



THE UNIVERSITY *of* EDINBURGH



Proceedings of the
20th International Conference
on Digital Audio Effects

5 – 9 September 2017
Edinburgh UK

**Acoustics &
Audio Group**



THE UNIVERSITY *of* EDINBURGH



Proceedings of the
20th International Conference
on Digital Audio Effects

5 – 9 September 2017
Edinburgh UK

**Acoustics &
Audio Group**

Credits:

Proceedings edited and produced by
Alberto Torin, Brian Hamilton, Stefan Bilbao, Michael Newton
Cover designed by Nicky Regan
L^AT_EX style by Paolo Annibale
DAFx17 logo by Michael Gill
Back cover photo by Michael Newton

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

www.dafx17.eca.ed.ac.uk
www.dafx.de

DAFx17 is proudly sponsored by



≡ Ableton

audiokinetic

K R O T O S



All rights reserved.

All copyrights of the individual papers remain with their respective authors.

Foreword

September is finally upon us, and we the local organising committee for DAFx17 would like to welcome you to Edinburgh for what is sure to be, as always, a mind- and ear-busting experience. We are especially happy to be able to host this 20th edition of the International Conference on Digital Audio Effects, which provides an opportune moment to reflect on what has changed since 1998, when DAFx was launched as a COST action, with its inaugural meeting in Barcelona.

Looking back at the proceedings from DAFx98, one may remark that, of the 35 accepted papers, there were none concerned with virtual analog or physical modelling, and only a handful concerned with time-frequency audio analysis and spatial audio. At DAFx17, approximately two thirds of the 68 accepted papers may be classified as such. DAFx has increasingly become the home of the sophisticated use of mathematics for audio—and at this point, there's no other conference quite like it.

Beyond this change in the scope of DAFx, what is perhaps most striking is that the average age of a DAFx paper author is, if anything, lower than it was in 1998. Why? Well, an unspoken rule of DAFx seems to be: we all love audio of course, but—it had better be fun too. Rest assured that DAFx17 will be no exception. Hosting a conference in Edinburgh makes this easy. We will be giving the conference participants lots of opportunity to see some of the best parts of this spectacular city, and to venture further afield in Scotland. As at all times here, there is a 70% chance of rain.

We have been very lucky to be able to secure some superb keynote and tutorial presenters from the academic world and from industry. Miller Puckette of UCSD, Julius Smith of CCRMA at Stanford, and Avery Wang from Shazam will deliver the keynotes, and tutorials will be presented by Dave Berners of Universal Audio, Jean-Marc Jot of Magic Leap, Julian Parker from Native Instruments, and Brian Hamilton from the Acoustics and Audio Group here at the University of Edinburgh. To top it all off, we are also delighted to be able to showcase new multichannel work by Trevor Wishart—one of the legends of electroacoustic music.

One sure sign of the success of a conference series is the level of industrial support through sponsorship. We are happy to report that for DAFx17, thirteen sponsors have lent a hand; they are listed on the previous page and on the back cover of these proceedings. A number of the sponsors are new in 2017, and, furthermore, many volunteered support before being asked! We are of course extremely grateful for this support, not only because it defrays operating costs, and permits the reduction of registration fees, but also because of the increasing importance of the industrial perspective in the research landscape—in digital audio, the academic/industrial boundary is more porous than ever before, and DAFx is very proud to serve as a gateway of sorts.

DAFx17 is hosted by the Acoustics and Audio Group at the University of Edinburgh. We are very grateful for the support from previous organisers, and especially Udo Zölzer, Vesa Välimäki, Damian Murphy, Sascha Disch, Victor Lazzarini, Joe Timoney, Sigurd Saue, and Pavel Rajmic, as well as the Journal of the Audio Engineering Society, which has agreed to invite best papers for submission. We are also grateful for support offered by the Edinburgh College of Art at the University of Edinburgh, and especially Nicky Regan, who took care of our graphic design work at very short notice, Jared de Bruin who looked after our financial affairs, and more generally, the Reid School of Music, the host department.

DAFx of course will continue, and we look forward to seeing you all again in Aveiro, Portugal in 2018.

The DAFx17 Local Organising Committee

Conference Committees

DAFx Board

Daniel Arfib (CNRS-LMA, Marseille, France)
Nicola Bernardini (Conservatorio di Musica “Cesare Pollini”, Padova, Italy)
Stefan Bilbao (Acoustics and Audio Group, University of Edinburgh, UK)
Francisco Javier Casajús (ETESIS Telecomunicación - Universidad Politécnica de Madrid, Spain)
Philippe Depalle (McGill University, Montréal, Canada)
Giovanni De Poli (CSC-DEI, University of Padova, Italy)
Myriam Desainte-Catherine (SCRIME, Université de Bordeaux, France)
Markus Erne (Scopein Research, Aarau, Switzerland)
Gianpaolo Evangelista (University of Music and Performing Arts, Vienna, Austria)
Simon Godsill (University of Cambridge, UK)
Pierre Hanna (Université de Bordeaux, France)
Robert Höldrich (IEM, University of Music and Performing Arts, Graz, Austria)
Jean-Marc Jot (Magic Leap, USA)
Victor Lazzarini (Maynooth University, Ireland)
Sylvain Marchand (L3i, University of La Rochelle, France)
Damian Murphy (University of York, UK)
Søren Nielsen (SoundFocus, Arhus, Denmark)
Markus Noisternig (IRCAM - CNRS - Sorbonne Universities / UPMC, Paris, France)
Luis Ortiz Berenguer (ETESIS Telecomunicación - Universidad Politécnica de Madrid, Spain)
Geoffroy Peeters (IRCAM - CNRS SMTS, France)
Rudolf Rabenstein (University Erlangen-Nuernberg, Erlangen, Germany)
Davide Rocchesso (University of Palermo, Italy)
Jørn Rudi (NoTAM, Oslo, Norway)
Mark Sandler (Queen Mary University of London, UK)
Augusto Sarti (DEI - Politecnico di Milano, Italy)
Lauri Savioja (Aalto University, Espoo, Finland)
Xavier Serra (Universitat Pompeu Fabra, Barcelona, Spain)
Julius O. Smith III (CCRMA, Stanford University, CA, USA)
Alois Sontacchi (IEM, University of Music and Performing Arts, Graz, Austria)
Todor Todoroff (ARTeM, Brussels, Belgium)
Jan Tro (Norwegian University of Science and Technology, Trondheim, Norway)
Vesa Välimäki (Aalto University, Espoo, Finland)
Udo Zölzer (Helmut-Schmidt University, Hamburg, Germany)

DAFx17 Local Organizing Committee

Stefan Bilbao
Roderick Buchanan-Dunlop
Charlotte Desvages
Michele Ducceschi
Brian Hamilton
Reginald Harrison-Harsley
Amaya López-Carromero
Michael Newton
Alberto Torin
Craig J. Webb

DAFx17 Programme Committee

Federico Avanzini (University of Padova, Italy)
Stefan Bilbao (Acoustics and Audio Group, University of Edinburgh, UK)
Marcelo Caetano (INESC Porto, Portugal)
Vasileios Chatzioannou (Institute of Music Acoustics, Vienna, Austria)
Philippe Depalle (McGill University, Montréal, Canada)
Giovanni De Poli (CSC-DEI, University of Padova, Italy)
Sascha Disch (Fraunhofer IIS, Erlangen, Germany)
Michele Ducceschi (Acoustics and Audio Group, University of Edinburgh, UK)
Gianpaolo Evangelista (University of Music and Performing Arts, Vienna, Austria)
Federico Fontana (University of Udine, Italy)
Brian Hamilton (Acoustics and Audio Group, University of Edinburgh, UK)
Thomas Hélie (IRCAM-CNRS-UPMC, France)
Robert Höldrich (IEM, University of Music and Performing Arts, Graz, Austria)
Martin Holters (Helmut Schmidt University, Germany)
Jean-Marc Jot (Magic Leap, USA)
Richard Kronland-Martinet (CNRS Marseille, France)
Victor Lazzarini (Maynooth University, Ireland)
Esteban Maestre (McGill University, Montréal, Canada)
Sylvain Marchand (L3i, University of La Rochelle, France)
Damian Murphy (University of York, UK)
Michael Newton (Acoustics and Audio Group, University of Edinburgh, UK)
Markus Noisternig (IRCAM - CNRS - Sorbonne Universities / UPMC, Paris, France)
Julian Parker (Native Instruments, Berlin, Germany)
Geoffroy Peeters (IRCAM - CNRS SMTS, France)
Rudolf Rabenstein (University Erlangen-Nuernberg, Germany)
Pavel Rajmic (Brno University of Technology, Brno, Czech Republic)
Josh Reiss (Queen Mary University of London, UK)
Jørn Rudi (NoTAM, Oslo, Norway)
František Rund (Czech Technical University, Prague, Czech Republic)
Sigurd Sauve (NTNU, Trondheim, Norway)
Lauri Savioja (Aalto University, Espoo, Finland)
Jiří Schimmel (Brno University of Technology, Brno, Czech Republic)
Stefania Serafin (Aalborg University, Denmark)
Xavier Serra (Universitat Pompeu Fabra, Barcelona, Spain)
Julius O. Smith III (CCRMA, Stanford University, CA, USA)
Alois Sontacchi (IEM, University of Music and Performing Arts, Graz, Austria)
Alex Southern (AECOM, UK)
Todor Todoroff (ARTeM, Brussels, Belgium)
Alberto Torin (Acoustics and Audio Group, University of Edinburgh, UK)
Jan Tro (Norwegian University of Science and Technology, Trondheim, Norway)
Vesa Välimäki (Aalto University, Espoo, Finland)
Maarten van Walstijn (Queen's University Belfast, UK)
Kurt Werner (Queen's University Belfast, UK)
Udo Zölzer (Helmut Schmidt University, Hamburg, Germany)

Contents

Foreword	iii
Conference Committees	iv
Keynotes	1
History of Virtual Musical Instruments and Effects Based on Physical Modeling <i>Julius O. Smith III</i>	1
Robust Indexing and Search in a Massive Corpus of Audio Recordings <i>Avery Wang</i>	1
Time-domain Manipulation via STFTs <i>Miller Puckette</i>	1
Tutorials	2
Efficient reverberation rendering for complex interactive audio scenes <i>Jean-Marc Jot</i>	2
Room Acoustic Simulation: Overview and Recent Developments <i>Brian Hamilton</i>	2
Modeling Circuits with Nonlinearities in Discrete Time <i>David Berners</i>	2
From Algorithm to Instrument <i>Julian D. Parker</i>	2
Poster Session 1	
Nonlinear Homogeneous Order Separation for Volterra Series Identification <i>Damien Bouvier, Thomas Hélie and David Roze</i>	3
Introducing Deep Machine Learning for Parameter Estimation in Physical Modelling <i>Leonardo Gabrielli, Stefano Tomassetti, Stefano Squartini and Carlo Zinato</i>	11
Real-time Physical Model of a Wurlitzer and Rhodes Electronic Piano <i>Florian Pfeifle</i>	17
A Mechanical Mapping Model for Real-time Control of a Complex Physical Modelling Synthesis Engine with a Simple Gesture <i>Fiona Keenan and Sandra Pauletto</i>	25
Simulating the Friction Sounds Using a Friction-based Adhesion Theory Model <i>Takayuki Nakatsuka and Shigeo Morishima</i>	32
Kinematics of Ideal String Vibration Against a Rigid Obstacle <i>Dmitri Kartofelev</i>	40
Doppler Effect of a Planar Vibrating Piston: Strong Solution, Series Expansion and Simulation <i>Tristan Lebrun and Thomas Hélie</i>	48
Latent Force Models for Sound: Learning Modal Synthesis Parameters and Excitation Functions from Audio Recordings <i>William J. Wilkinson, Joshua D. Reiss and Dan Stowell</i>	56

Oral Session 1: Physical Modeling I

Energy Shaping of a Softening Duffing Oscillator Using the Formalism of Port-Hamiltonian Systems <i>Marguerite Jossic, David Roze, Thomas Hélie, Baptiste Chomette and Adrien Mamou-Mani</i>	64
On Iterative Solutions for Numerical Collision Models <i>Vasileios Chatzioannou, Sebastian Schmutzhard and Stefan Bilbao</i>	72
A Numerical Scheme for Various Nonlinear Forces, Including Collisions, Which Does Not Require an Iterative Root Finder <i>Michele Ducceschi</i>	80
Trajectory Anti-aliasing on Guaranteed-passive Simulation of Nonlinear Physical Systems <i>Rémy Muller and Thomas Hélie</i>	87

Oral Session 2: Audio Processing and Effects

The Quest for the Best Graphic Equalizer <i>Juho Liski and Vesa Välimäki</i>	95
Audio Processing Chain Recommendation <i>Spyridon Stasis, Nicholas Jillings, Sean Enderby and Ryan Stables</i>	103
Investigation of a Drum Controlled Cross-adaptive Audio Effect for Live Performance <i>Saurjya Sarkar, Joshua D. Reiss and Øyvind Brandtsegg</i>	110
Real-time Pitch Tracking in Audio Signals with the Extended Complex Kalman Filter <i>Orchisama Das, Julius O. Smith III and Chris Chafe</i>	118

Poster Session 2

Efficient Anti-aliasing of a Complex Polygonal Oscillator <i>Christoph Hohnerlein, Maximilian Rest and Julian D. Parker</i>	125
Modeling Circuits with Operational Transconductance Amplifiers Using Wave Digital Filters <i>Ólafur Bogason and Kurt J. Werner</i>	130
Automatic Decomposition of Non-linear Equation Systems in Audio Effect Circuit Simulation <i>Martin Holters and Udo Zölzer</i>	138
WDF Modeling of a Korg MS-50 Based Non-linear Diode Bridge VCF <i>Maximilian Rest, Julian D. Parker and Kurt J. Werner</i>	145
Comparison of Germanium Bipolar Junction Transistor Models for Real-time Circuit Simulation <i>Ben Holmes, Martin Holters and Maarten van Walstijn</i>	152
Automatic Control of the Dynamic Range Compressor Using a Regression Model and a Reference Sound <i>Di Sheng and György Fazekas</i>	160

Oral Session 3: Virtual Analog

Fixed-rate Modeling of Audio Lumped Systems: A Comparison Between Trapezoidal and Implicit Midpoint Methods <i>François G. Germain</i>	168
Generalizing Root Variable Choice in Wave Digital Filters with Grouped Nonlinearities <i>Kurt J. Werner, Michael J. Olsen, Maximilian Rest and Julian D. Parker</i>	176
Block-oriented Gray Box Modeling of Guitar Amplifiers <i>Felix Eichas, Stephan Möller and Udo Zölzer</i>	184
Virtual Analog Buchla 259 Wavefolder <i>Fabián Esqueda, Henri Pöntynen, Vesa Välimäki and Julian D. Parker</i>	192
Network Variable Preserving Step-size Control in Wave Digital Filters <i>Michael J. Olsen, Kurt J. Werner and François G. Germain</i>	200

Poster Session 3

On the Design and Use of Once-differentiable High Dynamic Resolution Atoms for the Distribution Derivative Method <i>Nicholas Esterer and Philippe Depalle</i>	208
Nicht-negativeMatrixFaktorisierungnutzendenKlangsynthesenSystem (NiMFKS): Extensions of NMF-based Concatenative Sound Synthesis <i>Michael Buch, Elio Quinton and Bob L. Sturm</i>	215
Validated Exponential Analysis for Harmonic Sounds <i>Matteo Briani, Annie Cuyt and Wen-shin Lee</i>	222
A System Based on Sinusoidal Analysis for the Estimation and Compensation of Pitch Variations in Musical Recordings <i>Luís F. V. Carvalho and Hugo T. Carvalho</i>	228
Gradient Conversion Between Time and Frequency Domains Using Wirtinger Calculus <i>Hugo Caracalla and Axel Roebel</i>	234
Live Convolution with Time-variant Impulse Response <i>Øyvind Brandtsegg and Sigurd Saue</i>	239
Modal Audio Effects: A Carillon Case Study <i>Elliot K. Canfield-Dafilou and Kurt J. Werner</i>	247
LP-BLIT: Bandlimited Impulse Train Synthesis of Lowpass-filtered Waveforms <i>Sebastian Kraft and Udo Zölzer</i>	255

Oral Session 4: Audio Analysis and Source Separation

Redressing Warped Wavelets and Other Similar Warped Time-something Representations <i>Gianpaolo Evangelista</i>	260
REDS: A New Asymmetric Atom for Sparse Audio Decomposition and Sound Synthesis <i>Julian Neri and Philippe Depalle</i>	268
Harmonic-percussive Sound Separation Using Rhythmic Information from Non-negative Matrix Factorization in Single-channel Music Recordings <i>Francisco Canadas-Quesada, Derry Fitzgerald, Pedro Vera-Candeas and Nicolas Ruiz-Reyes</i>	276
Iterative Structured Shrinkage Algorithms for Stationary/Transient Audio Separation <i>Kai Siedenburg and Simon Doclo</i>	283

Oral Session 5: Physical Modeling II

An Explorative String-bridge-plate Model with Tunable Parameters <i>Maarten van Walstijn and Sandor Mehes</i>	291
Modal Based Tanpura Simulation: Combining Tension Modulation and Distributed Bridge Interaction <i>Jamie Bridges and Maarten van Walstijn</i>	299
Physically Derived Synthesis Model of a Cavity Tone <i>Rod Selfridge, Joshua D. Reiss and Eldad J. Avital</i>	307
A Continuous Frequency Domain Description of Adjustable Boundary Conditions for Multidimensional Transfer Function Models <i>Maximilian Schäfer and Rudolf Rabenstein</i>	315

Poster Session 4

EVERTims: Open Source Framework for Real-time Auralization in Architectural Acoustics and Virtual Reality <i>David Poirier-Quinot, Brian F.G. Katz and Markus Noisternig</i>	323
A Comparison of Player Performance in a Gamified Localisation Task Between Spatial Loudspeaker Systems <i>Joe Rees-Jones and Damian Murphy</i>	329

Accurate Reverberation Time Control in Feedback Delay Networks <i>Sebastian J. Schlecht and Emanuël A. P. Habets</i>	337
Binauralization of Omnidirectional Room Impulse Responses - Algorithm and Technical Evaluation <i>Christoph Pörschmann, Philipp Stade and Johannes M. Arend</i>	345
Pinna Morphological Parameters Influencing HRTF Sets <i>Slim Ghorbal, Théo Auclair, Catherine Soladié and Renaud Séguier</i>	353
2D Spatial Audio in a Molecular Navigator/Editor for Blind and Visually Impaired Users <i>Ian Rodrigues, Ricardo Teixeira, Sofia Cavaco, Vasco D. B. Bonifácio, Daniela Peixoto, Yuri Binev, Florbela Pereira, Ana M. Lobo and João Aires-de-Sousa</i>	360
Performance Portability for Room Acoustics Simulations <i>Larisa Stoltzfus, Alan Gray, Christophe Dubach and Stefan Bilbao</i>	367
On Restoring Prematurely Truncated Sine Sweep Room Impulse Response Measurements <i>Elliot K. Canfield-Dafilou and Jonathan S. Abel</i>	375
Oral Session 6: Spatial Audio and Artificial Reverberation	
Constrained Pole Optimization for Modal Reverberation <i>Esteban Maestre, Jonathan S. Abel, Julius O. Smith III and Gary P. Scavone</i>	381
Diffuse-field Equalisation of First-order Ambisonics <i>Thomas McKenzie, Damian Murphy and Gavin Kearney</i>	389
Improving Elevation Perception with a Tool for Image-guided Head-related Transfer Function Selection <i>Michele Geronazzo, Enrico Peruch, Fabio Prandoni and Federico Avanzini</i>	397
Velvet-Noise Decorrelator <i>Benoit Alary, Archontis Politis and Vesa Välimäki</i>	405
Parametric Acoustic Camera for Real-time Sound Capture, Analysis and Tracking <i>Leo McCormack, Symeon Delikaris-Manias and Ville Pulkki</i>	412
Poster Session 5	
Estimating Pickup and Plucking Positions of Guitar Tones and Chords with Audio Effects <i>Zulfadhl Mohamad, Simon Dixon and Christopher Harte</i>	420
Unsupervised Taxonomy of Sound Effects <i>David Moffat, David Ronan and Joshua D. Reiss</i>	428
The Mix Evaluation Dataset <i>Brecht De Man and Joshua D. Reiss</i>	436
The Snail: A Real-time Software Application to Visualize Sounds <i>Thomas Hélie and Charles Picasso</i>	443
Beat-aligning Guitar Looper <i>Daniel Rudrich and Alois Sontacchi</i>	451
A Method for Automatic Whoosh Sound Description <i>Eugene Cherny, Johan Lilius and Dmitry Mouromtsev</i>	459
Analysis and Synthesis of the Violin Playing Style of Heifetz and Oistrakh <i>Chi-Ching Shih, Pei-Ching Li, Yi-Ju Lin, Yu-Lin Wang, Alvin W. Y. Su, Li Su and Yi-Hsuan Yang</i>	466
Oral Session 7: Perceptual Audio and Applications	
A Nonlinear Method for Manipulating Warmth and Brightness <i>Sean Enderby and Ryan Stables</i>	474
Soundscape Categorisation and the Self-assessment Manikin <i>Francis Stevens, Damian Murphy and Stephen Smith</i>	481

Development of a Quality Assurance Automatic Listening Machine (QuAALM) <i>Daniel J. Gillespie, Woody Herman and Russell Wedelich</i>	489
Blind Upmix for Applause-like Signals Based on Perceptual Plausibility Criteria <i>Alexander Adami, Lukas Brand, Sascha Disch and Jürgen Herre</i>	496
Oral Session 8: DAFx	
Co-authorship and Community Structure in the DAFx Conference Proceedings: 1998–2016 <i>Alex Wilson</i>	502
Author Index	510

Keynotes

Julius O. Smith III

History of Virtual Musical Instruments and Effects Based on Physical Modeling

Abstract This presentation visits historical developments leading to today's virtual musical instruments and effects based on physical modeling principles. It is hard not to begin with Daniel Bernoulli and d'Alembert who launched the modal representation (leading to both "additive" and "subtractive" synthesis) and the traveling-wave solution of the wave-equation for vibrating-strings, respectively, in the 18th century. Newtonian mechanics generally suffices mathematically for characterizing physical musical instruments and effects, although quantum mechanics is necessary for fully deriving the speed of sound in air. In addition to the basic ballistics of Newton's Law $f = ma$, and spring laws relating force to displacement, friction models are needed for modeling the aggregate behavior of vast numbers of colliding particles. The resulting mathematical models generally consist of ordinary and partial differential equations expressing Newton's Law, friction models, and perhaps other physical relationships such as temperature dependence. Analog circuits are similarly described. These differential-equation models are then solved in real time on a discrete time-space grid to implement musical instruments and effects. The external forces applied by the performer (or control voltages, etc.) are routed to virtual masses, springs, and/or friction-models, and they may impose moving boundary conditions for the discretized differential-equation solver. To achieve maximum quality per unit of computation, techniques from digital signal processing are typically used to implement the differential-equation solvers in ways that are numerically robust, energy aware, and minimizing computational complexity. In addition to reviewing selected historical developments, this presentation will try to summarize some of the known best practices for computational physical modeling in existing real-time virtual musical instruments and effects.

Avery Wang

Robust Indexing and Search in a Massive Corpus of Audio Recordings

Abstract In this talk I will give an overview of the Shazam audio recognition technology. The Shazam service takes a query comprised of a short sample of ambient audio (as little as 2 seconds) from a microphone and searches a massive database of recordings comprising more than 40 million soundtracks. The query may be degraded with significant additive noise (< 0 dB SNR), environmental acoustics, as well as nonlinear distortions. The computational scaling is such that a query may cost as little as a millisecond of processing time. Previous algorithms could index hundreds of items, required seconds of processing time, and were less tolerant to noise and distortion by 20-30 dB SNR. In aggregate, the Shazam algorithm represents a leap of more than a factor of 1E+10 in efficiency over prior art. I will discuss the various innovations leading to this result.

Miller Puckette

Time-domain Manipulation via STFTs

Abstract Perhaps the most important shortcoming of frequency-domain signal processing results from the Heisenberg limit that often forces tradeoffs between time and frequency resolution. In this paper we propose manipulating sounds by altering their STFTs in ways that affect time spans smaller than the analysis window length. An example of a situation in which this could be useful is the algorithm of Griffin and Lim, which generates a time-domain signal that optimally matches a (possibly overspecified) short-time amplitude spectrum. We propose an adaptation of Griffin-Lim to simultaneously optimize a signal to match such amplitude spectra on two or more different time scales, in order to simultaneously manage both transients and tuning.

Tutorials

Jean-Marc Jot

Efficient reverberation rendering for complex interactive audio scenes

Abstract Artificial reverberation algorithms originated several decades ago with Schroeder's pioneering work in the late 60s, and have been widely employed commercially since the 80s in music and soundtrack production. In the late 90s, artificial reverberation was introduced in game 3D audio engines, which today are evolving into interactive binaural audio rendering systems for virtual reality. By exploiting the perceptual and statistical properties of diffuse reverberation decays in closed rooms, computationally efficient reverberators based on feedback delay networks can be designed to automatically match with verisimilitude the “reverberation fingerprint” of any room. They can be efficiently implemented on standard mobile processors to simulate complex natural sound scenes, shared among a multiplicity of virtual sound sources having different positions and directivity, and combined to simulate complex acoustical spaces. In this tutorial presentation, we review the fundamental assumptions and design principles of artificial reverberation, apply them to design parametric reverberators, and extend them to realize computationally efficient interactive audio engines suitable for untethered virtual and augmented reality applications.

Brian Hamilton

Room Acoustic Simulation: Overview and Recent Developments

Abstract Simulation of room acoustics has applications in architecture, audio engineering, and video games. It is also gaining importance in virtual reality applications, where realistic 3D sound rendering plays an integral part in creating a sense of immersion within a virtual space. This tutorial will give an overview of room acoustic simulation methods, ranging from traditional approaches based on principles of geometrical and statistical acoustics, to numerical methods that solve the wave equation in three spatial dimensions, including recent developments of finite difference time domain (FDTD) methods resulting from the recently completed five-year NESS project (www.ness-music.eu). Computational costs and practical considerations will be discussed, along with the benefits and limitations of each framework. Simulation techniques will be illustrated through animations and sound examples.

David Berners

Modeling Circuits with Nonlinearities in Discrete Time

Abstract Modeling techniques for circuits with nonlinear components will be discussed. Nodal analysis and K-method models will be developed and compared in the context of delay-free loop resolution. Standard discretization techniques will be reviewed, including forward- and backward-difference and bilinear transforms. Interaction between nonlinearities and choice of discretization method will be discussed. Piecewise functional models for memoryless nonlinearities will be developed, as well as iterative methods for solving equations with no analytic solution. Emphasis will be placed on Newton's method and related techniques. Convergence and seeding for iterative methods will be reviewed. Relative computational expense will be discussed for piecewise vs. iterative approaches. Time permitting, modeling of time-varying systems will be discussed.

Julian D. Parker

From Algorithm to Instrument

Abstract The discipline of designing algorithms for creative processing of musical audio is now fairly mature in academia, as evidenced by the continuing popularity of the DAFX conference. However, this large corpus of work is motivated primarily by the traditional concerns of the academic signal-processing community - that being technical novelty or improvement in quantifiable metrics related to signal quality or computational performance. Whilst these factors are extremely important, they are only a small part of the process of designing an inspiring and engaging tool for the creative generation or processing of sound. Algorithms for this use must be designed with as much thought given to subjective qualities like aesthetics and usability as to technical considerations. In this tutorial I present my own experiences of trying to bridge this gap, and the design principles I've arrived at in the process. These principles will be illustrated both with abstract examples and with case studies from the work I've done at Native Instruments.

NONLINEAR HOMOGENEOUS ORDER SEPARATION FOR VOLTERRA SERIES IDENTIFICATION

Damien Bouvier, Thomas Hélie, David Roze

S3AM team, IRCAM-CNRS-UPMC UMR 9912
1, place Igor Stravinsky, 75004 Paris, France
damien.bouvier@ircam.fr

ABSTRACT

This article addresses identification of nonlinear systems represented by Volterra series. To improve the robustness of some existing methods, we propose a pre-processing stage that separates nonlinear homogeneous order contributions from which Volterra kernels can be identified independently. The proposed separation method exploits phase relations between test signals rather than amplitude relations that are usually used. This method is compared with standard separation process. Its contribution to identification is illustrated on a simulated loudspeaker with nonlinear suspension.

1. INTRODUCTION

Volterra series are a representation formalism for dynamical system with memory and weak nonlinearities. In audio, it has been widely used in various applications such as simulation of nonlinear resonators (brass instruments [1], Moog ladder filter [2], etc), distortion effect [3, 4], audio transducer analysis [5]. For any purpose (analysis, simulation or even system control), the use of Volterra series has to begin with the computation of the Volterra kernels, either directly from the system's dynamical equation [2], or by identification from inputs/outputs measurement [6].

This paper focuses on the identification problem in blind context, that is, without assuming any particular structures (Hammerstein, Wiener-Hammerstein, etc) [7, 8] or parametric model [9]. To this end, we propose a pre-processing stage that separates nonlinear homogeneous order contributions from which Volterra kernels can be identified independently. Such an approach is commonly used (see [10–14]), but the separation process relies on amplitude discrimination, that rapidly leads to ill-conditioned problems. To improve robustness, a new method is proposed that also exploits phase relations. This pre-processing method is embedded into a kernel identification method proposed in [6].

This paper is organized as follows. Section 2 gives an overview of Volterra Series and standard order separation. In Section 3, the proposed separation method is presented, and its advantages and disadvantages are discussed. Finally, in Section 4, simulation of a loudspeaker with nonlinear suspension is used to compare the separation methods and their contribution to identification.

2. RECALLS ON VOLTERRA SERIES AND ORDER SEPARATION

2.1. Overview on Volterra Series

An overview of the Volterra formalism is given here; further and more thorough explanations can be found in [15, 16], among the vast literature on Volterra series.

Definition 1 (Volterra series). A nonlinear causal time-invariant system is described by a Volterra series $\{h_n\}_{n \geq \mathbb{N}^*}$ if, for all input signals u such that $\|u\|_\infty < \rho$, the output signal y is given by the following Volterra operator:

$$y = V[u] = \sum_{n=1}^{\infty} y_n \quad (1)$$

where, for continuous-time systems:

$$y_n(t) = \int_{\mathbb{R}_+^n} h_n(\tau_1, \dots, \tau_n) \prod_{i=1}^n u(t - \tau_i) d\tau_i \quad (2)$$

and for discrete-time systems:

$$y_n[l] = \sum_{\mathbb{N}^n} h_n[m_1, \dots, m_n] \prod_{i=1}^n u[l - m_i] \quad (3)$$

and with ρ the convergence radius of the characterising function $\Phi_h(x) = \sum_{n=1}^{+\infty} \|h_n\|_1 x^n$. The terms h_n are called the *Volterra kernels* of the system, and terms y_n the *nonlinear homogeneous order contributions* (or more simply the nonlinear orders).

In the following, for sake of notation, continuous-time signals and systems will be used; if not specified otherwise, results are also valid for their discrete-time counterparts.

Remark 1 (Convergence). It has been shown in [17, 18] that there exists a large class of well-posed systems for which we know how to compute the convergence radius of the Volterra series. In this work, we will always assume that convergence's conditions are met.

Remark 2 (Order and memory truncation). In numerical implementation for simulation or identification, it will be necessary to truncate both infinite sums (i.e. for nonlinear orders and memory). Thus, in practice, Volterra series can only be used to approximate systems with small nonlinearities (limited to the first few nonlinear orders) and with finite memory [19].

Remark 3 (Non-uniqueness of kernels). It can easily be seen from (2) that kernels are not uniquely defined. To circumvent this problem for identification purpose, uniquely-defined forms can be specified, such as the triangular or symmetric kernels (where the kernel is invariant to any permutation of its arguments).

Remark 4 (Frequency domain kernels). As it is common for linear filters, it is possible to work in the frequency domain by means of a Laplace or Fourier transform.

2.2. Volterra operator properties

In the following definition, $L^p(\mathbb{E}, \mathbb{F})$ denotes the standard Lebesgue-space of functions from vector spaces \mathbb{E} to \mathbb{F} with p -norm.

Definition 2 (Volterra operator of order n). Let $h_n \in L^1(\mathbb{R}_+^n, \mathbb{R})$ be a Volterra kernel, for $n \in \mathbb{N}^*$. Then we introduce the multi-linear operator $V_n : L^\infty(\mathbb{R}, \mathbb{R}) \times \dots \times L^\infty(\mathbb{R}, \mathbb{R}) \mapsto L^\infty(\mathbb{R}, \mathbb{R})$ such that function $V_n[u_1, \dots, u_n]$ is defined $\forall t \in \mathbb{R}$ by

$$V_n[u_1, \dots, u_n](t) = \int_{\mathbb{R}_+^n} h_n(\tau_1, \dots, \tau_n) \prod_{i=1}^n u_i(t - \tau_i) d\tau_i \quad (4)$$

If $u_1 = \dots = u_n = u$, $V_n[u, \dots, u] = y_n$.

Property 1 (Symmetry). Given a symmetric kernel h_n , the corresponding Volterra operator V_n is also symmetric, meaning

$$V_n[u_{\pi(1)}, \dots, u_{\pi(n)}](t) = V_n[u_1, \dots, u_n](t) \quad (5)$$

for any permutations π and $\forall t \in \mathbb{R}$.

In the following, symmetry of h_n and V_n will be supposed.

Property 2 (Multilinearity and homogeneity). Volterra operator V_n is multilinear; i.e. for any signals u_1, u_2 , and any scalars λ, μ ,

$$\begin{aligned} V_n[\lambda u_1 + \mu u_2, \dots, \lambda u_1 + \mu u_2](t) &= \\ \sum_{q=0}^n \binom{n}{q} \lambda^{n-q} \mu^q V_n[\underbrace{u_1, \dots, u_1}_{n-q}, \underbrace{u_2, \dots, u_2}_q](t) \end{aligned} \quad (6)$$

This also implies that V_n is a homogeneous operator of degree n , i.e. for any signal u and scalar α ,

$$V_n[\alpha u, \dots, \alpha u](t) = \alpha^n V_n[u, \dots, u](t) \quad (7)$$

2.3. State-of-the-art order separation

Nonlinear homogeneous order separation implies the ability to recover signals y_n from the output y of a system described by a Volterra series truncated to order N .

From (7), $V[\alpha u](t) = \sum_{n=1}^N \alpha^n y_n(t)$. Consider a collection of input signals $u_k(t) = \alpha_k u(t)$, with $\alpha_k \in \mathbb{R}^*, k = 1, \dots, N$ and note $z_k(t) = V[u_k](t)$ their corresponding output through the system; then, for all time t :

$$\begin{bmatrix} z_1 \\ z_2 \\ \vdots \\ z_N \end{bmatrix}(t) = \begin{bmatrix} \alpha_1 & \alpha_1^2 & \dots & \alpha_1^N \\ \alpha_2 & \alpha_2^2 & \dots & \alpha_2^N \\ \vdots & \vdots & \ddots & \vdots \\ \alpha_N & \alpha_N^2 & \dots & \alpha_N^N \end{bmatrix} \cdot \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_N \end{bmatrix}(t), \quad \alpha_k \in \mathbb{R}^* \\ \mathbf{Z}(t) = \mathbf{A} \cdot \mathbf{Y}(t) \quad (8)$$

Since \mathbf{A} is a Vandermonde matrix, it is invertible if and only if all α_k are different; hence it is possible to recover terms y_n .

But for real-valued α_k , this type of matrix is known for becoming rapidly ill-conditioned when its size grows (meaning small noise in the measured outputs would become a large error in the estimation); so robustness decreases rapidly when the truncation order N increases. In order to circumvent that, it is possible to solve the linear problem $\mathbf{Z} = \mathbf{AY}$ by using a Newton Recursive or Lagrange Recursive method [20, Algorithm 4.6.1 and 4.6.2].

But in practice, this approach is still very sensitive to the choice of amplitude factors α_k . Indeed, for small amplitudes, higher orders will be hidden in measurement noise, while high values of α_k will potentially overload the system, or simply lead it out of its Volterra convergence radius.

Despite those disadvantages, this separation method has been used intensively for simplifying identification process [13]; it is generally used in frequency domain (the previous equations and remarks remains valid) and jointly with frequency probing methods [10,11]; recently, a maximum order truncation estimation method has been constructed from it [14]. In the following, this method will be referred to as the *Amplitude Separation* (AS) method.

3. PHASE-BASED HOMOGENEOUS SEPARATION METHOD

The starting point of the proposed separation method are the following remarks:

- using the AS method with factor 1 and -1 , it is possible to separate odd and even orders by inverting a mixing matrix \mathbf{A} with optimum condition number;
- multiplying a signal by amplitude factor -1 is equivalent to taking its opposite phase;
- thus, for a system truncated to order $N = 2$, there exists a robust separation method that relies only on *phase* deconstruction and reconstruction between tests signals.

The main idea of this paper is to generalize the use of phase for robust separation method to Volterra systems with truncation $N > 2$.

3.1. Method for complex-valued input

This section proposes a theoretical separation method relying on the use of complex signals $u(t) \in \mathbb{C}$ as system inputs.

3.1.1. Principle

Using complex signals, factors α_k in the AS method are not limited to real scalar. So it would be possible to choose values which only differs by their phase (e.g. are on the unit circle instead of the real axis). Noticing that 1 and -1 are the two square root of unity, a natural extension of the toy-method proposed for order-2 systems would be to take the N -th roots of unity as factors α_k . Choosing $\alpha_k = w_N^k$ with $w_N = e^{j\frac{2\pi}{N}}$, (8) becomes, for all time t :

$$\begin{bmatrix} z_1 \\ z_2 \\ \vdots \\ z_N \end{bmatrix}(t) = \begin{bmatrix} w_N & w_N^2 & \dots & w_N^N \\ w_N^2 & w_N^4 & \dots & w_N^{2N} \\ \vdots & \vdots & \ddots & \vdots \\ w_N^N & w_N^{2N} & \dots & w_N^{N^2} \end{bmatrix} \cdot \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_N \end{bmatrix}(t), \quad w_N = e^{j\frac{2\pi}{N}} \\ \mathbf{Z}(t) = \mathbf{W}_N \cdot \mathbf{Y}(t) \quad (9)$$

where \mathbf{W}_N is the Discrete Fourier Transform (DFT) matrix of order N (after a column and row permutation¹). It is important to note that here the DFT does not apply on time but on the homogeneous nonlinear orders.

Since the DFT is invertible, order separation is possible. Furthermore, the DFT matrix is well-conditioned², and the solution

¹It suffices to consider vectors $\hat{\mathbf{Z}}(t) = [z_N, z_1, \dots, z_{N-1}]^T$ and $\hat{\mathbf{Y}} = [y_N, y_1, \dots, y_{N-1}]^T$ to recover the usual DFT matrix.

²Its condition number is even optimum, since it is 1 for any order N .

can be computed using the Fast Fourier Transform algorithm³. In the following, this method will be referred to as the *Phase Separation* (PS) method.

