Acustica united with Acustica*, vol. 95, no. 6, pp. 1093–1103, 2009.

[18] D. B. Sharp, *Acoustic Pulse Reflectometry for the Measurement of Musical Wind Instruments*, Ph.D. thesis, The University of Edinburgh, Edinburgh, UK, 1996.

[19] A. Farina, "Simultaneous measurement of impulse response and distortion with a swept-sine technique," in *Proc. Audio Eng. Soc. 108th Conv.*, Paris, France, Feb. 2000.

[20] P. Dutilleux, M. Holters, S. Disch, and U. Zölzer, "Filters and delays," in *DAFX: Digital Audio Effects, Second Edition*, U. Zölzer, Ed., pp. 47–81. Wiley, 2011.

[21] V. Välimäki, S. Bilbao, J. Smith, J. Abel, J. Pakarinen, and D. Berners, "Virtual analog effects," in *DAFX: Digital Audio Effects, Second Edition*, U. Zölzer, Ed., pp. 473–522. Wiley, 2011.

GENERATIVE TIMBRE SPACES: REGULARIZING VARIATIONAL AUTO-ENCODERS WITH PERCEPTUAL METRICS

*Philippe Esling** *Axel Chemla–Romeu-Santos, Adrien Bitton*

Institut de Recherche et Coordination Acoustique-Musique (IRCAM)
CNRS - UMR 9912, UPMC - Sorbonne Universite
1 Place Igor Stravinsky, F-75004 Paris, France
esling@ircam.fr

ABSTRACT

Timbre spaces have been used in music perception to study the perceptual relationships between instruments based on dissimilarity ratings. However, these spaces do not generalize to novel examples and do not provide an invertible mapping, preventing audio synthesis. In parallel, generative models have aimed to provide methods for synthesizing novel timbres. However, these systems do not provide an understanding of their inner workings and are usually not related to any perceptually relevant information.
Here, we show that Variational Auto-Encoders (VAE) can alleviate all of these limitations by constructing *generative timbre spaces*. To do so, we adapt VAEs to learn an audio latent space, while using perceptual ratings from timbre studies to regularize the organization of this space. The resulting space allows us to analyze novel instruments, while being able to synthesize audio from any point of this space. We introduce a specific regularization allowing to enforce any given similarity distances onto these spaces. We show that the resulting space provide almost similar distance relationships as timbre spaces. We evaluate several spectral transforms and show that the Non-Stationary Gabor Transform (NSGT) provides the highest correlation to timbre spaces and the best quality of synthesis. Furthermore, we show that these spaces can generalize to novel instruments and can generate any path between instruments to understand their timbre relationships. As these spaces are continuous, we study how audio descriptors behave along the latent dimensions. We show that even though descriptors have an overall non-linear topology, they follow a locally smooth evolution. Based on this, we introduce a method for *descriptor-based synthesis* and show that we can control the descriptors of an instrument while keeping its timbre structure.

1. INTRODUCTION

For the past decades, music perception research has tried to understand the perception of instrumental *timbre*. Timbre is the set of properties that distinguishes two instruments that play the same note at the same intensity. To do so, several studies [1] collected human dissimilarity ratings between pairs of audio samples inside a set of instruments. These ratings are organized by applying MultiDimensional Scaling (MDS), leading to *timbre spaces*, which exhibit the perceptual similarities between different instruments. By analyzing the dimensions of resulting spaces, the studies tried to correlate audio descriptors to the perception of timbre [2]. Although these spaces provided interesting avenues of analysis, they are inherently limited by the fact that ordination techniques (e.g. MDS) produce a fixed space, which has to be recomputed entirely for any new sample. Therefore, these spaces do not generalize to novel examples and do not provide an invertible mapping, precluding audio synthesis to understand their perceptual topology.

In parallel, recent developments in audio synthesis using *generative models* has seen great improvements with the introduction of approaches such as the *WaveNet* [3] and *SampleRNN* [4] architectures. These allow to generate novel high-quality audio matching the properties of the corpus they have been trained on. However, these models give little cue and control over the output or the features it results from. More recently, *NSynth* [5] has been proposed to synthesize audio by allowing to morph between specific instruments. However, these models still require very large number of parameters, long training times and a large number of examples. Amongst recent generative models, another key proposal is the *Variational Auto-Encoder* (*VAE*) [6]. In these, a *latent space* is learned that allows both to encode data for analysis, but also to sample from it in order to generate novel content. VAEs address the limitations of control and analysis through this latent space, while remaining simple and fast to learn with a small set of examples. Furthermore, VAEs seem able to disentangle underlying variation factors by learning independent latent variables accounting for distinct generative processes [7]. However, these latent dimensions are learned in an unsupervised way. Therefore, they are not related to perceptual properties, which might hamper their understandability or their use for audio analysis and synthesis.

Here, we show that we can bridge timbre perception analysis and perceptually-relevant audio synthesis by regularizing the learning of VAE latent spaces so that they match the perceptual distances collected from timbre studies. Our overall approach is depicted in Figure 1. First, we adapt the VAE to analyze musical audio content, by comparing the use of different spectral transforms as input to the learning. We show that, amongst the Short-Term Fourier Transform (STFT), Discrete Cosine Transform (DCT) and the Non-Stationary Gabor Transform (NSGT) [8], the NSGT provides the best reconstruction abilities and regularization performances. By training this model on a small database of spectral frames, it already provides a generative model with an interesting latent space, able to synthesize novel instrumental timbres. Then, we introduce a regularization to the learning objective inspired by the t-Stochastic Neighbors Embedding (t-SNE) [9], aiming to enforce that the latent space exhibits the same distances between instruments as those found in timbre studies. To do so, we build a model of perceptual relationships by analyzing dissimilarity ratings from five independent timbre studies [10, 11, 12, 13, 14]. We show that perceptually-regularized latent spaces are simultaneously coherent with perceptual ratings, while being able to synthe-

* This work was supported by project MAKIMOno 17-CE38-0015-01 funded by the French ANR and Canadian NSERC (STPG 507004-17) and the ACTOR Partnership funded by the Canadian SSHRC (895-2018-1023).