3.1.2. Nonlinear order aliasing and rejection factor

Given the N -periodicity of N -th root of unity w_n , the output of a Volterra system with no order truncation is:

$$V[w_N u](t) = \sum_{n=1}^N w_n^n \sum_{r=0}^{\infty} y_{n+rN}(t) \quad (10)$$

By applying the PS method, estimation \tilde{Y} of nonlinear orders 1 to N yields:

$$\tilde{Y}(t) = \begin{bmatrix} y_1 + \sum_{r=1} y_{1+rN} \\ y_2 + \sum_{r=1} y_{2+rN} \\ \vdots \\ y_N + \sum_{r=1} y_{N+rN} \end{bmatrix} (t) \quad (11)$$

Equation (11) reveals that estimation \tilde{y}_n is perturbed by a residual term $\sum_{r=1} y_{n+rN}$, which is structured as an aliasing with respect to the nonlinear order: we⁴ call this effect the *nonlinear order aliasing*.

For a band-limited input signal, presence of term y_{n+k} in the estimated signal \tilde{y}_n means presence of higher-frequency components than expected; therefore, this artefact can help detect a wrong truncation order N . But moreover, (11) permits to create higher-order rejection by using amplitude as a *contrast factor*. Taking $\alpha_k = \rho w_N^k$, where ρ is a positive real number less than 1, estimation \tilde{Y} using PS method becomes

$$\tilde{Y}(t) = \begin{bmatrix} \rho & & & 0 \\ & \rho^2 & & \\ & & \ddots & \\ 0 & & & \rho^N \end{bmatrix} \begin{bmatrix} y_1 + \sum_{r=1} \rho^{rN} y_{1+rN} \\ y_2 + \sum_{r=1} \rho^{rN} y_{2+rN} \\ \vdots \\ y_N + \sum_{r=1} \rho^{rN} y_{N+rN} \end{bmatrix} (t), \quad (12)$$

creating a ρ^N ratio between desired signal $y_n(t)$ and the first perturbation $y_{n+N}(t)$. Thus parameters N and ρ enables to reach a required Signal-to-Noise Ratio (SNR).

However, the need of complex input (and output) signals prevents the use of PS method in practice.

3.2. Application to real-valued input

Consider the real signal $v(t)$ constructed as follows:

$$v(t) = w u(t) + \overline{w u(t)} = 2 \operatorname{Re}[w u(t)] \quad (13)$$

where w is a complex scalar on the unit circle (such that $\overline{w} = w^{-1}$) and $u(t)$ a complex signal. Therefore, using property (6), and assuming symmetry for the operator V_n (meaning symmetry

³Even if the gain in time computation is not significant since generally N is not a power of 2 and is not very high.

⁴We and the anonymous reviewer that proposed this relevant expression.

for kernel h_n), the order n contribution is:

$$\begin{aligned} y_n(t) &= V_n[v, \dots, v](t) \\ &= V_n[w u + \overline{w} \bar{u}, \dots, w u + \overline{w} \bar{u}](t) \\ &= \sum_{q=0}^n \binom{n}{q} w^{n-q} \overline{w}^q V_n[\underbrace{u, \dots, u}_{(n-q) \text{ times}}, \underbrace{\bar{u}, \dots, \bar{u}}_q](t) \\ &= \sum_{q=0}^n \binom{n}{q} w^{n-2q} M_{n,q}(t) \end{aligned} \quad (14)$$

with $M_{n,q}(t) = V_n[\underbrace{u, \dots, u}_{n-q}, \underbrace{\bar{u}, \dots, \bar{u}}_q](t) \in \mathbb{C}$. This term represents the homogeneous contribution of order n for an equivalent multi-input system excited by combinations of u and \bar{u} .

Remark 5. By symmetry of V_n , there is $M_{n,q}(t) = \overline{M_{n,n-q}(t)}$ and term $M_{n,n/2}(t)$, for even n , is real. Therefore, from sum over q in (14), realness of $y_n(t)$ is recovered.

Remark 6. By their definition, terms $M_{n,0}(t)$ (respectively $M_{n,n}(t)$) are homogeneous contribution of order n of the system excited by the complex signal $u(t)$ (resp. $\bar{u}(t)$).

Equation (14) shows that, in the output term y_n , there is more than one characterising phase factor w . So PS method is not directly exploitable to separate terms y_n .

3.3. Method for real-valued input

Difficulty analysis on an example: Consider a system truncated to $N = 3$ with input the real signal described in (13); then, omitting temporal dependency, its nonlinear orders are:

$$\begin{aligned} y_1 &= w M_{1,0} + w^{-1} M_{1,1} \\ y_2 &= w^2 M_{2,0} + 2 M_{2,1} + w^{-2} M_{2,2} \\ y_3 &= w^3 M_{3,0} + 3 w M_{3,1} + 3 w^{-1} M_{3,2} + w^{-3} M_{3,3} \end{aligned} \quad (15)$$

Only 7 different phase terms appears (from w^{-3} to w^3). Consider a collection of $K = 7$ real signals $v_k(t) = w_K^k u(t) + \overline{w_K^k u(t)}$, where w_K is the first K -th root of unity, and $z_k(t) = V[v_k](t)$ their corresponding output through the system. Then:

$$\begin{bmatrix} z_1 \\ z_2 \\ z_3 \\ z_4 \\ z_5 \\ z_6 \\ z_7 \end{bmatrix} (t) = \mathbf{W}_K \cdot \begin{bmatrix} M_{2,1} \\ M_{1,0} + 3M_{3,1} \\ M_{2,0} \\ M_{3,0} \\ M_{3,3} \\ M_{2,2} \\ M_{1,1} + 3M_{3,2} \end{bmatrix} (t), \quad (16)$$

where \mathbf{W}_K is the DFT matrix⁵ of order K .

Therefore, by application of the PS algorithm (i.e. inversion of \mathbf{W}_K) on this of signals z_k , the right-hand side vector in (16) is recovered. Further separation could be made using two different amplitudes to discriminate between $M_{1,0}$ and $M_{3,1}$, and thus be able to reconstruct nonlinear orders y_n using (14).

⁵It is important to notice that the right-hand side vector has hermitian symmetry, due to its Fourier Transform (the left-hand side) being real.

Generalization: Using (14), it is possible to generalize phase grouping shown in (15) as follows (see Appendix 7 for detailed computation):

$$y(t) = \sum_{\substack{-N \leq p \leq N \\ p \text{ even}}} w^p \sum_{\substack{1 < |p| \leq n \leq N \\ n \text{ even}}} \binom{n}{\frac{n-p}{2}} M_{n, \frac{n-p}{2}}(t) + \sum_{\substack{-N \leq p \leq N \\ p \text{ odd}}} w^p \sum_{\substack{|p| \leq n \leq N \\ n \text{ odd}}} \binom{n}{\frac{n-p}{2}} M_{n, \frac{n-p}{2}}(t) \quad (17)$$

So, by applying a PS algorithm of order $K = 2N + 1$, we can separate the terms

$$Q_p(t) = \sum_{\substack{|p| \leq n \leq N \\ n \equiv p \pmod{2}}} \binom{n}{\frac{n-p}{2}} M_{n, \frac{n-p}{2}}(t) \quad (18)$$

with $-N \leq p \leq N$. Then, application of the AS algorithm on each Q_p (with $\lfloor N/2 \rfloor$ amplitudes) gives all $M_{n,q}$ terms; terms y_n are reconstructed using (14).

This concatenation of PS and AS algorithm constitutes the proposed phase-based method, which will be referred to as *Phase-Amplitude Separation* (PAS) method. As will be pointed out in Section 4.3, it can be more interesting to use directly terms $M_{n,q}$ for identification, instead of nonlinear orders y_n ; this alternative process will be referred to as raw-PAS (rPAS) method.

3.4. Condition number

In numerical analysis, condition number κ measures the method's sensibility to noise in the measured data; it only depends on the solving method itself (the matrix to invert in linear problems), and not the data it is applied to. In this section, condition number is used to compare AS and PAS robustness⁶.

For AS method, amplitude factors α_k are chosen equally-spaced in dB scale, with alternating signs:

$$\alpha_{2p} = \gamma^{p-1} \text{ and } \alpha_{2p+1} = -\alpha_{2p} \quad (19)$$

where γ is a chosen spacing gain.

Because the DFT matrix is optimally conditioned, PAS method overall conditioning only depends on the K applications of AS method. Amplitude factors are also chosen equally-spaced in dB scale; the need to separate terms with same order-parity prevents us from using alternating signs. Only the worst condition number for all K sub-problems is reported.

Figure 1 presents conditioning for AS and PAS methods. For all maximum order truncation and gain spacing, an improvement from amplitude-based to phase-based method is visible. Furthermore, for negative gain spacing, optimum conditioning is divided by a factor 2 between AS and PAS. But, for both methods, the same behaviour when truncation order N increases is remarked (which is unsurprising due to the fact that PAS relies partly on AS).

Those results indicates that PAS robustness to noisy measurements should be better than AS.

⁶As PAS and rPAS methods share the same steps, their condition number are equal.

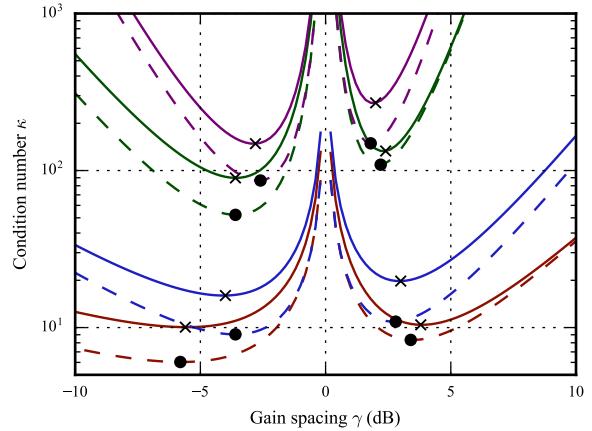


Figure 1: Condition number $\kappa = \|\mathbf{A}\| \|\mathbf{A}^{-1}\|$ (Frobenius norm): evolution w.r.t gain spacing γ and truncation order N (3 to 6, from bottom to top) for AS (solid line, \times indicating minima) and PAS (dashed line, \bullet indicating minima) methods.

4. APPLICATIONS

In order to test and compare the different separation method, data from simulation of a nonlinear device were used.

4.1. Simulation of a loudspeaker with nonlinear suspension

The simulated system was a loudspeaker with nonlinear suspension described by the modified Thiele & Small following equations:

$$u(t) = R_e i(t) + L \frac{di(t)}{dt} + Bl \frac{d\ell(t)}{dt} \quad (20a)$$

$$M \frac{d^2\ell(t)}{dt^2} = Bl i(t) - R_m \frac{d\ell(t)}{dt} - \sum_{n=1}^3 k_n \ell^n(t) \quad (20b)$$

where u is the voltage at the loudspeakers terminals, i the current flowing through it and ℓ the position of the diaphragm. The term $\sum_{n=1}^3 k_n \ell^n(t)$ corresponds to the nonlinear force that the suspension applies on the diaphragm.

Consider the state vector $\mathbf{x} = [i \ \ell \ d(\ell)/dt]^T$. Then, using state-space formalism, the system of input $u(t)$ and output $i(t)$ is written:

$$\begin{cases} \dot{\mathbf{x}} = \mathbf{Ax} + \mathbf{Bu} + \mathbf{K}_2(\mathbf{x}, \mathbf{x}) + \mathbf{K}_3(\mathbf{x}, \mathbf{x}, \mathbf{x}) \\ i = \mathbf{Cx} \end{cases} \quad (21)$$

where $\dot{\mathbf{x}}$ indicates the temporal derivative of \mathbf{x} , and:

$$\mathbf{A} = \begin{bmatrix} -\frac{R_e}{L} & 0 & -\frac{Bl}{L} \\ 0 & 0 & 1 \\ \frac{Bl}{M} & -\frac{k_1}{M} & -\frac{R_m}{M} \end{bmatrix}, \mathbf{B} = \begin{bmatrix} 1 \\ \frac{L}{0} \\ 0 \end{bmatrix}, \mathbf{C} = [1 \ 0 \ 0]$$

and where \mathbf{K}_2 et \mathbf{K}_3 are multilinear functions of \mathbf{x} such that:

$$\mathbf{K}_2(\mathbf{a}, \mathbf{b}) = \begin{bmatrix} 0 \\ 0 \\ -\frac{k_2}{M} a_2 b_2 \end{bmatrix}, \text{ and } \mathbf{K}_3(\mathbf{a}, \mathbf{b}, \mathbf{c}) = \begin{bmatrix} 0 \\ 0 \\ -\frac{k_3}{M} a_2 b_2 c_2 \end{bmatrix}$$

R_e	$5, 7 \Omega$	R_m	$4,06 \cdot 10^{-1} \text{ Nm}^{-1}\text{s}$
L	$1, 1 \cdot 10^{-1} \text{ H}$	k_1	$912, 3 \text{ kg} \cdot \text{s}^{-2}$
Bl	$2, 99 \text{ NA}^{-1}$	k_2	$611.5 \text{ kg} \cdot \text{m}^{-1}\text{s}^{-2}$
M	$1, 9 \cdot 10^{-3} \text{ kg}$	k_3	$8, 0 \cdot 10^7 \text{ kg} \cdot \text{m}^{-2}\text{s}^{-2}$

Table 1: Thiele & Small parameters of SICA loudspeaker model Z000900, given by constructor [21] (apart from Bl and the k_n , which were measured experimentally).

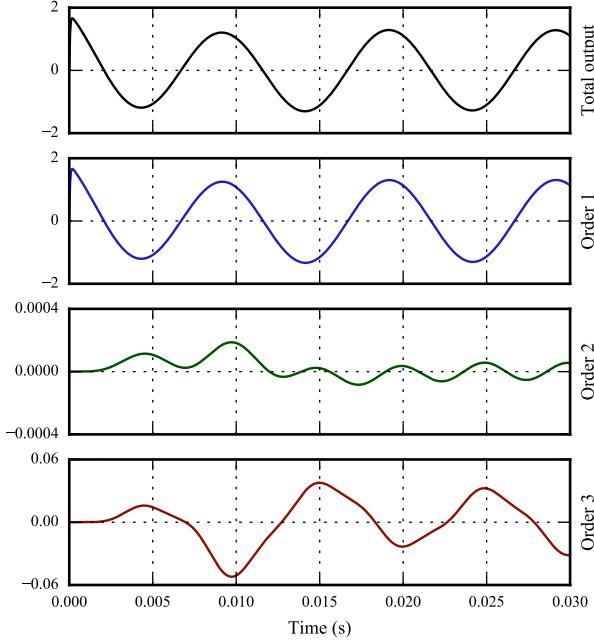


Figure 2: Total output and first three nonlinear orders y_n of the simulated loudspeaker for a sinusoidal input of frequency 100 Hz and amplitude 10 V.

Simulations were made using numerical methods described in Appendix 8, with parameters given in Table 1.

4.2. Error separation

Order separation method are compared using the relative error ϵ_n of each order n for noisy measurements of system outputs, where

$$\epsilon_n = \frac{\text{RMS}(\tilde{y}_n[k] - y_n[k])}{\text{RMS}(y_n[k])} \quad (22)$$

with y_n the true nonlinear homogeneous contribution, \tilde{y}_n their estimation, and RMS the standard Root-Mean Square measure.

Simulation were made at a sampling frequency of 20000 Hz, and with a truncation order $N = 5$. All signals were 1 second long.

Figure 2 shows the total output and first three nonlinear orders of the system. Its linear behaviour is predominant: y_1 is one (respectively four) magnitude order above y_3 (resp. y_2). This large difference of amplitude between signals y (or equivalently y_1) and y_2 means that the quadratic order could be hidden by noise in relatively high SNR.

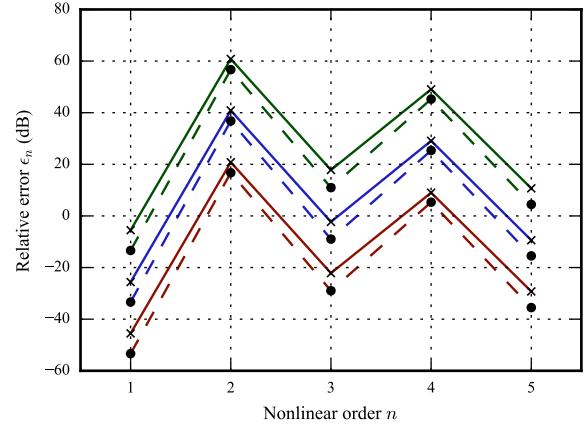


Figure 3: Relative error ϵ_n (in dB) of separation w.r.t. nonlinear order n and SNR (80 dB, 60 dB and 40 dB from bottom to top) for AS (solid line with \times) and PAS (dashed line with \bullet) methods. Input signals are linear sweep of amplitude 10 V going from 30 Hz to 200 Hz.

Figure 3 compares the separation measure ϵ_n for both AS and PAS methods applied to noisy output data with different SNR. Estimation on clean data output is not shown here; in this case, both methods perform similarly and give the true homogeneous contribution y_n (within machine accuracy). The high values of relative errors ϵ_2 and ϵ_4 is due to the smaller amplitude of signals y_2 and y_4 in respect to the other orders, which makes their estimation more sensitive to measurement noise.

Furthermore, Figure 3 shows that PAS outperforms AS method in presence of noise. For same order n and SNR, the gain in ϵ_n is around 6 dB.

4.3. Kernel identification using order separation

4.3.1. Identification algorithms

The standard Volterra identification method used in this paper is the Korenberg's algorithm for Least-Squares problem (KLS) as described in [6].

It consists of solving the following linear-in-the-parameters problem:

$$\mathbf{y} = \Phi \mathbf{f} \quad (23)$$

where:

- vector $\mathbf{y} = [y[0], \dots, y[L-1]]^T$ is the concatenation of all output samples, with L the signal length (in samples);
- $\Phi = [\phi[0], \dots, \phi[L-1]]^T$ is the input combinatorial matrix, where vector $\phi[k]$ regroups all cross-product $u[k-k_1] \dots u[k-k_n]$ of the input signal at time k (for all orders n);
- vector \mathbf{f} is the concatenation of all kernels values (for all orders n).

Using order separation (AS or PAS method), (23) is separated into N smaller problems:

$$\mathbf{y}_n = \Phi_n \mathbf{f}_n \quad (24)$$

Identification is then carried on separately for each kernel; this leads to AS-KLS and PAS-KLS algorithm.

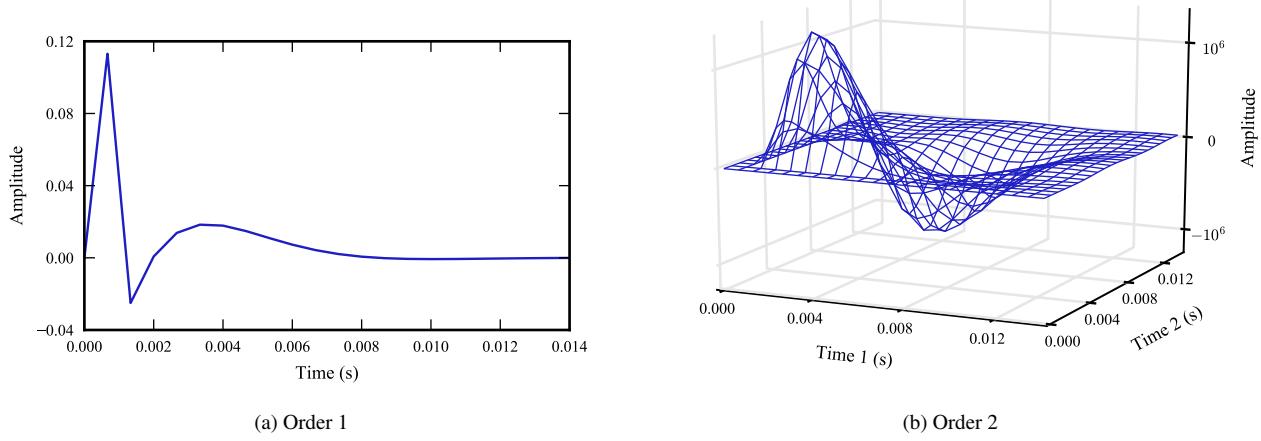


Figure 4: First and second-order kernels computed from (30) and (31) in Appendix 8.2.

Furthermore, using rPAS method, (23) is separated into the following problems

$$\mathbf{M}_{n,q} = \Phi_{n,q} \mathbf{f}_n \quad (25)$$

for $n = 1, \dots, N$ and $q = 1, \dots, \lfloor n/2 \rfloor$ ⁷. Identification is then carried on separately for each couple (n, q) ; kernels coefficients are taken as the mean of the $\lfloor n/2 \rfloor$ estimations. This gives the proposed rPAS-KLS identification algorithm.

4.3.2. Identification results

The aforementioned identification methods were tested on the nonlinear loudspeaker, truncated to order $N = 3$.

Simulation were made at a sampling frequency of 1500 Hz. Gaussian noise signals of amplitude 10 V and 2 second length were used for input test signals. Kernel memory length was supposed to be equal to $M = 22$ samples (equivalent to 14 ms memory).

First two kernels of the nonlinear loudspeaker are given in Figure 4. Both linear and quadratic kernels shows exponential-like decay; the chosen memory length proves to be adequate at this sampling rate.

For all algorithms, relative identification error (computed as $\xi_n = \text{RMS}(h_n - h_n)/\text{RMS}(h_n)$) are given in Figure 5.

When using clean output data, the addition of a prior order separation stage before identification improves greatly overall estimation process: error is reduced of more than 30 dB for the first order and 50 dB for the second. In this case, AS-KLS and PAS-KLS performs similarly; this is due to the similar separation results of AS and PAS methods with clean data.

Likewise, the better separation robustness of PAS over AS (around 6 dB gain) induces PAS-KSL to be more robust to noise than AS-KLS (around 2 dB gain).

⁷We consider only one case for the complex conjugates couples (n, q) and $(n, n - q)$.

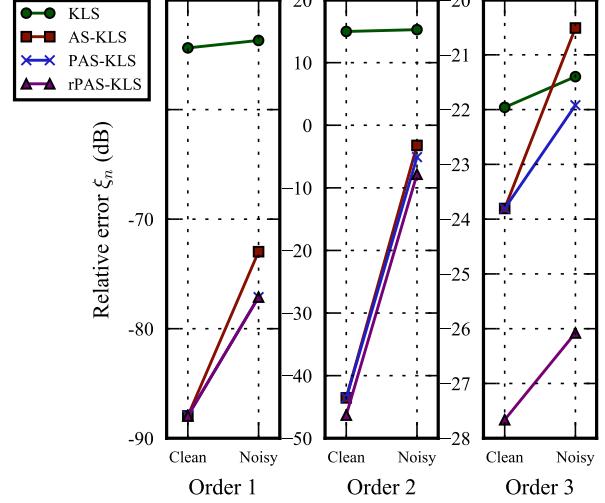


Figure 5: Relative error ξ_n (in dB) of identification w.r.t. order n for all identification methods using clean and noisy data (SNR = 80 dB)

Furthermore, for second and third-order, rPAS-KLS algorithm offers a little improvement over PAS-KLS. This amelioration is the direct consequence of using terms $M_{n,q}$ rather than nonlinear orders y_n directly; indeed, averaging over all estimated $f_{n,q}$ for identification of kernel h_n can be seen as a way to artificially increase the data size used for identification, thus leading to better estimates.

The first two estimated kernels using rPAS-KLS on clean data are shown in Figure 6. We can see that the second-order kernel, which has a relative error $\xi_2 = -46$ dB, is well-identified using this method.

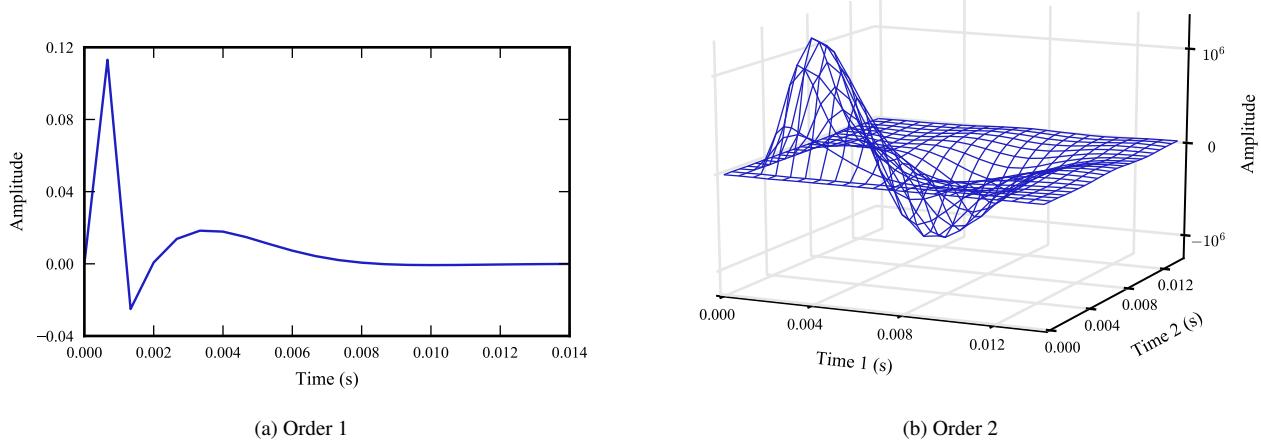


Figure 6: First and second-order kernels estimated with rPAS-KLS.

5. CONCLUSIONS

In this paper, a new method of nonlinear homogeneous order separation for Volterra series was proposed, based on phase dissimilarity. This method has been shown to be better conditioned, and thus more robust to noise, than amplitude-based separation algorithm.

Furthermore, two Volterra kernels identification methods were constructed combining the proposed separation process and a Least Squares-based identification algorithm from literature. Those methods have been shown to greatly improve kernel estimation.

Future works will be on the reproduction, with real signals, of the product's complex behaviour that enables usage of the theoretical PS method. An other interest would be to derive another identification method from Korenberg's algorithm that could be used directly on the phase-grouped terms Q_p , without needing to use AS method for further separation.

6. REFERENCES

- [1] Thomas Hélie and Vanessa Smet, “Simulation of the weakly nonlinear propagation in a straight pipe: application to a real-time brassy audio effect,” in *16th Mediterranean Conference on Control and Automation*. IEEE, 2008, pp. 1580–1585.
- [2] Thomas Hélie, “Volterra series and state transformation for real-time simulations of audio circuits including saturations: Application to the Moog ladder filter,” *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 18, no. 4, pp. 747–759, 2010.
- [3] Finn T. Agerkvist, “Volterra Series Based Distortion Effect,” in *Audio Engineering Society Convention 129*. Audio Engineering Society, 2010.
- [4] Lamberto Tronchin, “The emulation of nonlinear time-invariant audio systems with memory by means of Volterra series,” *Journal of the Audio Engineering Society*, vol. 60, no. 12, pp. 984–996, 2013.
- [5] Arie J.M. Kaizer, “Modeling of the nonlinear response of an electrodynamic loudspeaker by a Volterra series expansion,” *Journal of the Audio Engineering Society*, vol. 35, no. 6, pp. 421–433, 1987.
- [6] Yves Goussard, William C Krenz, Lawrence Stark, and Guy Demoment, “Practical identification of functional expansions of nonlinear systems submitted to non-Gaussian inputs,” *Annals of biomedical engineering*, vol. 19, no. 4, pp. 401–427, 1991.
- [7] Antonin Novak, Laurent Simon, Pierrick Lotton, and Frantisek Kadlec, “Modeling of nonlinear audio systems using swept-sine signals: Application to audio effects,” in *Proc. of the 12th Int. Conference on Digital Audio Effects (DAFx-09)*, 2009, pp. 1–4.
- [8] Marc Rébillat, Romain Hennequin, Etienne Corteel, and Brian F. G. Katz, “Identification of cascade of Hammerstein models for the description of nonlinearities in vibrating devices,” *Journal of Sound and Vibration*, vol. 330, no. 5, pp. 1018–1038, 2011.
- [9] Stephen A. Billings, *Nonlinear system identification: NARMAX methods in the time, frequency, and spatio-temporal domains*, John Wiley & Sons, 2013.
- [10] Stephen P. Boyd, Y. S. Tang, and Leon O. Chua, “Measuring volterra kernels,” *Circuits and Systems, IEEE Transactions on*, vol. 30, no. 8, pp. 571–577, 1983.
- [11] Delphine Bard and Göran Sandberg, “Modeling of Nonlinearities in Electrodynamic Loudspeakers,” in *Audio Engineering Society Convention 123*. Audio Engineering Society, 2007.
- [12] Finn Agerkvist, Antoni Torras-Rosell, and Richard McWalter, “Eliminating transducer distortion in acoustic measurements,” in *Audio Engineering Society Convention 137*. Audio Engineering Society, 2014.
- [13] Russell H. Lambert, “Vandermonde Method for Separation of Nonlinear Orders and Measurement of Linear Response,”

- in *Audio Engineering Society Convention 141*. Audio Engineering Society, 2016.
- [14] B. Zhang and S.A. Billings, “Volterra series truncation and kernel estimation of nonlinear systems in the frequency domain,” *Mechanical Systems and Signal Processing*, vol. 84, pp. 39–57, 2017.
 - [15] Wilson J. Rugh, *Nonlinear system theory*, Johns Hopkins University Press Baltimore, 1981.
 - [16] Stephen P. Boyd, Leon O. Chua, and Charles A. Desoer, “Analytical foundations of Volterra series,” *IMA Journal of Mathematical Control and Information*, vol. 1, no. 3, pp. 243–282, 1984.
 - [17] Thomas Hélie and Béatrice Laroche, “Computation of convergence bounds for Volterra series of linear-analytic single-input systems,” *Automatic Control, IEEE Transactions on*, vol. 56, no. 9, pp. 2062–2072, 2011.
 - [18] Thomas Hélie and Béatrice Laroche, “Computable convergence bounds of series expansions for infinite dimensional linear-analytic systems and application,” *Automatica*, vol. 50, no. 9, pp. 2334–2340, 2014.
 - [19] Stephen Boyd and Leon Chua, “Fading memory and the problem of approximating nonlinear operators with Volterra series,” *IEEE Transactions on circuits and systems*, vol. 32, no. 11, pp. 1150–1161, 1985.
 - [20] Gene H. Golub and Charles F. Van Loan, *Matrix computations*, vol. 3, JHU Press, 2012.
 - [21] SICA, “Constructor datasheet for SICA loudspeaker model Z000900,” <https://www.sica.it/media/Z000900.pdf>, Accessed: 2017-04-07.

7. APPENDIX: DETAILED COMPUTATION OF (17)

Consider a system truncated to $N = 3$ with real input signal described in (13). From (17), we have:

$$y_n(t) = \sum_{q=0}^n \binom{n}{q} w^{n-2q} M_{n,q}(t) \quad (26)$$

So, using (1), the overall output of the system is:

$$\begin{aligned} y(t) &= \sum_{n=1}^N y_n(t) \\ &= \sum_{n=1}^N \sum_{q=0}^n \binom{n}{q} w^{n-2q} M_{n,q}(t) \\ &= \sum_{\substack{1 \leq n \leq N \\ n \text{ even}}} \sum_{q=0}^n \binom{n}{q} w^{n-2q} M_{n,q}(t) \\ &\quad + \sum_{\substack{1 \leq n \leq N \\ n \text{ odd}}} \sum_{q=0}^n \binom{n}{q} w^{n-2q} M_{n,q}(t) \\ &= \sum_{\substack{1 \leq n \leq N \\ n \text{ even}}} \sum_{\substack{-n \leq p \leq n \\ p \text{ even}}} \binom{n}{(n-p)/2} w^p M_{n,(n-p)/2}(t) \\ &\quad + \sum_{\substack{1 \leq n \leq N \\ n \text{ odd}}} \sum_{\substack{-n \leq p \leq n \\ p \text{ odd}}} \binom{n}{(n-p)/2} w^p M_{n,(n-p)/2}(t) \end{aligned}$$

by posing $p = n - 2q$. Taking into account the fact that both sum are finite, it is possible to inverse their orders, and thus:

$$\begin{aligned} y(t) &= \sum_{\substack{-N \leq p \leq N \\ p \text{ even}}} w^p \left(\sum_{\substack{1 < |p| \leq n \leq N \\ n \text{ even}}} \binom{n}{(n-p)/2} M_{n,(n-p)/2}(t) \right) \\ &\quad + \sum_{\substack{-N \leq p \leq N \\ p \text{ odd}}} w^p \left(\sum_{\substack{|p| \leq n \leq N \\ n \text{ odd}}} \binom{n}{(n-p)/2} M_{n,(n-p)/2}(t) \right) \end{aligned} \quad (27)$$

8. APPENDIX: SYSTEM NUMERICAL APPROXIMATION

8.1. Numerical simulation

The input signal $u(t)$ is approximated at sampling rate f_s with a zero-order holder; then, following the state-space representation in (21), output current i at sample time l is given by:

$$i[l] = \sum_{n=1}^N \mathbf{C} \mathbf{x}_n[l] \quad (28)$$

where \mathbf{x}_n are the states of nonlinear homogeneous order n ; first three orders are given by the following recursive equations:

$$\mathbf{x}_1[l+1] = e^{\mathbf{A}T_s} \mathbf{x}_1[l] + \Delta_0 \mathbf{B} u[l] \quad (29a)$$

$$\mathbf{x}_2[l+1] = e^{\mathbf{A}T_s} \mathbf{x}_2[l] + \Delta_0 \mathbf{K}_2(\mathbf{x}_1[l], \mathbf{x}_1[l]) \quad (29b)$$

$$\begin{aligned} \mathbf{x}_3[l+1] &= e^{\mathbf{A}T_s} \mathbf{x}_3[l] + \Delta_0 \mathbf{K}_3(\mathbf{x}_1[l], \mathbf{x}_1[l], \mathbf{x}_1[l]) \\ &\quad + 2\Delta_0 \mathbf{K}_2(\mathbf{x}_1[l], \mathbf{x}_2[l]) \end{aligned} \quad (29c)$$

where $T_s = 1/f_s$ is the sampling time and $\Delta_0 = \mathbf{A}^{-1}(e^{\mathbf{A}T_s} - \mathbf{I})$ is a bias term due to sampling.

8.2. Discrete-time kernel computation

Discrete-time kernels h_n corresponding to the loudspeaker simulation using (28) and (29) are given by:

$$h_n[m_1, \dots, m_n] = \sum_{n=1}^N \mathbf{C} \mathbf{g}_n[m_1, \dots, m_n] \quad (30)$$

where \mathbf{g}_n are the input-to-states kernels of order n ; first three orders are given by:

$$\mathbf{g}_1[m] = (1 - \delta_{m,0}) e^{\mathbf{A}T_e(m-1)} \Delta_0 \quad (31a)$$

$$\mathbf{g}_2[m_1, m_2] = \sum_{l=0}^{\max\{m_1, m_2\}} e^{\mathbf{A}T_e l} \Delta_0 \mathbf{K}_2(\mathbf{g}_1[m_1-l], \mathbf{g}_1[m_2-l]) \quad (31b)$$

$$\begin{aligned} \mathbf{g}_3[m_1, m_2, m_3] &= \sum_{l=0}^{\max\{m_1, m_2, m_3\}} e^{\mathbf{A}T_e l} \Delta_0 \left(\mathbf{K}_3(\mathbf{g}_1[m_1-l], \mathbf{g}_1[m_2-l], \mathbf{g}_1[m_3-l]) \right. \\ &\quad \left. + 2\mathbf{K}_2(\mathbf{g}_1[m_1-l], \mathbf{g}_2[m_2-l, m_3-l]) \right) \end{aligned} \quad (31c)$$

with $\delta_{m,n}$ the Kronecker delta.

INTRODUCING DEEP MACHINE LEARNING FOR PARAMETER ESTIMATION IN PHYSICAL MODELLING

Leonardo Gabrielli *

A3LAB,
Università Politecnica delle Marche
Ancona, IT
l.gabrielli@univpm.it

Stefano Tomassetti *

A3LAB,
Università Politecnica delle Marche
Ancona, IT
tomassetti.ste@gmail.com

Stefano Squartini

A3LAB,
Università Politecnica delle Marche
Ancona, IT
s.squartini@univpm.it

Carlo Zinato

Viscount International SpA,
c.zinato@viscount.it

ABSTRACT

One of the most challenging tasks in physically-informed sound synthesis is the estimation of model parameters to produce a desired timbre. Automatic parameter estimation procedures have been developed in the past for some specific parameters or application scenarios but, up to now, no approach has been proved applicable to a wide variety of use cases. A general solution to parameters estimation problem is provided along this paper which is based on a supervised convolutional machine learning paradigm. The described approach can be classified as “end-to-end” and requires, thus, no specific knowledge of the model itself. Furthermore, parameters are learned from data generated by the model, requiring no effort in the preparation and labeling of the training dataset. To provide a qualitative and quantitative analysis of the performance, this method is applied to a patented digital waveguide pipe organ model, yielding very promising results.

1. INTRODUCTION

Almost all sound synthesis techniques require a nontrivial effort in the selection of the parameters, to allow for expressiveness and obtain a specific sound. The choice of the parameters depends on tone pitch, control dynamics, interpretation, and aesthetic criteria, with the aim of producing all the nuances required by musicians and their taste. Hereby, interest is given to the so-called *physically-informed* sound synthesis, a family of algorithms [1, 2] usually inspired by acoustic physical systems or derived from the transform in the digital domain of their formulation in the continuous-time domain. Such acoustic systems (e.g. strings, bores, etc.) often require simplifying hypotheses to limit the modeling complexity and to separate the acoustic phenomenon into different components. Notwithstanding this, the number of *micro*-parameters that control the sound and its evolution may be extremely large (see e.g. [3]) and if the effects of the parameters is intertwined the estimation effort may grow.

In the past some algorithms have been proposed to estimate some of the parameters of a physical model in an algorithmic fashion (see e.g. [4, 5]). These, however, require specific knowledge of physics, digital signal processing and psychoacoustic in order

to provide an estimate in a white-box approach. Furthermore, specific estimation algorithms must be devised for each parameter. To solve these issues, a black-box approach could be undertaken to provide a good estimate of all the parameters at once. The goal of this preliminary work is to support the thesis that adequate machine learning techniques can be identified to satisfactorily estimate a whole set of model parameters without specific physical knowledge or model knowledge.

In the past some works extended the use of early machine learning techniques to the parametrization of nonlinearities in physical models [6, 7], or employed nonlinear recursive digital filters as physical models and employed parameter estimation techniques mutuated from the machine learning literature for the estimate of the coefficients [8, 9]. More in the spirit of this paper comes the work of Cemgil et al. on the calibration of a simple physical model employing artificial neural networks [10]. This work, however, to the best of our knowledge saw no continuation. Recently another computational intelligence approach for the estimation of a synthesizer parameters using a multiple linear regression model has been proposed, employing hand-crafted features [11]. To the best of our knowledge, however, no further attempt has been made to the estimation of a physical model parameters for sound synthesis employing other machine learning approaches. From this point of view, the swift development of deep machine learning techniques, and the exciting results obtained by these in a plethora of application scenarios, including musical representation and regression [12] suggests their application to the problem at hand. Following the recent advances of deep neural networks in audio applications, we propose here an end-to-end approach to the parameter estimation of acoustic physical models for sound synthesis based on Convolutional Neural Networks (CNN). The training can be conducted in a supervised fashion, since the model itself can provide audio and ground-truth parameters in an automated fashion. To evaluate the approach, this concept is applied to a valuable use case, i.e. a commercial flue pipe organ physical model, detailed in [13]. The estimation yields promising results which call for more research work.