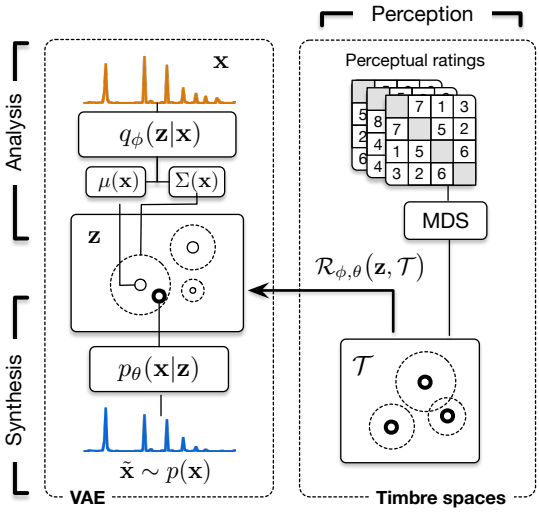


Figure 1: (*Left*) VAEs can model a spectral frame $\mathbf{x}$ of an audio sample by learning an encoder $q_\phi(\mathbf{z} \mid \mathbf{x})$ which maps them to a Gaussian $\mathcal{N}(\mu(\mathbf{x}), \sigma(\mathbf{x}))$ inside a latent space $\mathbf{z}$. The decoder $p_\theta(\mathbf{x} \mid \mathbf{z})$ samples from this Gaussian to generate a reconstruction $\tilde{x}$ of the spectral frame. (*Right*) Perception studies use similarity ratings to construct *timbre spaces* exhibiting perceptual distances between instruments. Here, we develop a regularization $\mathcal{R}(\mathbf{z}, \mathcal{T})$ enforcing that the variational model finds a topology of latent space $\mathbf{z}$ that matches the topology of the timbre space $\mathcal{T}$.

size high-quality audio distributions. Hence, we drive the learning of latent spaces to match the topology of given target spaces.

We demonstrate that these spaces can be used for generating novel audio content, by analyzing their reconstruction quality on a test dataset. Furthermore, we show that paths in the latent space (where each point corresponds to a single spectral frame) provide sound synthesis with continuous evolutions of timbre. We also show that these spaces generalize to novel samples, by encoding a set of instruments that were not part of the training set. Therefore, the spaces could be used to predict the perceptual similarities of novel instruments. Finally, we study how traditional audio descriptors are organized along the latent dimensions. We show that even though descriptors behave in a non-linear way across space, they still follow a locally smooth evolution. Based on this smoothness property, we introduce a method for *descriptor-based path synthesis*. We show that we can modify an instrumental distribution so that it matches a given target evolution of audio descriptors, while remaining perceptually smooth. The source code, audio examples and animations are available on a supporting repository[1].

2. STATE-OF-ART

2.1. Variational auto-encoders

Generative models are a flourishing class of learning approaches, which aim to find the underlying probability distribution of the data $p(\mathbf{x})$ [15]. Formally, based on a set of examples in a high-dimensional space $\mathbf{x} \in \mathbb{R}^{d_x}$, we assume that these follow an unknown distribution $p(\mathbf{x})$. Furthermore, we consider a set of *latent variables* defined in a lower-dimensional space $\mathbf{z} \in \mathbb{R}^{d_z}$ ($d_z \ll d_x$). These latent variables help govern the generation of the data and enhance the *expressivity* of the model. Thus, the complete model is defined by the joint probability distribution $p(\mathbf{x}, \mathbf{z}) = p(\mathbf{x} \mid \mathbf{z})p(\mathbf{z})$. We could find $p(\mathbf{x})$ through its relation to the posterior distribution $p(\mathbf{z} \mid \mathbf{x})$ given by Bayes' theorem. However, for complex non-linear models (such as those that we will consider in this paper), this posterior can not be found in closed form.

For decades, the dominant paradigm for approximating $p(\mathbf{x})$ has been *sampling* methods [16]. However, the quality of this approximation depends on the number of sampling operations, which might be extremely large before we have an accurate estimate. Recently, *variational inference* (VI) [15] has been proposed to solve this problem through *optimization* rather than sampling. VI assumes that if the distribution is too complex to find, we could find a simpler approximate distribution that still models the data, while trying to minimize its difference to the real distribution. Formally, VI specifies a family $\mathcal{Q}$ of approximate densities, where each member $q(\mathbf{z} \mid \mathbf{x}) \in \mathcal{Q}$ is a candidate approximation to the exact $p(\mathbf{z} \mid \mathbf{x})$. Hence, the inference problem can be transformed into an optimization problem by minimizing the Kullback-Leibler (KL) divergence between the approximation and original density

$$q^*(\mathbf{z} \mid \mathbf{x}) = \underset{q(\mathbf{z} \mid \mathbf{x}) \in \mathcal{Q}}{\arg\min}\, \mathcal{D}_{KL}\big[q(\mathbf{z} \mid \mathbf{x}) \parallel p(\mathbf{z} \mid \mathbf{x})\big] \quad (1)$$

The complexity of the family $\mathcal{Q}$ will both determine the quality of the approximation, but also the complexity of this optimization. Hence, the major issue of VI is to choose $\mathcal{Q}$ to be flexible enough to closely approximate $p(\mathbf{z} \mid \mathbf{x})$, while being simple enough to allow efficient optimization. Now, if we expand the KL divergence that we need to minimize and rely on Bayes' rule to replace $p(\mathbf{z}|\mathbf{x})$, we obtain the following expression

$$D_{KL}\big[q(\mathbf{z} \mid \mathbf{x}) \parallel p(\mathbf{z} \mid \mathbf{x})\big] = \mathbb{E}_{q(\mathbf{z})}\big[\log q(\mathbf{z} \mid \mathbf{x}) - \log p(\mathbf{x} \mid \mathbf{z}) - \log p(\mathbf{z}) + \log p(\mathbf{x})\big] \quad (2)$$

Noting that the expectation is over $q(\mathbf{z}|\mathbf{x})$ and that $p(\mathbf{x})$ does not depend on it, we can get this term out of the expectation and then observe that the remaining equation can be rewritten as another KL divergence leading to

$$\log p(\mathbf{x}) - D_{KL}\big[q(\mathbf{z} \mid \mathbf{x}) \parallel p(\mathbf{z} \mid \mathbf{x})\big] = \mathbb{E}_{\mathbf{z}}\big[\log p(\mathbf{x} \mid \mathbf{z})\big] - D_{KL}\big[q(\mathbf{z} \mid \mathbf{x}) \parallel p(\mathbf{z})\big] \quad (3)$$