The paper outline follows. Section 2 provides a mathematical formulation of the problem and the machine learning techniques employed to provide a general solution to it. Section 3 describes a real-world use case for validation of the proposed techniques.

* This work is partly supported by Viscount International SpA

Section 4 reports the implementation details and the experiments conducted, while Section 5 provides the results of these experiments and discusses them. Finally in Section 6 conclusions are drawn and open avenues for research are suggested.

2. THE PROPOSED METHOD

A physical model solves a set of differential equations that model a physical system and requires a set of parameters θ to generate a discrete time audio sequence $s(n)$. The goal of the model is to approximate acoustic signals generated by the physical system in some perceptual sense. If the model provides a mapping from the parameters to the discrete sequence $s(n)$, the problem of estimating the parameters θ that yield a specific audio sequence identified as a target (e.g. an audio sequence sampled from the physical system we are approximating), is equivalent to finding the model inverse. Finding an inverse mapping (or an approximation thereafter) for the model is a challenging task to face, and a first necessary condition is the existence of the inverse for a given $s(n)$. Usually, however, in physical modelling applications, requirements are less strict, and generally it is only expected that audio signals match in perceptual or qualitative terms, rather than on a sample-by-sample basis. This means that, although, a signal $r(n)$ cannot be obtained from the model for any θ , a sufficiently close estimate $\hat{r}(n)$ may. Evaluating the distance between the two signals in psychoacoustic terms is a rather complex task and is out of the scope of this work.

Artificial neural networks, and more specifically, deep neural networks of recent introduction, are well established to solve a number of inverse modelling problems. Here, we propose the application of a convolutional neural network that, provided with an audio signal of maximum length L in a suitable time-frequency representation, can estimate model parameters $\hat{\theta}$ that fed to the physical model obtain an audio signal $\hat{s}(n)$ close to $s(n)$.

To achieve this, the inverse of the model must be learned employing deep machine learning techniques. If a supervised training approach is employed, the network must be fed with audio sequences and the related model parameters, also called *target*. The production of a dataset D of such tuples $D = \{(\theta_i, s_i(n)), i = 1, \dots, M\}$ allows the network to try learn the mapping that connects these. The production of the dataset is often a lengthy task and may require human effort. However, in this application, the model is known and, once implemented, it can be employed to automatically generate a dataset D in order to train the neural network.

The neural network architecture proposed here allows for end-to-end learning and is based on convolutional layers. Convolutional neural networks globally received attention from a large number of research communities and found application into commercial applications, especially in the field of image processing, classification, etc. They are also used with audio signals, where, usually, the signal is provided to the CNN in a suitable time-frequency representation, obtained by means of a Short-Time Fourier Transform (STFT) with appropriate properties of time-frequency localization. The architecture of a CNN is composed of several layers in the following form,

$$\mathbf{Z}^{(m)} = h(\sigma(\mathbf{Q}^{(m)})), \quad (1)$$

$$\mathbf{Q}^{(m)} = W^{(m)} * \mathbf{Z}^{(m-1)}, \quad (2)$$

and $\mathbf{Z}^{(0)} \equiv \mathbf{X}$, where M denotes the total number of layers,

$W^{(m)}$, $m = 1, \dots, M$ are the filter weights to be learned, $\sigma(\cdot)$ is a non-linear sigmoid activation function, $\mathbf{Z}^{(m-1)}$ is the output of layer $m - 1$, called *feature map*, $\mathbf{Q}^{(m)}$ is the result of convolution on the previous feature map and $h(\cdot)$ is a *pooling* function that reduces the feature map dimensionality. After M convolutional layers, one or more fully connected layers are added. The final layer has size p and outputs an estimate of the model parameters $\hat{\theta}$.

Learning is conducted according to an update rule, which is based on the evaluation of a loss function, such as

$$\ell(W, e) = \|\theta - \hat{\theta}^{(e)}\|_2 \quad (3)$$

where e is the training epoch. Training is iterated until a convergence criterion is matched or a maximum number of epochs has passed. To avoid overfitting and reduce training times early stopping by validation can be performed, which consists in evaluating after a constant number of epochs the loss, called validation loss, calculated against a validation set, i.e. a part of the dataset that is not used for training and is, hence, new to the network. Even if the training loss may still be improving, if the validation loss does not improve after some training epochs, the training may stop avoiding a network overfit.

Finally, once the network is trained, it can be fed with novel audio sequences to estimate the physical model parameters that can obtain a result close to the original. In the present work we employ additional audio sequences generated by the model in order to measure the distance in the parameter space between the parameters θ_i and $\hat{\theta}_i$. If non-labelled audio sequences are employed (e.g. sequences sampled from the real-world), it is not straightforward to validate the generated result, that is why in the present work no attempt has been made to evaluate the estimation performance of the network with real-world signals.

3. USE CASE

The method described in the previous section has been applied to a specific use case of interest, i.e. a patented digital pipe organ physical model. A pipe organ is a rather complex system [14, 15], providing a challenging scenario for physical modelling itself. This specific model, already employed on a family of commercial digital pipe organs, exposes 58 macro-parameters to be estimated for each key, some of which are intertwined in a non linear fashion and are acoustic-wise non-orthogonal (i.e. jointly affect some acoustic features of the resulting tone).

We introduce here some key terms for later use. A pipe organ has one or more keyboards (*manuals* or *divisions*), each of which can play several *stops*, i.e. a set of pipes, typically one or more per key, which can be activated or deactivated at once by means of a *drawstop*. When a stop is activated, air is ready to be conveyed to the embouchure of each pipe, and when a key is pressed, a valve is opened to allow air flow into the pipe. Each stop has a different timbre and pitch (generally the pitch of the central C is expressed in feet measuring the pipe length). From our standpoint, the concept of stop is very important, since each key in a stop will sound similar to the neighboring ones in terms of timbre and envelope, and each key will trigger different stops which may have similar pitch but different timbre. In a pipe organ it can be expected that pipes in a stop have consistent construction features (e.g. materials, geometry, etc.) and a physical model that mimics that pipe stop may have correlated features along the keys but this is not

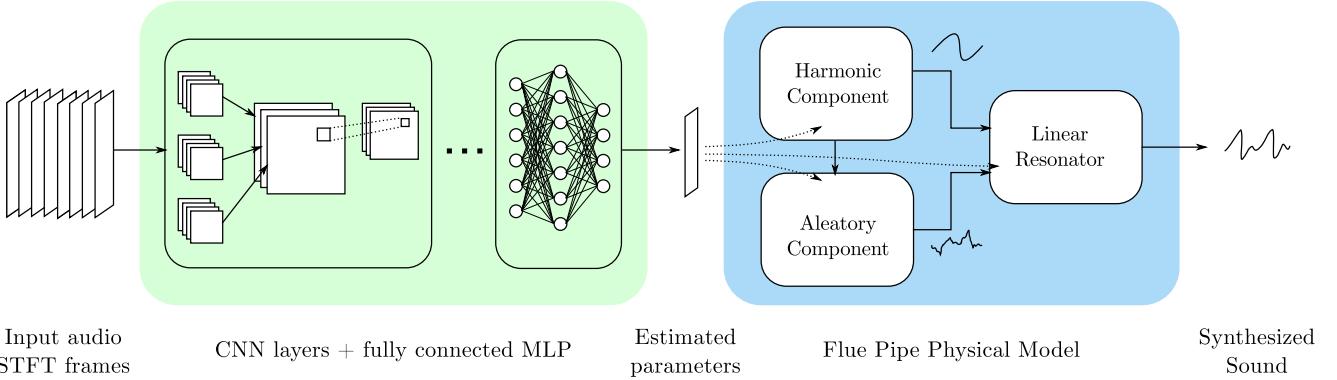


Figure 1: Overview of the proposed system including the neural network for parameter estimation and the physical model for sound synthesis.

an assumption that can be done, so it is necessary to conduct an estimate of the parameters for each key in a stop.

The physical model employed in this work is meant to emulate the sound of flue pipes and is described in detail in the related patent. To summarize, it is constituted by three main parts:

1. exciter: models the wind jet oscillation that is created in the embouchure and gives rise to an air pressure fluctuation,
2. resonator: a digital waveguide structure that simulates the bore,
3. noise model: a stochastic component that simulates the air noise modulated by the wind jet.

The parameters involved in the sound design are widely different in range and meaning, and are e.g. digital filters coefficients, non-linear functions coefficients, gains, etc. The diverse nature of the parameters requires a normalization step which is conducted on the whole training set and maps each parameter in a range [-1, 1], in order for the CNN to learn all parameters employing the same arithmetic. A de-normalization step is required, thus, to remap the parameters to their original range.

Figure 1 provides an overview of the overall system for parameter estimation and sound generation including the proposed neural network and the physical model.

4. IMPLEMENTATION

The CNN and the machine learning framework has been implemented as a python application employing Keras¹ libraries and Theano² as a backend, running on a Intel i7 Linux machine equipped with 2 x GTX 970 graphic processing units. The physical model is implemented as both an optimized DSP algorithm and a PC application. The application has been employed in the course of this work and has been modified to allow producing batches of audio sequences and labels for each key in a stop (e.g. to produce the dataset, given some specifications). Each audio sequence contains a few seconds of a tone of specific pitch with given parameters.

A dataset of 30 *Principal* pipe organ stops, each composed of 74 audio files, one per note, has been created taking the parameters from a database of pre-existing stops hand-crafted by expert musicians to mimic different hystoric styles and organ builders. The

dataset has been split by 90% and 10% for the training and validation sets respectively, for a total of 1998 samples for the former and 222 for the latter. The only pre-processing conducted is the normalization of the parameters and the extraction of the STFT. A trade-off has been selected in terms of resolution and hop size to allow a good tracking of harmonics peaks and attack transient. Figure 2 shows the input STFT for a A4 tone used for training the network.

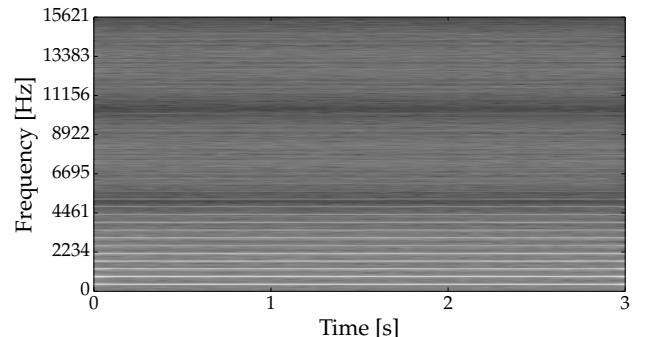


Figure 2: The STFT for an organ sound in the training set. The tone is a A4.

The CNN architecture is composed of up to 6 convolutional layers, with optional batch normalization [16] and max pooling, up to 3 fully connected layers and a final activation layer. For the training, stochastic gradient descent (SGD), Adam and Adamax optimizers have been tested. The training made use of early stopping on the validation set. A coarse random search has been performed first to pinpoint some of the best performing hyperparameters. A finer random search has been, later, conducted keeping the best hyperparameters from the previous search constant. Tests have been conducted with other stops not belonging to the training set and averaged for all the keys in the stops.

5. RESULTS

The loss used in training, validation and testing is the Mean Square Error (MSE) calculated at the output of the network with respect

¹<http://keras.io>

²<http://deeplearning.net/software/theano/>

to the target, before de-normalization. Results are, therefore, evaluated in terms of MSE.

Table 2 reports the 15 best hyperparameters combinations in the fine random search. The following activation function combinations are reported in Table 2:

1. A: employs *tanh* for all layers,
2. B: employs *tanh* for all layers besides the last one, that uses a Rectified Linear Unit (*ReLU*)[17],
3. C: employs *ReLU* functions for all layers,
4. all other combinations of the two aforementioned activation functions obtained higher MSE score and are not included here.

Results are provided against a test set of 222 samples from three organ stops, and have same learning rate ($1E-5$), momentum max (0.9), pool sizes (2x2 for each convolutional layer), receptive field sizes (3x3 for each convolutional layer) and optimizer (Adamax [18]). These fixed parameters have been selected as the best ones after the coarse random search.

Figure 4 shows the training and validation loss plots for the first 200 training epochs for the first combination in Table 2. The loss is based on the MSE for all parameters before denormalization. This means all parameters contribute to the MSE with the same weight and makes results clearer to evaluate. Indeed, if the MSE would be evaluated after de-normalization, parameters with larger excursion ranges would have a larger effect on the loss (e.g. a delay line length versus a digital filter pole coefficient). Validation Early Stopping is performed when the minimum validation loss is achieved to prevent overfitting and reduce training times. In Figure 4, e.g., the validation loss minimum (0.027) occurs at epoch 122, while the training loss minimum (0.001) occurs at epoch 198. Two spectra and their waveforms are shown in Figure 3 showing the similarity of the original tone and the estimated one, both obtained from the flue pipe model.

Results provided in terms of MSE, unfortunately, are not acoustically motivated: not all errors have the same effect, since parameters affect different perceptual features, thus large errors on some parameters may not result as easily perceived as small errors on other parameters. To the best of the authors’ knowledge there is no agreed method in literature to objectively evaluate the degree of similarity of two musical instruments spectra. Previous works suggested the use of subjective listening tests [19, 20, 21, 22], but an objective way to measure this distance is still to be addressed.

In order to provide the reader with some cues on how to evaluate these results, we draw from the psychoacoustic literature, as an example, the work from Caclin et al. [23], where spectral irregularity is proposed as a salient feature in sound similarity rating. Spectral irregularity is modelled, in their work, as the attenuation of even harmonics in dB (EHA). The perceptual extremes are chosen to be a tone with a regular spectral decay and a tone with all even harmonics attenuated by 8dB. The mean squared error calculated for these two tones (HMSE) for the first 20 harmonics (as done in their work) is 32dB. In our experiments, results vary greatly, depending on the pipe sounds to be modeled by the CNN. As an example, Figure 3 shows time and frequency plots of two experiments. They both present two A4 signals created by the physical model with two different parameter configurations hand-crafted by an expert musician, called respectively “Stentor” and “HW-DE”. The peak amplitude of the harmonics for the tones in

	HMSE10	HMSE20	HMSE35
Stentor	5.2 dB	9.4 dB	18.9 dB
HW-DE	0.3 dB	10.1 dB	12.3 dB

Table 1: Harmonics MSE (HMSE) for the first 10, 20 and all 35 harmonics for the tones shown in Figure 3.

Activations class	Minibatch size	Internal layers size	MSE
A	40	(16, 16, 512, 58)	0.261
B	50	(4, 6, 8, 10, 512, 58)	0.203
A	40	(16, 16, 32, 32, 512, 58)	0.164
A	40	(16, 16, 32, 32, 512, 58)	0.139
A	25	(16, 16, 32, 32, 1024, 58)	0.161
A	40	(16, 16, 32, 32, 1024, 58)	0.266
A	25	(16, 16, 32, 32, 512, 58)	0.156
A	40	(16, 16, 32, 32, 1024, 58)	0.252
A	50	(4, 6, 8, 10, 512, 58)	0.166
A	40	(16, 16, 32, 32, 256, 58)	0.179
A	25	(2, 2, 4, 4, 128, 58)	0.254
C	740	(16, 16, 32, 32, 512, 58)	0.252
B	50	(4, 6, 8, 10, 512, 58)	0.214
C	740	(16, 16, 32, 32, 512, 58)	0.257
B	40	(16, 16, 512, 58)	0.179

Table 2: The best 15 results of the fine random hyperparameters search. Activation classes are described in the text. The MSE is evaluated before denormalization, thus, all parameters have the same weight. Please note: all the layers are convolutional with kernel size as indicated, exception made for the second to last which is a fully connected layer. The output layer has fixed size equal to the number of model parameters.

Figure 3 are evaluated in terms of HMSE for the first 10, 20 and 35 harmonics in Table 1³.

The first tone, shown in Figure 3(a) has a spectrum with a good match for the first harmonics, but with some outliers and a generally bad match for harmonics higher than 12. The latter, shown in Figure 3(b) has a good match, especially in its first 10 partials, but the error raises with higher partials, especially from the 12th up. This is reflected by an HMSE10 of 5.2 dB vs. 0.3 dB and an error on the whole spectrum (HMSE35) of 18.9 dB vs. 12.3 dB. HMSE20 values for the two tones do not differ significantly, due to the averaging done on spectral ranges with different results, but we leave them to the reader so that they can be compared to the experiments of Caclin et al. The HMSE20 values are somewhere between the two extremes, “same” and “different”, tending more towards the former. Informal listening tests conducted with expert musicians suggest that the estimated “Stentor” tone does not match well to the original, while the “HW-DE” does match sufficiently. We hypothesize that the spectral matching of the first harmonics is more relevant in psychoacoustic terms to assess similarity, but we leave this to more systematic studies as a future work. The tones are made available to the reader online⁴.

³The sampling frequency of the tones is 31250 Hz, thus, 35 is the highest harmonic for a A4 tone.

⁴<http://a3lab.dii.univpm.it/research/10-projects/84-ml-phymod>

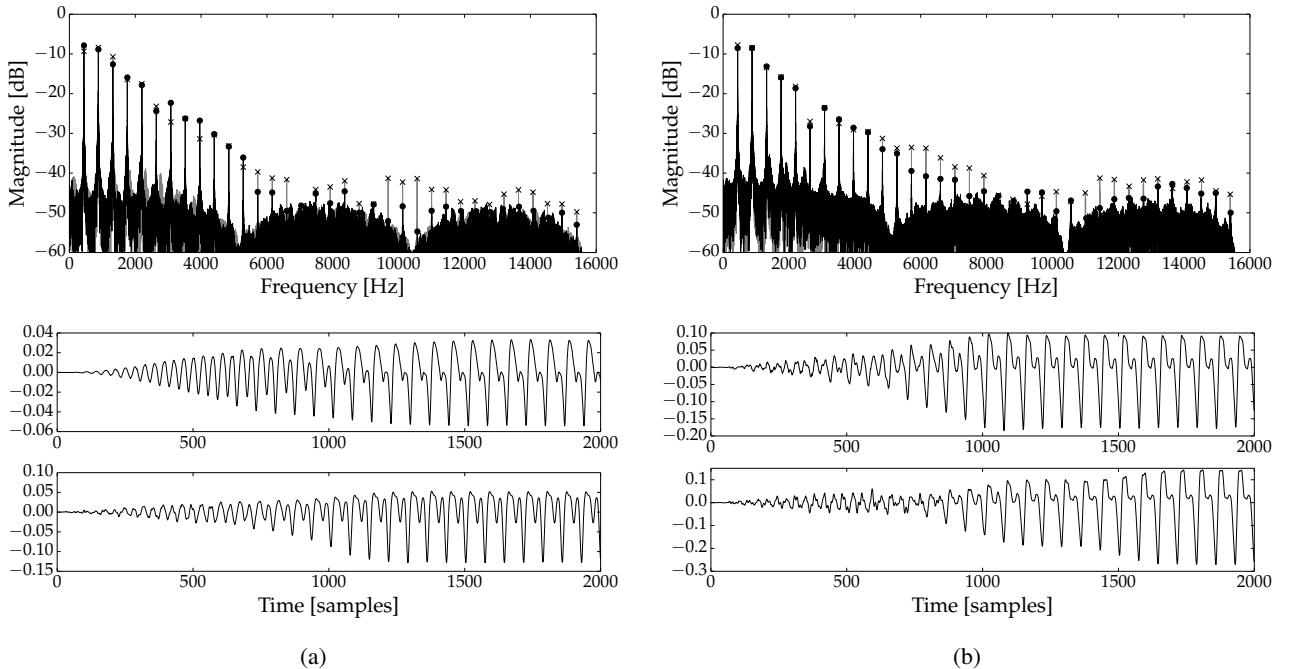


Figure 3: Spectra and harmonic content for two A4 tones from (a) *Principal* stop named “Stentor”, and (b) *Principal* stop named “HW-DE”. The gray lines and crosses show the spectrum and the harmonic peaks of $\hat{s}(n)$, while the black line and dots show the spectrum and the harmonic peaks of $s(n)$. In the waveform plots, the upper ones are obtained by the target parameters, while the lower ones are obtained with the estimated parameters.

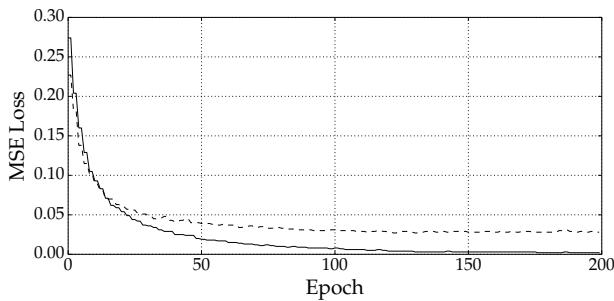


Figure 4: Training (solid line) and validation loss (dashed line) for the best combination reported in Table 2. Please note that the validation loss minimum (0.027) occurs at epoch 122, while the training loss minimum (0.001) occurs at epoch 198.

6. CONCLUSIONS

In this paper a machine learning paradigm that is general and flexible is proposed and applied to the problem of estimating the parameters for a physical model for sound synthesis. To validate the idea a specific use case of a flue pipe physical model has been employed. Results in term of MSE are good and tones spectra have a good match in terms of harmonic content, although results vary. Such results, coming from a real-world application scenario motivate the authors in believing that a machine learning paradigm can be employed with success for the problem at hand. Nonetheless, this first achievement calls for more research works. First of all a

validation is required with data sampled from a real pipe organ for further assessment and to evaluate the robustness of this method to noise, reverberation and such.

During the evaluation of the results, it has been discovered that results may greatly vary depending on the stops acoustic character. A rigorous approach to machine learning requires understanding whether the training set, which is a sampling of the probability distribution of all flue pipe stops obtained by the model, is representative of that probability distribution. Furthermore, it can be expected that the organ stops that can be obtained from the model are a subset of all organ stops that could physically built, due to model limitations and simplifying hypotheses. On the other hand, due to its digital implementation, the model could circumvent some physical limitation of real flue pipes, thus, yielding stops that are not physically feasible. This calls for a better understanding of how different stops are related to each others in the modelled and the physical realms, to understand before trying the machine learning approach, whether satisfying results can be obtained. As a last remark, these are general issues that apply also to other physical model or musical instruments.

7. REFERENCES

- [1] Vesa Välimäki, Jyri Pakarinen, Cumhur Erkut, and Matti Karjalainen, “Discrete-time modelling of musical instruments,” *Reports on progress in physics*, vol. 69, no. 1, pp. 1, 2005.
- [2] Julius O. Smith, “Virtual acoustic musical instruments: Re-

- view and update,” *Journal of New Music Research*, vol. 33, no. 3, pp. 283–304, 2004.
- [3] Stefano Zambon, Leonardo Gabrielli, and Balazs Bank, “Expressive physical modeling of keyboard instruments: From theory to implementation,” in *Audio Engineering Society Convention 134*. Audio Engineering Society, 2013.
- [4] Janne Riionheimo and Vesa Välimäki, “Parameter estimation of a plucked string synthesis model using a genetic algorithm with perceptual fitness calculation,” *EURASIP Journal on Advances in Signal Processing*, vol. 2003, no. 8, 2003.
- [5] Vasileios Chatzioannou and Maarten van Walstijn, “Estimation of clarinet reed parameters by inverse modelling,” *Acta Acustica united with Acustica*, vol. 98, no. 4, pp. 629–639, 2012.
- [6] Carlo Drioli and Davide Rocchesso, “Learning pseudo-physical models for sound synthesis and transformation,” in *Systems, Man, and Cybernetics, 1998. 1998 IEEE International Conference on*. IEEE, 1998, vol. 2, pp. 1085–1090.
- [7] Aurelio Uncini, “Sound synthesis by flexible activation function recurrent neural networks,” in *Italian Workshop on Neural Nets*. Springer, 2002, pp. 168–177.
- [8] Alvin WY Su and Liang San-Fu, “Synthesis of plucked-string tones by physical modeling with recurrent neural networks,” in *Multimedia Signal Processing, 1997., IEEE First Workshop on*. IEEE, 1997, pp. 71–76.
- [9] Alvin Wen-Yu Su and Sheng-Fu Liang, “A new automatic IIR analysis/synthesis technique for plucked-string instruments,” *IEEE transactions on speech and audio processing*, vol. 9, no. 7, pp. 747–754, 2001.
- [10] Ali Taylan Cemgil and Cumhur Erkut, “Calibration of physical models using artificial neural networks with application to plucked string instruments,” *PROCEEDINGS INSTITUTE OF ACOUSTICS*, vol. 19, pp. 213–218, 1997.
- [11] Katsutoshi Itoyama and Hiroshi G Okuno, “Parameter estimation of virtual musical instrument synthesizers,” in *Proc. of the International Computer Music Conference (ICMC)*, 2014.
- [12] Soroush Mehri, Kundan Kumar, Ishaaq Gulrajani, Rithesh Kumar, Shubham Jain, Jose Sotelo, Aaron Courville, and Yoshua Bengio, “SampleRNN: an unconditional end-to-end neural audio generation model,” in *5th International Conference on Learning Representations (ICLR 2017)*, 2017.
- [13] C. Zinato, “Method and electronic device used to synthesise the sound of church organ flue pipes by taking advantage of the physical modeling technique of acoustic instruments,” Oct. 28 2008, US Patent 7,442,869.
- [14] NH Fletcher, “Sound production by organ flue pipes,” *The Journal of the Acoustical Society of America*, vol. 60, no. 4, pp. 926–936, 1976.
- [15] Neville H Fletcher and Suszanne Thwaites, “The physics of organ pipes,” *Scientific American*, vol. 248, no. 1, pp. 94–103, 1983.
- [16] S. Ioffe and C. Szegedy, “Batch normalization: Accelerating deep network training by reducing internal covariate shift,” *arXiv preprint arXiv:1502.03167*, 2015.
- [17] 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.
- [18] Diederik Kingma and Jimmy Ba, “Adam: A method for stochastic optimization,” *arXiv preprint arXiv:1412.6980*, 2014.
- [19] Simon Wun and Andrew Horner, “Evaluation of weighted principal-component analysis matching for wavetable synthesis,” *J. Audio Engineering Society*, vol. 55, no. 9, pp. 762–774, 2007.
- [20] H. M. Lehtonen, H. Penttilä, J. Rauhala, and V. Välimäki, “Analysis and modeling of piano sustain-pedal effects.” *J. Acoustical Society of America*, vol. 122, pp. 1787–1797, 2007.
- [21] Brahim Hamadicharef and Emmanuel Ifeachor, “Objective prediction of sound synthesis quality,” *115th Convention of the AES, New York, USA*, p. 8, October 2003.
- [22] L. Gabrielli, S. Squartini, and V. Välimäki, “A subjective validation method for musical instrument emulation,” in *131st Audio Eng. Soc. Convention, New York*, 2011.
- [23] Anne Caclin, Stephen McAdams, Bennett K Smith, and Suzanne Winsberg, “Acoustic correlates of timbre space dimensions: A confirmatory study using synthetic tones a,” *The Journal of the Acoustical Society of America*, vol. 118, no. 1, pp. 471–482, 2005.

REAL-TIME PHYSICAL MODEL OF A WURLITZER AND RHODES ELECTRIC PIANO

Florian Pfeifle

Systematic Musicology,
University of Hamburg
Hamburg, DE

Florian.Pfeifle@uni-hamburg.de

ABSTRACT

Two well known examples of electro-acoustical keyboards played since the 60s to the present day are the Wurlitzer electric piano and the Rhodes piano. They are used in such diverse musical genres as Jazz, Funk, Fusion or Pop as well as in modern Electronic and Dance music. Due to the popularity of their unique sound and timbre, there exist various hardware and software emulations which are either based on a physical model or consist of a sample based method for sound generation. In this paper, a real-time physical model implementation of both instruments using field programmable gate array (FPGA) hardware is presented. The work presented herein is an extension of simplified models published before. Both implementations consist of a physical model of the main acoustic sound production parts as well as a model for the electromagnetic pickup system. Both models are compared to a series of measurements and show good accordance with their analog counterparts.

1. INTRODUCTION

Electromechanical and analog electric systems used for musical sound generation were building blocks of early types of electronic music instruments from the late 19th century well into the second half of the 20th century. Largely driven by the advances in science and engineering as well as the rising capabilities of music recording, transmission and reproduction systems, these instruments were crucial for the evolution of various modern music styles and were formative ingredients for multiple music genres. Two prominent electromechanical keyboard instruments from the 1960s and 1970s that are still being used in modern music productions are the Fender Rhodes and the Wurlitzer electronic piano. Their unique sound can be heard in many well known songs from genres such as Jazz, Funk, Pop, Rock and modern electronic music as well. Due to the fact that central parts of modern music production, recording and transmission gradually shifted from analog to fully digital processing chains from the late 20th century on, these two instruments are popular options for digital emulations in synthesizers, digital keyboards and hardware/software samplers of differing product generations and vendors. Notwithstanding the availability of a multitude of different emulations on various hardware and software platforms, there is an ongoing effort to optimize models of musical instrument towards physical plausibility and realistic sounding simulations of analog instruments in general as well as the Rhodes and Wurlitzer in special.

In this paper, a methodology and implementation of a Fender Rhodes and Wurlitzer e-piano's sound production system implemented on field programmable gate array (FPGA) hardware is presented. The implementation is based on a physical model published in [18], [17], [19] and uses a similar hardware implemen-

tation methodology as is published in [21]. This work aims at extending the existing physical models of mentioned publications in two regards by (1) implementing them on a FPGA for real-time synthesis and (2) making the physical model more accurate when compared to physical measurements as is discussed in more detail in section 4 and 5.

2. RELATED WORK

Scientific research regarding acoustic and electro-mechanic properties of both instruments is comparably sparse. Freely available user manuals as well as patents surrounding the tone production of the instruments give an overview of basic physical properties of both instrument [5]; [7]; [8]; [13]; [4]. The operation manual of the Rhodes contains several important aspects of its construction and explains influences of mechanical properties and the resulting effects on the sound of the instrument, see [11]. The Wurlitzer's manual gives a comprehensive overview and reference on the construction of the instruments and the resulting sound, see [25]. As is shown in [18], some of these publications fail to explain the influence of certain physical effects in the formation of the sound.

Regarding a scientific classification of the acoustic and electronic properties of the Rhodes and Wurlitzer piano there are several works that can be highlighted here. A thesis, available in German only, highlights acoustic features of the Rhodes by focussing on the vibration of the tine and the resulting sound [10]. Some non-linear properties and effects of the rhodes' tone production is published in [19] and [17]. A physical model of Rhodes' tone production using a Port-Hamiltonian approach is presented in [27]. Acoustic properties taken from high-speed camera measurements and finite difference models of both instruments are published in [18].

3. PHYSICAL PROPERTIES

This section gives an overview on on physical properties of the main sound producing parts of both instruments. The measurements presented here are based on work published in [18] as well as more recent measurements performed on the same instruments.

3.1. Tone production mechanism in the Fender Rhodes

The Fender Rhodes measured in this work is used as a foundation for the model presented in section 5. It is comparable to most Rhodes' electronic pianos from the late 60s to the early 80s.

The mechanical part consists of a rod made of spring steel shrunk into an aluminium block on one side, making the resulting system comparable to a cantilever beam. The length and circumference of the rod as well as the position of a small tuning

spring, adding mass, determines its fundamental frequency. The rod, which in the case of the Rhodes piano is called a tine, is excited by hammer having a neoprene tip. The key action mechanism is a simplified single action as described in [6], it can be compared to a *Viennese* or *German* piano action because the hammer is in direct contact with the key. Depending on the year of construction the key and hammer mechanisms are crafted from wood or, as is generally used in newer models, of synthetic materials. Every tine is damped by an individual felt damper that is in contact with the tines from below. The fixation of the tine, the aluminium block, is tightly connected to a, sometimes $\frac{\pi}{2}$ twisted, brass bar which acts as the second prong of the patented Rhodes' "tuning fork" system.

The harmonic oscillations of the mechanic part of the Rhodes' tone production is converted to an alternating voltage by an electromagnetic pickup that consists of a wound permanent magnet. This setup is comparable to a pickup of a guitar in its overall structure but differs in terms of the geometry of the magnets tip as is depicted in Figure 1.

The geometry of the pickup's iron tip shapes the specific distribution of the magnetic field in which the tine vibrates. The motion of the ferromagnetic tine changes the flux of the magnetic field which in turn produces a change in the electromotive force of the pickup. This results in an alternating voltage which then can be amplified by an external amplifier. The copper wire winding of each pick up is divided into two sections, connected in opposite phase for hum cancelling.

The timbre of a Rhodes note can be altered by changing the position of the tine in respect to the magnet as schematically depicted in Figure 7a and Figure 7b.

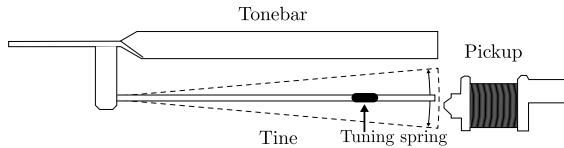


Figure 1: The Rhodes Tuning Fork assembly with electromagnetic pickup.

3.2. Tone production mechanism in the Wurlitzer

Compared to the Rhodes piano, the tone production mechanism in the Wurlitzer piano is based on a different physical principle . In contrast to the Rhodes' picks-up which reacts to changes in the magnetic field (H-field), the Wurlitzer's pickup system is designed to detect changes in the E-field distribution over the sound generators geometry. In special, it is designed as a time-varying capacitor which consists of a loaded static plate and a moving plate, called a reed [25], having zero potential (connected to ground). Mechanically the reed is fixed at one side and free on all others. According to the manual, the high potential has a voltage 170V which has been found to be considerably lower (130V) in the measured instrument used in this work. Effectively, the time varying capacitance induces a time-varying current which in turn induces a time-varying voltage which is being amplified by the subsequent amplification circuit.

There are two factors determining the fundamental frequency f_0 of every reed, the physical dimensions of the reed itself and the amount of solder on the tip of the reed. By removing or adding lead to the tip of the reed its f_0 is increased or lowered respectively.

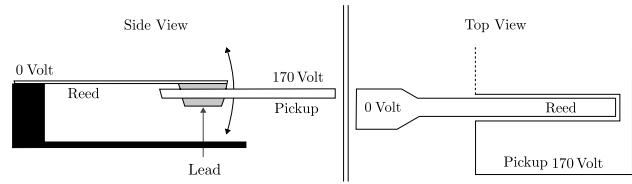


Figure 2: Structural depiction of the Wurlitzers pickup system. A side view on the left, top view on the right. Both showing the high potential plate and the low potential reed.

As is depicted in Figure 2, the charged plate has cutouts at the position of the reed for each note of the instrument. The reeds are designed to vibrate freely between the symmetric cutouts, providing a surface area large enough to produce a measurable change in capacity. The air gaps between plate and reed act as dielectric material. Analogous to the case of a plate capacitor or the diaphragm of a condenser microphone, the capacity varies inversely proportional to the distance between the two electrodes, in this case: reed and fixed plate.

The key action mechanism of the Wurlitzer piano consists of a miniaturized London style piano action that can be regulated like a grand piano action. Every reed of the Wurlitzer electric piano is excited by an individual ply maple hammer that has a felt tip [25]. Comparable to the playing dynamics of the Rhodes piano, depressing the keys with higher velocity results in a richer harmonic sound of the Wurlitzer than playing softly.

4. MEASUREMENTS

4.1. Measurement tools

All measurements of visibly moving parts of both instrument's primary sound production parts are performed using a *Vision Research* V711 high-speed camera. The recorded motion is tracked with sub-pixel accuracy using *Innovision System's MaxTraq2D* software. The traced trajectories are exported as time series which are post-processed and evaluated using the high-level language *julia*. Electronic properties are measured using a measurement amplifier and converter *LTT24* by *Tasler*. Analog sound outputs of both instruments are recorded using *Logic Pro X* software and a *Focusrite* interface running at 44,1kHz and 24 bit.

4.2. Rhodes measurements

As depicted in Figure 3, the tip of the tine vibrates in an approximately sinusoidal motion, the direct-out signal measured behind the pick-up has a more complex waveform, showing the influence of the magnetic field of the pick-up.

As an extension to the measurements in one plane, the same rhodes tine is measured with the camera in two horizontal dimensions. Figure 4 shows non-planar motion in the transversal direction. The tine's vibration in the vertical plane u_{ver} , which is the direction of the hammer impact, is larger when compared to that in the horizontal plane u_{hor} . The horizontal motion is excited either through coupling effects on the tine or due to imperfections in the hammer tip.

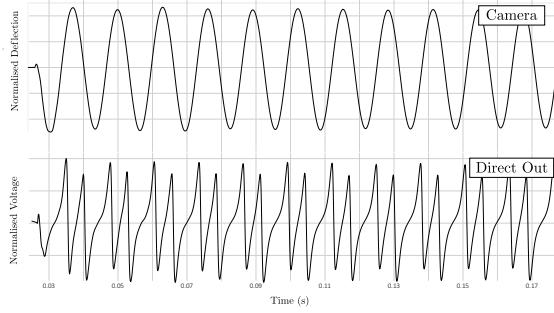


Figure 3: Measurement of one Rhodes tine. The upper row shows the deflection of the tine the lower the measured direct-out signal behind the pick-up.

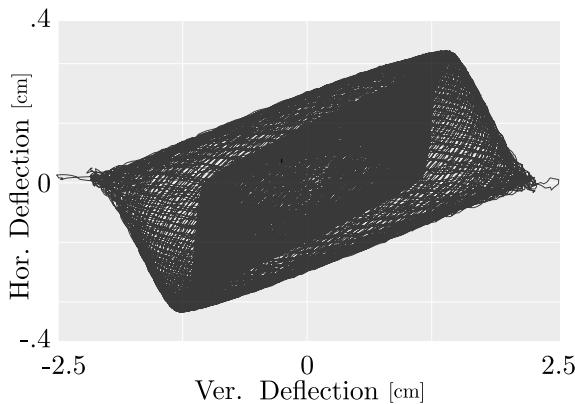


Figure 4: Phase plot of the two polarisations of the tine deflection of one tine.

4.3. Wurlitzer measurements

Figure 5 shows the first 12 milliseconds of the Wurlitzer's reed motion and the measured sound at the output.

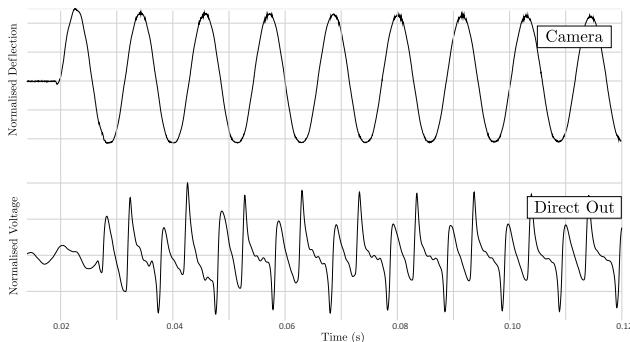


Figure 5: The upper graph shows the tracked camera signal having approximately sinusoidal motion. The lower graph shows the voltage measured behind the pick-up system before the amplification circuitry.