This formulation describes the logarithm of the quantity that we want to maximize $\log p(\mathbf{x})$ minus the error we make by using an approximate q instead of p. Therefore, we can optimize this alternative objective, called the *evidence lower bound* (ELBO) as

$$\log p(\mathbf{x}) = D_{KL}\big[q(\mathbf{z} \mid \mathbf{x}) \parallel p(\mathbf{z} \mid \mathbf{x})\big] + ELBO(q). \quad (4)$$

and the KL is non-negative, so $\log p(\mathbf{x}) \geq ELBO(q), \forall q(\mathbf{z})$. Now, to optimize this objective, we will rely on parametric distributions $q_\phi(\mathbf{z} \mid \mathbf{x})$ and $p_\theta(\mathbf{x} \mid \mathbf{z})$. Therefore, optimizing our generative model will amount to optimize these parameters $\{\theta, \phi\}$

$$\mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\phi}) = \mathbb{E}_{q_\phi(\mathbf{z})}\big[\log p_\theta(\mathbf{x}|\mathbf{z})\big] - D_{KL}\big[q_\phi(\mathbf{z}|\mathbf{x}) \parallel p_\theta(\mathbf{z})\big] \quad (5)$$

[1] https://github.com/acids-ircam/variational-timbre

We can see that this equation involves $q_\phi(\mathbf{z} \mid \mathbf{x})$ which *encodes* the data $\mathbf{x}$ into the latent representation $\mathbf{z}$ and a *decoder* $p(\mathbf{x} \mid \mathbf{z})$, which generates a data $\mathbf{x}$ given a latent configuration $\mathbf{z}$. Hence, this whole structure defines the *Variational Auto-Encoder* (VAE), which is depicted in Figure 1 (Left).

The VAE objective can be interpreted intuitively. The first term increases the likelihood of the data generated given a configuration of the latent, which amounts to minimize the *reconstruction error*. The second term represents the error made by using a simpler distribution $q_\phi(\mathbf{z} \mid \mathbf{x})$ rather than the true distribution $p_\theta(\mathbf{z})$. Therefore, this *regularizes* the choice of approximation q so that

$$\mathcal{L}_{\boldsymbol{\theta},\boldsymbol{\phi}} = \underbrace{\mathbb{E}_{q_\phi(\mathbf{z})}\left[\log p_\theta(\mathbf{x}|\mathbf{z})\right]}_{\text{reconstruction}} - \beta \cdot \underbrace{D_{KL}\left[q_\phi(\mathbf{z}|\mathbf{x}) \parallel p_\theta(\mathbf{z})\right]}_{\text{regularization}} \quad (6)$$

The first term can be optimized through a usual maximum likelihood estimation, while the second term requires that we define the prior $p(\mathbf{z})$. While the easiest choice is to choose $p(\mathbf{z}) \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$, it also adds the benefit that this term has a simple closed solution for computing the optimization, as detailed in [6]. Here we introduced a weight β to the KL divergence, which leads to the β-VAE formulation [7]. This has been shown to improve the capacity of the model to disentangle factors of variations in the data. However, it has later been shown that an appropriate way to handle this parameter was to perform *warm-up* [17], where the β parameter is linearly increased in the first epochs of training.

Finally, we need to select a family of variational densities $\mathcal{Q}$. One of the most widespread choice is the *mean-field variational family* where latent variables are independent and are each parametrized by a distinct variational parameter

$$q(\mathbf{z}) = \prod_{j=1}^{m} q_j(z_j) \quad (7)$$

Therefore, each dimension of the latent space will be governed by an independent Gaussian distribution with its own mean and variance depending on the input data $q_j(z_j) = \mathcal{N}(\mu_j(\mathbf{x}), \Sigma_j(\mathbf{x}))$.

VAEs are powerful representation learning frameworks, while remaining simple and fast to learn without requiring large sets of examples [17]. Their potential for audio applications have been only scarcely investigated yet and mostly in topics related to speech processing such as blind source separation [18] and speech transformation [19]. However, to the best of our knowledge, the use of VAE and their latent spaces to perform musical audio analysis and generation has yet to be investigated.

2.2. Timbre spaces and auditory perception

For several decades, music perception research has tried to understand the mechanisms leading to the perception of *timbre*. Several studies have shown that timbre could be partially described by computing various audio descriptors [13]. To do so, most studies relied on the concept of *timbre spaces* [2], a model that organize audio samples based on perceptual dissimilarity ratings. In these studies, pairs of sounds are presented to subjects that are asked to rate their perceptual dissimilarities inside a given set of instruments. Then, these ratings are compiled into a set of dissimilarity matrices that are analyzed with Multi-Dimensional Scaling (MDS). The MDS algorithm provides a timbre space that exhibits the underlying perceptual distances between different instruments (Figure 1 (Right)). Here, we briefly detail corresponding studies and redirect interested readers to the full articles for more details. In his seminal paper, Grey [10] performed a study with 16 instrumental sound samples. Each of the 22 subjects had to rate the dissimilarity between all pairs of sounds on a continuous scale from 0 (most similar) to 1 (most dissimilar). This lead to the first construction of a timbre space for instrumental sounds. They further exhibit that the dimensions explaining these dissimilarities could be correlated to the spectral centroid, spectral flux and attack centroid. Several studies followed this research by using the same experimental paradigm. Krumhansl [11] used 21 instruments with 9 subjects on a discrete scale from 1 to 9, Iverson et al. [12] with 16 samples and 10 subjects on a continuous scale from 0 to 1, McAdams et al. [13] with 18 orchestral instruments and 24 subjects on a discrete scale from 1 to 16 and, finally, Lakatos [14] with 17 subjects on 22 harmonic and percussive samples on a continuous scale from 0 to 1. Each of these studies shed light on different aspects of audio perception, depending on the aspect being scrutinized and the interpretation of the space by the experimenters. However, all studies have led to different spaces with different dimensions. The fact that different studies correlate to different audio descriptors prevents a generalization of the acoustic cues that might correspond to timbre dimensions. Furthermore, timbre spaces have been explored based on MDS to organize perceptual ratings and correlate spectral descriptors [13]. Therefore, these studies are inherently limited by the fact that

- ordination techniques (such as MDS) produce fixed spaces that must be recomputed for any new data point
- these spaces do not generalize nor synthesize audio between instruments as they do not provide an invertible mapping
- interpretation is bounded to the *a posteriori* linear correlation of audio descriptors to the dimensions rather than analyzing the topology of the space itself

As noted by McAdams et al. [1], critical problems in these approaches are the lack of an objective distance model based on perception and general dimensions for the interpretation of timbral transformation and source identification. Here, we show that relying on VAE models to learn unsupervised spaces, while regularizing the topology of these spaces to fit given perceptual ratings can allow to alleviate all of these limitations.

3. REGULARIZING LATENT SPACE TOPOLOGY

In this paper, we aim to construct a latent space that could both analyze and synthesize audio content, while providing the underlying perceptual relationships between audio samples. To do so, we show that we can influence the organization of the VAE latent space $\mathbf{z}$ so that it follows the topology of a given target space $\mathcal{T}$. Here, we will rely on the MDS space constructed from perceptual ratings as a target space $\mathcal{T}$. However, it should be noted that this idea can be applied to any given target space that provides a set of distances between the elements used for learning the VAE space.

To further specify our problem, we consider a set of audio samples, where each x_i can be encoded in the latent space as $\mathbf{z}_i$ and have an equivalent in the target space $\mathcal{T}_i$. In order to relate the elements of the audio dataset to the perceptual space, we consider that each sample is labeled with its instrumental class $\mathcal{C}_i$, that has an equivalent in the timbre space. Therefore, we will match the properties of the classes between the latent and target spaces (note that we could use element-wise properties for finer control).

Here, we propose to regularize the learning by introducing the perceptual similarities through an additive term $\mathcal{R}(\mathbf{z}, \mathcal{T})$. This *penalty* imposes that the properties of the latent space $\mathbf{z}$ are similar to that of the target space $\mathcal{T}$. The optimization objective becomes

$$\mathbb{E}\big[\log p_\theta(\mathbf{x}|\mathbf{z})\big] - \beta D_{KL}\big[q_\phi(\mathbf{z}|\mathbf{x}) \parallel p_\theta(\mathbf{z})\big] + \alpha\mathcal{R}\big(\mathbf{z}, \mathcal{T}\big) \quad (8)$$

where α is an hyper-parameter that allows us to control the influence of the regularization. Hence, amongst two otherwise equal solutions, the model is pushed to select the one that comply with the penalty. In our case, we want the distances between instruments to follow perceptual timbre distances. Therefore, we need to minimize the differences between the set of distances in the latent space $\mathcal{D}^{\mathbf{z}}_{i,j} = \mathcal{D}(\mathbf{z}_i, \mathbf{z}_j)$ and the distances in target space $\mathcal{D}^{\mathcal{T}}_{i,j} = \mathcal{D}(\mathcal{T}_i, \mathcal{T}_j)$. Therefore, the regularization criterion will try to minimize the overall differences between these sets of distances. To compute these sets, we take inspiration from the *t-Stochastic Neighbor Embedding* (t-SNE) algorithm [9]. Indeed, as their goal is to map the distances from one (high-dimensional) space into a target (low-dimensional) space, it is highly correlated to our task. However, we can not simply apply the t-SNE algorithm on the latent space as this would lead to a non-invertible mapping. Instead, we aim to steer the learning in a similar way. Hence, we compute the relationships in the latent space $\mathbf{z}$ by using the conditional Gaussian density that i would choose j as its neighbor

$$\mathcal{D}^{\mathbf{z}}_{i,j} = \frac{\exp\big(-\|\mathbf{z}_i - \mathbf{z}_j\|^2 / 2\sigma_i^2\big)}{\sum_{k\neq i}\exp\big(-\|\mathbf{z}_i - \mathbf{z}_k\|^2 / 2\sigma_i^2\big)} \quad (9)$$

where σ_i is the variance of the Gaussian centered on $\mathbf{z}_i$, defined as $\sigma_i = 1/\sqrt{2}$. Then, to relate the points in the timbre space $\mathcal{T}$, we use a Student-t distribution to define the distances in this space as

$$\mathcal{D}^{\mathcal{T}}_{i,j} = \frac{\big(1 + \|\mathcal{T}_i - \mathcal{T}_j\|^2\big)^{-1}}{\sum_{k\neq l}\big(1 + \|\mathcal{T}_k - \mathcal{T}_l\|^2\big)^{-1}} \quad (10)$$

Finally, we rely on the sum of KL divergences between the two distributions of distances in different spaces to define our complete regularization criterion

$$\mathcal{R}\big(\mathbf{z}, \mathcal{T}\big) = \sum_i \mathcal{D}_{KL}\big[\mathcal{D}^{\mathbf{z}}_i \parallel \mathcal{D}^{\mathcal{T}}_i\big] = \sum_i\sum_j \mathcal{D}^{\mathbf{z}}_{i,j}\log\frac{\mathcal{D}^{\mathbf{z}}_{i,j}}{\mathcal{D}^{\mathcal{T}}_{i,j}}$$

Hence, instead of applying a distance minimization a posteriori, we steer the learning to find a configuration of the latent space $\mathbf{z}$ that displays the same distance properties as the space $\mathcal{T}$, while providing an invertible mapping.

4. EXPERIMENTS

4.1. Datasets

Timbre studies. We rely on the perceptual ratings collected across five independent timbre studies [10, 11, 12, 13, 14]. As discussed earlier, even though all studies follow the same experimental protocol, there are some discrepancies in the choice of instruments, rating scales and sound stimuli. However, here we aim to obtain a consistent set of properties to define a common timbre space. Therefore, we computed the maximal set of instruments for which we had ratings for all pairs. To do so, we collated the list of instruments from all studies and counted their co-occurences, leading to a set of 12 instruments (Piano, Cello, Violin, Flute, Clarinet, Trombone, French Horn, English Horn, Oboe, Saxophone, Trumpet, Tuba) with pairwise ratings. Then, we normalized the raw dissimilarity data (keeping all instruments of that study) so that it maps to a common scale from 0 to 1. Finally, we extracted the set of ratings that corresponds to our selected instruments. This leads to a total of 1217 subject ratings for all instruments, amounting to 11845 pairwise ratings. Based on this set of ratings, we compute an MDS space to ensure the consistency of our normalized perceptual space on the selected set. The results of this analysis are displayed in Figure 2. We can see that even though the ratings come from different studies, the resulting space remains very coherent, with the distances between instruments remaining coherent with the original perceptual studies.

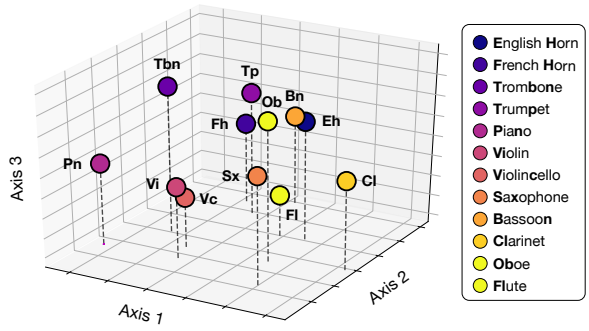


Figure 2: Multi-dimensional scaling (MDS) of the combined and normalized set of perceptual ratings from different studies.