5. PHYSICAL MODEL

5.1. Overview

The physical models presented in this section are based on the measured properties presented in section 4, qualitative observations of the FEM models published in [18] and some assumptions regarding material properties of the Wurlitzer's reed and the hammer tip of both instruments. The model of the Rhodes e-piano includes a formulation for the hammer impact, a model of a beam vibrating in two polarisations subject to large effects and a pre-processed approximation of the magnetic field distribution over the tip of the pick-up's magnet core. The model of the Wurlitzer EP200 shares conceptual similarities with the Rhodes model but is adapted to the different geometry of the sound production. The tine of the Wurlitzer is modeled as a non-uniform cantilever beam with large-deflection effects and a spatial transfer function describing the change in capacitance resulting from the motion of the tine.

5.2. Hammer model

A hammer impact including viscoelastic material properties of the hammer tip can be simulated by using a hysteretic hammer model as presented in [22] and [14]. This impact model is able to simulate hammer impacts of different materials showing viscoelastic behaviour. Based on the formulation in [14], a distributed force exerted by a hammer impact follows the relationship

$$F([\mathbf{x}], t) = \begin{cases} k \cdot \mathbf{x}(t)^\alpha + \lambda \cdot \mathbf{x}(t)^\alpha \cdot x_t(t) & \text{if } \sum_{xL} \mathbf{x} > 0 \\ 0 & \text{for } \sum_{xL} \mathbf{x} \leq 0 \end{cases} \quad (1)$$

with \sum_{xL} indicating a weighted sum over the contact area. This model is based on a model for hammer impacts developed by Hunt & Crossly [33], that has been shown to yield good results for models of hammer impacts with moderate impact velocities and plain geometries [14]; [29]. Here, α is the nonlinearity exponent depending on the geometry of the contact area and λ is a material dependent damping term that dissipates energy in dependence to the velocity of the changing hammer-tip compression written as x_t .

5.3. Rhodes tine model

In an earlier work ([18]), the tine of the Rhodes was modeled as simple harmonic oscillator approximating its fundamental motion. Here, the tine is modeled as a cantilever beam with non-planar motion, large-deflection effects, and inclusion of the tuning spring as an additional mass.

Large deflection effects are included by taking shearing effects in the beam into account. Trail & Nash [26] showed that the shear beam is a better approximation for the vibrations of the fundamental frequency than the more popular Euler-Bernoulli beam and less computationally complex as the similar accurate Timoshenko beam model. Coupling between two polarisations for large deflections of the beam can be included as proposed in [?]. Compared to its diameter, the deflection of the Rhodes' tine is large. Thus it is feasible to include high deflection effects into the formulation of the model. As shown in [28] the inclusion of shear effects to the Euler-Bernoulli beam raises the accuracy of the fundamental frequency as well as the accuracy of higher partials. Following the consideration in [28], the differential equation for a round beam

exhibiting large and non-planar deflections, having non homogeneous mass, and not considering the angle of rotation can be written as

$$\begin{aligned} \rho \mathbf{u}_{tt} + & [EI\mathbf{u}_{xx}]_{xx} - EA \frac{1}{2} \mathbf{u}_{xx} \cdot K(\mathbf{u}) \\ - & \kappa \mathbf{u}_{2x2t} - F(\mathbf{u}^V[x], t) = 0 \end{aligned} \quad (2)$$

Here, \mathbf{u} is the deflection in two transverse polarisations (H, V), E is the Young's modulus I is the radius of gyration and A is the cross-sectional area. $F(\mathbf{u}^V[x], t)$ is the forcing function of the hammer model, impacting the bar in the vertical direction indicated by u^V . $K(u, w)$ is a nonlinear Kirchhoff-like term given as

$$K(\mathbf{u}) = \int_0^l (\mathbf{u}_x^H)^2 + (\mathbf{u}_x^V)^2 dx \quad (3)$$

5.4. Rhodes pickup model

As depicted in Figure 1 the tip of the tine vibrates in close proximity to the electromagnetic pickup and the FEM simulations given in Figure 8 of [18] highlight that only a small part of the tip is influenced by the magnetic field. For a derivation of the magnetic field distribution in two dimensions, the tip is approximated as a finite point oscillating over an idealised geometry of the magnetic pick-up tip.

The electromagnetic effects of the Rhodes' pickup system can be reduced from Maxwell's equations for transient electromagnetic effects to a more tractable formulation known as Faraday's law of induction. As shown above, the pickup consists of a magnetized steel tip and a coil wrapped permanent magnet; leaving reciprocal magnetic effects of the induced current in the coil out of our consideration, the voltage induced over the pickup is equivalent to the change of the magnetic flux in the field produced by the magnet

$$\epsilon = - \frac{\partial \Psi_B}{\partial t} \quad (4)$$

with ϵ the electromotive force and Ψ_B the magnetic flux due to the change in the magnetic field given by

$$\Psi_B = \int \vec{B} \cdot d\vec{S} \quad (5)$$

with B the magnetic field strength integrated over surface S . Using these equalities, the induced voltage directly depends on the change of magnetic field strength which depends solely on the position of the tine disturbing the field as shown in Figure 7.

The following derivation of the magnetic field distribution uses the unphysical assumption that there exist magnetic monopoles which produce a distributed magnetic field.¹ As is shown in [16] this approach yields good approximations of notional magnetic induction fields produced by guitar pickups (see also [15]). Consisting of a plainer geometry, the tip of a guitar pickup bar magnet can be simplified to a circular, magnetically charged disc with a certain cross-section, which reduces the problem to a position-dependent integration of the field over the pickup. Due to the specific pickup geometry of the Rhodes, a different approach is taken here to calculate the induction field strength above the tip of the magnet. As

¹This assumption proposes an equivalence between the effective causes of electric fields and magnetic fields and can be used as a mathematical modeling tool, see: [31, pp. 174 ff].

depicted in Figure 6 our derivation makes use of several simplifying assumptions facilitating the computation of the magnetic field distribution over the magnet's tip that are

1. The tine vibrates in a sinusoidal motion in both planes in front of the pickup.
2. The tip of the tine vibrates on the trajectory of an ideal circle with the center at its fixation point.

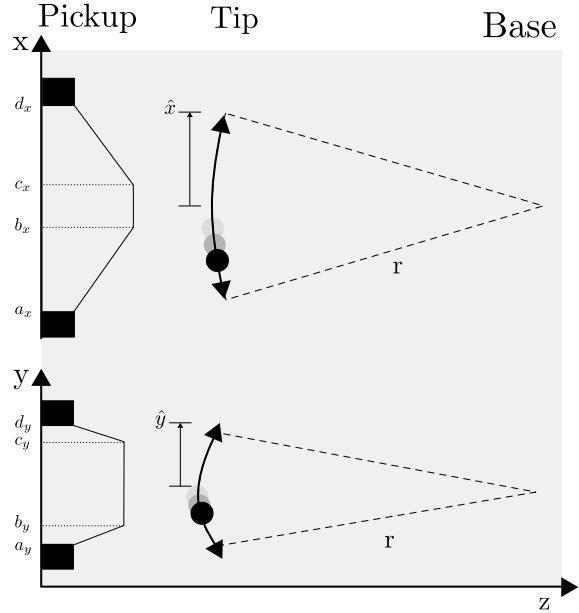


Figure 6: Simplified geometry of the pickup system and the vibrating tine in the x- and y-plane orthogonal to the magnetic pick-up.

Defining an imaginary magnetic point charge which induces a change in the magnetic flux in the direction of z

$$\mathbf{B}_z = B_0 \frac{\Delta z}{|r_{21}|^3} \quad (6)$$

The magnetic field for position (x', z') in front of the steel tip can thus be written as a three-part integral

$$\begin{aligned} \mathbf{B}_z(x', z') = & |\mathbf{B}_{tine}| \\ & \cdot \left[\int_a^b \frac{\sigma(z' - z(x))x}{[(x' - x)^2 + (z' - z(x))^2]^{3/2}} dx \right. \\ & + \int_b^c \frac{\sigma(z' - z_k)x}{[(x' - x)^2 + (z' - z_k)^2]^{3/2}} dx \\ & \left. + \int_c^d \frac{\sigma(z' - z(x))x}{[(x' - x)^2 + (z' - z(x))^2]^{3/2}} dx \right] \end{aligned} \quad (7)$$

with σ the constant magnetic charge density the magnetic field distribution for position (y', z') can be computed accordingly. Integrating this formula for all points on a trajectory given by the position of the Rhodes' tine tip

$$\begin{aligned} z' &= r - \sqrt{r^2 - (x')^2} \\ x' &= \hat{x} \cdot \sin(2\pi f_{tine} t) \end{aligned} \quad (8)$$

with f_{tine} the fundamental frequency of the tine, leads to a magnetic potential function characterising the magnitude of relative magnetic field change.

An idealised form of the magnetic field in front in one plane of the Rhodes pickup is depicted in Figure 7a and 7b, it is comparable to the measurements results published in [16].

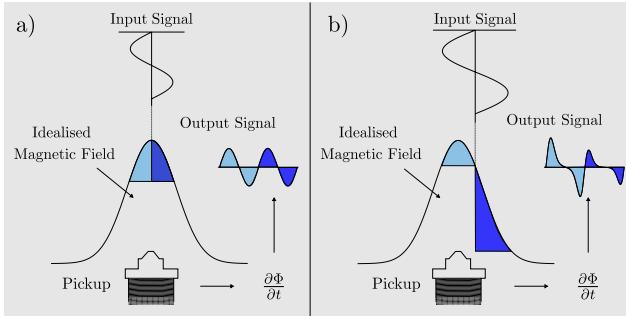


Figure 7: An idealised schematic depiction of the pickup system of the Rhodes E-piano. The sinusoidal motion of the vibrating tine induces a c a) A low amplitude input of a sinusoidal vibration of the magnetic flux weighted by the magnet fields distribution. By differentiating the magnetic flux in respect to time, the alternating voltage present at the output is calculated. b) A similar model setup as before consisting of a slightly displaced mid-point for the input motion resulting in a different weighting function of the magnetic field. The output shows a different form than before. This condition is close to a realistic playing condition found in Rhodes E-pianos.

5.5. Wurlitzer reed model

The reed of the Wurlitzer is modeled using a similar PDE as for the Rhodes' tine using only one direction of motion u , the coupling term between the two polarisations is omitted. The use of a beam model instead of a plate model is justifiable here because typical effects found in plates were not measured using the high-speed camera setup and thus are either not present or small compared to the transversal deflection of the fundamental mode. In addition to that, the measurements show that the influence of higher modes are comparably small or non-existent even under extreme playing conditions, thus the primary mode of vibration can be approximated by the reeds first natural frequency which coincides with a cantilever beam of similar dimensions.

$$\rho u_{tt} + [EIu_{xx}]_{xx} - \kappa u_{2x2t} - f(x, t) = 0$$

with the same variables as introduced before. Again Equation 9 does not explicitly depend on the shear angle α (see [28]) thus it is not regarded here any further. Again omitting the shear angle, the boundary conditions for the fixed/free beam are

$$\begin{aligned} u|_0 &= 0 \\ k' G A u_x|_L &= 0. \end{aligned} \quad (9)$$

5.6. Wurlitzer pickup model

The influence of the Wurlitzer's pick-up system can be characterised as a change in capacitance of a time varying capacitor induces an alternating voltage which is amplified as the instruments

sound. Using a basic definition of time-varying capacitance that induces a current i we get

$$i(t) = C(t) \frac{\partial u(t)}{\partial t} + u(t) \frac{\partial C(t)}{\partial t} \quad (10)$$

with u the voltage and C the capacitance both depending on time t . For the derivation of the influence function of the capacitor we take two simplifying assumptions.

1. The time dependent charging / discharging curve of the capacitor is linear in the considered range.
2. The supply voltage stays constant during a capacity change cycle of the capacitor.

Using both definitions, we can write the time-dependent current resulting from a changing capacitance as

$$i(t) = u_0 \frac{\partial C(t)}{\partial t} \quad (11)$$

This alternating current induces an alternating voltage over a resistor that drives an amplification circuit.

To calculate the capacitance curve due to the deflection of the Wurlitzer's reed, a number of i planes through the geometry are taken and the electric field strength is computed for each resulting slice simplifying the 3-dimensional problem to a 2-dimensional. The capacitance for each slice can be computed from the electric field by

$$C_i = \frac{Q_i}{U_i} \quad (12)$$

with $Q_i = \epsilon_t \oint_A \vec{E} \cdot d\vec{A}$ the charge defined as the integral of the electric field over the surfaces of the geometries using Gauss's theorem and ϵ_t an electric field constant for the material and free field properties. Three exemplary positions for the computation of the capacitance are depicted in Figure 8.

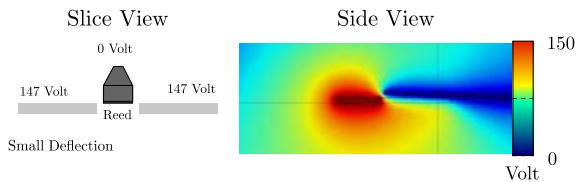


Figure 8: Distribution of the electric field for one exemplary reed deflections. On the left hand side one slice of geometry on the right hand side the results from an FEM model.

6. HARDWARE MODEL

6.1. Overview

The finite difference implementations of the physical models described in section 5 are implemented on an FPGA development board consisting of a XILINX Virtex-7 690T. This section gives a short overview on the methodology and the implementation of the real-time hardware models. A more detailed account is published in [21] or [34]

6.2. FPGA hardware overview

The real-time implementations presented in this work, make use of the parallel structure of an FPGA using a concurrent iterative approach for basic computations of all numerical schemes.

The parallel processing capabilities of FPGAs are mainly due to their inherently parallel hardware structure. The core parts of FPGAs consist of freely programmable, interconnected logic cells that can perform several Boolean operations (AND, OR, XOR...) depending on input and configurations signals. Most modern FPGAs employ Look-Up-Tables (LUTs) which act as addressable function generators having a number of logic inputs and logic outputs performing a specified Boolean operation [?]. Interconnecting these basic logic blocks, more complex logic functions can be realized ranging from basic arithmetic operations to specialised processing units.

6.3. FD Models on FPGA Hardware

To take benefit from the inherently parallel structure of FPGA hardware central parts of the FD models introduced in this work are processed in parallel. The numerical schemes developed in (5) can be split into sequential and parallel parts. The sequential computation is necessary for the interdependent computations of the velocities and the deflection of one discretised FD node on a geometry. When processed with synchronous timing, all FD nodes can be computed concurrently leading to an numerical scheme which can be efficiently computed on FPGA hardware.

The real-time implementations of the proposed FD models make use of a structured design approach consisting of a layer model developed to combine functional parts of FD models and assort them according to their respective functionality. All partitions of the FPGA implementations are categorised into five different sub-layers, each encapsulating specific functionality, specific data types and a specific communication protocol [21]. This layer model enables the reuse of structurally similar parts of different models minimizing implementation efforts for new FD designs.

Both instrument models are implemented by transferring the FD models developed before to a hardware level. This can be realised by rewriting the finite difference schemes using a hardware operator notation explained below. All basic numerical computations are implemented on the Algorithmic Layer, the structure of the modeled geometry is initialised and parametrised on the Model Routing Layer. The functionality of the other layers include signal routing, timing and synchronisation of the computation and simulated signals.

A structural flow diagram given in Figure 9 shows that both models share similarities regarding their processing steps.

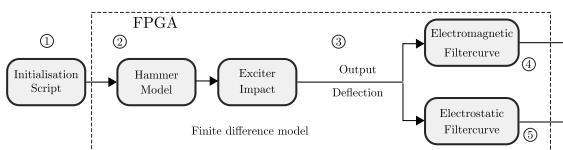


Figure 9: Schematic depiction of the processing chain of the model. ① The respective model is initialised regarding its physical properties and boundary condition. ② Computation of the finite difference models on FPGA hardware. ③ Output of the respective exciter model. ④ Rhodes model output. ⑤ Wurlitzer model output.

6.4. Discrete FD Operators

As extension to the well-established FD operator notation (see [32]; [30] or [29] a discrete operator formalism is used in this work. The operator notation allows to transfer several mathematical operation into a simpler notation. In the following, this concept is extended to an even lower abstraction level by resolving the underlying mathematical operations to the specific operations depending on the data type and underlying hardware structure. Assuming a signed two's complement data type, a centered finite difference operator can be expressed with following statement

$$\hat{\delta}_x = T_\Delta \cdot [\hat{e}_{\Delta x+}, -\hat{e}_{\Delta x-}] \quad (13)$$

with $\hat{e}_{\Delta x+/ \Delta x-}$ = a read operation from a register a finite difference cell right (+) or left (-) of the actual cell and T_Δ a multiplicand which depends on the stride of the discrete grid in the spatial domain. A second order centered FD operator in vector notation can be written as

$$\hat{\delta}_{xx} = T_\Delta \cdot [(\hat{e}_{\Delta x-}), (< 1), (\hat{e}_{\Delta x+})] \quad (14)$$

with $T_\Delta = \frac{1}{\Delta x}$ and (< 1) indicating a shift operation. This shift operation can be used to replace a multiplication by 2 in fixed-point arithmetic. A higher order digital FD-operator used for the fourth order differential equation of the beam can be constructed by a convolution of two second order digital FD operators

$$\hat{\delta}_{4x} = \hat{\delta}_{xx} * \hat{\delta}_{xx}. \quad (15)$$

This can be extended to higher spatial order difference operators leading to a specific numbers of digital operations for the respective operator given in Table 1. These basic operators can be em

??

Table 1: Digital operations for FD operators used in this work

Operator	Reg. Op.	Shift Op.	Mult.	Add/Sub.
$\hat{\delta}_x$	2	0	1	2
$\hat{\delta}_{xx}$	3	1	1	2
$\hat{\delta}_{4x}$	5	4	1	5
$\hat{\delta}_{2x2y}$	5	1	1	4

ployed for temporal as well as spatial discretisation and can be extended by including variable as well as static multiplicands into the formulation. For reasons of brevity, all material parameters of the following models are included in the operator notation as multiplicands and are preprocessed during model initialisation. Hence, a second order operator still needs one multiplication including material and geometry dependant weighting.

6.5. Finite difference hardware models

The exciter models of the Rhodes and the Wurlitzer pianos are discretised applying standard finite difference approximations using a symplectic Euler scheme for iteration in time. The discretisation method and the scheme are published in more detail in [21] and [20]. Applying the presented hardware FD approximations from Table 1 and splitting the PDE by introducing $v_1 = u_{-t}$, the transverse part of Equation 2 can be written as

$$\begin{aligned} \hat{\delta}_t v &= [-\hat{\delta}_{xx} + \hat{\delta}_{tt}] \hat{\delta}_{xx}^M \mathbf{u} + K(\mathbf{u}, \mathbf{v}, t) + F([\mathbf{x}], t) \\ \hat{\delta}_t u &= v \end{aligned} \quad (16)$$

7. SIMULATION RESULTS

7.1. Interaction with the model

Both real-time models can be controlled and parametrized using a script file. Besides controlling physical properties of the tone generators, initial values of the hammer can be set. A keyboard interface implementing the OSC protocol to interact with the models is a work in progress.

7.2. Simulation results Rhodes model

The following examples show simulation results of the real-time model parametrized with different initial conditions.

Increasing hammer velocities

Figure 10 shows five simulated Rhodes notes with varying start velocities the hammer model. This simulation shows that increasing the impact velocity of the hammer, increases the complexity of the Rhodes' due to the effects shown in Figure 7.

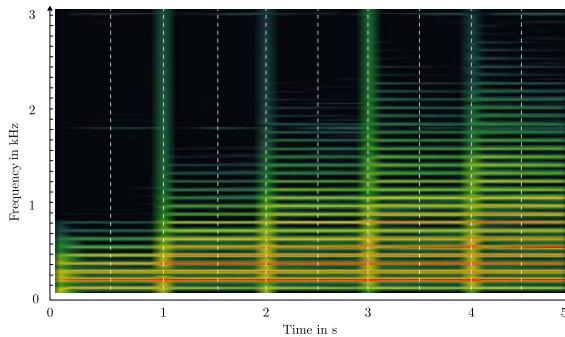


Figure 10: Spectrogram of a Rhodes model impacted with increasing hammer velocity.

2 dimensional polarisation

One of the effects that can be modeled using the extended formulation of the Rhodes' tine presented in this paper is the possibility to simulate two-dimensional vibrations of the impacted Rhodes tine

Figure 11 depicts the simulated deflection of both horizontal polarizations of the tine. When compared to the measurements given in Figure 4 it is obvious that the simulated coupling mechanism is capable of representing the measured behaviour of the tine.

Tuning of a Wurlitzer reed

In this simulation, the amount of solder on the Wurlitzers reed is changed by altering the mass distribution at the tip of the reed (see Figure 2 for a schematic depiction of the reed). As shown in Figure 12, changing the amount of solder leads to a detuning of the fundamental vibration frequency.

8. CONCLUSIONS

In this paper a real-time implementation of main sound producing parts of the Rhodes and Wurlitzer electronic pianos was presented.

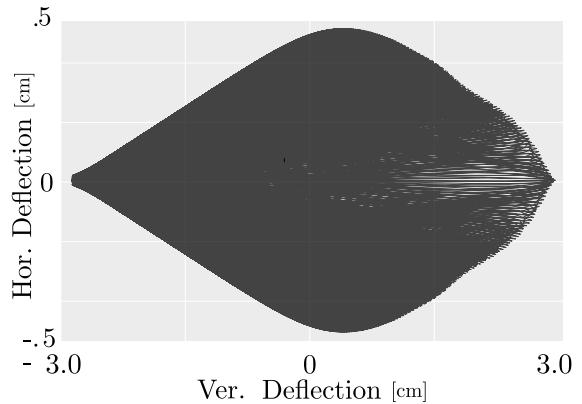


Figure 11: Horizontal vs. vertical deflection of an hammer impacted Rhodes tine.

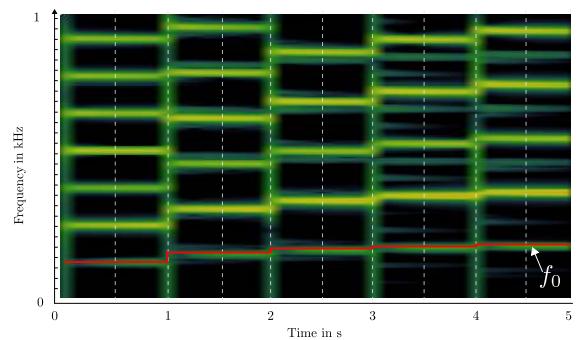


Figure 12: Spectrogram of five synthesized Wurlitzer notes. Every second, the mass on the tip is increased, thereby simulating the tuning procedure as described in section 3.

Both instruments are based on physical models and are computed on a an FPGA board which is connected to a standard personal computer and can be played and parametrized in real-time.

The presented models are able to capture salient features of the instruments and make it possible to interact with physical properties and parameters. Regarding the expressive range, both modeled instruments capture essential parts of the sound characteristics and can help in understanding specific features of both instruments in more detail.

Nonetheless, there are several parts in the model formulation which work with simplifications especially the physical properties of the respective pick-up systems. A further step to enhance this work would be an inclusion of a geometrically accurate model of the Rhodes electromagnetic pick-up and the Wurlitzer's electrostatic pick-up system.

9. ACKNOWLEDGMENTS

I would like to thank the anonymous reviewers. This work was partially funded by the *Deutsche Forschungsgemeinschaft DFG* hence it gives me great pleasure to acknowledge their support.

10. REFERENCES

- [1] A. Kovetz, "The Principles of Electromagnetic Theory", Cambridge University Press, 1990.
- [2] Jianming Jin, "The Finite Element Method in Electromagnetics", 2nd ed., Wiley-IEEE Press, May 2002.
- [3] D.K. Cheng, "Field and Wave Electromagnetics, 2nd ed.", Addison-Wesley, 1991.
- [4] Anthony C. Ippolito, "Electronic piano feedback reduction", US 3435122 A, 1965
- [5] Harold B. Rhodes, "Electrical musical instrument in the nature of a piano", U.S. Patent 2,972,922, 1961.
- [6] Harold B. Rhodes, "Piano Action", U.S. Patent 4,338,848, 1982.
- [7] Harold B. Rhodes and James B. Murphy, "Multiple Voice Electric Piano And Method", U.S. Patent 4,342,246, 1982.
- [8] Harold B. Rhodes and Steven J. Woodyard, "Tuning Fork Mounting Assembly In Electromechanical Pianos", U.S. Patent 4,373,418, 1983.
- [9] Greg Shear and Matthew Wright, "The electromagnetically sustained Rhodes piano", NIME Proceedings, Oslo, 2011.
- [10] Torsten Wendland, "Klang und Akustik des Fender Rhodes E-Pianos", Technische Universität Berlin, Berlin, 2009.
- [11] Rhodes Keyboard Instruments, "Service Manual", CBS Musical Instruments a Division of CBS Inc., Fullerton CA, 1979.
- [12] Benjamin F. Miessner, "Method and apparatus for the production of music", US Patent 1,929,027, 1931.
- [13] C.W. Andersen, "Electronic Piano", US 2974555, 1955.
- [14] Federico Avanzini and Davide Rocchesso. Physical modeling of impacts: theory and experiments on contact time and spectral centroid. In *Proceedings of the Conference on Sound and Music Computing*, pages 287–293, 2004.
- [15] Tatsuya Furukawa, Hideaki Tanaka, Hideaki Itoh, Hisao Fukumoto, and Masashi Ohchi. Dynamic electromagnetic Analysis of Guitar Pickup aided by COMSOL Multiphysics. In *Proceedings of the COMSOL Conference Tokyo 2012*, Tokyo, 2012. COMSOL.
- [16] Nicholas G. Horton and Thomas R. Moore. Modeling the magnetic pickup of an electric guitar. *American Journal of Physics*, 77(2):144, 2009.
- [17] Malte Muenster and Florian Pfeifle. Non-Linear Behaviour in Sound Production of the Rhodes Piano. In *Proceedings of the International Symposium of Musical Acoustics (ISMA) 2014*, pages 247–252, Le Mans, France., 2014.
- [18] Florian Pfeifle and Malte Muenster. Tone Production of the Wurlitzer and Rhodes E-Pianos. In Albrecht Schneider, editor, *Studies in Musical Acoustics and Psychoacoustics* volume 5 of *Current Research in Systematic Musicology*, pages 75–107. Springer International Publishing, 2017.
- [19] Malte Muenster, Florian Pfeifle, Till Weinrich, and Martin Keil. Nonlinearities and self-organization in the sound production of the Rhodes piano. *The Journal of the Acoustical Society of America*, 136(4):2164–2164, 2014.
- [20] Florian Pfeifle. Multisymplectic Pseudo-Spectral Finite Difference Methods for Physical Models of Musical Instruments. In Rolf Bader, editor, *Sound - Perception - Performance*, volume 1 of *Current Research in Systematic Musicology*, pages 351–365. Springer International Publishing, 2013.
- [21] Florian Pfeifle and Rolf Bader. Real-Time Finite-Difference Method Physical Modeling of Musical Instruments Using Field-Programmable Gate Array Hardware. *J. Audio Eng. Soc*, 63(12):1001–1016, 2016.
- [22] Anatoli Stulov. Hysteretic model of the grand piano hammer felt. *The Journal of the Acoustical Society of America*, 97(4):2577–2585, 1995.
- [23] Roald K. Wangness. *Electromagnetic fields*. Wiley, New York, 2nd ed edition, 1986.
- [24] Steve Espinola. Wurlitzer Electric Piano models: a list. | Paleophone. Blog entry Available at: http://paleophone.net/?page_id=923. (Accessed: 23rd May 2016)
- [25] Wurlitzer Company, The Electric Pianos Series 200 and 200A Service Manual Available at: <http://manuals.fdiskc.com/flat/Wurlitzer%20Series%20200%20Service%20Manual.pdf> (Accessed: 17.02.2016)
- [26] R. W. Traill-Nash and A. R. Collar. The effects of shear flexibility and rotatory inertia on the bending vibrations of beams. *The Quarterly Journal of Mechanics and Applied Mathematics*, 6(2):186–222, 1953.
- [27] Antoine Falaize and Thomas HÃ©lie. Passive simulation of the nonlinear port-Hamiltonian modeling of a Rhodes Piano. *Journal of Sound and Vibration*, 390:289–309, March 2017.
- [28] Seon M. Han, Haym Benaroya, and Timothy Wei. Dynamics of transversely vibrating beams using four engineering theories. *Journal of Sound and Vibration*, 225(5):935–988, September 1999.
- [29] Stefan D. Bilbao. *Numerical sound synthesis: finite difference schemes and simulation in musical acoustics*. Wiley, Chichester, 2009.
- [30] Charles Jordan. *Calculus of finite differences*. Chelsea Publ. Co, New York, 1 edition, 1950.
- [31] John D. Jackson. *Classical Electrodynamics* John Wiley & sons, Inc., 3rd edition, 1998
- [32] John C. Strikwerda. *Finite Difference Schemes and Partial Differential Equations, Second Edition*. Society for Industrial and Applied Mathematics, Philadelphia, USA, 2nd ed edition, 2004.
- [33] K. H. Hunt and F. R. E. Crossley. Coefficient of restitution interpreted as damping in vibroimpact. *Journal of applied mechanics*, 42(2):440–445, 1975.
- [34] F. Pfeifle *Physical Model Real-time Auralisation of Musical Instruments: Analysis and Synthesis* Ph.D. Dissertation. University of Hamburg. 2014.

A MECHANICAL MAPPING MODEL FOR REAL-TIME CONTROL OF A COMPLEX PHYSICAL MODELLING SYNTHESIS ENGINE WITH A SIMPLE GESTURE

Fiona Keenan

Department of Theatre, Film, Television and Interactive Media
University of York
Baird Lane, Heslington East
York, UK
fiona.keenan@york.ac.uk

Sandra Pauletto

Department of Theatre, Film, Television and Interactive Media
University of York
Baird Lane, Heslington East
York, UK
sandra.pauletto@york.ac.uk

ABSTRACT

This paper describes the design and control of a digital synthesis engine developed to imitate the sound of an acoustic wind machine, a historical theatre sound effect first designed in the nineteenth century. This work is part of an exploration of the potential of historical theatre sound effects as a resource for Sonic Interaction Design (SID). The synthesis engine is based on a physical model of friction and is programmed using the Sound Designer's Toolkit (SDT) suite of physical modelling objects in Max/MSP. The program is controlled in real-time with a single stream of rotation data from a rotary encoder and Arduino, with complexity achieved through a mapping strategy that recreates the mechanical process at the heart of the acoustic wind machine's sound production. The system is outlined, along with a discussion of the possible application of this approach to the modeling of other historical theatre sound effects.

1. INTRODUCTION

The design and live performance of sound with acoustic materials and mechanical devices has a long history in the theatre space [1]. This research explores this area to inform new strategies for Sonic Interaction Design (SID). SID researches the performative and multisensory aspects of sonic experience to design new sonic interactions [2]. Theatre wind machines were first used in the nineteenth century [3], and this particular sound effect method was chosen for closer investigation to facilitate the exploration of a continuous action-sound coupling [4] in an experimental setting. This work was informed by Serafin and de Götzen's [5] approach to replicating a historical acoustic device as a digital synthesis engine and user interface. It is proposed to deconstruct the action-sound coupling afforded by the acoustic wind machine in order to examine the importance of various modes of feedback (sonic, tactile, kinesthetic) while performing with sound [7]. A digital synthesis engine created to imitate the sonic response and performative action of the acoustic wind machine as closely as possible will enable this deconstruction to be achieved.

The first section of this article briefly describes the design of the prototype digital sound synthesis engine that physically models the workings of the acoustic wind machine; the second section describes in detail the digital modelling of the action (a simple rotational gesture) and its mapping and complex effect on the sound engine. In the third and final section of this article, it is proposed that this digital action model can be applied to the reproduction of other historical theatre sound effects, and more broadly to the design and control of real-time physical modelling

synthesis systems that require a simple gesture and a complex audio output.

2. DIGITAL SOUND AND ACTION

2.1. Sound: The Digital Synthesis Engine

An acoustic wind machine consists of a wooden slatted cylinder mounted on a frame so that it can rotate freely about its axle. The performer activates this rotation with a crank handle coupled to the cylinder. The wooden slats rub against an encompassing cloth as the cylinder moves, and the friction created by this interaction produces a wind-like sound. Historical research has revealed that the basic acoustic wind machine design that became popular in the nineteenth century was reinterpreted and adjusted with each iteration by many different practitioners of the craft [8]. As such, there is no single definitive version of this device. Rather than attempt to create an ideal version of a wind machine through physical modeling synthesis, an acoustic version was first constructed to enhance the descriptions available in historical texts and make them more concrete [9]. This version was designed through a synthesis of different designs, following a similar process to that of historical practitioners. A specific example was then available to model in software.

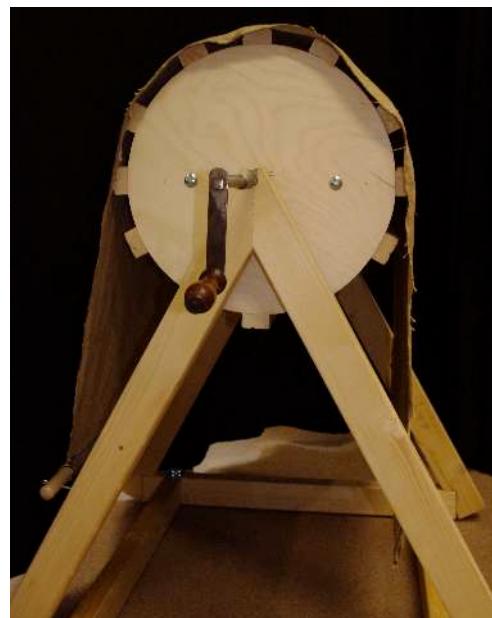


Figure 1: The acoustic wind machine example.

To facilitate real-time performance with a digital wind sound while providing the same multisensory feedback, the acoustic wind machine itself was fitted with a rotary encoder mechanically coupled to its moving cylinder with 3D printed gearing. The movement of the encoder was read with an Arduino prototyping board connected to the computer via USB, sending a single stream of rotation data scaled to a range of 0° - 360°. Max/MSP¹ was chosen as the platform for a design-led approach to programming and the production of a functional prototype of a digital wind sound using the Sound Designer's Toolkit (SDT)² suite of physics-based sound synthesis algorithms. The SDT algorithms are computationally efficient and designed for advanced control mapping, a particularly relevant feature for this project. While Max/MSP is an ideal platform for prototyping this effect, it is envisaged that the overall model could in the future be ported to another platform with similar real-time audio processing and physical modelling functions.

To model the digital wind sound, the acoustic wind machine's stages of sound production were first deconstructed and described as an entity-action model [6]. The acoustic wind machine creates sound through friction, but instead of just two surfaces (one cylinder, one cloth) in contact, there are twelve slat 'rubbers' at work. As the cylinder of the acoustic wind machine is rotated, each of its twelve slats come into contact with the encompassing cloth and rub it to produce sound through friction. Each slat falls silent for part of its rotation as it moves out of range of the cloth, and only seven of the slats contact the cloth and produce sound at any one time (Figure 2). This mechanical process, which gives an individual envelope to the friction sound produced by each slat of the cylinder, informed the programming. Twelve instances of a physical model of nonlinear friction, the most recent iteration of the SDT's dynamic friction model [11], were created, giving one voice for each slat on the acoustic wind machine example. The physical model, which simulates the friction between a sliding probe and an inertial object with a modal resonator, used the following SDT algorithms:

- sdt.scraping~: A scraping texture generator, which outputs a force to be applied to a resonator or another solid. Affords real-time control over the grain, velocity and force of the scrape.
- sdt.friction~: An elasto-plastic friction model, which takes its rubbing force input from sdt.scraping~.
- sdt.inertial~: Inertial object model
- sdt.modal~: Modal resonator

The parameters to these objects were chosen through a process of comparison and evaluation of the sonic outputs of the acoustic wind machine and the digital synthesis engine [as reported in 12, 10] with the aim of creating a digital sound as close as possible to the acoustic sound (Table 1).

Table 1: Parameters to the Sound Designer's Toolkit (SDT) objects in Max/MSP for each 'digital slat' [12, 10].

<i>Parameters to sdt.scraping~</i>	<i>Surface profile (A signal)</i>	Noise
	<i>Grain (Density of micro- impacts)</i>	<i>0.080596, modulated by encoder data</i>
	<i>Velocity (m/s)</i>	<i>Real-time optical encoder data</i>
	<i>Force (N)</i>	<i>0.546537, modulated by encoder data</i>
<i>Parameters to sdt.friction~</i>	<i>External rubbing force</i>	Signal from sdt.scraping~
	<i>Bristle stiffness (Evolution of mode lock-in)</i>	500.
	<i>Bristle dissipation (Sound bandwidth)</i>	40.
	<i>Viscosity (Speed of timbre evolution and pitch)</i>	1.2037
	<i>Amount of sliding noise (Perceived surface roughness)</i>	0.605833
	<i>Dynamic friction coefficient (High values reduce sound bandwidth)</i>	0.159724
	<i>Static friction coefficient (Smoothness of sound attack)</i>	0.5 (for Hemp cloth and Wood)
	<i>Breakaway coefficient (Transients of elasto- plastic state)</i>	0.174997
	<i>Stribeck velocity (Smoothness of sound attacks)</i>	0.103427
	<i>Pickup index of object 1 (Contact point)</i>	0
	<i>Pickup index of object 2 (Contact point)</i>	0
	<i>Mass of inertial object (Kg)</i>	0.01
<i>Parameters to sdt.inertial~</i>	<i>Fragment size (To simulate crumpling)</i>	1
	<i>Frequency factor</i>	1
	<i>Frequency of each of three modes (Hz)</i>	380, 836, 1710
	<i>Decay factor</i>	0.005
	<i>Decay of each mode (s)</i>	0.8, 0.45, 0.09
	<i>Pickup factor</i>	2.2
	<i>Pickup0_1</i>	50.
	<i>Pickup0_2</i>	100.
	<i>Pickup0_3</i>	80.
<i>Parameters to sdt.modal~</i>	<i>Fragment size</i>	1

¹ <http://www.cycling74.com/>

² For Max/MSP and Pure Data: <http://soundobject.org/SDT/>

To ensure the efficiency of the digital synthesis engine and keep demands on the computer's CPU low, a strategy to pause computation of the friction model while the slat is not producing sound was devised. In Max/MSP this can be achieved by housing each digital slat inside a poly~ object. Designed to manage polyphony and DSP, poly~ also affords downsampling and individual voice muting. This ensured that each digital slat voice could be muted when its acoustic counterpart would be out of range of the cloth and therefore silent. Downsampling within each slat voice ensured that the program was responsive and efficient in real-time performance while running twelve nonlinear friction models.

To increase the responsiveness of the resulting sound of the digital wind machine, the grain and force parameters to each sdt.scraping~ object were modulated with real-time encoder data. The grain parameter was programmed to vary with the angular position of the digital slat, with the highest value corresponding to the top position of the acoustic wind machine (at 180°), where the slat is most fully in contact with the cloth. The acceleration data modulated the force parameter to each sdt.scraping~, reflecting the increased effort required for the first part of each rotation when overcoming inertia.