Audio datasets. In order to learn the distribution of instrumental sounds directly from the audio signal, we rely on the Studio On Line (SOL) database [20]. We selected 2,200 samples to represent the 11 instruments for which we extracted perceptual ratings. We normalized the range of notes used by taking the whole tessitura and dynamics available (to remove effects from the pitch and loudness). All recordings were resampled to 22050 Hz for the experiments. Then, as we intend to evaluate the effect of different spectral distributions as input to our proposed model, we computed several invertible transforms for each audio sample. First, we compute the Short-Term Fourier Transform (STFT) with a Hamming window of 40ms and a hop size of 10ms. Then, we compute the Discrete Cosine Transform (DCT) with the same set of parameters. Finally, we compute the Non-Stationary Gabor Transform (NSGT) [8] mapped either on a Constant-Q scale of 48 bins per octave and a Mel scale or ERB scale of 400 bins, all from 30 to 11000 Hz. For all transforms, we only keep the magnitude of the distribution to train our models. We perform a corpus-wide normalization to preserve the relative intensities of the samples (normalizing all distributions by the maximal value found across samples). Then, we extract a single temporal frame from the sustained part of the representation (200 ms after the beginning of the sample) to represent a given audio sample. Finally, the dataset is randomly split across notes to obtain a training (90%) and test (10%) set.

Audio reconstruction. To perform audio synthesis, we consider paths inside the latent space, where each point corresponds to a single spectral frame. We sample along a given path and concatenate the spectral frames to obtain the magnitude distribution. Then, we apply the Griffin-Lim algorithm in order to recover the phase distribution and synthesize the corresponding waveform.

4.2. Models

Here, we rely on a simple VAE architecture to show the efficiency of the proposed method. The encoder is defined as a 3-layer feed-forward neural network with Rectified Linear Units (ReLU) activation functions and 2000 units per layer. The last layer maps to a given dimensionality d of the latent space. In our experiments, we analyzed the effect of relying on different latent spaces and empirically selected latent spaces with 64 dimensions. The decoder is defined in a symmetrical way, with the same architecture and units, mapping back to the dimensionality of the input transform. For learning the model, we use a value of $\beta = 2$, which is linearly increased from 0 to its final value during the first 100 epochs (following the *warmup* procedure [17]). In order to train the model, we rely on the ADAM [21] optimizer with an initial learning rate of 0.0001. In a first stage, we train the model without perceptual regularization ($\alpha = 0$) for a total of 5000 epochs. Then, we introduce the perceptual regularization ($\alpha = 0.1$) and train for another 1000 epochs. This allows the model to first focus on the quality of the reconstruction, and then to converge towards a solution with perceptual space properties. We found in our experiments that this two-step procedure is critical to the success of the regularization.

5. RESULTS

5.1. Latent spaces properties

In order to visualize the 64d latent spaces, we apply a simple Principal Component Analysis (PCA) to obtain a 3d representation. Using a PCA ensures that the visualization is a linear transform of the original space. Therefore, this preserves the real distances inside the latent space. Furthermore, this will allow to recover an exploitable representation when we will use this space to generate novel audio content. The results of learning regularized latent spaces for different spectral transforms are displayed in Figure 3.

As we can see, in VAEs without regularization (small space), the relationships between instruments do not match perceptual ratings. Furthermore, the variance of distributions show that the model rather tries to spread the information across the latent space to help the reconstruction. However, the NSGT provides a better unregularized space with different instrumental distributions already well separated. Now, if we compare to the regularized spaces, we can clearly see the effect of the criterion, which provides a larger separation of distribution. This effect and final result is particularly striking for the NSGT (c), which provides the highest correlation to the distances in our combined timbre space (Figure 2). Interestingly, the instrumental distributions might be shuffled around space in order to comply with the reconstruction objective. However, the pairwise distances reflecting perceptual relations are well matched as indicated by the KL divergence. By looking at the test set reconstructions, we can see that enforcing the perceptual topology on the latent spaces do not impact the quality of audio reconstruction for the NSGT, where the reconstruction provides an almost perfectly matching distribution. In the case of the STFT, we can see that the model is impacted by the regularization and mostly match the overall density of the distribution rather than its exact peak information. Finally, it seems that the DCT model diverged in terms of reconstruction, being unable to reconstruct the distributions. However, we can see that the KL fit to timbre distances is better than the STFT, indicating an overfit of the learning towards the regularization criterion. This generative evaluation is quantified and confirmed in the next section.

Method		$\log p(\mathbf{x})$	$\|\mathbf{x} - \tilde{\mathbf{x}}\|^2$
Unregularized (NSGT)	PCA	-	2.2570
	AE	-1.2008	1.6223
	VAE	-2.3443	0.1593
Regularized (VAE)	STFT	-1.9237	0.2412
	DCT	4.3415	2.2629
	NSGT-CQT	-2.8723	0.1610
	NSGT-MEL	-2.9184	0.1602
	NSGT-ERB	**-2.9212**	**0.1511**

Table 1: Generative capabilities evaluated by the log likelihood and mean quality of reconstructed representations on the test set.

5.2. Generative capabilities

We quantify the generative capabilities from the latent spaces by computing the log likelihood and mean difference between the original and reconstructed spectral representations on the test set. We compare these results for different transforms and without regularization, which are presented in Table 1.

As we can see, the unregularized VAE trained on the NSGT distribution provides a very good reconstruction capacity, and still generalizes very well. This can be seen in its ability to generate spectral distributions from the test set almost perfectly. Interestingly, regularizing the latent space does not seem to affect the quality of the reconstruction at all. It even seems that the generalization increases with the regularized latent space. This could however be explained by the fact that the regularized models are trained for twice as much epochs based on our two-fold procedure.

It clearly seems that NSGTs provide both better generalization and reconstruction abilities, while the DCT seems to provide only a divergent model. This can be explained by the fact that NSGT frequency axis is organized on a logarithmic scale. Furthermore, their distribution are well spread across this axis, whereas STFT and DCT tends to have most of their informative dimensions in the bottom half of the spectrum. Therefore, NSGTs provide a more informative input. Finally, there only seems to be a marginal difference between the results of different NSGT scales. However, for all remaining experiments, we select the NSGT-ERB as it is more coherent with our perceptual endeavor.

Thanks to the decoder and its generative capabilities, we can now directly synthesize the audio corresponding to any point inside the latent space, but also any paths between two given instruments. This allows us to turn our analytical spaces into audio synthesizers. Furthermore, as shown in Figure 5 (Bottom right), synthesizing audio along these spaces lead to smooth evolution of spectral distributions and perceptually continuous synthesis (as discussed extensively in the next section). In order to perform subjective evaluation of the audio reconstruction, generated samples from the latent space are available on the supporting repository.

5.3. Generalizing perception, audio synthesis of timbre paths

Given that the encoder of our latent space is trained directly on spectral distributions, it is able to analyze samples belonging to new instruments that were not part of the original perceptual studies. Furthermore, as the learning is regularized by perceptual ratings, we could hope that the resulting position would predict the perceptual relationships of this new instrument to the existing in-

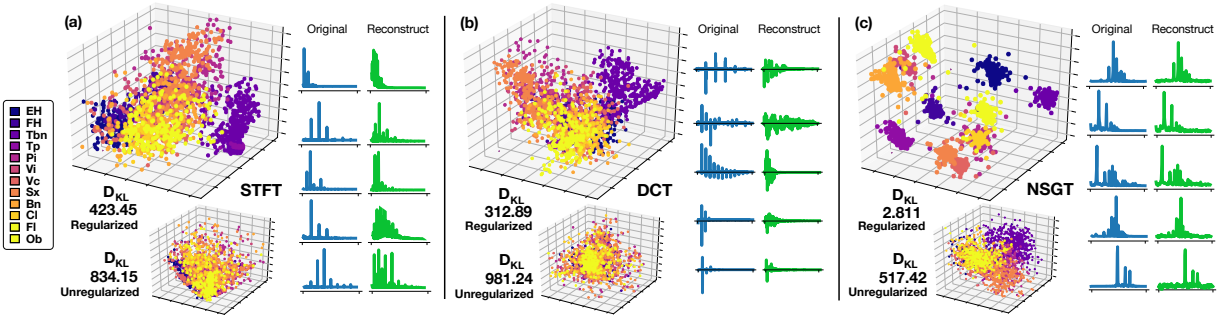


Figure 3: Comparing the regularized VAE latent spaces (large) for the STFT (a), DCT (b) and NSGT (CQT) (c) transforms. For each transform, we plot the corresponding unregularized space (small) and their respective $\mathcal{D}_{KL}$ divergence to the timbre space distances. We plot a set of VAE decoder reconstructions of instrumental spectral frame distributions from the test set directly from the regularized spaces

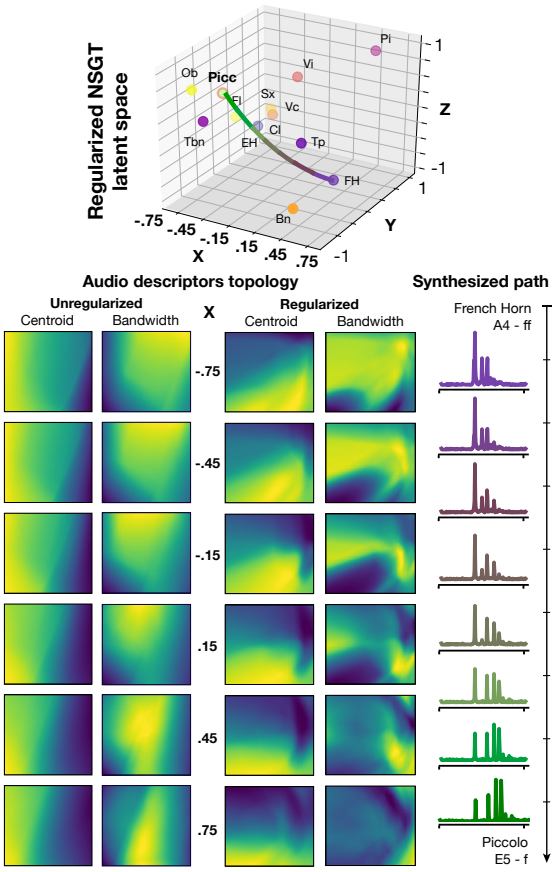


Figure 4: (Top) Projecting new instruments inside the regularized latent space allow to see their perceptual relations to others. (Bottom right) We can generate any path between instruments in the space and synthesize the corresponding perceptually-smooth audio evolution. (Bottom, left) We define 6 equally-spaced projection planes across the x axis and sample points on a 50x50 grid. We reconstruct their audio distribution to compute their spectral *centroid* and *bandwidth*. We compare the resulting descriptor space topology for unregularized (left) and regularized (right) spaces.

struments. This could potentially feed further perceptual studies, to refine timbre understanding. To evaluate this hypothesis, we extracted a set of *Piccolo* audio samples to evaluate their behavior in latent space. We perform the same processing as for the training dataset (Section 4.1) and encode these new samples in the latent space to study the *out-of-domain* generalization capabilities of our model. The results of this analysis are presented in Figure 5 (Top).

Here, we can see that new samples (represented by their centroid for clarity) are encoded in a coherent position in the latent space, as they group with their families, even though they were never presented to the model during learning. However, obtaining a definitive answer on the perceptual inference capabilities of these spaces would require a complete perception experiment, that we leave to future work. Now, as argued previously, one of the key property of the latent spaces is that they provide an invertible non-linear mapping. Therefore, we could thrive on this property to truly understand what are the perceptual relations between instruments based on the behavior of spectral distributions *between* the points in the timbre space. To exhibit this capability, we encode the position in the latent space of a *Piccolo* sample playing an E5-f. Then, based on the position of a *French Horn* playing an A4-ff, we perform an interpolation between these latent points to obtain the path between these two instruments in latent space. We then sample and decode the spectral distributions at 6 equally spaced positions along the path, which are displayed in Figure 5 (Right). As we can see, the resulting audio distributions demonstrate a smooth evolution between timbral structures of both instruments. Furthermore, the resulting interpolation is clearly more complex than a linear change between one structure to the other. Hence, this approach could be used to understand more deeply the timbre relationships between instruments. Also, this provides a model able to perform perceptually-relevant synthesis of novel timbres, while sharing the properties of multiple instruments.