The acoustic wind machine's cloth is its main resonator, and the sdt.modal~ object in each digital slat adds modal resonance to the nonlinear friction model. To account for the cloth's role in dispersing the sound, a bidirectional digital waveguide model [13] was adapted. The cloth is pulled tight on one side of the acoustic wind machine, and hangs freely on the other (Figure 3). The tight side of the cloth, which is coupled to a wooden pole 'bridge' pinned to the a-frame, is similar to a bowed string, with the slatted cylinder 'bowing' the cloth during rotation. A digital waveguide in series with a low-pass filter and an all-pass filter was used to simulate dispersion of the friction sound through the tight side of the cloth, allowing for some damping due to its coupling to the acoustic wind machine's 'bridge.' Dispersion through the freely hanging side of the cloth was modeled using the most basic method, without damping, of a delay line in series with an all-pass filter [14].

2.2. Action: The Gesture Mapping

The mapping strategy focused on creating complexity from the encoder's single smooth data stream in order to recreate the gesture afforded by the acoustic wind machine as closely as possible. As previously outlined, the trajectory of the rotary encoder coupled to the movement of the acoustic wind machine's cylinder was mapped to a 360° rotation. Velocity and acceleration were also calculated from the variation in time of the encoder's data. A one-to-many mapping strategy was implemented [15]. To drive sound production from each slat model, the single stream of encoder data was parsed into eleven further streams, each placed out of phase with the original to correspond with the degree placement of the slats on the original acoustic wind machine (Figure 2). This provided twelve individual data streams, each corresponding to the movement of one of the slats on the acoustic machine, allowing the position of each to be tracked and its digital voice activated accordingly. These rotational data streams were mapped to the activation of each digital slat voice, muting it if it passed out of range of the cloth's position on the acoustic wind machine, and activating it again when it came into range (Figure 3).

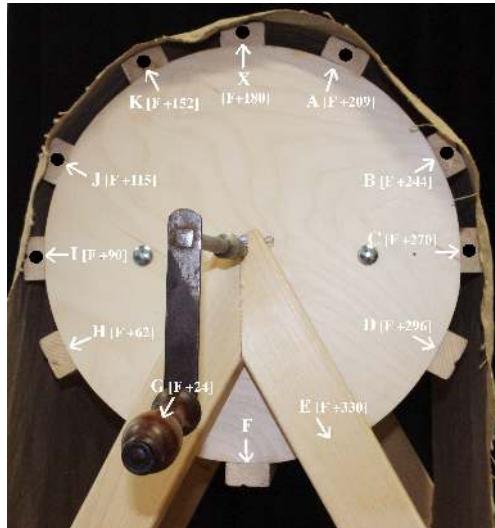


Figure 2: Placement of slats on the acoustic wind machine. A black dot marks each of the seven positions where the slats make contact with the cloth at any one time.

The rotational gesture afforded by the acoustic wind machine is comprised of two distinct stages due to the way the cloth has been fixed to one side of the frame, and the influence of gravity. This creates a rotational gesture that at first requires more effort to overcome oppositional force from gravity and the cloth's tension on the upstroke, and then requires less effort with the loose cloth and assistance from gravity on the downstroke (Figure 3). This produces an amplitude envelope that rises in the first half of the rotation, and then decreases in the second half [12]. This amplitude envelope is most pronounced at slow speeds, when the overall amplitude of the wind sound is lower. In addition, as the cylinder gains speed in rotation, and overcomes inertia, it moves much more freely, reducing the difference in amplitude between the upstroke and downstroke of each rotation, while the overall amplitude of the acoustic wind machine is higher.

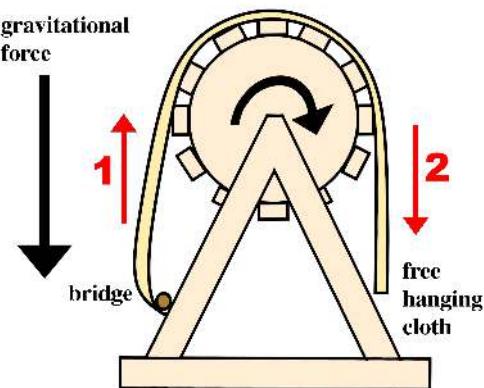


Figure 3: The two-part rotational gesture afforded by the acoustic wind machine [10].

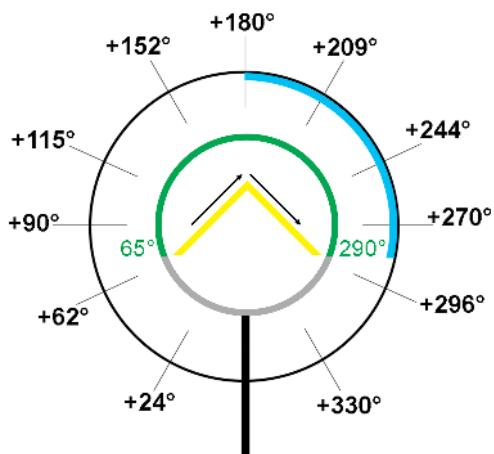


Figure 4: Angular starting positions of each digital slat, with their voice activation region (65° - 290° , in green), amplitude offset region (blue) and envelope of grain and force modulation (peaking at 180° , in yellow) highlighted.

The velocity parameter to `sdt.scraping~` activates each friction model and shapes its amplitude envelope. The 360° rotation data for each digital slat was mapped to this parameter, and additional steps were taken to reproduce the complexity of the rotational gesture afforded by the acoustic wind machine within the digital domain:

1. The rotation data mapped to activate the velocity parameter to `sdt.scraping~` was split in two, and offset (slightly reduced) during the second half of the rotation (Figure 4) to ensure higher amplitude during the first half of each rotation, and slightly lower amplitude during the second half of each rotation (Figure 5)
2. The rotation data was then filtered through a simple inertia model [10] to imitate the effect of the oppositional forces on the rotational gesture, effectively recreating the upstroke and downstroke.
3. In addition, the amplitude of the summed outputs of the twelve digital slats (0. – 1.) was scaled by the velocity value of the encoder data stream, ensuring that the digital wind sound (produced by any number of rotations) had lower average amplitude at slower speeds and higher average amplitude at faster speeds, similar to its acoustic counterpart.

The development of the complexity of the digital wind sound is ongoing, and further parameters to the digital slats will be explored for real-time manipulation as required, but the current prototype is already quite effective in producing a digital audio simulation of the mechanical wind machine in action. A diagram detailing signal flow and mapping within Max/MSP is shown at Figure 6.

2.3. Evaluation

A full evaluation of the acoustic wind machine and digital synthesis engine in performance was conducted, with both systems simultaneously recorded to facilitate analysis. This has

been published elsewhere [10], and a summary of the main results now follows.

The digital wind sound begins and ends appropriately when controlled with the acoustic wind machine interface. The digital action model ensures that the amplitude envelope of the digital wind sound is similar to that of the acoustic wind sound (Figure 5), but the inertia model requires some further calibration to ensure it is not overly delaying the progress of the digital sound during repeated rotations.

The digital wind sound is perceivably wind-like, and sounds like a rotating machine during a repetitive rotational gesture. It does, however, lack power at those high frequencies responsible for the acoustic wind machine's characteristic 'whistling.' Further real-time modulation of parameters to the friction model and bidirectional cloth model will be investigated to brighten the digital sound at these frequencies during rotation to imitate the acoustic wind machine more fully.

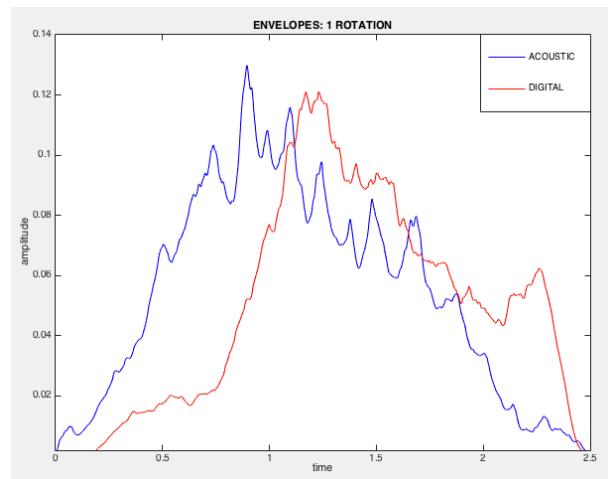


Figure 5: A comparison of the amplitude envelopes of the acoustic wind machine (blue) and digital synthesis engine (red) during the same single rotation [10].

3. FURTHER APPLICATIONS OF THIS ACTION MODEL

A detailed description of a physical model of an acoustic wind machine has been presented, which includes both a model of the sound and a model of the mechanical action that modulates the sound in time. In this mechanical action model, one stream of data from a single rotary encoder, coupled with a one-to-many mapping strategy, is effective in producing a complex continuous digital sound from a simple gesture.

3.1. Modelling Other Historical Designs

Historical theatre sound effects designers produced mechanical devices for a variety of complex sound creations that could be activated by the same simple rotational gesture used by the wind machine. For example:

1. Rain machines produced sound from multiple impacts and rolling. Various designs were based on the rotation

- of metal shot, dried peas or other small materials rotated inside a sealed barrel [16].
- 2. Crash machines produced sound from multiple impacts and rolling, with pieces of masonry or large rocks rotated inside a sealed barrel [17]. Some variations of the design were based on a large ratchet mechanism, and produced sound through impacts between pieces of wood [18].
 - 3. Thunder barrels produced sound from rolling metal cannonballs inside a metal-lined barrel or metal container [19].
 - 4. The crackling of a fire was produced by slowly rotating a wooden ratchet or clapper [20].
 - 5. Creaking was produced by rotating a clay pot inside a larger clay pot [18].
 - 6. Designs to imitate machinery, vehicles or motors produced sounds through repetitive impacts between different materials, such as an aeroplane sound achieved by plucking catgut strings with a toothed wheel [21].
 - 7. A simple gesture of rotation was also used to control the production of sound that had previously been achieved manually as devices became more complex in the early twentieth century. A mechanism for the sound of horses' hooves [22] triggered impacts on wooden cups, which would have been usually held in each hand and hit on a wooden surface. The Allefex machine, which offered mechanisms for many kinds of sounds, produced a series of gunshots, thunder and a steam engine controlled with crank handles [23].

The gesture mapping model (using the SDT and data from a rotary encoder and Arduino configuration) could be easily adapted to be used in the digital replication of many of these mechanical sound effects. For example, recreating a cylinder 'structure' from the rotation data using degree values in a similar way to the digital wind machine described here could trigger a series of `sdt.crumpling~` (a model of a stochastic sequence of impacts) and `sdt.friction~` objects to model the sliding and multiple impact sounds of an acoustic rain machine in rotation. As the cloth works against the rotational motion in the acoustic wind machine, so could the inertia of the material inside the rain machine's barrel be factored into the data stream through the proposed inertia model. Modelling these historical methods of sound production may reveal more useful strategies to increase the potential of simple rotational gestures in the control of physical modelling synthesis systems, as well as the design and DSP management of those systems for real-time performance.

3.2. A Mechanical Approach to Mapping: Extending Rotation

The mapping strategy implemented for the digital wind machine takes a mechanical approach, and recreates the stages of sound

production of the acoustic wind machine outlined in the entity-action model. This enables the digital system to take the individuality of the acoustic wind machine into account through the use of the specific degree placement of its wooden slats. This design could be extended to model other specific examples of acoustic wind machines through the incorporation of the detail of their own structures into the digital domain.

Parsing a single stream of rotation data into twelve distinct streams extends the possibilities for the real-time control of synthesis with a simple gesture of rotation. This is particularly pertinent in the case of real-time performance with physical modelling synthesis, where action and resulting sound should be tightly coupled. Other simple single-gesture controllers could be investigated and extended in a similar way, perhaps to control physical models of other historical sound effects devices. For example, a digital fader could set off a chain of impact events, each triggered by a particular position on its travel.

More broadly, this approach opens a new design space for real-time performance with physical modelling synthesis. By building on historical techniques for making sound with acoustic materials and simple mechanisms, new interactions between virtual materials can be devised and explored while simultaneously affording a simple, but meaningful and rich control interface to the performer. This approach might also be applied in a meaningful way to other synthesis methods, enabling (as in the case of the crank handle example) the control of envelopes and triggers in real-time performance to bring obvious, or more 'mechanical,' affordances to potentially more abstract methods of sound creation. This is an area currently under investigation as part of this work.

4. CONCLUSIONS

This paper has proposed a strategy for real-time control over physical modeling synthesis informed by the design and operation of historical theatre sound effects. A description of the design of a digital wind machine synthesis engine based on the Sound Designer's Toolkit (SDT) in Max/MSP was detailed, along with the mechanical mapping model to add complexity to a single stream of rotational data for its performance in real-time. Further applications of this approach and suggestions for future work were also outlined.

5. ACKNOWLEDGMENTS

This research is supported by the Arts and Humanities Research Council (AHRC) through the White Rose College of the Arts and Humanities (WRoCAH) Doctoral Training Partnership.

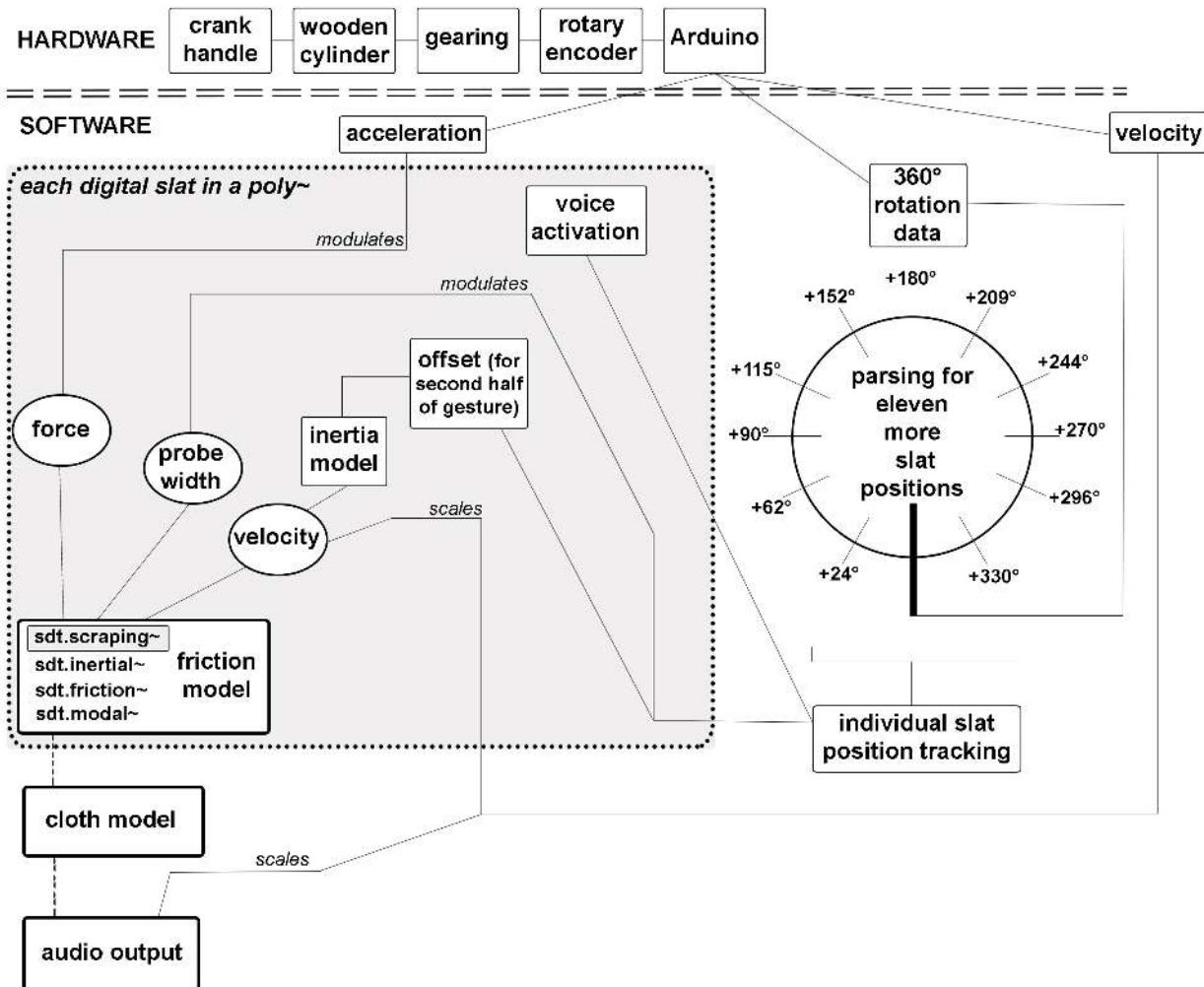


Figure 6: Diagram of signal flow and mapping for the digital wind machine in Max/MSP.

6. REFERENCES

- [1] R. Brown, *Sound: A Reader in Theatre Practice*: Palgrave Macmillan, 2010.
- [2] K. Franinović and S. Serafin, *Sonic Interaction Design*: MIT Press, 2013.
- [3] J. Moynet, A. S. Jackson, and M. G. Wilson, *French Theatrical Production in the Nineteenth Century: "L'Envers du Théâtre" by M.J. Moynet, 1873* vol. 10. State University of New York, USA: Max Reinhardt Foundation with the Center for Modern Theatre Research, 1976.
- [4] A. R. Jensenius, "Action-sound: Developing Methods and Tools to Study Music-Related Body Movement," Department of Musicology, University of Oslo, 2007.
- [5] S. Serafin and A. De Götzen, "An Enactive Approach to the Preservation of Musical Instruments Reconstructing Russolo's Intonarumori," *Journal of New Music Research*, vol. 38, pp. 231-239, 2009.
- [6] A. Farnell, *Designing Sound*: MIT Press Cambridge, 2010.

- [7] F. Keenan and S. Pauletto, "["Listening Back": Exploring the Sonic Interactions at the Heart of Historical Sound Effects Performance](#)," *The New Soundtrack*, vol. 7, pp. 15-30, 2017.
- [8] F. Keenan, "[A Theatre Wind Machine as Interactive Sounding Object](#)," presented at the International Conference on Live Interfaces (ICLI), Doctoral Colloquium, University of Sussex, 2016.
- [9] D. Elliott, R. MacDougall, and W. J. Turkel, "New old things: Fabrication, physical computing, and experiment in historical practice," *Canadian Journal of Communication*, vol. 37, 2012.
- [10] F. Keenan and S. Pauletto, "[Design and Evaluation of a Digital Theatre Wind Machine](#)," in *Proceedings of The 17th International Conference on New Interfaces for Musical Expression (NIME 17)*, Copenhagen, Denmark, 2017.
- [11] S. Delle Monache, D. D. C. Drioli, F. Fontana, S. Papetti, P. Polotti, and D. Rocchesso, "Closing the loop of sound evaluation and design (CLOSED): Deliverable 2.1, algorithms for ecologically-founded

- [12] sound synthesis: Library and documentation," Technical report, UNIVERONA (Verona)2007.
- [13] F. Keenan and S. Pauletto, "[An Acoustic Wind Machine and its Digital Counterpart: Initial Audio Analysis and Comparison](#)," presented at the Interactive Audio Systems Symposium (IASS), University of York, York, UK, 2016.
- [14] M. Karjalainen, V. Välimäki, and T. Tolonen, "Plucked-string models: From the Karplus-Strong algorithm to digital waveguides and beyond," *Computer Music Journal*, vol. 22, pp. 17-32, 1998.
- [15] J. O. Smith, *Physical audio signal processing: For virtual musical instruments and audio effects*: W3K Publishing, 2010.
- [16] A. Hunt, M. M. Wanderley, and M. Paradis, "The importance of parameter mapping in electronic instrument design," *Journal of New Music Research*, vol. 32, pp. 429-440, 2003.
- [17] D. Collison, "The Sound of Theatre," ed: London: Professional Lighting and Sound Association, 2008.
- [18] F. Napier, *Noises Off: A Handbook of Sound Effects*: London: JG Miller, 1936.
- [19] G. H. Leverton, *The Production of Later Nineteenth Century American Drama: A Basis for Teaching*: AMS Press, 1936.
- [20] M. K. Culver, "A History of Theatre Sound Effects Devices to 1927," University of Illinois at Urbana-Champaign, 1981.
- [21] A. E. Peterson, "Stage Effects and Noises Off," *Theatre and Stage*, vol. 1 and 2 1934.
- [22] A. Rose, *Stage Effects: How to Make and Work Them*. Chatham: Mackays Ltd, 1928.
- [23] V. D. Browne, *Secrets of Scene Painting and Stage Effects*: Routledge, 1913.
- [24] F. A. A. Talbot, *Moving Pictures: How they are made and worked*: JB Lippincott Company, 1914.

SIMULATING THE FRICTION SOUNDS USING A FRICTION-BASED ADHESION THEORY MODEL

Takayuki Nakatsuka

Department of Pure and Applied Physics,
Waseda University
Tokyo, Japan
t59nakatsuka@fuji.waseda.jp

Shigeo Morishima

Waseda Research Institute for Science and Engineering
Tokyo, Japan
shigeo@waseda.jp

ABSTRACT

Synthesizing a friction sound of deformable objects by a computer is challenging. We propose a novel physics-based approach to synthesize friction sounds based on dynamics simulation. In this work, we calculate the elastic deformation of an object surface when the object comes in contact with other objects. The principle of our method is to divide an object surface into microrectangles. The deformation of each microrectangle is set using two assumptions: the size of a microrectangle (1) changes by contacting other object and (2) obeys a normal distribution. We consider the sound pressure distribution and its space spread, consisting of vibrations of all microrectangles, to synthesize a friction sound at an observation point. We express the global motions of an object by position based dynamics where we add an adhesion constraint. Our proposed method enables the generation of friction sounds of objects in different materials by regulating the initial value of microrectangular parameters.

1. INTRODUCTION

Friction and its sound are familiar phenomena. We encounter such sounds almost constantly in our daily lives: sounds from brushing of hand bags and garments, footsteps, rustling of leaves, and rolling of tires are some examples of friction sounds. The sounds in computer animations (movies, video games, etc.) are created by Foley artists. Foley artists use two approaches to produce sound effects: recording actual but not actual sounds (a conspicuous example is rolling adzuki beans on a basket to create a sound of waves) and using a synthesizer to compose sounds. However, creation of sound effects by these approaches places a heavy burden on Foley artists to create various ingenious plans to create ideal sounds. On the other hand, friction is a complicated phenomenon on a macroscale as well as a microscale and physical parameters vary for each material. Also, extracting the theoretical principle of friction using numerical calculation is challenging. Simulating its sound is then difficult without considering multi-scale of the material.

In this paper, we suggest novel techniques to synthesize friction sounds based on physics simulation. Our proposed method has wide applicability and can create sounds of various objects from rigid to elastic. In related works, objects are limited to only rigid materials [1, 2] or databases constructed by recorded friction sounds are used [3–5]. Our method adopts a manner of subsuming the adhesion theory [6] into computer animation and uses position based dynamics (PBD) [7] framework, which is widely accepted in the field of computer animation because of its robustness and simplicity for simulation.

2. RELATED WORK

On a microscale, an object surface is usually rough with asperities which is composed of a mass of atoms and molecules. Each asperity forms an elevation irregularly on a surface. Therefore, we consider the irregularities of the actual contact point positions formed by the distribution of asperities. The computation of friction at the actual contact point on the object surface is modeled by direct numerical solution [8,9] and a method specialized in computation [10]. However, some aspects of friction need further understanding to model them exactly. Because a precise friction model is not available at present, prediction of friction sounds that are sensitive to friction parameters is fraught with errors. Avanzini et al. [11] and Desvages et al. [12] focused on friction in a bowed string. Akay et al. [13] have compiled past studies about friction sounds. These studies explain the cause of friction sounds in great detail for different target bodies. In this paper, we suggest a friction model which has a wide application.

In regard to computer animation, Takala et al. [14] and Gaver [15] suggested a series of frameworks to express a sound (sound rendering) for the first time. The series of frameworks of sound rendering synthesize the sound according to the motion of objects and compute a spread of the sound. Based on these frameworks, Van den Doel and Pai [16, 17] generated a plausible sound of objects by performing linear modal analysis of the input shape. Those studies were the first to apply computer mechanics to sound synthesis but they could handle only simple shapes. Other approaches synthesize the sound of specific musical instruments [18–21]. O’Brien et al. [22, 23] and Van den Doel et al. [1] proposed a solution that can be generalized to all rigid objects. These works made it possible to put knowledge to practical use. As a pioneering work in virtual reality of sounds, Pai et al. [24] built an interactive system to generate sounds synthesized based on the motion of the user at the time of contact to the material body [25–27]. In addition, several works have reported synthesis of sounds of collision or contact with objects based on linear modal analysis and speed-up and optimization of such synthesis [28–35]. It is difficult to adopt the linear modal model to thin-shell objects where a non-linearity exists, but Chadwick et al. [34], Bilbao [20] and Cirio et al. [36] have suggested improved techniques to generate high-quality sounds in such cases. Most of these studies aimed at rigid bodies and provided outputs that are plausible sounds of “rolling friction” or “sticking collisions,” which are modeled by friction models [37, 38], but not of “sliding friction,” which is caused by sliding objects. The reason these techniques are not good at synthesizing the sounds of sliding friction is that sliding friction involves continuous collisions in contrast with rolling friction, which involves discrete collisions.

Van den Doel et al. [1] generated friction sounds to remark on the fractal characteristics of asperities. However, the model is applicable only to simple and homogeneous shapes of material body. Ren et al. [2] synthesized friction sounds using the techniques of Raghuvanshi et al. [28] and a surface model of the object in three phases of scales following [1]. However, the model could not treat objects with nonlinear properties such as deformation of clothes. On the other hand, An et al. [3] made friction sounds synthesized to the cloth animation using a database of recorded actual sounds. The velocity of the cloth mesh top is matched with entries in the database and the corresponding fragment of friction sound is returned, and the fragments are joined to synthesize the sound. Making of such a database requires a large amount of time and labor to record the sounds of rubbing clothes at various speeds. Our proposed method solves these problems partially with a novel surface model using a physically well-grounded simulation and enables synthesis of friction sounds for deformable objects.

3. BACKGROUND

3.1. Principle of Friction

Friction is a complicated phenomenon to solve analytically because it is an irreproducible, microscale phenomenon (an object surface slightly deforms by friction). As shown in Fig. 1, when an external force is applied on an object in contact with another object in order to move the first object, friction acts to impede the motion. The force caused by friction is called friction force. The magnitude of friction force depends on the objects in contact.

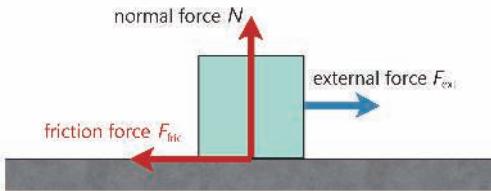


Figure 1: Forces during motion of two objects in contact. The friction force F_{fric} is the resistance between the relative motion of the two bodies.

The classical laws of friction derived from observing objects, known as Amonton-Coulomb laws of friction, are as follows.

- Amonton's first law: magnitude of friction force is independent of contact area.
- Amonton's second law: magnitude of friction force is proportional to magnitude of normal force.
- Coulomb's law of friction: magnitude of kinetic friction is independent of speed of slippage.

A phenomenological discussion of Amonton-Coulomb laws is incorporated in the theory of adhesion [6], which is deeply rooted in the field of tribology. Adhesion theory describes a friction phenomenon based on Amonton-Coulomb model of friction, especially with a focus on the second property. The main principle of adhesion theory is that the contact between two object is present at points. The surface roughness of an object is on a sufficiently small scale. In particular, the convex portion of the object surface is termed as an asperity. Friction is considered to be caused by contact between asperities of the two objects (called actual contact

points). At each actual contact point, there is adhesion (or bond) of two objects due to the interaction between molecules or atoms of the object surface. The sum of the areas of the actual contact points is called actual contact area. The force required to overcome the adhesion at actual contact points is the friction force, which is represented by the following equation:

$$F_{fric} = \sigma_s \times A_r \quad (1)$$

where F_{fric} is the friction force, σ_s is the shear strength, and A_r is the actual contact area. Assuming A_r is related to the load W as $A_r = aW$ with a constant of proportionality a , the friction coefficient μ is given by the following equation.

$$\mu = \frac{F_{fric}}{W} = \frac{\sigma_s \times A_s}{W} = a\sigma_s \quad (2)$$

Therefore, the friction coefficient μ does not depend on the apparent area of contact (Amonton's first law of friction). An actual contact point is formed at the point surrounded by the green rectangle in Fig. 2, forming a "stick" state. The adhesion at the pair of asperities is cut by the elastic energy and the actual contact point collapses a few moments later, termed as a "slip" state, when the two objects move against each other in a horizontal direction. The energy dissipated in this series of processes is equal to the work done by the friction force. The stick-slip phenomenon is repetition of the process of adhesion and cutting the pair of asperities. The global motion of the object and the stick-slip phenomenon at actual contact points are not related. The actual contact points change with time by the motion of the object. Even though the object continues the sliding movement at a regular velocity, asperities on the object surface are in stick state with pairs of asperities on the different object surfaces forming the actual contact points. An asperity in the stick state is deformed by the motion of an object, and the elastic energy is accumulated around the asperity. The actual contact point beyond the limit of elasticity slip by the elastic energy and the neighborhood of an asperity vibrate around the equilibrium state. The stick-slip motion is repeated locally within the system during the movement of an object at a constant velocity. Since the frequency of local slip per unit time is proportional to the sliding velocity, the energy dissipation per unit time is also proportional to the sliding velocity. Therefore, the friction force does not depend on the sliding velocity as the energy dissipation per unit time is equal to the friction force times the sliding velocity. Such a localized stick-slip motion explains the third Amonton-Coulomb law.

3.2. Sound Propagation

The sound propagates in air as a wave. In general, a sound propagated through the space Ω can express using a wave equation (3):

$$\nabla^2 \phi(\vec{x}, t) - \frac{1}{c^2} \frac{\partial^2 \phi(\vec{x}, t)}{\partial t^2} = 0, \quad \vec{x} \in \Omega \quad (3)$$

where c is the speed of sound in air (at standard temperature and pressure, it is empirically described as $c = 331.5 + 0.6t$ [m/s] with temperature t [$^{\circ}\text{C}$]). In the study of acoustics, acoustic pressure $p(\vec{x}, t)$ and particle velocity $\mathbf{u}(\vec{x}, t)$ are used as a variable of the wave equation as follows.

$$\left(\nabla^2 - \frac{1}{c^2} \frac{\partial^2}{\partial t^2} \right) p(\vec{x}, t) = 0 \quad (4)$$

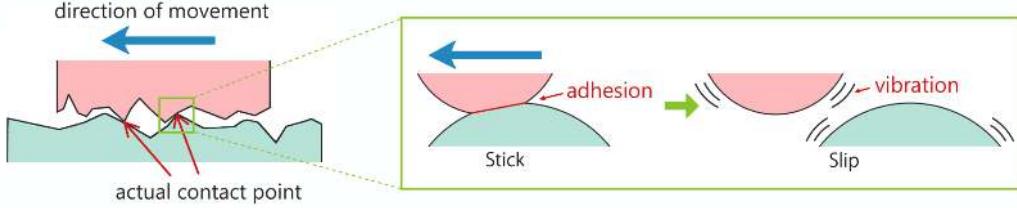


Figure 2: Illustration of contact between two objects and the stick-slip phenomenon at an actual contact point. Due to the roughness of the surface, objects are in contact with each other at points, instead of a large area.

$$\left(\nabla^2 - \frac{1}{c^2} \frac{\partial^2}{\partial t^2} \right) \mathbf{u}(\vec{x}, t) = 0 \quad (5)$$

The relation between acoustic pressure and particle velocity is described as first-order partial differential equation (PDE):

$$\rho \frac{\partial \mathbf{u}(\vec{x}, t)}{\partial t} = -\nabla p(\vec{x}, t) \quad (6)$$

where ρ is the density of the air (at standard temperature and pressure, it is empirically described as $\rho = 1.293/(1 + 0.00367t)$ [kg/m³] with temperature t [°C]).

4. GENERATE FRICTION SOUND

Our physics-based method comprises three steps (see Fig. 3). First, we separate an object surface into microrectangles (local shape) around each vertex of the object and define their initial size. The initial size of the microrectangles is decided by the object texture (e.g., in the case of a cloth, the initial size is the thread diameter) and is deformed on the basis of two assumptions: the deformation (1) depends on the velocity of the contacting vertex and (2) the initial size obeys normal distributions. The deformation of a microrectangle is given by the following equation:

$$D\nabla^4 w + \rho \frac{\partial^2 w}{\partial t^2} = 0 \quad (7)$$

where D is the flexural rigidity, w is the displacement, and ρ is the density of the object [39]. We then determine the global shape of the object using PBD [7]. As constraints, we adopt a distance constraint between the vertices of the object and an adhesion constraint between two objects. Finally, we synthesize the sounds generated by each microrectangle at the observation point by using a wave equation [40]. The synthesized sound can be expressed as a linear sum of sounds because the sound waves are independent.

4.1. Determination of Vibration Area

The spectrum of friction sound depends on the velocity of the object. Therefore, An et al. [3] recorded several friction sounds at different speeds and made a friction sound database consisting of pairs of a sound spectrum and a velocity. We also conducted experiments to record friction sounds as described in their method [3] and confirmed the relation between spectrum and velocity (see Fig. 4). To reflect the dependence on velocities of the object, we define relationships between sides a and b of the rectangular domain and the velocity v of an actual contact point as follows:

$$\frac{1}{a} = \alpha v, \quad \frac{1}{b} = \beta v \quad (8)$$

where α and β are constants. The roughness of the object surface is expressed as dispersions σ_a and σ_b , respectively, of the normal distributions of lengths of a and b .

4.2. Determination of Global Shape and Amplitude

Several studies have used elastic body simulation for computer animation. There are three main methods used in computer graphics. Finite element method (FEM) is the one of most popular methods and is based on physics. The spring mass model [41] sacrifices accuracy to achieve a low costs of computation. In addition, there is a specialized technique for computer animation [42]. To determine the global transition of the body shape and the amplitude which of vibration of the rectangular domains by friction, we handle the mesh vertices of the material body as true actual contact points and control their positions. This is because PBD is widely known in the field of computer graphics and has the advantages of robustness and simplicity. Let us postulate there are N particles with positions \mathbf{x}_i and inverse masses $g_i = 1/m_i$. For a constraint C , the positional corrections $\Delta \mathbf{q}_i$ is calculated as

$$\Delta \mathbf{q}_i = -sg_i \nabla_{\mathbf{q}_i} C(\mathbf{q}_1, \dots, \mathbf{q}_n) \quad (9)$$

where

$$s = \frac{C(\mathbf{q}_1, \dots, \mathbf{q}_n)}{\sum_j g_j \nabla_{\mathbf{q}_i} C(\mathbf{q}_1, \dots, \mathbf{q}_n)}. \quad (10)$$

As constraints with respect to the position of the object vertices, we adapt constraints representing the shape deformation of the body and the adhesion at actual contact points: the respective terms defined as $C_{Deformation}$ and $C_{Adhesion}$. Then, the total constraint C is expressed by

$$C = C_{Deformation} + C_{Adhesion}. \quad (11)$$

In our method, we adopt a distance constraint between the vertices of the body for $C_{Deformation}$ and a relational expression for $C_{Adhesion}$, which is given as

$$C_{Adhesion} = \begin{cases} \frac{Z}{r} & (r > d) \\ Z & (r \leq d) \end{cases}. \quad (12)$$

where r is the distance between a vertex of the object and another object, Z is the adhesion degree depended on material properties and d is the effective distance of the adhesion. Equation (12) denotes represents the fact that the attracting forces between the objects are given as a Coulomb potential. Then, the amplitude A of the rectangular amplitude is determined by the following equation:

$$A = \frac{1}{\#X} (\mathbf{x}_i(t + \Delta t) - \mathbf{x}_i(t)) \quad (13)$$

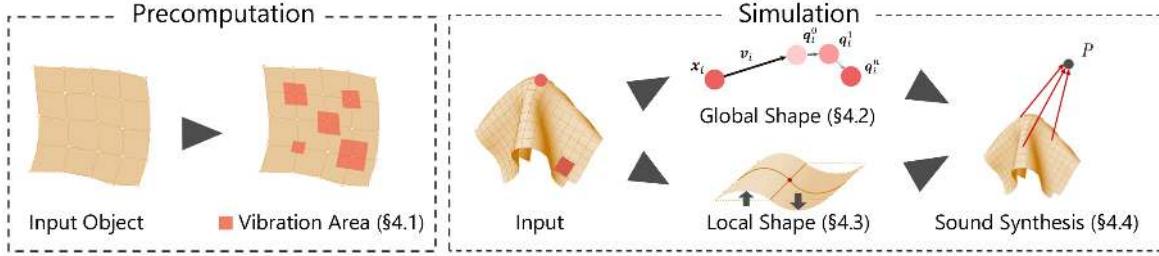


Figure 3: **Overview:** For the input object, we first define the vibration area which deform independently with regard to each surrounding vertex of the object surface. We then compute a global shape of the object and an amplitude of each vibration area to use PBD, and a local shape of each rectangle. Finally, we synthesize the sounds which is generated by each surrounding vertex of the object at the observation point P based on wave equation.

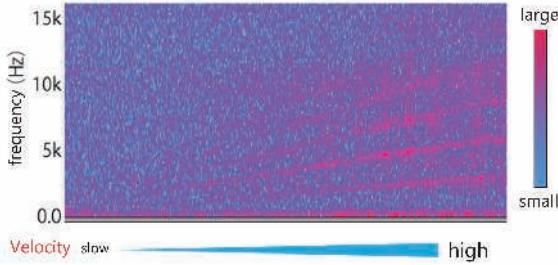


Figure 4: A sound spectrogram of the friction sounds of cupra fiber. The red parts of the figure show that spectrum and velocity are related.

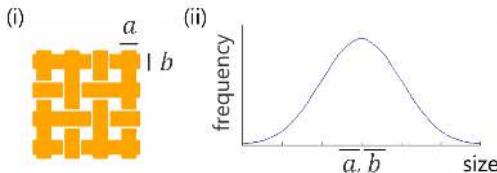


Figure 5: An example of setting initial parameters of a and b . In the case of a cloth, we use the diameter of the cloth's string and a normal distribution of their sizes.

where $\mathbf{x}_i(t)$ is the i -th vertex position of the object at time t and X is the direct product of the sets M and N defined as

$$M = \{m \mid m = 2k + 1\} \quad (14)$$

$$N = \{n \mid n = 2l + 1\} \quad (15)$$

where m and n are the mode orders of the vibration and k and l are natural numbers $\mathbf{N} \cup \{0\}$.