5.4. Topology of audio descriptors

Here, we analyze the topology of signal descriptors across the latent space. As the space is continuous, we do so by sampling uniformly the PCA space and then using the decoder to generate audio samples at a given point. Then, we compute the audio descriptors of this sample. In order to provide a visualization, we select 6 equally-distant planes across the x dimension, at $\{-.75, -.45, -.15, .15, .45, .75\}$, which define an uniform 50x50

grid between $[-1, 1]$ on other dimensions. We compare the results between unregularized or regularized NSGT latent spaces in Figure 5 (Bottom left) for the *spectral centroid* and *spectral bandwidth*. Animations of continuous traversals of the latent space are available on the supporting repository. As we can see, the audio descriptors behave following overall non-linear patterns for both unregularized and regularized latent spaces. However, they still exhibit locally smooth properties. This shows that our model is able to organize audio variations. In the case of unregularized spaces, the organization of descriptors is spread out in a more even fashion. The addition of perceptual ratings to regularize the learning seems to require that this space is organized with a more complex topology. This could be explained by the fact that, in the unregularized case, the VAE only needs to find a configuration of the distributions that maximizes their reconstruction. Oppositely, the regularization requires that instrumental distances follow the perceptual dissimilarity ratings, prompting the need for a more complex relationship between descriptors. This might underline the fact that linear correlations between MDS dimensions and audio descriptors is insufficient to truly understand the dimensions related to timbre perception. However, the audio descriptors topology overall still provide locally smooth evolutions. Finally, a very interesting observation comes from the topology of the centroid. Indeed, all perceptual studies underline its correlation to timbre perception, which is partly confirmed by our model (by projecting on the y axis). This tends to confirm the perceptual relevance of our regularized latent spaces. However, this also shows that the relation between centroid and timbre might not be linear.

5.5. Descriptor-based synthesis

As shown in the previous section, the audio descriptors are organized in a smooth locally linear way across the space. Furthermore, as discussed in Section 5.1, we have seen that the instrumental distributions are grouped across spaces depending on perceptual relations. Based on these two findings, we hypothesize that we can find paths inside these spaces that modify a given audio distribution to follow a target descriptor, while remaining perceptually smooth. Hence, we propose a simple method for perceptually-relevant *descriptor-based path synthesis* presented in Algorithm 1.

Based on the latent space $\mathbf{z}$ (with corresponding encoder q and decoder p) and a given origin spectrum $\mathbf{x}_0$, the goal of this algorithm is to find the succession of spectral distributions that match a given target evolution $\mathbf{t} \in \mathbb{R}^N$ for a descriptor d. First, we find the position of the origin distribution in latent space $\mathbf{z}_0$ and evaluate its descriptor value $\mathbf{d}_0$ (lines 1-4). Then for each point i, we compute the descriptor values $\mathbf{D}_i$ in the neighborhood of the current latent point (lines 6-10) by decoding their audio distributions. Note that the the neighborhood is defined as the set of close latent points, and its size directly defines the complexity of the optimization. Then, we select the neighboring latent point $\mathbf{z}_i$ that provides the evolution of descriptor closest to the target evolution $t[i]$ (lines 11-14). Finally, we obtain the spectral distribution $S[i]$ by decoding the latent position $\mathbf{z}_i$. The results of applying this algorithm to a given instrumental distribution is presented in Figure 5.

Here, we start from the NSGT distribution of a Clarinet-Bb playing a G#4 in *fortissimo*. We apply our algorithm twice from the same origin point, either on a descending target shape for the spectral centroid (top), or an ascending log shape for the spectral bandwidth (bottom). In both cases, we plot the synthesized NSGT distributions at different points of the optimized path, and the neighboring descriptor space. As we can see, the resulting descriptor evolution closely match the input target in both cases. Furthermore, we can see by visual inspection of the spectrum evolution, that the corresponding distributions are indeed sharply modified to match the desired descriptors. Interestingly, the optimization of different target shapes on different descriptors lead to widely different paths in the latent space. However, the overall timbre structure of the original instrument still seems to follow a smooth evolution. Here, we note that the algorithm is quite rudimentary, and could benefit from more global neighborhood information, as witnessed from the slightly erratic local selection of latent points.

Algorithm 1: Descriptor-based path synthesis

Data: space $\mathbf{z}$, encoder $q_\phi(\mathbf{z}|\mathbf{x})$, decoder $p_\theta(\mathbf{x}|\mathbf{z})$
Data: origin spectrum $\mathbf{x}_0$, target series $\mathbf{t}_{1..N}$, descriptor d
Result: spectral distrib. $S \in \mathbb{R}^{N \times F}$

1 // Find origin position in latent space
2 $\mathbf{z}_0 = q_\phi(\mathbf{x}_0)$
3 // Evaluate origin descriptor
4 $\mathbf{d}_0 = evaluate(\mathbf{x}_0, d)$
5 **for** $i \in [1, N]$ **do**
6 // Latent 3-d neighborhood of current point
7 $\mathbf{N}_i = neighborhood(\mathbf{z}_{i-1})$
8 // Sample and evaluate descriptors
9 $\mathbf{X}_i = q_\phi(\mathbf{N}_i)$
10 $\mathbf{D}_i = evaluate(\mathbf{X}_i, d)$
11 // Compute difference to target
12 $\Delta_i = \|(\mathbf{D}_i - \mathbf{d}_{i-1}) - (t[i] - t[i-1])\|^2$
13 // Find next latent point
14 $\mathbf{z}_i = argmin(\Delta_i)$
15 // Decode distribution
16 $S[i] = p_\theta(\mathbf{z}_i)$
17 **end**

6. CONCLUSION

Here, we have shown that regularizing VAEs with perceptual ratings provides timbre spaces that allow for high-level analysis and audio synthesis directly from these spaces. The organization of these perceptually-regularized latent spaces prove the flexibility of these systems, and provides a latent space from which generation of novel audio content is straightforward. These spaces allow to extrapolate perceptual results on new sounds and instruments without the need to collect new measurements. Finally, by analyzing the behavior of audio descriptors across the latent space, we have shown that even though they follow a non-linear evolution, they still exhibit some locally smooth properties. Based on these, we introduced a method for descriptor-based path synthesis that allow to synthesize audio that match a target descriptor shape, while retaining the timbre structure of instruments. Future work on these latent spaces would be to perform perceptual experiments to confirm their perceptual topology.

7. REFERENCES

[1] Stephen McAdams, Bruno L. Giordano, Patrick Susini, Geoffroy Peeters, and Vincent Rioux, "A meta-analysis of

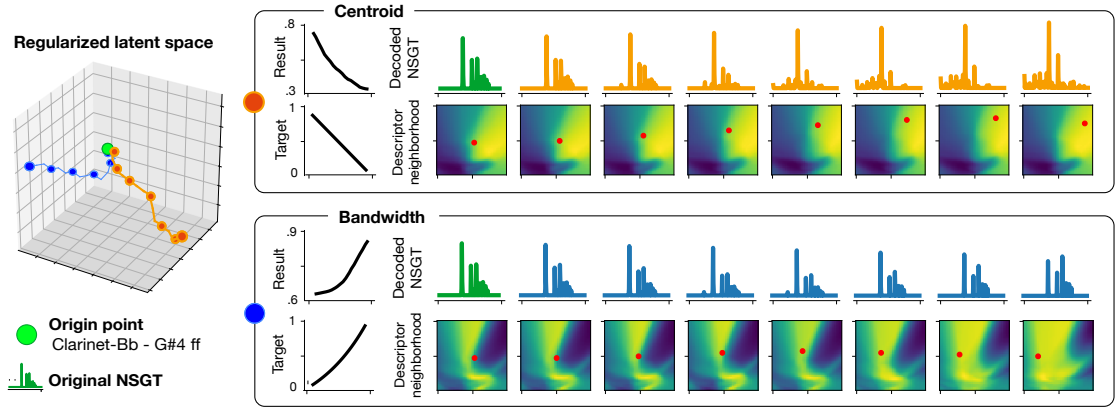


Figure 5: *Descriptor-based synthesis*. Given an origin point in latent space (Clarinet-Bb G#4 ff), we apply our algorithm either on a descending target shape for the spectral centroid (top), or an ascending log shape for the spectral bandwidth (bottom). In both cases, we plot the decoded NSGT distributions and neighboring descriptor space information along the optimized path

acoustic correlates of timbre dimensions," *Journal of the Acoustical Society of America*, vol. 120, no. 5, 2006.

[2] John M Grey and John W Gordon, "Perceptual effects of spectral modifications on musical timbres," *The Journal of the Acoustical Society of America*, vol. 63, no. 5, 1978.

[3] Aaron Van Den Oord, Sander Dieleman, Heiga Zen, Karen Simonyan, Oriol Vinyals, Alex Graves, Nal Kalchbrenner, Andrew Senior, and Koray Kavukcuoglu, "Wavenet: A generative model for raw audio," *arXiv preprint arXiv:1609.03499*, 2016.

[4] Soroush Mehri, Kundan Kumar, Ishaan Gulrajani, Rithesh Kumar, Shubham Jain, Jose Sotelo, Aaron Courville, and Yoshua Bengio, "Samplernn: An unconditional end-to-end neural audio generation model," *ICLR Conference*, 2017.

[5] Jesse Engel, Cinjon Resnick, Adam Roberts, Sander Dieleman, Douglas Eck, Karen Simonyan, and Mohammad Norouzi, "Neural audio synthesis of musical notes with wavenet autoencoders," *arXiv preprint:1704.01279*, 2017.

[6] Diederik P Kingma and Max Welling, "Auto-encoding variational bayes," *ICLR Conference*, 2014.

[7] Irina Higgins, Loic Matthey, Arka Pal, Christopher Burgess, Xavier Glorot, Matthew Botvinick, Shakir Mohamed, and Alexander Lerchner, "beta-vae: Learning basic visual concepts with a constrained variational framework," *ICLR Conference*, 2016.

[8] Peter Balazs, Monika Dörfler, Florent Jaillet, Nicki Holighaus, and G Velasco, "Theory, implementation and applications of nonstationary gabor frames," *Journal of computational and applied mathematics*, vol. 236, no. 6, 2011.

[9] Laurens van der Maaten and Geoffrey Hinton, "Visualizing data using t-sne," *Journal of machine learning research*, vol. 9, no. Nov, pp. 2579–2605, 2008.

[10] John M Grey, "Multidimensional perceptual scaling of musical timbres," *the Journal of the Acoustical Society of America*, vol. 61, no. 5, pp. 1270–1277, 1977.

[11] Carol L Krumhansl, "Why is musical timbre so hard to understand," *Structure and perception of electroacoustic sound and music*, vol. 9, pp. 43–53, 1989.

[12] Paul Iverson and Carol L Krumhansl, "Isolating the dynamic attributes of musical timbrea," *The Journal of the Acoustical Society of America*, vol. 94, no. 5, pp. 2595–2603, 1993.

[13] Stephen McAdams, Suzanne Winsberg, Sophie Donnadieu, Geert De Soete, and Jochen Krimphoff, "Perceptual scaling of synthesized musical timbres: Common dimensions, specificities, and latent subject classes," *Psychological research*, vol. 58, no. 3, pp. 177–192, 1995.

[14] Stephen Lakatos, "A common perceptual space for harmonic and percussive timbres," *Perception & psychophysics*, vol. 62, no. 7, pp. 1426–1439, 2000.

[15] Christopher M Bishop and Tom M Mitchell, "Pattern recognition and machine learning," 2014.

[16] Keith Hastings, "Monte carlo sampling methods using markov chains and their applications," *Biometrika*, vol. 57, no. 1, pp. 97–109, 1970.

[17] Casper Kaae Sønderby, Tapani Raiko, Lars Maaløe, Søren Kaae Sønderby, and Ole Winther, "How to train deep variational autoencoders and probabilistic ladder networks," *arXiv preprint arXiv:1602.02282*, 2016.

[18] Jen-Tzung Kuo and Kuan-Ting Chien, "Variational recurrent neural networks for speech separation," *INTERSPEECH 2017*.

[19] Wei-Ning Hsu, Yu Zhang, and James Glass, "Learning latent representations for speech generation and transformation," *arXiv preprint arXiv:1704.04222*, 2017.

[20] Guillaume Ballet, Riccardo Borghesi, Peter Hoffmann, and Fabien Levy, "Studio online 3.0: An internet "killer application" for remote access to ircam sounds and processing tools," *Journee Informatique Musicale (JIM)*, 1999.

[21] Diederik P Kingma and Jimmy Ba, "Adam: A method for stochastic optimization," *arXiv preprint:1412.6980*, 2014.

Author Index

Sponsored by

Eventide®

steinberg

ARTURIA®
YOUR EXPERIENCE • YOUR SOUND

YAMAHA

audiokinetic®

iZOTOPE

imaginando

Sonnox

Ableton

NEWFANGLED AUDIO

EURAL

AVEIRO
CÂMARA MUNICIPAL

TEATRO AVEIRENSE

AROUCA

AROUCA GEOPARK
UNESCO
Organização das Nações Unidas para a Educação, a Ciência e a Cultura
Arouca Geoparque Mundial da UNESCO