4.3. Determination of Local Shape

This section describes the main principle of our method to define the vibration area on an object surface. Waves propagate from the actual contact point when friction occurs. We consider that these waves spread out a rectangular domain around the actual contact point. In general, the propagating waves are attenuated in the body or prevented from spreading to the neighboring actual contact points. Therefore, the vibration of the rectangular domain

around the actual contact point can be expressed by adopting an appropriate boundary condition. In this paper, we take simple support (SS) by several asperities as a boundary condition on an object surface. Given a vibration area, let S denote its rectangular domain as follows:

$$S = \left\{ (x, y, z) \mid -\frac{a}{2} \leq x \leq \frac{a}{2}, -\frac{b}{2} \leq y \leq \frac{b}{2}, -\frac{h}{2} \leq z \leq \frac{h}{2} \right\} \quad (16)$$

where a and b are lengths of each side which is parallel to X or Y-axis. Then, the deformation (displacement w_{mn} and natural angular frequency ω_{mn}) of the rectangle domain can be described analytically by following equations:

$$w_{mn} = A \sin \frac{m\pi(x + \frac{a}{2})}{a} \times \sin \frac{n\pi(y + \frac{b}{2})}{b} e^{-R+i\omega_{mn}t} \quad (17)$$

$$\omega_{mn} = \pi^2 \left\{ \left(\frac{m}{a} \right)^2 + \left(\frac{n}{b} \right)^2 \right\} \sqrt{\frac{Eh^3}{12\rho(1-\nu^2)}} \quad (18)$$

where A is the amplitude, m and n are the mode orders, R is the attenuation coefficient, E is Young's modulus, h is the thickness of domain S , ρ is the density per unit volume, and ν is Poisson's ratio (see Appendix A).

4.4. Sound Synthesis

The wave equation with sound sources is generally solved to analyze sound propagation in air. These equations have high computational cost due to the complex boundary conditions. In our case, a rectangle sound source can be approximated by a point sound source, because a and b (sides of the rectangle) are sufficiently small compared with the distance $|\mathbf{r}|$ between the actual contact point and the observation point. Therefore, we take the limit of $a, b \rightarrow 0$ to approximate a plane sound source with a point sound source. Then, the spatial distribution of the sound pressure $p(r, t)$, where r is the position and t is the time, is denoted as

$$p(r, t) = i\rho c V_0 \frac{k}{4\pi r} e^{-ikr} \quad (19)$$

where i is the imaginary unit, ρ is the air density, c is the acoustic velocity, V_0 is the vibration velocity of the object surface, and k is

the wave number (see Appendix B). The vibration velocity V_0 can be expressed as

$$V_0 = iA\omega_{mn}\alpha_{mne}^{i\omega_{mn}t} \quad (20)$$

where

$$\alpha_{mn} = \begin{cases} 1 & m+n \equiv 0 \pmod{2} \\ -1 & m+n \equiv 1 \pmod{2} \end{cases}. \quad (21)$$

Finally, we synthesize the friction sounds caused by the vibration of each actual contact point at the observation point. Since the sound waves are independent, the synthesized sound at the observation point can be expressed as a linear sum of the sound made by each actual contact point. Therefore, the synthesized sound can be written as

$$sound = \sum_i p_i(r_i, t) \quad (22)$$

where p_i is the sound pressure of i -th vertex under friction.

5. RESULTS

We show the parameters that we use at the time of physical simulation in table 1. In the simulation, we utilize literature values [43, 44] in terms of shear modulus E , Poisson ratio ν and density per unit area ρ . The execution environment when simulating is CPU: Intel(R) core(TM) i7 CPU 2.93GHz, GPU: NVIDIA GeForce GT 220, RAM: 4GB and OS: Windows 7. Also, the time step t of the physical simulation is 1/44100 [s], the refresh rate of animations is 60 [fps] and the audio sampling rate of sounds by friction is 44100 [Hz].

5.1. Examples

cloth: In our results, we synthesize friction sounds in the case of pulling a cloth (cotton) over the sphere (see Fig. 6 and Fig. 7). We only change the velocity of a cloth in cloth1 and cloth 2. Then, our method can generate a friction sound matched to its speed in each case. Looking at the figures, there are differences between cloth 1 and 2. In particular, the results show that (i) fluctuating power of the sound denotes stick-slip and (ii) friction phenomena lead to various consequences on each frame.

metal: In this result, we generate the friction sounds of metal (copper) sliding on the floor (see Fig. 8). As a result, we succeed to create a friction sound of the metal. The spectrum shows natural frequency of a copper on 0.31 [s] and 0.54 [s], which is caused by a picking action.

In this study, we propose the examples - a cloth pulled over the sphere and a copper sliding on the floor. In particular, the results synthesizing friction sounds for clothes are the first illustrations. In addition, our proposed method is able to manage metal like materials such as a copper. In each case, our method can synthesize a friction sound as an actual sound. In our method, we focus on the only the friction sounds. However, our method can be used in combination with other methods owing to the postulation that object surface vibrates independently, and expected to bring about sounds improving effect of correcting.

6. CONCLUSION

In this study, we presented a novel method to generate friction sounds that do not depend on the material body. The method involves physical simulation of friction on the basis of the adhesion

theory. We synthesized friction sounds and confirmed that clear differences in tone appear by varying the parameters. As future works, synthesis of friction sounds with high quality in complicated scenes and body shapes is important. In particular, we would like to consider the transfer of sound waves between the object and the observation point to reflect the spatial properties affecting the sound, such as reflection, refraction, and diffraction. Further, we want to evaluate the similarity between the synthesized sounds and the actual sounds. In addition, we want to parallelize the calculations using GPUs for each actual contact point to speed up the synthesis.

7. ACKNOWLEDGMENTS

This work was supported by JST ACCEL Grant Number JPM-JAC1602, Japan.

8. REFERENCES

- [1] Kees Van Den Doel, Paul G Kry, and Dinesh K Pai, “Fo-leyautomatic: physically-based sound effects for interactive simulation and animation,” in *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*. ACM, 2001, pp. 537–544.
- [2] Zhimin Ren, Hengchin Yeh, and Ming C Lin, “Synthesizing contact sounds between textured models,” in *Virtual Reality Conference (VR), 2010 IEEE*. IEEE, 2010, pp. 139–146.
- [3] Steven S An, Doug L James, and Steve Marschner, “Motion-driven concatenative synthesis of cloth sounds,” *ACM Transactions on Graphics (TOG)*, vol. 31, no. 4, pp. 102–111, 2012.
- [4] Camille Schreck, Damien Rohmer, Doug James, Stefanie Hahmann, and Marie-Paule Cani, “Real-time sound synthesis for paper material based on geometric analysis,” in *Eurographics/ACM SIGGRAPH Symposium on Computer Animation (2016)*, 2016.
- [5] Andrew Owens, Phillip Isola, Josh McDermott, Antonio Torralba, Edward H Adelson, and William T Freeman, “Visually indicated sounds,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 2405–2413.
- [6] Lieng-Huang Lee, *Fundamentals of adhesion*, Springer Science & Business Media, 2013.
- [7] Matthias Müller, Bruno Heidelberger, Marcus Hennix, and John Ratcliff, “Position based dynamics,” *Journal of Visual Communication and Image Representation*, vol. 18, no. 2, pp. 109–118, 2007.
- [8] JT Oden and JAC Martins, “Models and computational methods for dynamic friction phenomena,” *Computer methods in applied mechanics and engineering*, vol. 52, no. 1, pp. 527–634, 1985.
- [9] JAC Martins, JT Oden, and FMF Simoes, “A study of static and kinetic friction,” *International Journal of Engineering Science*, vol. 28, no. 1, pp. 29–92, 1990.
- [10] Yu A Karpenko and Adnan Akay, “A numerical model of friction between rough surfaces,” *Tribology International*, vol. 34, no. 8, pp. 531–545, 2001.

Table 1: List of conditions at the time of the physical simulation

	Shear modulus E [GPa]	Poison ratio ν	Density ρ [kg/m^3]	a, b [m]	Dispersion σ^2	Vertex	Time [min]
cloth1	928.7	0.85	1.54×10^1	5.0×10^{-4}	1.0×10^{-12}	900	31
cloth2	928.7	0.85	1.54×10^1	5.0×10^{-4}	1.0×10^{-12}	900	59
copper	129.8	0.343	8.94	1.0×10^{-8}	1.0×10^{-7}	242	72

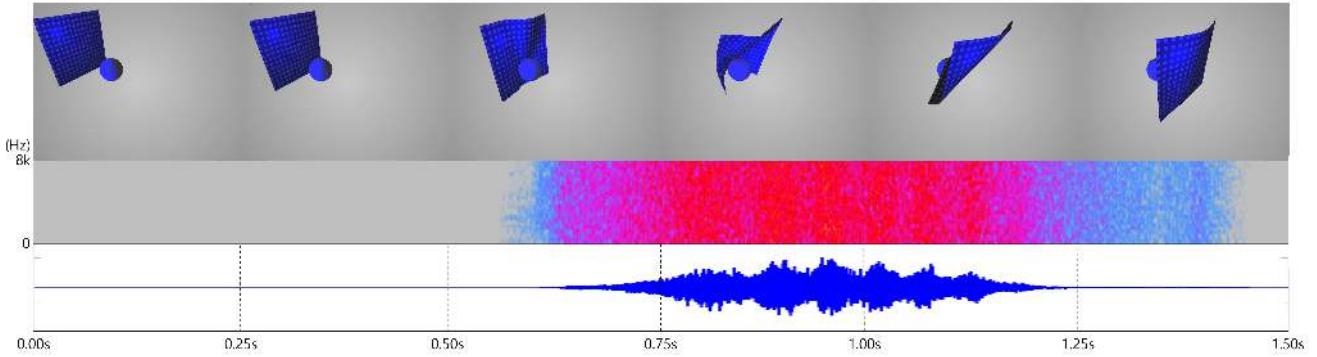


Figure 6: Key frames and sound waveform of a cloth (cotton). In this example, pull a cloth toward the front and slide it over the sphere.

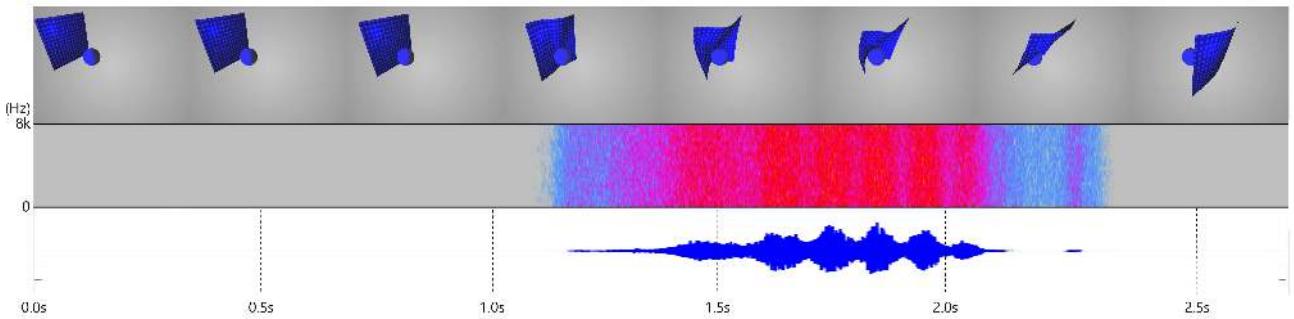


Figure 7: Key frames and sound waveform of a cloth (cotton). In this example, change the velocity of pulling a cloth ($\times 1/2$).

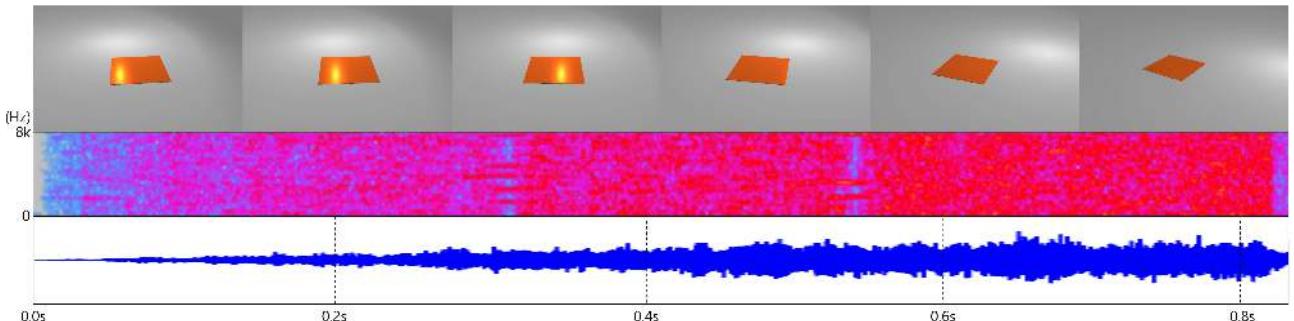


Figure 8: Key frames and sound waveform of a metal (copper). In this example, slide a copper on the floor.

- [11] Federico Avanzini, Stefania Serafin, and Davide Rocchesso, “Modeling interactions between rubbed dry surfaces using an elasto-plastic friction model,” in *Proc. DAFX*, 2002.
- [12] Charlotte Desvages and Stefan Bilbao, “Two-polarisation finite difference model of bowed strings with nonlinear contact and friction forces,” in *Int. Conference on Digital Audio Effects (DAFx-15)*, 2015.
- [13] Adnan Akay, “Acoustics of friction,” *The Journal of the Acoustical Society of America*, vol. 111, no. 4, pp. 1525–1548, 2002.
- [14] Tapio Takala and James Hahn, “Sound rendering,” in *ACM*

- SIGGRAPH Computer Graphics*. ACM, 1992, vol. 26, pp. 211–220.
- [15] William W Gaver, “Synthesizing auditory icons,” in *Proceedings of the INTERACT’93 and CHI’93 conference on Human factors in computing systems*. ACM, 1993, pp. 228–235.
- [16] Kees van de Doel and Dinesh K Pai, “Synthesis of shape dependent sounds with physical modeling,” 1996.
- [17] Kees van den Doel and Dinesh K Pai, “The sounds of physical shapes,” *Presence: Teleoperators and Virtual Environments*, vol. 7, no. 4, pp. 382–395, 1998.
- [18] Kevin Karplus and Alex Strong, “Digital synthesis of plucked-string and drum timbres,” *Computer Music Journal*, vol. 7, no. 2, pp. 43–55, 1983.
- [19] Perry R Cook, *Real sound synthesis for interactive applications*, CRC Press, 2002.
- [20] Stefan Bilbao, *Numerical sound synthesis: finite difference schemes and simulation in musical acoustics*, John Wiley & Sons, 2009.
- [21] Andrew Allen and Nikunj Raghuvanshi, “Aerophones in flatland: interactive wave simulation of wind instruments,” *ACM Transactions on Graphics (TOG)*, vol. 34, no. 4, pp. 134, 2015.
- [22] James F Director-O’Brien, “Synthesizing sounds from physically based motion,” in *ACM SIGGRAPH 2001 video review on Animation theater program*. ACM, 2001, pp. 59–66.
- [23] James F O’Brien, Chen Shen, and Christine M Gatchalian, “Synthesizing sounds from rigid-body simulations,” in *Proceedings of the 2002 ACM SIGGRAPH/Eurographics symposium on Computer animation*. ACM, 2002, pp. 175–181.
- [24] Dinesh K Pai, Kees van den Doel, Doug L James, Jochen Lang, John E Lloyd, Joshua L Richmond, and Som H Yau, “Scanning physical interaction behavior of 3d objects,” in *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*. ACM, 2001, pp. 87–96.
- [25] Cynthia Bruyns, “Modal synthesis for arbitrarily shaped objects,” *Computer Music Journal*, vol. 30, no. 3, pp. 22–37, 2006.
- [26] Nobuyuki Umetani, Jun Mitani, and Takeo Igarashi, “Designing custom-made metallophone with concurrent eigen-analysis,” in *NIME*. Citeseer, 2010, vol. 10, pp. 26–30.
- [27] Gaurav Bharaj, David IW Levin, James Tompkin, Yun Fei, Hanspeter Pfister, Wojciech Matusik, and Changxi Zheng, “Computational design of metallophone contact sounds,” *ACM Transactions on Graphics (TOG)*, vol. 34, no. 6, pp. 223, 2015.
- [28] Nikunj Raghuvanshi and Ming C Lin, “Interactive sound synthesis for large scale environments,” in *Proceedings of the 2006 symposium on Interactive 3D graphics and games*. ACM, 2006, pp. 101–108.
- [29] Doug L James, Jernej Barbič, and Dinesh K Pai, “Pre-computed acoustic transfer: output-sensitive, accurate sound generation for geometrically complex vibration sources,” in *ACM Transactions on Graphics (TOG)*. ACM, 2006, vol. 25, pp. 987–995.
- [30] Nicolas Bonneel, George Drettakis, Nicolas Tsingos, Isabelle Viaud-Delmon, and Doug James, “Fast modal sounds with scalable frequency-domain synthesis,” in *ACM Transactions on Graphics (TOG)*. ACM, 2008, vol. 27, pp. 24–32.
- [31] Jeffrey N Chadwick, Steven S An, and Doug L James, “Harmonic shells: a practical nonlinear sound model for near-rigid thin shells,” in *ACM Transactions on Graphics (TOG)*. ACM, 2009, vol. 28, pp. 119–128.
- [32] Changxi Zheng and Doug L James, “Rigid-body fracture sound with precomputed soundbanks,” in *ACM Transactions on Graphics (TOG)*. ACM, 2010, vol. 29, pp. 69–81.
- [33] Changxi Zheng and Doug L James, “Toward high-quality modal contact sound,” *ACM Transactions on Graphics (TOG)*, vol. 30, no. 4, pp. 38–49, 2011.
- [34] Jeffrey N Chadwick, Changxi Zheng, and Doug L James, “Precomputed acceleration noise for improved rigid-body sound,” *ACM Transactions on Graphics (TOG)*, vol. 31, no. 4, pp. 103–111, 2012.
- [35] Timothy R Langlois, Steven S An, Kelvin K Jin, and Doug L James, “Eigenmode compression for modal sound models,” *ACM Transactions on Graphics (TOG)*, vol. 33, no. 4, pp. 40–48, 2014.
- [36] Gabriel Cirio, Dingzeyu Li, Eitan Grinspun, Miguel A Otaduy, and Changxi Zheng, “Crumpling sound synthesis,” *ACM Transactions on Graphics (TOG)*, vol. 35, no. 6, pp. 181, 2016.
- [37] Matthias Rath, “Energy-stable modelling of contacting modal objects with piece-wise linear interaction force,” *DAFx-08, Espoo, Finland*, 2008.
- [38] Stefano Papetti, Federico Avanzini, and Davide Rocchesso, “Energy and accuracy issues in numerical simulations of a non-linear impact model,” in *Proc. Of the 12th Int. Conference on Digital Audio Effects*, 2009.
- [39] Arthur W Leissa, “Vibration of plates,” Tech. Rep., DTIC Document, 1969.
- [40] Miguel C Junger and David Feit, *Sound, structures, and their interaction*, vol. 225, MIT press Cambridge, MA, 1986.
- [41] Xavier Provot, “Deformation constraints in a mass-spring model to describe rigid cloth behaviour,” in *Graphics interface*. Canadian Information Processing Society, 1995, pp. 147–147.
- [42] Matthias Müller, Bruno Heidelberger, Matthias Teschner, and Markus Gross, “Meshless deformations based on shape matching,” in *ACM Transactions on Graphics (TOG)*. ACM, 2005, vol. 24, pp. 471–478.
- [43] Shigeta Fujimoto, “Modulus of rigidity as fiber properties,” *Textile Engineering*, vol. 22, no. 5, pp. 369–376, 1969.
- [44] National Astronomical Observatory of Japan, Ed., *Chronological Science Tables 2016 A.D.*, Maruzen, 2015.

A. VIBRATION OF PLATE

Suppose a micro rectangle is an elastic isotropic plate and there are no normal forces N_i , inplane shearing forces N_{ij} and external

forces q . Then, displacement $w(\vec{x}, t)$ of the plate is expressed in the following equation:

$$D\nabla^4 w + \rho \frac{\partial^2 w}{\partial t^2} = 0 \quad (23)$$

where D is flexural rigidity defined as follow,

$$D = \frac{Eh^3}{12(1-\nu^2)} \quad (24)$$

where E is Young's modulus, h is the thickness of plate, and ν is Poisson's rate. A general solution of the expression (23) is given by the following formula:

$$w = W(x, y)e^{i\omega t} \quad (25)$$

$$W(x, y) = X(x)Y(y) \quad (26)$$

where ω is angular frequency. And, $W(x, y)$ is calculated from the following formula:

$$\left(\nabla^4 - \frac{\rho\omega^2}{D} \right) W(x, y) = 0 \quad (27)$$

Then, define the domain S of a plate as follows:

$$S = \left\{ (x, y, z) \mid 0 \leq x \leq a, 0 \leq y \leq b, -\frac{h}{2} \leq z \leq \frac{h}{2} \right\} \quad (28)$$

Solve equation (27) assuming that boundary conditions of a plate are Simple Support for four sides under the following conditions:

$$w = 0, M_x = 0 \quad (\text{for } x = 0, a) \quad (29)$$

$$w = 0, M_y = 0 \quad (\text{for } y = 0, b) \quad (30)$$

where M_i is bending moment. With normal stress σ_i , bending moment M_i is expressed by the following formula.

$$M_i = \int_{-h/2}^{h/2} \sigma_i z dz \quad (\text{for } i = x, y) \quad (31)$$

Finally, we obtain the equations which is indicated in Section 4.3 as below.

$$W(x, y) = A \sin \frac{m\pi x}{a} \sin \frac{n\pi y}{b} \quad (32)$$

$$\omega_{mn} = \sqrt{\frac{D}{\rho}} \left\{ \left(\frac{m\pi}{a} \right)^2 + \left(\frac{n\pi}{b} \right)^2 \right\} \quad (33)$$

B. POINT SOUND SOURCE ANALYSIS

Let convert a wave equation into polar coordinates system from Descartes coordinate system to think about the spread of the sounds from a point sound source as follow:

$$\nabla^2 \phi(r, \theta, \varphi) - \frac{1}{c^2} \frac{\partial^2 \phi(r, \theta, \varphi)}{\partial t^2} = 0 \quad (34)$$

where

$$\begin{aligned} \nabla^2 &= \frac{1}{r^2} \frac{\partial}{\partial r} \left(r^2 \frac{\partial}{\partial r} \right) \\ &\quad + \frac{1}{r^2 \sin \theta} \frac{\partial}{\partial \theta} \left(\sin \theta \frac{\partial}{\partial \theta} \right) \\ &\quad + \frac{1}{r^2 \sin^2 \theta} \frac{\partial^2}{\partial \varphi^2} \end{aligned} \quad (35)$$

The sounds from a point sound source spread symmetrically and spherically. Therefore, we should only think about a component of radius direction. Then, we obtain the general solution of a wave equation in the polar coordinates system as below:

$$\phi(r) = \frac{A}{r} e^{i(\omega t - kr)} \quad (36)$$

where A is the constant. As mentioned in Section 3.2, acoustic pressure $p(r, t)$ and particle velocity $\mathbf{u}(r, t)$ are made use of a variable in acoustics as follows:

$$\left(\nabla^2 - \frac{1}{c^2} \frac{\partial^2}{\partial t^2} \right) p(r, t) = 0 \quad (37)$$

$$\left(\nabla^2 - \frac{1}{c^2} \frac{\partial^2}{\partial t^2} \right) \mathbf{u}(r, t) = 0 \quad (38)$$

where c is the acoustic velocity. Now, to derive the expression (19) using a general solution (36), suppose that the acoustic pressure $p(r, t)$ is as follow:

$$p(r, t) = \frac{A}{r} e^{i(\omega t - kr)} \quad (39)$$

Then, particle velocity $\mathbf{u}(r, t)$ is indicated by the following formula:

$$\begin{aligned} \mathbf{u}(r, t) &= -\frac{1}{\rho} \int dt \frac{\partial p}{\partial r} \\ &= -\frac{A}{\rho} \int dt \frac{\partial}{\partial r} \left\{ \frac{1}{r} e^{i\omega(t - \frac{r}{c})} \right\} \\ &= \frac{A}{\rho c} \frac{1}{r} \left(1 + \frac{1}{ikr} \right) e^{i(\omega t - kr)} \end{aligned} \quad (40)$$

From the initial condition, the constant A is given by the following:

$$\begin{aligned} \mathbf{u}(a, 0) &= U_0 = \frac{A}{\rho c} \frac{1}{a} \left(1 + \frac{1}{ika} \right) e^{-ika} \\ A &= \rho a c U_0 \frac{ika}{1 + ika} e^{ika} \end{aligned} \quad (41)$$

Finally, to take the limit of $a, b \rightarrow 0$, we can get the acoustic pressure $p(r, t)$ of a point sound source as below:

$$\begin{aligned} p(r, t) &= \frac{A}{r} e^{i(\omega t - kr)} \\ &= \frac{\rho c U_0}{4\pi} \frac{ika}{1 + ika} e^{i(\omega t - k(r-a))} \\ &\rightarrow i \rho c V_0 \frac{k}{4\pi r} e^{i(\omega t - kr)} \end{aligned} \quad (42)$$

where V_0 is the oscillate speed as follow.

$$U_0 = 4\pi a^2 V_0 \quad (43)$$

KINEMATICS OF IDEAL STRING VIBRATION AGAINST A RIGID OBSTACLE

Dmitri Kartofelev

Department of Cybernetics, School of Science,
Tallinn University of Technology,
Tallinn, Estonia
dima@ioc.ee

ABSTRACT

This paper presents a kinematic time-stepping modeling approach of the ideal string vibration against a rigid obstacle. The problem is solved in a single vibration polarisation setting, where the string's displacement is unilaterally constrained. The proposed numerically accurate approach is based on the d'Alembert formula. It is shown that in the presence of the obstacle the lossless string vibrates in two distinct vibration regimes. In the beginning of the nonlinear kinematic interaction between the vibrating string and the obstacle the string motion is *aperiodic* with constantly evolving spectrum. The duration of the aperiodic regime depends on the obstacle proximity, position, and geometry. During the aperiodic regime the fractional subharmonics related to the obstacle position may be generated. After relatively short-lasting aperiodic vibration the string vibration settles in the *periodic* regime. The main general effect of the obstacle on the string vibration manifests in the widening of the vibration spectra caused by transfer of fundamental mode energy to upper modes. The results presented in this paper can expand our understanding of timbre evolution of numerous stringed instruments, such as, the guitar, bray harp, tambura, veena, sitar, etc. The possible applications include, e.g., real-time sound synthesis of these instruments.

1. INTRODUCTION

Interaction and collision of a vibrating string with spatially distributed obstacles, such as fretboard or bridge, plays a significant role in the mechanics of numerous stringed musical instruments. One elegant example is the Medieval and Renaissance bray harp which has small bray-pins which provide a metal surface for the vibrating string to impact, increasing the upper partial content in the tone and providing a means for the harp to be audible in larger spaces and in ensemble with other instruments [1]. It is evident that for realistic physics-informed modeling of this instrument such nonlinearity inducing interactions need to be properly examined and understood.

The string–obstacle interaction modeling is a long-standing problem in musical acoustics. In the early twentieth century, Raman was first to study the problem and identify the veena bridge as the main reason for distinctive sounding of the tambura and veena. He noted that all string frequencies in these instruments are excited irrespective of the location of the excitation thereby violating the Young-Helmholtz law which states that the vibrations of a string do not contain the natural modes which have a node at the point of excitation. He notes that this is caused by the geometry of the bridge but did not explain the reason behind the inapplicability of the Young-Helmholtz law [2]. Since then, much effort has been devoted to modeling the collision dynamics of a vibrating string with various obstacles or boundary barriers. Over the

years many authors have solved this problem using different approaches. The problem was considered by Schatzman [3] and Cabannes [4], who used the method of characteristics and assumed that the string does not lose energy when it hits an obstacle. Burridge *et al.* [5] assumed an inelastic constraint where the string is losing kinetic energy during contact. Ducceschi *et al.* [6], Krishnaswamy and Smith [7], Han and Grosenbaugh [8], Bilbao *et al.* [9], Bilbao [10], Chatzioannou and Walstijn [11], and Taguti [12] used a finite difference method to study the string interaction with the obstacle. Issanchou *et al.* [13], van Walstijn and Bridges [14], and van Walstijn *et al.* [15] used a modal analysis approach. Vyasarayani *et al.* [1], Mandal and Wahi [16], and Singh and Wahi [17] described the movement of the sitar string with partial differential equations or sets of partial differential equations. Rank and Kubin [18], Evangelista and Eckerholm [19], and Siddiq [20] used a waveguide modelling approach to study the plucked string vibration with nonlinear limitation effects.

This paper proposes an idealised approach for modeling the nonlinear string–obstacle interaction. We consider lossless ideal string vibration and assume that the obstacle is absolutely rigid. The interaction between the vibrating string and the obstacle is modeled in terms of kinematics, i.e., we study the string motion without considering its mass nor the possible inertial and reactive forces acting between the string and the obstacle during the collisions. The heuristics of our approach is directly determined by the d'Alembert formula (travelling wave solution). The results presented here are a continuation of the work published in [21].

The organisation of the paper is as follows. In Sec. 2, the string vibration modeling is explained. In Sec. 3 the problem description and the kinematic numerical model of string–obstacle interaction is presented and explained. Section 4 presents the results and analysis of three case studies, where the effects of the obstacle geometry, proximity, and position are considered. In Sec. 5 the presented results and the accuracy and efficiency of the numerical model are briefly commented on. Section 6 presents the main results and conclusions.

2. MODELING STRING VIBRATION

Let us consider the vibration of lossless ideal string in a single vibration polarisation setting. The one-dimensional equation of motion, called the wave equation, is in the following form:

$$\frac{\partial^2 u}{\partial t^2} = c^2 \frac{\partial^2 u}{\partial x^2}, \quad (1)$$

where $u(x, t)$ is the transverse displacement of the string, $c = \sqrt{T/\mu}$ is the speed of the waves travelling on the string, where T is the tension and μ is the linear mass density of the string, having the dimension [kg/m]. In the context of a real string Eq. (1)

can be used as an approximation of thin homogeneous elastic (and lossless) string vibration under a small amplitude restriction. In this case wave speed $c = \sqrt{T/(\rho A_\circ)}$, where ρ is the volumetric density, $A_\circ = \pi r^2$ is the cross-section area of a cylindrical string, and T is the tension. Equation (1) is studied, here, in a normalised and dimensionless form for increased clarity. We normalise the fundamental frequency f_0 by setting $c = 1$ and by introducing the following dimensionless variables:

$$t = \frac{\mathcal{T}}{P}, \quad x = \frac{X}{\lambda}, \quad u = \frac{U}{\lambda}, \quad (2)$$

where P is the fundamental period and λ is the corresponding wavelength. The dimensional quantities \mathcal{T} , X , and U are the time, space, and string displacement, respectively. Additionally, the speaking length L of the string is chosen to be a half of the wavelength, i.e., $L = 1/2$ [d.u.] (dimensionless units). This ensures that the fundamental frequency $f_0 = 1$ because for Eq. (1) it holds that

$$f_0 = \frac{c}{2L} = \frac{c}{\lambda}. \quad (3)$$

In order to further simplify the frequency domain analysis presented in Sec. 4 the following initial condition is selected: at $t = 0$ the string is freely released and the initial displacement is selected to be equal to the shape of the fundamental mode

$$u_0(x) = u(x, 0) = A \sin 2\pi x, \quad x \in [0, L], \quad (4)$$

$$\frac{\partial}{\partial t} u(x, 0) = 0, \quad x \in [0, L], \quad (5)$$

where $A = 1$ is the amplitude. This selection results in a sinusoidal (standing wave) vibration of all string points with fundamental frequency $f_0 = 1$ determined by the normalised Eq. (1). Figure 1 shows the initial condition (4).

It is well known that the Eq. 1 has an analytical solution known as the d'Alembert formula. For infinite string (ignoring boundary conditions for now) and for initial conditions (4), (5) the solution takes the following form:

$$u(x, t) = \frac{1}{2} (u_0(x - ct) + u_0(x + ct)). \quad (6)$$

This solution represents a superposition of two travelling waves: $u_0(x - ct)/2$ moving to the right (positive direction of the x -axis), and $u_0(x + ct)/2$ moving to the left. The function $u_0/2$ describing the shape of these waves stays constant with respect to x -axis, as they are translated in opposite directions at speed $c = 1$.

In the general case and under our assumptions a wave on any arbitrary segment of the string can be understood as a sum of two travelling waves that do not need to be equal. This means that one can write

$$u(x, t) = r(x - ct) + l(x + ct), \quad (7)$$

where $r(x - ct)$ is the travelling wave moving to the right and $l(x + ct)$ is the travelling wave moving to the left. Conveniently, one-dimensional advection equations

$$\frac{\partial r}{\partial t} + c \frac{\partial r}{\partial x} = 0, \quad (8)$$

$$\frac{\partial l}{\partial t} - c \frac{\partial l}{\partial x} = 0, \quad (9)$$

have similar general solutions. The travelling wave $r(x - ct)$, on its own, is also a solution to Eq. (8) and the travelling wave $l(x + ct)$

is a solution to Eq. (9). This result is not surprising because Eq. (1) can be factored into

$$\left[\frac{\partial}{\partial t} - c \frac{\partial}{\partial x} \right] \left[\frac{\partial}{\partial t} + c \frac{\partial}{\partial x} \right] u = 0. \quad (10)$$

Advection Eqs (8) and (9) are used below to numerically model the propagation of travelling waves. The finite difference method is used to approximate the solutions of these equations.

We discretise xt -plane into $n \times m$ discrete samples using a grid with equal step sizes in x and t directions. We discretise the x -axis with grid spacing $\Delta x = L/n$ and the t -axis with grid spacing $\Delta t = \Delta x = t_{\max}/m$, where $m = 2n$. We let $x_i = i\Delta x$, where $0 \leq i \leq n$ and $t^j = j\Delta t$, where $0 \leq j \leq m$. From here it follows that $u_i^j = u(x_i, t^j)$, $r_i^j = r(x_i, t^j)$, and $l_i^j = l(x_i, t^j)$. A combination of step forward finite difference (FD) approximations of first order derivatives

$$\frac{\partial u}{\partial t} \approx \frac{u_i^{j+1} - u_i^j}{\Delta t}, \quad \frac{\partial u}{\partial x} \approx \frac{u_{i+1}^j - u_i^j}{\Delta x}, \quad (11)$$

and step backwards FD approximations

$$\frac{\partial u}{\partial t} \approx \frac{u_i^j - u_i^{j-1}}{\Delta t}, \quad \frac{\partial u}{\partial x} \approx \frac{u_i^j - u_{i-1}^j}{\Delta x}, \quad (12)$$

are used to approximate Eqs (8) and (9). Resulting FD approximations are in the following form:

$$r_i^{j+1} - r_i^j + c \frac{\Delta t}{\Delta x} (r_i^j - r_{i-1}^j) = 0, \quad (13)$$

$$l_i^{j+1} - l_i^j - c \frac{\Delta t}{\Delta x} (l_{i+1}^j - l_i^j) = 0. \quad (14)$$

Because $c = 1$ and $\Delta x = \Delta t$, in our case, the Courant number $c\Delta t/\Delta x = 1$ and the above expressions are simplified

$$r_i^{j+1} = r_{i-1}^j, \quad (15)$$

$$l_i^{j+1} = l_{i+1}^j. \quad (16)$$

By applying these rules for all grid points i and j one gets a simple translation of numerical values r_i^j and l_i^j propagating in opposite directions with respect to the x -axis (i -axis). This result agrees with the d'Alembert formula (7) and can be understood as a digital waveguide based on travelling wave decomposition and use of delay lines. The equivalence between FD approximation used here and digital waveguide modeling is shown in [22].

So far we have not addressed the boundary conditions of Eq. (1). We assume that the string is fixed at both ends. The following boundary conditions apply:

$$u(0, t) = u(L, t) = 0, \quad t \in [0, t_{\max}], \quad (17)$$

where $t_{\max} = 10P$ is the maximum integration time (size of the temporal domain). By applying boundary conditions (17) to the general solution (7) of the initial equation one finds for $x = 0$ the reflected travelling wave in the following form:

$$u(0, t) = r(-ct) + l(ct) = 0 \Rightarrow r(-ct) = -l(ct), \quad (18)$$

and similarly for $x = L$:

$$\begin{aligned} u(L, t) &= r(L - ct) + l(L + ct) = 0 \Rightarrow \\ &\Rightarrow l(L + ct) = -r(L - ct). \end{aligned} \quad (19)$$

These results are discretised according to the FD discretisation scheme discussed above. The travelling wave (18) reflected from the left boundary at $x = 0$ is thus

$$r_0^j = -l_0^j, \quad j \in [0, m], \quad (20)$$

and the travelling wave (19) reflected from the right boundary at $x = L$ is

$$l_n^j = -r_n^j, \quad j \in [0, m]. \quad (21)$$

In order to obtain the resulting string displacement u_i^j , for the selected initial and boundary conditions, a superposition of travelling waves (15), (16), (20), and (21) is found in accordance with general solution (7)

$$u_i^j = r_i^j + l_i^j, \quad \{i, j\} \in [0, n] \times [0, m]. \quad (22)$$

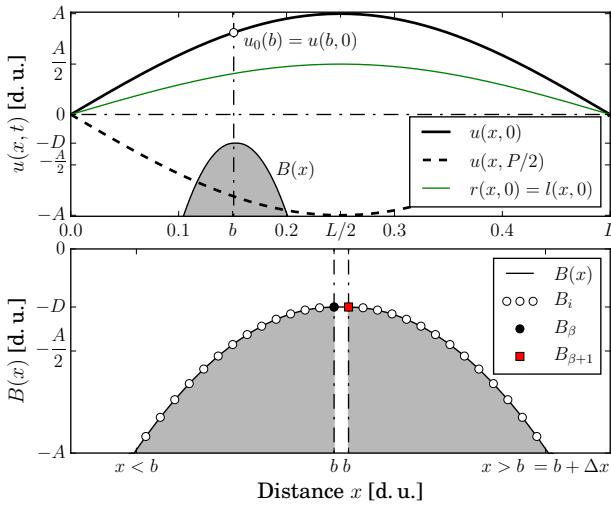


Figure 1: Top: Schematic of the problem studied. The initial condition (4) and the corresponding travelling waves are presented. The obstacle is shown with the grey formation at $x = b$. The resting position of the string is shown with the horizontal dash-dotted line. Bottom: The cross-section profile $B(x)$ of the rigid parabolic obstacle and its discrete samples B_i , where $B_\beta = B_{\beta+1} = -D$.

3. STRING–OBSTACLE INTERACTION KINEMATICS

Let us consider a smooth and absolutely rigid impenetrable obstacle. The obstacle is placed near the vibrating string so that it is able to obscure string's displacement $u(x, t)$. We select an obstacle with parabolic cross-section profile

$$B(x) = - \left[\frac{(x - b)^2}{2R} + D \right], \quad (23)$$

where $x = b$ is the position of the obstacle along the string, D is the vertical proximity of the obstacle to the string at its rest position, and R is the curvature radius of parabola at its apex. The function $B(x)$ is discretised according to the FD discretisation scheme presented in the previous Section. We let $B_i = B(x_i)$. Figure 1 shows the schematic drawing of the problem studied, the

cross-section profile function $B(x)$ of the obstacle, and its discretised samples B_i .

The kinematic modeling of the string–obstacle interaction is a twofold problem. First, one needs to consider travelling wave $r(x - ct)$ approaching the obstacle from the left side. Second, one considers the travelling wave $l(x + ct)$ approaching the obstacle from the right side.

3.1. Reflection of travelling wave $r(x - ct)$

The heuristics of the following approach is strictly determined by the d'Alembert formula (6) or (7). Any change in string displacement $u(x, t)$ imposed by the obstacle must involve both travelling waves. The reflection of the travelling wave $r(x - ct)$ approaching the obstacle from the left is determined by a change in the travelling wave $l(x + ct)$. We assume that the travelling wave $l(x + ct)$ resulting from the interaction first appears, or already existing one is modified, only at the point $x = x^* \leq b$, where the amplitude of string displacement $u(x^*, t) < B(x^*)$, i.e., the string attempts to penetrate the obstacle. The position of this point x^* is determined by the obstacle profile geometry $B(x)$. Because, the obstacle is in fact impenetrable we must have $u(x^*, t) = B(x^*)$ at the moment of collision. This condition determines the shape of a *reflected* travelling wave

$$\hat{l}(x^*, t) = B(x^*) - u(x^*, t). \quad (24)$$

Once one has determined the shape of the *reflected* travelling wave (24) for given time moment. It is used inside the same time moment as a travelling wave $l(x + ct)$ or to modify the already existing travelling wave moving to the left in the following manner:

$$l(x^*, t) = l(x^*, t) + \hat{l}(x^*, t). \quad (25)$$

The above modification of the travelling wave $l(x + ct)$ simply ensures that the resulting string displacement $u(x^*, t) = r(x^*, t) + l(x^*, t)$ does not geometrically penetrate the obstacle during the collision.

In determining the points x^* one does not need to consider points $x > b$. In the absence of any waves approaching from the right and for $x^* = b$ the string displacement $u(b, t) = -D$ (caused by the reflection process explained in the next Subsection). This means that the string displacement $u(x, t)$ becomes truncated and equal to the maximum value of the obstacle as the propagating wave passes over its apex. This phenomenon is shown in Fig. 2 in the case of a short wavelength bell-shaped pulse reflecting from the obstacle during the first period of string vibration. Because the obstacle profile is an inverted parabola with its maximum at $x = b$ then for $x > b$ it holds that $u(x, t) > B(x) \Rightarrow x \neq x^*$.

Numerical implementation of the procedure is straightforward. We let $x_\beta = \beta \Delta x = b$ (see Fig. 1) and $x_{i^*} = i^* \Delta x = x^*$. Using this notation the *reflected* travelling wave (24) takes the following form:

$$\hat{l}_{i^*}^j = B_{i^*} - u_{i^*}^j, \quad i \in [0, \beta]. \quad (26)$$

The resulting reflection and consequent string shape according to (22) and (25) for grid points i^* becomes

$$u_{i^*}^j = r_{i^*}^j + l_{i^*}^j, \quad i \in [0, \beta], \quad (27)$$

where

$$l_{i^*}^j = l_{i^*}^j + \hat{l}_{i^*}^j, \quad i \in [0, \beta]. \quad (28)$$

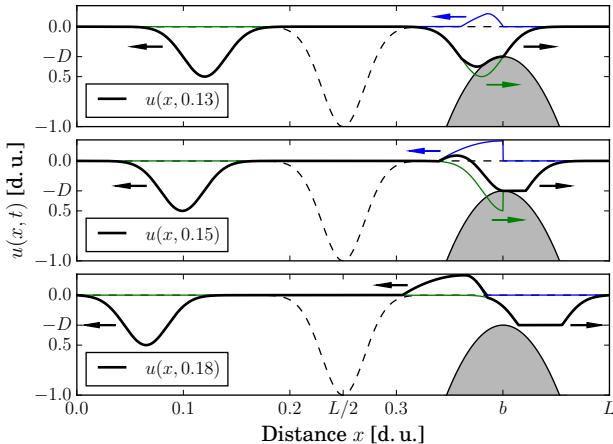


Figure 2: Reflection of travelling wave $r(x - ct)$, shown with the thin green line, from the obstacle that is shown with the grey formation. The reflected travelling wave $\hat{r}(x + ct)$ is shown with the thin blue line. Arrows indicate the directions of wave propagation. Bell-shaped initial condition is shown with the dashed line.

3.2. Reflection of travelling wave $l(x + ct)$

The determination of the reflection of travelling wave $l(x + ct)$ approaching the obstacle from the right side is similar to the previous case. In fact, it is a mirror image of that problem with symmetry axis at $x = b$. However, there is a slight difference in the region where it is necessary to evaluate and determine the points $x = x^*$. This difference stems from the selection and mathematical definition of the obstacle cross-section profile (23), namely, the function $B(x)$, being a unimodal function, has one maximum $\max B(x) = -D$ at $x = b$. The problem arises from the fact that one has already evaluated this point inside the same time moment t and used it to calculate the reflection of travelling wave $r(x - ct)$. By using this point again one would introduce a discontinuity in the string displacement $u(x, t)$. In principle it is not possible to use some other closely located neighbouring point, e.g., $x = b' = b + \Delta x$ (see Fig. 1). This selection, too, would introduce a small discontinuity due to $B(b) \neq B(b')$, and more importantly, in a continuous domain of real numbers there is no such thing as a neighbouring number (point) because one can always find infinitely many numbers between any two real numbers. One way of resolving this problem is to slightly modify the discretised approximation of the obstacle profile B_i as shown in Fig. 1. One simply squeezes in a second maximum point $B_{\beta+1} = -D$ after grid point $i = \beta$ corresponding to $x_\beta = \beta\Delta x = b$. Because $\Delta x \ll 1$ the overall change introduced in B_i compared to $B(x)$ remains negligibly small.

Let us formalised the result. Similarly to the previous case only at the points $x = x^* \geq b' \simeq b$, where by definition and during the collision $u(x^*, t) < B(x^*)$, a reflected travelling wave moving to the right is introduced or existing one is modified

$$r(x^*, t) = r(x^*, t) + \hat{r}(x^*, t), \quad (29)$$

where

$$\hat{r}(x^*, t) = B(x^*) - u(x^*, t). \quad (30)$$

Numerical implementation, using the notation proposed in the previous Subsection, takes the following form: the resulting reflec-

tion and consequent string shape according to (22), (29), and (30) for grid points i^* is

$$u_{i^*}^j = r_{i^*}^j + l_{i^*}^j, \quad i \in (\beta, n], \quad (31)$$

where

$$r_{i^*}^j = r_{i^*}^j + \hat{r}_{i^*}^j \text{ and } \hat{r}_{i^*}^j = B_{i^*} - u_{i^*}^j, \quad \text{if } i \in (\beta, n], \quad (32)$$

4. RESULTS

Next, three case studies are considered. The effects of varying the values of the obstacle radius R (Sec. 4.1), obstacle proximity D (Sec. 4.2), and position b (Sec. 4.3), while keeping other system parameters constant, on the string vibration are analysed. Table 1 displays the values of parameters used in the case studies.

Table 1: Values of the parameters used in the case studies. The parameter that is being varied is indicated with the grey background.

Sec.	Radius R [d.u.]	Proximity D [d.u.]	Position b [d.u.]
4.1	$1 \cdot 10^{-5}$ $6 \cdot 10^{-3}$	$0.5 u_0(b) = 0.29$	$L/5 = 1/10$
4.2	$3 \cdot 10^{-3}$	$0.3 u_0(b) = 0.26$ $0.1 u_0(b) = 0.09$	$L/3 = 1/6$
4.3	$4 \cdot 10^{-3}$	$0.35 u_0(b) = 0.30$ $0.35 u_0(b) = 0.21$	$L/3 = 1/6$ $L/5 = 1/10$

The proximity D of the obstacle to the string at its rest position is expressed in terms of the initial condition (4). Namely, the string displacement $u_0(x) = u(x, 0)$ at obstacle location $x = b$ (see Fig. 1). This is done in order to make the results comparable to each other. All frequency domain result are calculated using time series of the string displacement $u(x, t)$ recorded at $x = x_r = 0.47L = 0.235$. This point is close to a node at $x = L/2$ shared by all even numbered modes (harmonics or overtones). The reader must keep in mind that this selection results in relatively small amplitude values of even numbered modes in the spectra presented below. On the other hand, this selection ensures that the amplitude of fundamental mode is nearly unity for the linear case where the obstacle is absent. This, in turn, will aid in drawing the conclusions. The spectrograms of power spectra, amplitude spectra, spectral centroid $\langle f \rangle$, and instantaneous spectral centroid $\langle f' \rangle(t)$ are calculated using fast Fourier transform algorithm which is based on the Fourier transform. In calculating spectrograms a sliding window approach, in combination with the Hanning window function, is used. Here, window size $t = 3P$ and window overlap value is 50% of the window size. Instantaneous spectral centroid $\langle f' \rangle(t)$ is also calculated using the windowing approach with window size $t = P$ and with window overlap value 70% of the window size. Video animations of the string vibrations for all case studies presented below can be downloaded at the accompanying web-page of this paper¹. The web-page also includes an additional example of a symmetric case where the obstacle is positioned at the midpoint of the string at $b = L/2$.

4.1. Effect of the obstacle radius

Figure 3 shows the influence of varying the value of obstacle radius R on the string shape $u(x, t)$ for the duration of the first period

¹<http://www.ioc.ee/~dima/dafx17.html>

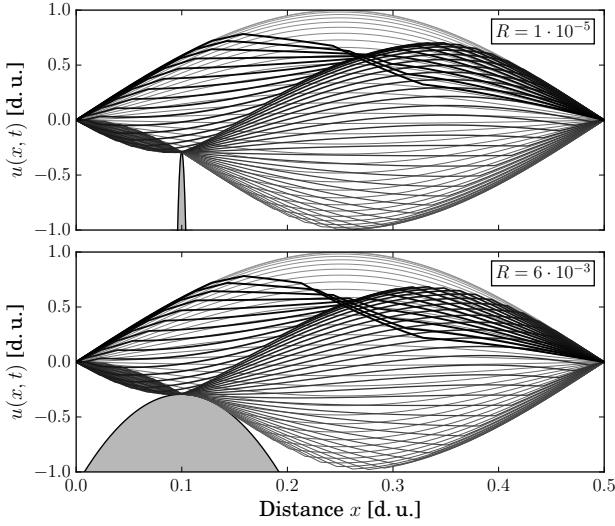


Figure 3: Stroboscopic plot of the string displacement during the first period of vibration where $t \in [0, P]$. Showing 68 time steps. The thickness of the lines is proportional to the direction of time flow. The obstacles with radius $R = 1 \cdot 10^{-5}$ (top) and $R = 6 \cdot 10^{-3}$ (bottom) are positioned at $b = L/5$ and $D = u_0(b)/2$.

only. The results of frequency domain analysis of the resulting vibration are shown in Fig. 4. Visual inspection of the time series $u(x_r, t)$ reveals that the string vibrates in two distinct vibration regimes (generally true for all presented case studies). The initial strong influence of the obstacle is manifested in the constantly evolving spectrum that prolongs for $t = t_p$. We call this regime the *aperiodic* regime. After $t = t_p$ the string vibration settles in the *periodic* regime, where the spectrum remains constant. Time moment t_p corresponds to the latest time instance where point x^* is determined.

During the aperiodic regime the value of instantaneous spectral centroid grows with the time (generally true for all presented case studies) resulting in the value of spectral centroid that is one octave (two times) greater compared to the linear case where the obstacle is absent (in linear case $\langle f' \rangle(\infty) = \langle f \rangle = f_0 = 1$). The growth of the value of spectral centroid is driven by nonlinear widening of the spectrum caused by transfer of the fundamental mode energy to higher modes. For both presented cases and according to the amplitude spectra, shown in Fig. 4, approximately 65% of initial fundamental mode amplitude $A = 1$ is redistributed to and between the higher modes. Mutual comparison of the cases shows that the case with larger radius R results in approximately one-third shorter-lasting aperiodic regime and slightly higher value of spectral centroid. A relatively large change in the obstacle radius R has a moderate effect (compared to the other case studies) on the final string vibration—at least for the given parameter values. This is best seen in roughly equal amplitude spectra.

4.2. Effect of the obstacle proximity

Figure 5 shows the influence of varying the value of the obstacle proximity D on the string shape $u(x, t)$ for the duration of the first period only. The results of frequency domain analysis of the vibration are shown in Fig. 6. Inspection of the aperiodic regime

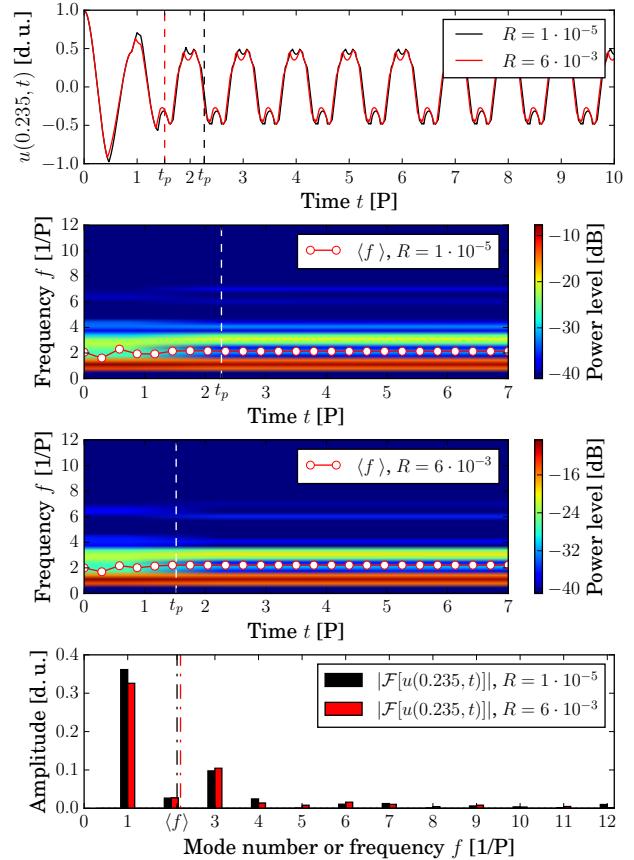


Figure 4: Top: Time series $u(x_r, t)$ shown for two values of the obstacle radius R . Onset time t_p of the periodic regime shown with the colour-coded dashed line. Middle: Power spectrograms. Instantaneous spectral centroid $\langle f' \rangle$ is shown with the solid red line marked with bullets. Onset time t_p of the periodic regime is shown with the dashed white line. Bottom: Amplitude spectra in the periodic regimes where $t \in [[t_p], t_{max}]$. Spectral centroid $\langle f \rangle$ is shown with the colour-coded dash-dotted line.

of both cases shows that a short-lasting large amplitude fractional subharmonic at $f = 1.5$ is generated. This partial does not survive the aperiodic regime. Based on the relationship (3) one can conclude that this frequency is related to the fact that the obstacle is promoting a node at $b = L/3$ by effectively dividing the string into two vibrating segments with lengths $L/3$ and $2L/3$.

Comparison of the periodic regime vibration to the linear case shows that the resulting value of spectral centroid is increased by almost four to five octaves depending on the case. As in the previous case study the string–obstacle interaction has widened the spectrum at the expense of the fundamental mode. The resulting fundamental mode amplitude, for both cases, is approximately 0.02. This means that during the aperiodic regime more than 97% of initial mode amplitude $A = 1$ is redistributed to higher modes. The greatest relative and absolute change in mode amplitude, caused by the reduction of the distance D between the string and the obstacle, is seen for the third and tenth modes. The amplitude of third mode is grown two times from approximately 0.02 to approximately 0.04. The amplification of the third mode is

related to the obstacle position b discussed above. Mutual comparison of the presented cases shows that the case where the obstacle is closer to the string results in approximately three times longer-lasting aperiodic regime and in the value of spectral centroid that is greater by an octave.

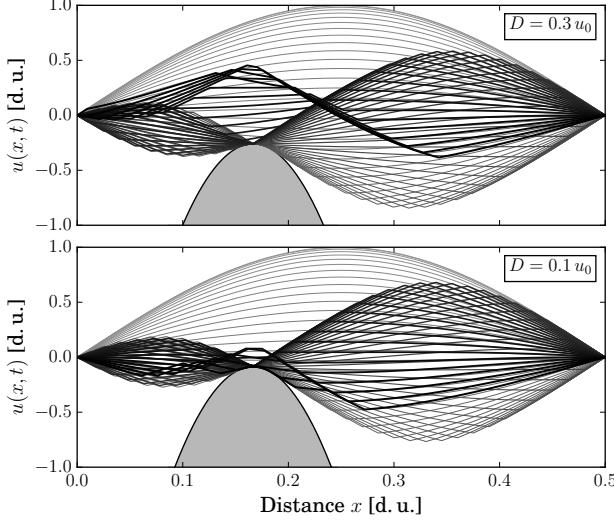


Figure 5: Stroboscopic plot of the string displacement during the first period of vibration where $t \in [0, P]$. Showing 68 time steps. The thickness of the lines is proportional to the direction of time flow. The obstacle with radius $R = 3 \cdot 10^{-3}$ is positioned at $b = L/3$. The obstacle with proximity $D = 0.3 u_0(b)$ (top) and $D = 0.1 u_0(b)$ (bottom) are presented.

4.3. Effect of the obstacle position along the string

Figure 7 shows the influence of varying the value of the obstacle position b on the string shape $u(x, t)$ for the duration of the first period only. The results of frequency domain analysis of the vibration are shown in Fig. 8. Inspection of the aperiodic regime shows, similarly to the previous case study, that short-lasting subharmonic at $f = 1.5$ and at $f = 1.25$ are generated for the cases where $b = L/3$ and $b = L/5$, respectively. The cause for these large amplitude partials is the same as discussed in the previous case study.

Comparison of the periodic regime to the linear case shows that the resulting value of spectral centroid has increased by approximately three octaves, in both cases. In contrast to the previous case study the inspection of the spectra in the periodic regime shows no significant amplification of third and fifth modes related to the obstacle positions b . In fact we see a significant reduction of the fifth mode as we move the obstacle closer to the string edge ($x = 0$). In addition to the nonlinearity, the absence of these modes may be explained by obstacle's ability (related to the profile geometry) to elongate the segments of travelling waves that kinematically reflect from it, see Fig. 2 [21]. If this is the case the obstacle is redistributing the energy of these modes to neighbouring lower modes. Also, the amplitudes of these expected modes may be masked by a process of constant and progressive trimming of the amplitudes of travelling waves during the aperiodic regime discussed and shown in Sec. 3.1 and Fig. 2, respectively. This result is to demonstrate that, when dealing with nonlinear systems,

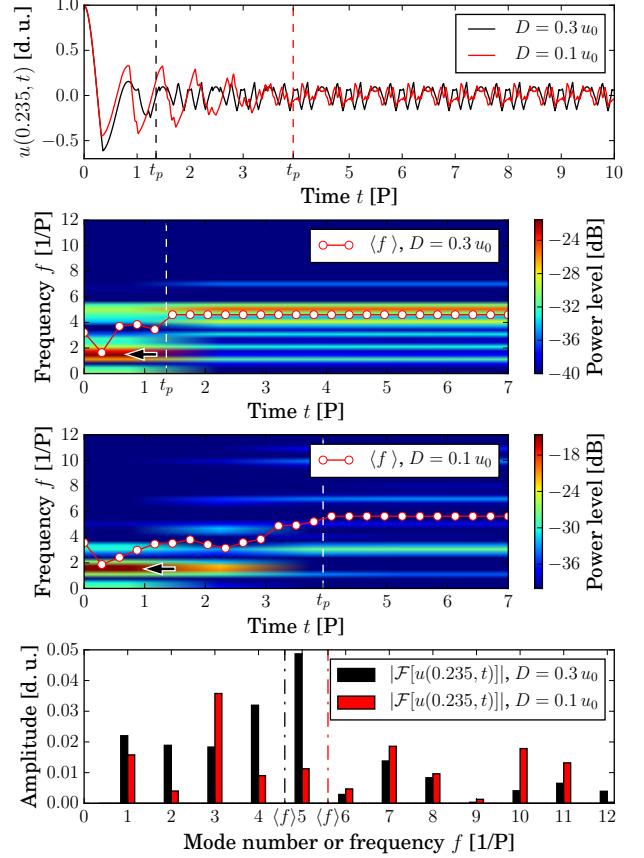


Figure 6: Top: Time series $u(x_r, t)$ shown for two values of the obstacle proximity D . Onset time t_p of the periodic regime shown with the colour-coded dashed line. Middle: Power spectrograms. Instantaneous spectral centroid $\langle f' \rangle$ is shown with the solid red line marked with bullets. Onset time t_p of the periodic regime is shown with the dashed white line. Subharmonic at $f = 1.5$ is shown with the bold arrow. Bottom: Amplitude spectra in the periodic regime where $t \in [[t_p], t_{max}]$. Spectral centroid $\langle f \rangle$ is shown with the colour-coded dash-dotted line.

each problem needs to be studied on a case by case basis.

Mutual comparison of presented cases shows that the final value of spectral centroid differs by half of an octave. The case where the obstacle is positioned closer to the string edge results in approximately two times longer-lasting aperiodic regime. The resulting fundamental mode amplitude is approximately 0.05 for $b = L/3$ and 0.1 for $b = L/5$ meaning that during the aperiodic regime approximately 90 to 95% of initial mode amplitude $A = 1$ is redistributed between the higher modes.

5. DISCUSSION

The method of modeling the string–obstacle interaction presented in this paper is probably the most simplified and idealised approach that is still able to retain scientific relevance and provide a useful insight into more realistic problems. The idealised nature of the method guarantees numerical accuracy and allows for efficient and simple implementation. This fact cannot be ignored when dealing

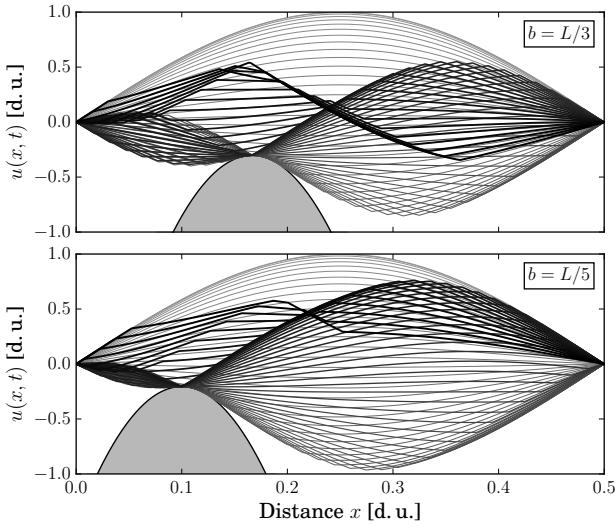


Figure 7: Stroboscopic plot of the string displacement during the first period of vibration where $t \in [0, P]$. Showing 68 time steps. The thickness of the lines is proportional to the direction of time flow. The obstacle with radius $R = 4 \cdot 10^{-3}$ is positioned at $b = L/3$ (top) and $b = L/5$ (bottom), and in both cases $D = 0.35 u_0(b)$.

with nonlinear problems of this type. Physically more comprehensive and thus more realistic mathematical descriptions of string–obstacle collision problems often result in rather complicated nonlinear equations of motion or in systems of equations. Modeling of these problems then rely on numerical integration of these equations. This in itself can pose a great technical challenge and the obtained solutions may be contaminated by numerical dispersion and other accumulative round-off or approximation errors inherent in most iterative numerical methods. For example in [10, 14] the iterative solvers are avoided altogether when solving the problem of lossy stiff string vibration against an obstacle.

The numerical accuracy of the proposed approach is mainly a result of an adept FD discretisation of the problem domain (xt -plane), discussed in Sec. 2, which ensures that at no iteration step does the method rely on multiplication or division of rounded-off numerical values. This guarantees that the resulting numerical solution lacks any accumulative approximation errors and the round-off errors are minimal. Travelling waves are modeled as translations of numerical values along the iteration axes, according to (15) and (16). The string interaction with the obstacle is calculated, according to (24) and (30), using only subtraction or addition operations. In fact, assuming perfect arithmetics, i.e., assuming no round-off, one could predict the motion of a string, undergoing a kinematic string–obstacle interaction, for infinite number of iterations without losing any accuracy.

The proposed approach can be applied for estimating the effect that an obstacle with various cross-section profiles has on a string vibration. In principle the function $B(x)$ can be chosen arbitrarily as long as one remembers to modify its FD discretisation at maximum or local maxima (in the case of more general multimodal and/or discontinuous functions) as explained in Sec. 3.2. Additionally, the model can be used to study the effects of different initial and plucking conditions (including dynamic ones) on the

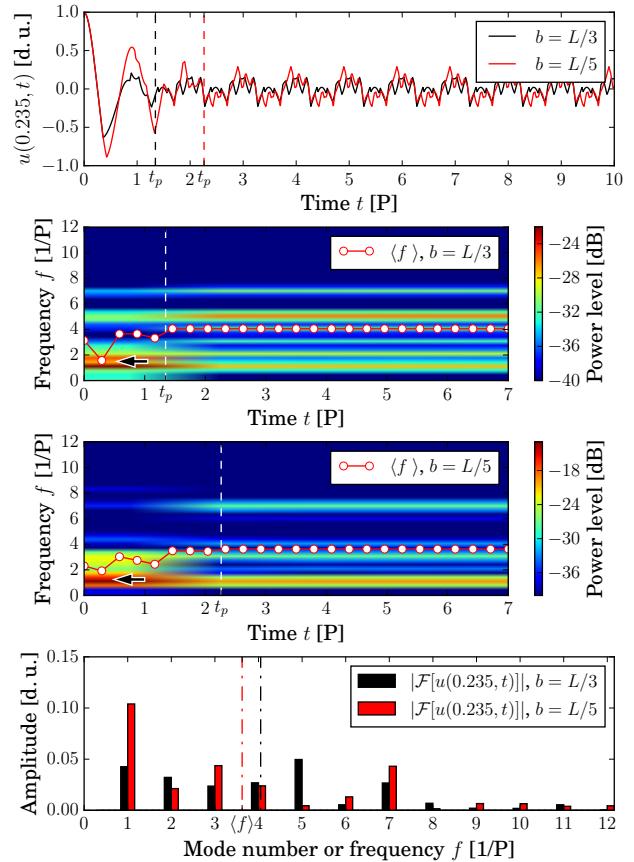


Figure 8: Top: Time series $u(x_r, t)$ shown for two values of obstacle position b . Onset time t_p of the periodic regime shown with the colour-coded dashed line. Middle: Power spectrograms. Instantaneous spectral centroid $\langle f' \rangle$ is shown with the solid red line marked with bullets. Onset time t_p of the periodic regime shown with the dashed white line. Subharmonics at $f = 1.5$ and $f = 1.25$ are shown with the bold arrows. Bottom: Amplitude spectra in the periodic regime where $t \in [t_p, t_{max}]$. Spectral centroid $\langle f' \rangle$ is shown with the colour-coded dash-dotted line.

string–obstacle system. It is reasonable to assume that proposed model can generate these results faster compared to more sophisticated models, which is often desired in real-time simulation applications. In connection with other methods of signal processing and sound synthesis the presented model can be used to synthesise timbres of stringed instruments that are equipped with nonlinearity inducing obstacles, such as, frets, bridges, and nuts.

6. CONCLUSIONS

In this paper the kinematics of ideal string vibration against an absolutely rigid obstacle was modeled using the approach based on the application of the d’Alembert formula as explained in Sec. 3. The presented numerical method is accurate and efficient lacking numerical dispersion caused by accumulative approximation errors. The effect of the obstacle proximity D on the string vibration and on the mean level of upper mode amplitudes was clearly evi-

dent and this was to confirm that the problem is nonlinear.

It was shown that the ideal lossless string interacting with the obstacle vibrates in two distinct vibration regimes. In the beginning of the kinematic interaction between the vibrating string and the obstacle the string motion is *aperiodic* with constantly evolving spectrum. After some time of aperiodic vibration the string vibration settles in the *periodic* regime where the string motion is repetitious in time. The duration of the aperiodic regime depends on the obstacle proximity D , position b , and geometry (curvature radius R). The comparison of the resulting spectra in the periodic regime with the linear case where the obstacle was missing showed that the general effect of the obstacle manifests in the widening of the spectrum caused by transfer of fundamental mode energy to upper modes. The analysis of the relatively short-lasting aperiodic regime showed that the obstacle position b may generate temporary fractional subharmonics related to the node point at $x = b$. In conclusion, the results presented in this paper can expand our understanding of timbre evolution of numerous stringed instruments, such as, the guitar, bray harp, tambura, veena, sitar, etc. The possible applications include, e.g., sound synthesis of these instruments.

7. ACKNOWLEDGMENTS

This research was supported by the Estonian Ministry of Education and Research, Project IUT33-24, and by Doctoral Studies and Internationalisation Programme DoRa Plus Action 1.1 (Archimedes Foundation, Estonia) through the ERDF. The author is grateful to the anonymous reviewers of this paper—their suggestions improved the manuscript greatly.

8. REFERENCES

- [1] C. P. Vyasarayani, S. Birkett, and J. McPhee, “Modeling the dynamics of a vibrating string with a finite distributed unilateral constraint: Application to the sitar,” *The Journal of the Acoustical Society of America*, vol. 125, no. 6, pp. 3673–3682, 2009.
- [2] C. V. Raman, “On some Indian stringed instruments,” *Proceedings of the Indian Association for the Cultivation of Science*, vol. 7, pp. 29–33, 1921.
- [3] M. Schatzman, “A hyperbolic problem of second order unilateral constraints: the vibrating string with a concave obstacle,” *Journal of Mathematical Analysis and Applications*, vol. 73, no. 1, pp. 138–191, 1980.
- [4] H. Cabannes, “Presentation of software for movies of vibrating strings with obstacles,” *Applied Mathematics Letters*, vol. 10, no. 5, pp. 79–84, 1997.
- [5] R. Burridge, J. Kappraff, and C. Morshedi, “The sitar string, a vibrating string with a one-sided inelastic constraint,” *SIAM Journal on Applied Mathematics*, vol. 42, no. 6, pp. 1231–1251, 1982.
- [6] M. Ducceschi, S. Bilbao, and C. Desvages, “Modelling collisions of nonlinear strings against rigid barriers: Conservative finite difference schemes with application to sound synthesis,” in *Proc. of 22nd International Congress on Acoustics (ICA 2016)*, Buenos Aires, Argentina, Sept. 5–9 2016, pp. 1–11.
- [7] A. Krishnaswamy and J. O. Smith, “Methods for simulating string collisions with rigid spatial obstacles,” in *Proc. of IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*, New Paltz, NY, USA, 2003, pp. 233–236.
- [8] S. M. Han and M. A. Grosenbaugh, “Non-linear free vibration of a cable against a straight obstacle,” *Journal of Sound and Vibration*, vol. 237, pp. 337–361, 2004.
- [9] S. Bilbao, A. Torin, and V. Chatzioannou, “Numerical modeling of collisions in musical instruments,” *Acta Acustica United With Acustica*, vol. 101, no. 1, pp. 155–173, 2012.
- [10] S. Bilbao, “Numerical modeling of string/barrier collisions,” in *Proc. of International Symposium on Musical Acoustics (ISMA 2014)*, Le Mans, France, Jul. 7–12 2014, pp. 1–6.
- [11] V. Chatzioannou and M. van Walstijn, “Energy conserving schemes for the simulation of musical instrument contact dynamics,” *Journal of Sound and Vibration*, vol. 339, pp. 262–279, March 2015.
- [12] T. Taguti, “Dynamics of simple string subject to unilateral constraint: A model analysis of sawari mechanism,” *Acoustical Science and Technology*, vol. 29, no. 3, pp. 203–214, 2008.
- [13] C. Issanchou, S. Bilbao, J.-L. Le Carrou, C. Touzé, and O. Doaré, “A modal-based approach to the nonlinear vibration of strings against a unilateral obstacle: Simulations and experiments in the pointwise case,” *Journal of Sound and Vibration*, vol. 393, pp. 229–251, April 2017.
- [14] M. van Walstijn and J. Bridges, “Simulation of distributed contact in string instruments: a modal expansion approach,” in *Proc. of 24th European Signal Processing Conference (EUSIPCO-2016)*, Budapest, Hungary, Aug. 29–Sept. 2 2016, pp. 1–5.
- [15] M. van Walstijn, J. Bridges, and S. Mehes, “A real-time synthesis oriented tanpura model,” in *Proc. of 19th International Conference on Digital Audio Effects (DAFx-16)*, Brno, Czech Republic, Sept. 5–9 2016, pp. 175–182.
- [16] A. K. Mandal and P. Wahi, “Natural frequencies, mode-shapes and modal interactions for strings vibrating against an obstacle: Relevance to sitar and veena,” *Journal of Sound and Vibration*, vol. 338, pp. 42–59, March 2015.
- [17] H. Singh and P. Wahi, “Non-planar vibrations of a string in the presence of a boundary obstacle,” *Journal of Sound and Vibration*, vol. 389, pp. 326–349, February 2017.
- [18] E. Rank and G. Kubin, “A waveguide model for slapbass synthesis,” in *Proc. of IEEE International Conference on Acoustics, Speech, and Signal Processing*, Munich, Germany, 1997, pp. 443–446.
- [19] G. Evangelista and F. Eckerholm, “Player-instrument interaction models for digital waveguide synthesis of guitar: touch and collisions,” *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 18, no. 4, pp. 822–832, 2010.
- [20] S. Siddiq, “A physical model of the nonlinear sitar string,” *Archives of Acoustics*, vol. 37, no. 1, pp. 73–79, 2012.
- [21] D. Kartofelev, A. Stulov, H.-M. Lehtonen, and V. Välimäki, “Modeling a vibrating string terminated against a bridge with arbitrary geometry,” in *Proc. of 4th Stockholm Music Acoustics Conference (SMAC 2013)*, Stockholm, Sweden, Jul. 30–Aug. 3 2013, pp. 626–632.
- [22] J. O. Smith, *Physical Audio Signal Processing for Virtual Musical Instruments and Audio Effects*, W3K Publishing, 2010.

DOPPLER EFFECT OF A PLANAR VIBRATING PISTON: STRONG SOLUTION, SERIES EXPANSION AND SIMULATION

Tristan Lebrun

S3AM team, STMS, IRCAM-CNRS 9912-UPMC
1 place Igor Stravinsky, 75004 Paris, France
tristan.lebrun@ircam.fr

Thomas Hélie

S3AM team, STMS, IRCAM-CNRS 9912-UPMC
1 place Igor Stravinsky, 75004 Paris, France
thomas.helie@ircam.fr

ABSTRACT

This article addresses the Doppler effect of a planar vibrating piston in a duct, as a plane wave radiation approximation generated by a loudspeaker membrane. This physical model corresponds to a nonlinear problem, because the linear propagation is excited by a moving boundary condition at the piston face: this introduces a varying propagation time between the piston and a fixed receiver. The existence of a regular function that solves the problem (a so-called “strong” solution) is proven, under a well-posed condition that guarantees that no shock occurs. This function satisfies an implicit equation to be solved. An algorithm based on the perturbation method is proposed, from which an exact solution can be built using power series. The convergence of the power series is numerically checked on several examples. Simulations derived from a truncated power series provide sound examples with audible intermodulation and distortion effects for realistic loudspeaker excursion and speed ranges.

1. INTRODUCTION

The Doppler effect in loudspeakers is due to the membrane motion: this introduces a varying propagation time between the piston and a fixed receiver. For large frequency range speakers, this results in distortion effects. This phenomenon has been highlighted by Beers and Belar [1], who recommend the use of a multi-way system to reduce its influence.

In [2], Butterweck proposes a simplified 1D model based on plane wave propagation generated by a moving piston. He proposes to approximate the solution by a truncated series expansion, from which a criterion to evaluate the distortion is built.

This paper restates the 1D model introduced in [2] and investigates the well-posedness of the problem. A necessary and sufficient condition is presented for the existence of a regular solution, characterized by an implicit equation. This equation admits a unique solution, and its regularity order is proven to be related to that of the membrane displacement function. The proof relies on the method of characteristics. This so-called moving boundary problem has already been investigated, and analytical solutions have been established [3, 4]. In this paper, infinite regular displacement functions are considered, and the perturbation method proposed by [2] is adopted to derive the power series expansion in an exact recursive way. The series expansion corresponds to a Volterra series and its terms involve the partial Bell polynomials. Finally, simulations are carried out by truncation of the series expansion and both harmonic and intermodulation distortions are evaluated.

This paper is organized as follows. Section 2 introduces the physical model and the equations to solve. Then Section 3 presents the strong solution derived from the method of characteristics, and a

recursive algorithm based on power series expansion is described in Section 4. Finally the simulations are presented in Section 5.

2. PROBLEM STATEMENT

2.1. Description

Consider a semi-infinite duct excited by a vibrating planar piston (see Figure 1), which follows those four hypotheses:

- (H1) conservative linear acoustic plane waves propagation in an adiabatic homogeneous gas initially at rest,
- (H2) no wave coming from the right side of the duct,
- (H3) no shockwave propagates,
- (H4) piston position described by the function $t \mapsto \xi(t)$ at the left side of the duct, initially at rest ($\xi(t) = 0$ for $t \leq 0$).

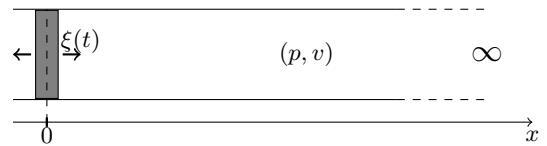


Figure 1: A piston vibrates around $x = 0$ in a semi-infinite duct.

2.2. Physical model

Following (H1), the wave propagation in the duct is described by

$$\rho_0 \partial_t v(x, t) + \partial_x p(x, t) = 0, \quad (1)$$

$$\partial_t p(x, t) + \rho_0 c_0^2 \partial_x v(x, t) = 0, \quad (2)$$

where p and v denote the acoustic pressure and the particle velocity, x and t are the space and time variables and ρ_0 and c_0 are the air density and the sound velocity, respectively.

The general solution (1-2) is decomposed into traveling waves as follows

$$\begin{aligned} v(x, t) &= v^+(t - x/c_0) - v^-(t + x/c_0), \\ p(x, t) &= \rho_0 c_0 (v^+(t - x/c_0) + v^-(t + x/c_0)), \end{aligned}$$

where functions $t \mapsto v^\pm(t)$ represent forward and backward waveforms.

Condition **(H2)** implies that there is no backward wave ($v^- = 0$), so that

$$v(x, t) = v^+(t - x/c_0), \quad (3)$$

$$p(x, t) = \rho_0 c_0 v^+(t - x/c_0), \quad (4)$$

only propagates from left to right.

Condition **(H3)** implies that v^+ must be a regular function. In the following, this class of solutions is called the class of “strong solutions” (contrary to the case of weak solutions that can admit jumps or singularities).

Finally, because of **(H4)**, the particle velocity at position ξ is the piston velocity ξ' . This means that the waves are driven by the piston face according to

$$v(\xi(t), t) = \xi'(t), \quad (5)$$

and that they propagate from left to right in the space-time domain

$$\mathbb{K}_\xi = \{(x, t) \in \mathbb{R}^2 \text{ s.t. } x \geq \xi(t)\}, \quad (6)$$

according to (3-4).

The problem described by **(H1-H4)** can be restated and reduced to the following question: given a piston motion function $t \mapsto \xi(t)$, does there exist a regular waveform function $t \mapsto v^+(t)$ such that

$$v^+(t - \xi(t)/c_0) = \xi'(t), \quad (7)$$

and such that, on domain (6), (3-4) is a solution of (1-2) ?

3. EXISTENCE OF STRONG SOLUTIONS

This section addresses the existence of regular solutions based on the method of characteristics. It provides a necessary and sufficient condition on C^1 -regular functions ξ .

In the following, the exponent “ a ” denotes quantities related to the arrival time of the acoustic wave at the observer point, whereas “ d ” denotes the departure time of the wave from the piston face.

Definition 1 (Characteristic) Let us define the regular functions

$$\begin{aligned} \tau_\xi^a : \quad \mathbb{K}_\xi &\mapsto \mathbb{R} \\ (x, t) &\mapsto t + \frac{x - \xi(t)}{c_0}, \end{aligned} \quad (8)$$

and

$$\begin{aligned} K_\xi : \quad \mathbb{K}_\xi &\mapsto \mathbb{K}_\xi^a \\ (x, t) &\mapsto (x, \tau_\xi^a(x, t)) \end{aligned} \quad (9)$$

where the image set \mathbb{K}_ξ^a is

$$\mathbb{K}_\xi^a = \{(x, \tau_\xi^a(x, t)) \text{ for } (x, t) \in \mathbb{K}_\xi\} \quad (10)$$

In this definition (see Figure 2), $T = \tau_\xi^a(x = X, t = \theta)$ provides the arrival time $t = T$ at position $x = X$ of an acoustic wave emitted at time $t = \theta$ and position $x = \xi(\theta)$, according to (3). Indeed, the particle velocity v at $(x, t) = (X, T)$ is

$$v^+(\tau_\xi^a(X, \theta) - X/c_0) = v^+(\theta - \xi(\theta)/c_0),$$

which corresponds to the particle velocity at $(x, t) = (\xi(\theta), \theta)$.

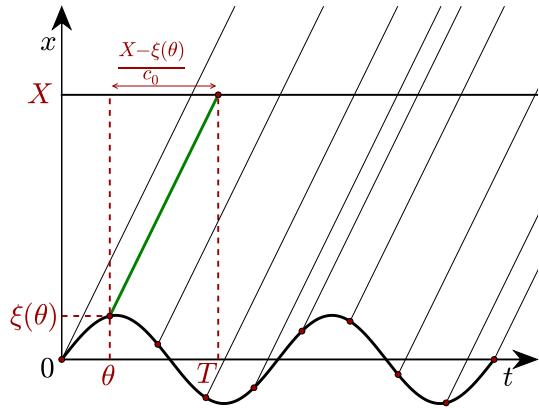


Figure 2: Illustration of the characteristics for a sinusoidal piston motion: the lines $L_\xi(t) = \{K_\xi(x, t), x \geq \xi(t)\}$ are the space-time locus on which the acoustic waves are constant. This figure details the case for the wave emitted at $(x, t) = (\xi(\theta), \theta)$ and arriving at $(x, t) = K_\xi(X, \theta) = (X, T)$, with $T = \tau_\xi^a(X, \theta)$.

Figure 3 is a 3D illustration of the wave propagation from the piston face $\xi(t)$ to the observer point x through the characteristic K_ξ . The piston displacement $\xi(t)$ is a sine function and its velocity $\xi'(t)$ describes an helicoidal trajectory, highlighting the moving character of the boundary condition.

This results in various lengths of characteristic lines (represented by arrows) and so various times of propagation to reach the position x . Thus, the waveform of the particle velocity observed at x is not the exact copy of the piston motion.

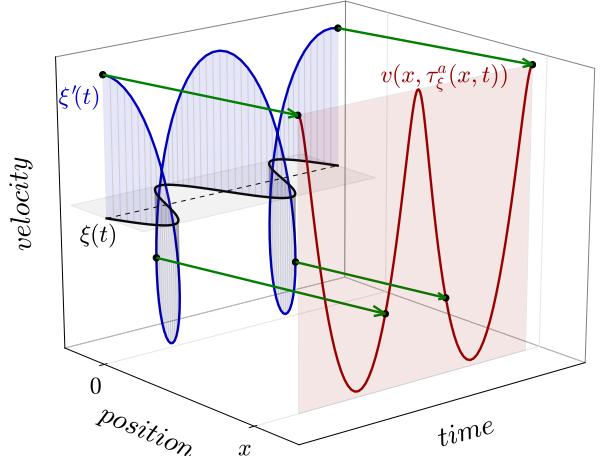


Figure 3: Perspective view of an acoustic wave generated at $(\xi(t), t)$ and observed at $(x, \tau_\xi^a(x, t))$. The piston displacement is described by the curve labelled $\xi(t)$, and the velocity by the helicoidal trajectory $\xi'(t)$. Four lines of characteristic K_ξ are represented by arrows, connecting $\xi'(t)$ with the particle velocity v at position x .

The characteristic defined in (9) is based on the emission time of the acoustic wave (propagation occurs at time $\tau_\xi^a(x, t) > t$). Then the expression of the particle velocity observed at x is $v(x, \tau_\xi^a(x, t))$. However we seek a solution based on the observation time, so that the velocity is $v(x, t)$. Thus, the existence of strong solutions involves the reciprocal application K_ξ^{-1} .

The main result of this section is given by the following theorem.

Theorem 1 (Strong solution). *Let ξ be a C^{n+1} -regular function with $n \in \mathbb{N}$. Suppose that its (signed) Mach number is strictly less than 1, namely,*

$$\forall t \in \mathbb{R}, \quad \xi'(t)/c_0 < 1. \quad (11)$$

Then, the following results hold:

- (i) K_ξ is a C^n -regular diffeomorphism from \mathbb{K}_ξ to \mathbb{K}_ξ^a ,
- (ii) The strong solution v of (3-6-7) is C^n -regular and given by

$$\begin{aligned} v : \mathbb{K}_\xi^a &\longrightarrow \mathbb{R} \\ (x, t) &\longmapsto \xi' \circ \tau_\xi^d(x, t) \end{aligned} \quad (12)$$

where the (unique) departure time $\tau_\xi^d(x, t)$ of the travelling wave observed at (x, t) is given by the C^n -regular function

$$\begin{aligned} \tau_\xi^d : \mathbb{K}_\xi^a &\longrightarrow \mathbb{R} \\ (x, t) &\longmapsto [K_\xi^{-1}(x, t)]_2, \end{aligned} \quad (13)$$

where $[K_\xi^{-1}(x, t)]_2$ denotes the second component of the function.

Consequently, if ξ is C^∞ -regular, then all the above-defined functions are also C^∞ -regular.

The proof of this theorem is given in appendix B. It relies on the two following propositions.

Proposition 1 (K_ξ is a bijection). *Let ξ be a C^1 -regular function. Then, function $K_\xi : \mathbb{K}_\xi \rightarrow \mathbb{K}_\xi^a$ is a bijection if function ξ satisfies condition (11).*

Proof. (i) By construction of the image set \mathbb{K}_ξ^a (see (10)), K_ξ is a surjective function.

(ii) *Injectivity.* ξ is a C^1 -regular function, so that τ_ξ^a and then K_ξ are also C^1 -regular functions. For all $(x, t) \in \mathbb{K}_\xi$, the jacobian of K_ξ is given by

$$J_{K_\xi}(x, t) = \begin{pmatrix} 1 & 0 \\ 1/c_0 & 1 - \xi'(t)/c_0 \end{pmatrix},$$

in which $1 - \xi'(t)/c_0$ is strictly positive. Therefore $K_\xi(x, t)$ is strictly increasing with respect to x and t . The monotonicity proves that K_ξ is injective, which concludes the proof. \square

Remark 1 (Condition on the Mach number). *(11) is a sufficient condition to ensure the bijectivity of K_ξ . To obtain a necessary and sufficient condition, (11) must be modified as follows:*

- the set $E = \{t \in \mathbb{R} \text{ s.t. } \xi'(t)/c_0 = 1\}$ has a zero measure, and
- $\xi'(t)/c_0 < 1$ for all $t \in \mathbb{R} \setminus E$.

Indeed, as E has a zero measure, the monotonicity is still fulfilled so that this condition is still sufficient. This condition is also necessary because if E has a non zero measure, there exists an open subset of \mathbb{K}_ξ on which the jacobian of K_ξ has a zero determinant, making K_ξ locally constant and so non injective.

Remark 2 (Relations between \mathbb{K}_ξ^a and \mathbb{K}_ξ). *By construction of \mathbb{K}_ξ^a (see (10)), $\mathbb{K}_\xi \subseteq \mathbb{K}_\xi^a$: the characteristics lines cover \mathbb{K}_ξ (see $L_\xi(t)$ in Figure 2.) Moreover, if the Mach number condition (11) is fulfilled, the characteristics lines $L_\xi(t)$ start from the boundary $\partial\mathbb{K}_\xi$ of \mathbb{K}_ξ at point $(\xi(t), t)$, $t \in \mathbb{R}$ and do not recross this boundary $\partial\mathbb{K}_\xi = (\xi(t), t)$, $t \in \mathbb{R}$. Therefore, $\mathbb{K}_\xi^a = \mathbb{K}_\xi$.*

Proposition 2 (K_ξ is a C^n -diffeomorphism). *Let ξ be a C^1 -regular function that satisfies condition (11). If function ξ is also C^{n+1} -regular with $n \in \mathbb{N}$, then function K_ξ is a C^n -regular diffeomorphism. Consequently, if ξ is C^∞ -regular, then K_ξ is a C^∞ -regular diffeomorphism.*

Proof. Let $n \in \mathbb{N}$ and consider $\xi \in C^{n+1}$ -regular. Suppose that condition (11) is satisfied. Then, from Proposition 1, function $\tau_\xi^d(x, t) : (x, t) \in \mathbb{K}_\xi \mapsto [K_\xi^{-1}(x, t)]_2 \in \mathbb{R}$ exists and is continuous.

Now, we prove by induction that τ_ξ^d is C^p -regular for $1 \leq p \leq n$.

• Case $p = 1$: $\tau_\xi^d \in C^1$.

From (8), τ_ξ^a is C^1 -regular so that $D_2\tau_\xi^a$ is continuous, where D_2 stands for the derivative with respect to the second component of the function. Moreover, $\forall (x, t) \in \mathbb{K}_\xi^a$, $(x, \tau_\xi^d(x, t)) \in \mathbb{K}_\xi$ and $f(x, t) = D_2\tau_\xi^a(x, \tau_\xi^d(x, t)) = 1 - \xi'(\tau_\xi^d(x, t))/c_0$ defines a function $f : \mathbb{K}_\xi^a \rightarrow \mathbb{R}$ which is:

(i) **continuous**, because it is a composition of the continuous functions τ_ξ^d and $D_2\tau_\xi^a$,

(ii) **strictly positive** because (11) is satisfied.

The jacobian of K_ξ^{-1} given by

$$\begin{aligned} J_{K_\xi^{-1}} : \quad \mathbb{K}_\xi^a &\longrightarrow \mathcal{M}_{2,2}(\mathbb{R}) \\ (x, t) &\longmapsto \begin{pmatrix} 1 & 0 \\ \frac{-1}{c_0 f(x, t)} & \frac{1}{f(x, t)} \end{pmatrix}, \end{aligned} \quad (14)$$

is then a continuous function. Hence, K_ξ^{-1} and then τ_ξ^d (see (13)) are C^1 -regular.

• Case $p \geq 2$: If τ_ξ^d is C^p -regular, then τ_ξ^d is C^{p+1} -regular.

From (14), the jacobian $J_{K_\xi^{-1}}$ is C^p -regular. Therefore τ_ξ^d is C^{p+1} -regular, which concludes the proof. \square

From Proposition 1, there exists a function $\tau_\xi^d : \mathbb{K}_\xi^a \mapsto \mathbb{R}$, such that for all $(x, t) \in \mathbb{K}_\xi^a$, equation (8) reads

$$\tau_\xi^d(x, t) = t - \frac{x + \xi \circ \tau_\xi^d(x, t)}{c_0}. \quad (15)$$

The calculation of this space-time function can be reduced to that of a simpler time function, because of the following property.

Property 1 (Translational symmetry). *Let $\delta > 0$ and $T_\delta : (x, t) \mapsto (x + \delta, t + \delta/c_0)$ be the space-time translation operator of space-time shift $(\delta, \delta/c_0)$.*

Then, for any arrival space-time point $(x, t) \in \mathbb{K}_\xi^a$, the translated point $\mathcal{T}_\delta(x, t)$ is in \mathbb{K}_ξ^a and the departure time of this translated point is unchanged:

$$\forall (x, t) \in \mathbb{K}_\xi^a, \delta > 0, \quad \tau_\xi^d \circ \mathcal{T}_\delta(x, t) = \tau_\xi^d(x, t). \quad (16)$$

and $v(\mathcal{T}_\delta(x, t)) = \xi' \circ \tau_\xi^d(x, t)$.

The proof is straightforward from (3).

Consequently, the next section addresses the derivation of a solver for the reduced problem given by

$$\tau_\xi(t) = t + \frac{\xi \circ \tau_\xi(t)}{c_0}, \quad (17)$$

where $\tau_\xi(t) = \tau_\xi^d(x=0, t)$, and from which the strong solution is derived

$$v(x, t) = \xi'(\tau_\xi(t) - x/c_0). \quad (18)$$

Remark 3 (Equivalent Eulerian description). Consider function $V : (x, t) \mapsto \xi'(\tau_\xi(t - x/c_0))$ defined on \mathbb{R}^2 . This function coincides with v on \mathbb{K}_ξ^a (see (12)) and is such that $V \circ \mathcal{T}_\delta$ is invariant with respect to δ . Consequently, function V extends solution v on the space-time domain \mathbb{R}^2 and $t \mapsto V(0, t)$ stands for the equivalent source description at $x = 0$ (Eulerian description).

4. RECURSIVE METHOD BASED ON POWER SERIES

First, a dimensionless version of the model is established, so that the representation as power series is simplified. Second, a method based on series expansion is described, in order to carry out computation of (17-18).

4.1. Dimensionless model

Consider the following change of variable, given in Table 1.

Table 1: Dimensionless Model.

Variables	Functions
$\tilde{x} = x/c_0$	$\tilde{p} = p/(\rho_0 c_0)$
$\tilde{t} = t$	$\tilde{v} = v/c_0$
	$\tilde{\xi} = \xi/c_0$

Replacing equations (3-6-7) by their versions denoted with a "tilde" yields

$$\tilde{p}(\tilde{x}, \tilde{t}) = \tilde{v}(\tilde{x}, \tilde{t}) = \tilde{v}^+(\tilde{t} - \tilde{x}), \quad (19)$$

the time-space propagation domain becomes

$$\tilde{\mathbb{K}}_\xi = \{(\tilde{x}, \tilde{t}) \in \mathbb{R}^2 \text{ s.t. } \tilde{x} > \tilde{\xi}(\tilde{t})\}, \quad (20)$$

and the left boundary

$$\tilde{v}^+(\tilde{t} - \tilde{\xi}(\tilde{t})) = \tilde{\xi}'(\tilde{t}). \quad (21)$$

The strong solution becomes

$$\tilde{v}(\tilde{x}, \tilde{t}) = \tilde{\xi}'(\tilde{\tau}_\xi(\tilde{t}) - \tilde{x}) \in (\tilde{\mathbb{K}}_\xi, \mathbb{R}), \quad (22)$$

where

$$\tilde{\tau}_\xi(\tilde{t}) = \tilde{t} + \tilde{\xi} \circ \tilde{\tau}_\xi(\tilde{t}). \quad (23)$$

For the sake of readability, the symbols "tilde" are omitted in the sequel.

4.2. Perturbation method and series expansion

Consider $\xi \in \mathcal{C}^\infty$ (so that $\tau_\xi \in \mathcal{C}^\infty$ from the theorem 1), and let

$$\epsilon(t) = \tau_\xi(t) - t, \quad (24)$$

which corresponds to the variation of propagation time due to source motion. Reformulating (22-23) with $\epsilon(t)$ yields

$$v(x, t) = \xi'(t - x + \epsilon(t)), \quad (25)$$

$$\epsilon(t) = \xi(t + \epsilon(t)). \quad (26)$$

Now, we solve the implicit equation (26) using a perturbation method. Consider that (26) describes a system of input ξ and output $\epsilon(t)$, and apply the change of variable $\xi = \alpha \cdot u$, with $\alpha \geq 0$. The method consists in writing the output as a formal power series in α , such that

$$\epsilon(t) = \sum_{n=0}^{\infty} \frac{\epsilon_n(t)}{n!} \alpha^n. \quad (27)$$

Then (26) becomes

$$\sum_{n=0}^{\infty} \alpha^n \frac{\epsilon_n(t)}{n!} = \alpha \cdot u \left(t + \sum_{n=0}^{\infty} \alpha^n \frac{\epsilon_n(t)}{n!} \right). \quad (28)$$

Now function u is developed into Taylor series at point t :

$$\sum_{n=0}^{\infty} \alpha^n \frac{\epsilon_n(t)}{n!} = \alpha \sum_{n=0}^{\infty} \frac{u^{(n)}(t)}{n!} \left(\sum_{m=0}^{\infty} \alpha^m \frac{\epsilon_m(t)}{m!} \right)^n. \quad (29)$$

Given that the right-hand side of (29) is multiplied by α , the term $\epsilon_0(t)$ (corresponding to α^0) is vanished. Then (29) reads

$$\sum_{n=1}^{\infty} \alpha^n \frac{\epsilon_n(t)}{n!} = \alpha \sum_{n=0}^{\infty} \frac{u^{(n)}(t)}{n!} \left(\sum_{m=1}^{\infty} \alpha^m \frac{\epsilon_m(t)}{m!} \right)^n. \quad (30)$$

The right-hand side is a series composition and can be developed using the Fa  di Bruno power series formula:

$$\sum_{n=1}^{\infty} \frac{\epsilon_n(t)}{n!} \alpha^n = \alpha \cdot u(t) + \sum_{n=1}^{\infty} c_n(t) \cdot \alpha^{n+1}, \quad (31)$$

where

$$c_n(t) = \sum_{k=1}^n \frac{u^{(k)}(t)}{n!} B_{n,k}(\epsilon_1(t), \epsilon_2(t), \dots, \epsilon_{n-k+1}(t)) \quad (32)$$

and $B_{n,k}$ are the partial Bell polynomials.

Now,

$$\sum_{n=1}^{\infty} \alpha^n \left[\frac{\epsilon_n(t)}{n!} - u(t) \delta_{1,n} - c_{n-1}(t) \right] = 0, \quad (33)$$

hence, for all $\alpha > 0$,

$$\begin{aligned} \epsilon_1(t) &= \xi(t), \\ \epsilon_n(t) &= n \cdot \sum_{k=1}^{n-1} \xi^{(k)}(t) B_{n-1,k}(\epsilon_1(t), \dots, \epsilon_{n-k}(t)), \end{aligned} \quad (34)$$

$$\forall n > 1,$$

where the amplitude α has been dropped, so that $u = \xi$.

Property 2 (Multivariate polynomial). $\epsilon_n(t)$ has the form

$$C_n : \quad \epsilon_n(t) = P_n(\xi(t), \xi'(t), \dots, \xi^{(n-1)}(t)), \quad (35)$$

where P_n is a homogeneous multivariate polynomial of degree n .

The proof of this property is given in Appendix C.

Now, the expression of the particle velocity (25) is also expanded into Taylor series, at point $t - x$:

$$v(x, t) = \sum_{n=0}^{\infty} \frac{\xi^{(n+1)}(t-x)}{n!} \left(\sum_{m=1}^{\infty} \frac{\epsilon_m(t)}{m!} \right)^n. \quad (36)$$

Applying again the Fa  di Bruno formula for power series composition finally yields the expression of the particle velocity,

$$v(x, t) = \sum_{n=1}^{\infty} v_n(x, t), \quad (37)$$

where

$$\begin{aligned} v_1 &= \xi^{(1)}(t-x), \\ v_n &= \sum_{k=1}^{n-1} \frac{\xi^{(k+1)}(t-x)}{k!(n-1)!} B_{n-1,k}(\epsilon_1, \dots, \epsilon_{n-k}), \quad (38) \\ \forall n > 1. \end{aligned}$$

The first orders of v_n are listed below, where the dimensionless model has been dropped:

$$v_1(x, t) = \xi^{(1)}(t-x/c_0), \quad (39a)$$

$$v_2(x, t) = \frac{1}{c_0} \cdot \xi(t-x/c_0) \cdot \xi^{(2)}(t-x/c_0), \quad (39b)$$

$$\begin{aligned} v_3(x, t) &= \frac{1}{c_0^2} \left[\frac{1}{2} \xi(t-x/c_0)^2 \cdot \xi^{(3)}(t-x/c_0) \right. \\ &\quad \left. + \xi(t-x/c_0) \cdot \xi^{(1)}(t-x/c_0) \cdot \xi^{(2)}(t-x/c_0) \right]. \quad (39c) \end{aligned}$$

Remark 4 (Link with Volterra series). From the property 2, $\epsilon_n(t)$ has homogeneous order n with respect to $\xi(t)$ and its derivatives. Then the system of input $\xi(t)$, on which the perturbation method is applied, and output $\epsilon(x, t)$ (and by extension $v(x, t)$) can be represented by a Volterra series expansion [5], formally in the space of distributions.

Remark 5 (Convergence). Some results about the convergence of Volterra series are available in [6, 7] for L^∞ input signals: they involve L^1 Volterra kernels. In the present problem, the convergence and the class of admissible inputs are a more complicated issue: because of the time-derivative in (38), kernels are not in L^1 . In addition to (11), some properties about asymptotic behaviour (bounds on time derivatives or on frequency characteristics) have to be examined to set the convergence condition: this future work will provide the class of admissible waveforms.

Remark 6 (Practical implementation). For implementation purpose, the expansion (37) is truncated at a given order N . Moreover, for a signal in the frequency range f_{\max} , a frequency oversampling by a factor of N is applied so that $f_s = N \cdot (2f_{\max})$, ensuring no aliasing (the frequency range of v_n is then $n f_{\max} \leq N f_{\max}$).

4.3. Numerical evaluation of the convergence

Although the convergence domain of (38) is not tackled from the theoretical side, this section presents simulations of the acoustic output for a 40Hz, 1s sinusoidal velocity input at various Mach numbers. Figure 4 shows the results for different truncation orders, from $N = 1$ to $N = 15$.

Divergence of the series is noted at Mach 1 and above, which is consistent with the condition of existence of the strong solution (11). Since the usual range of membrane velocities is far below this limit, the convergence criterion should be met, at least for sinusoidal motions.

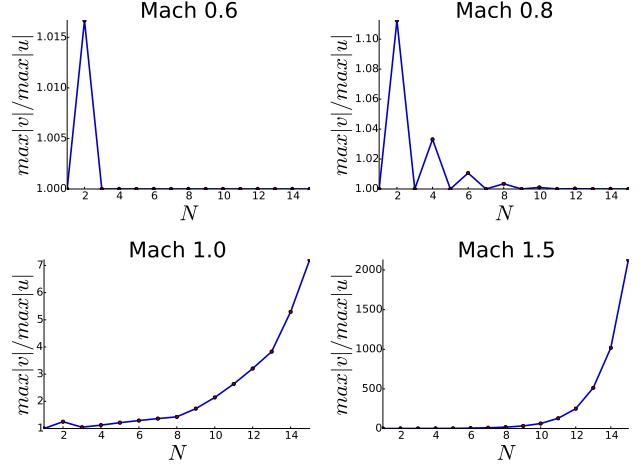


Figure 4: Convergence computation for various sinusoidal input velocities. Simulations at Mach 0.6 and Mach 0.8 converge, contrary to simulations at Mach 1 and Mach 1.5.

5. SIMULATION AND DISTORTION EVALUATION

5.1. Harmonic distortion

The harmonic distortion is evaluated for a sinusoidal piston velocity with constant amplitude over the frequency spectrum (loudspeaker with a flat frequency response), so that the input displacement takes the form

$$\xi(t) = \frac{A}{2\pi f_1} \sin(2\pi f_1 t). \quad (40)$$

Two simulations are implemented, listed in Table 2 below. The velocity amplitude A corresponds to a high velocity of the driver's membrane, T is the simulation duration and N is the truncation order of (38). Results are presented in Figures 5 and 6.

Table 2: Simulation parameters used for harmonic distortion simulations.

Name	f_1	A	T	N	f_s
HD1	40 Hz	1 m/s	50 s	3	$N \times 44100$ Hz
HD2	1 kHz	1 m/s	50 s	3	$N \times 44100$ Hz

- The same harmonic distortion is observed for **HD1** and **HD2**, respectively in Figure 5 and Figure 6, with the apparition of the second harmonic around -55dB. Injecting the expression of $\xi(t)$ in the first orders equations (39) leads to

$$\begin{aligned} v_1(x, t) &= A \sin(2\pi f_1 t), \\ v_2(x, t) &= \frac{A^2}{2c_0} (1 - \cos(4\pi f_1 t)). \end{aligned}$$

It appears that harmonic distortion only depends on piston velocity amplitude, as confirmed by the simulations. Moreover, the amplitude ratio between v_1 and v_2 is of -56,7 dB, which is consistent with the numerical result.

- A THD¹ of 0.14 % is noted. In most cases, harmonic distortion can then be neglected. This is in agreement with previous studies [2, 8]. A truncation at order 2 appears to be sufficient to characterise this type of distortion (in the loudspeaker velocities range).

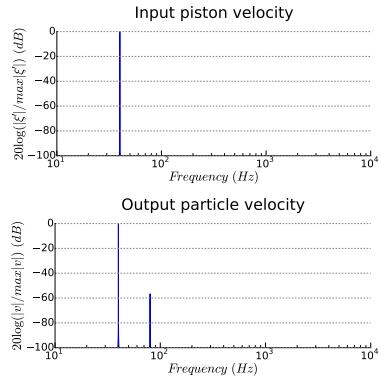


Figure 5: Results of simulation **HD1**. The amplitude spectrum of the input piston velocity (top) and the output particle velocity (bottom) are shown. Harmonic distortion is observed at $2f_1$.

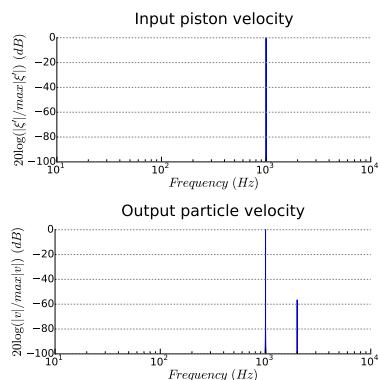


Figure 6: Results of simulation **HD2**. The amplitude spectrum of the input piston velocity (top) and the output particle velocity (bottom) are shown. Harmonic distortion is observed at $2f_1$.

¹Total Harmonic Distortion is calculated as the ratio of RMS amplitude of the harmonics to the RMS amplitude of the fundamental.

5.2. Intermodulation distortion

In this section, the intermodulation distortion is examined by simulating the particle velocity for an input displacement of the form

$$\xi(t) = \frac{A}{2\pi f_1} \sin(2\pi f_1 t) + \frac{A}{2\pi f_2} \sin(2\pi f_2 t).$$

Given the weak amplitude of harmonic distortion, it is assumed that both phenomena can be analyzed separately (harmonic distortion that occurs in the following computations is neglected). The simulation parameters are listed in Table 3. Simulation results are presented in Figures 7 and 8.

Table 3: Simulation parameters used for intermodulation distortion simulation.

Name	f_1	f_2	A	T	N	f_s
IMD1	40 Hz	3 kHz	1 m/s	50 s	5	$N \times 44100$ Hz
IMD2	40 Hz	6 kHz	1 m/s	50 s	5	$N \times 44100$ Hz

- An increase of intermodulation distortion is observed between both simulations with a noticeable gain in sidebands amplitude for **IMD2**. The Intermodulation Factors² are 7.5% and 11.4% for **IMD1** and **IMD2**, respectively. This confirms that the intermodulation effect, which rises with the value of f_2 , should be taken into account for large frequency range speakers.

- Truncation at order 5 was sufficient for both simulations to capture the distortion phenomena in a dynamic of 100dB. However, higher orders might be necessary in case of sound synthesis with audio signal as input, since intermodulation distortion can be much higher with complex signals.

- Finally, Figure 9 shows the intermodulation distortion for simulation **IMD2** in the time domain, resulting in phase modulation of the high frequency component.

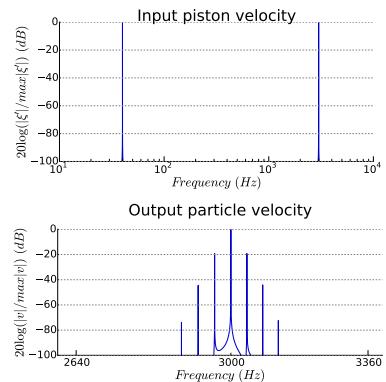


Figure 7: Amplitude spectrum of the particle velocity (bottom) for simulation **IMD1** (top). The bottom figure is rescaled around the frequency of interest, f_2 . Intermodulation distortion is observed at $f_2 - p.f_1$ and $f_2 + p.f_1$ for $p = 1, 2, 3$.

²Intermodulation Factor is calculated as the ratio of the RMS amplitude of the sidebands to the RMS amplitude of the carrier frequency.

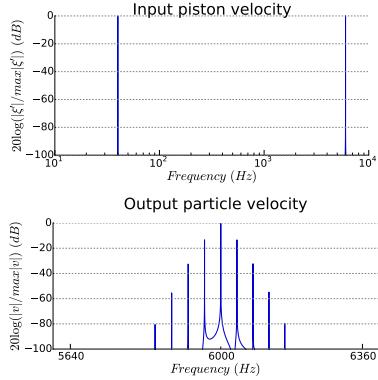


Figure 8: Amplitude spectrum of the particle velocity (bottom) for simulation **IMD2** (top). The bottom figure is rescaled around the frequency of interest, f_2 . Intermodulation distortion is observed at $f_2 - p.f_1$ and $f_2 + p.f_1$ for $p = 1, 2, 3, 4$.

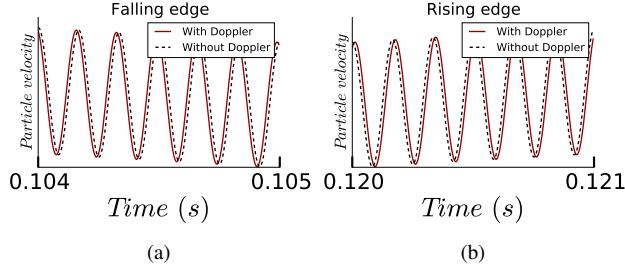


Figure 9: Time samples of simulation **IMD2**. Phase modulation is observed, since the high frequency component is phase-shifted on the falling edges and conversely on the rising edges.

In order to highlight the influence of Doppler effect on more complex signals, a final simulation is performed with an input signal composed of two chirps (50Hz to 3kHz) and one pure tone (1kHz), at constant amplitude 1 m/s. Spectrograms of the particle velocities are shown without Doppler effect on the upper part of Figure 10, which is an exact copy of the input (delayed with x/c_0). Bottom Figure 10 shows the acoustic output with the Doppler effect, truncated at $N = 5$. Both harmonic and intermodulation distortion are clearly visible.

6. CONCLUSION

The model and simulations presented in this paper confirm that the Doppler effect causes audible intermodulation distortion, as stated in [1, 2, 9]. The numerical examination of expansion terms shows that: (i) the convergence is satisfied quickly, (ii) the first expansion terms can have an audible impact up to the orders 3 or 4. Future work is concerned with:

- the convergence proof of the series expansion and the establishment of a truncation error bound,
- a corrector that compensates the Doppler effect based on series inversion and the control of the truncation order on the inverse series,

- the improvement of the model by including the convection phenomenon.

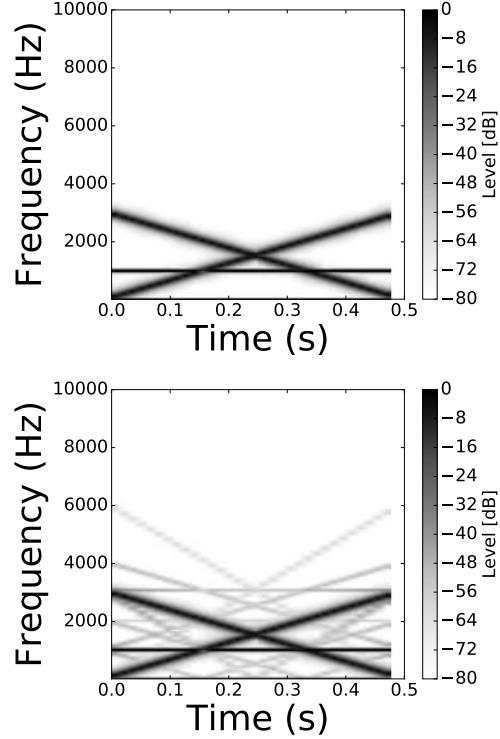


Figure 10: Spectrograms of the particle velocities, normalized in dB, simulated with (bottom) and without (top) Doppler effect. The input piston velocity is composed of two chirps and one pure tone.

A. REFERENCES

- [1] G.L. Beers and H. Belar, "Frequency-modulation distortion in loudspeakers," *Proceedings of the Institute of Radio Engineers*, vol. 31, no. 4, pp. 132–138, 1943.
- [2] HJ Butterweck, "About the doppler effect in acoustic radiation from loudspeakers," *Acta Acustica united with Acustica*, vol. 63, no. 1, pp. 77–79, 1987.
- [3] Nandor L. Balazs, "On the solution of the wave equation with moving boundaries," *Journal of Mathematical Analysis and Applications*, vol. 3, no. 3, pp. 472–484, 1961.
- [4] Lakhdar Gaffour, "Analytical method for solving the one-dimensional wave equation with moving boundary," *Progress In Electromagnetics Research*, vol. 20, pp. 63–73, 1998.
- [5] Thomas Hélie, "Modélisation physique d'instruments de musique et de la voix: systèmes dynamiques, problèmes directs et inverses," *Habilitation à Diriger des Recherches*, pp. 42–43, 2013.
- [6] Thomas Hélie and Béatrice Laroche, "Computation of convergence bounds for volterra series of linear-analytic single-input systems," *IEEE Transactions on automatic control*, vol. 56, no. 9, pp. 2062–2072, 2011.

- [7] Thomas Hélie and Béatrice Laroche, “Computable convergence bounds of series expansions for infinite dimensional linear-analytic systems and application,” *Automatica*, vol. 50, no. 9, pp. 2334–2340, 2014.
- [8] Wolfgang Klippel, “Loudspeaker nonlinearities—causes, parameters, symptoms,” in *Audio Engineering Society Convention 119*. Audio Engineering Society, 2005.
- [9] DW van Wulfften Palthe, “Doppler effect in loudspeakers,” *Acta Acustica united with Acustica*, vol. 28, no. 1, pp. 5–11, 1973.
- [10] Henry C. Kessler Jr, “Equivalent eulerian boundary conditions for finite-amplitude piston radiation,” *The Journal of the Acoustical Society of America*, vol. 34, no. 12, pp. 1958–1959, 1962.
- [11] Bronislaw Zoltogórski, “Moving boundary conditions and nonlinear propagation as sources of nonlinear distortions in loudspeakers,” *Journal of Audio Engineering Society*, vol. 41, no. 9, pp. 691–700, 1993.
- [12] Guy Lemarquand and Michel Bruneau, “Nonlinear intermodulation of two coherent acoustic progressive waves emitted by a wide-bandwidth loudspeaker,” *Journal of the Audio Engineering Society*, vol. 56, no. 1/2, pp. 36–44, 2008.

B. PROOF OF THE THEOREM 1

Proof. Let ξ be a C^{n+1} -regular function with $n \in \mathbb{N}$ that satisfies (11). Point (i) results from propositions 1 and 2.

Proof of point (ii) is divided into two steps:

a. Boundary condition: $v(\xi(t), t) = \xi'(t)$

Because of (i), τ_ξ^d and $v = \xi' \circ \tau_\xi^d$ are C^n -regular. From (9), the following equality can be computed:

$$K_\xi(\xi(t), t) = (\xi(t), t), \quad \forall t \in \mathbb{R}. \quad (41)$$

Moreover, by definition of the reciprocal function K_ξ^{-1} ,

$$\forall (x, t) \in \mathbb{K}_\xi^a, \quad (x, t) = K_\xi^{-1} \circ K_\xi(x, t), \quad (42)$$

so that, replacing (x, t) by $(\xi(t), t)$,

$$(\xi(t), t) = K_\xi^{-1}(\xi(t), t). \quad (43)$$

Taking the second component of (43), where $[K_\xi^{-1}]_2 = \tau_\xi^d$, yields

$$\tau_\xi^d(\xi(t), t) = t. \quad (44)$$

Therefore the solution $\xi' \circ \tau_\xi^d(\xi(t), t)$ verifies the boundary condition (5).

b. Propagation: $v(x, t) = v^+(t - x/c_0)$

Define the function

$$\begin{aligned} L : \quad \mathbb{K}_\xi^0 &\longmapsto \mathbb{K}_\xi^a \\ (x, t) &\longmapsto (x, t + x/c_0), \end{aligned} \quad (45)$$

where $K_\xi^0 = \{(x, t) \in \mathbb{R}^2 | (x, t + x/c_0) \in \mathbb{K}_\xi^a\}$. Moreover, isolating the second component of

$$\begin{aligned} L(x, t) &= K_\xi \circ K_\xi^{-1} \circ L(x, t) \\ &= \left(x, \tau_\xi^a(x, \tau_\xi^d \circ L(x, t)) \right), \end{aligned} \quad (46)$$

leads to, for all $(x, t) \in \mathbb{K}_\xi^0$,

$$t + \frac{x}{c_0} = \tau_\xi^d \circ L(x, t) + \frac{x - \xi \circ \tau_\xi^d \circ L(x, t)}{c_0}. \quad (47)$$

This equation is equivalent to, for all $(x, t) \in \mathbb{K}_\xi^0$,

$$t = F_\xi(\tau_\xi^d \circ L(x, t)), \quad (48)$$

where

$$\begin{aligned} F_\xi : \quad \mathbb{R} &\longmapsto \mathbb{R} \\ t &\longmapsto t - \xi(t)/c_0. \end{aligned} \quad (49)$$

Now, for all $t \in \mathbb{R}$, $F'_\xi(t) = 1 - \xi'(t)/c_0 > 0$ and $\lim_{t \rightarrow \pm\infty} F_\xi(t) = \pm\infty$ so that F_ξ is a strictly increasing bijective function. It follows that

$$\tau_\xi^d \circ L(x, t) = F_\xi^{-1}(t), \quad (50)$$

proving that $\tau_\xi^d \circ L(x, t)$ does not depend on x .

Consequently, the strong solution (12) can be written, for all $(x, t) \in \mathbb{K}_\xi^a$,

$$\begin{aligned} v(x, t) &= \xi' \circ \tau_\xi^d(x, t) \\ &= \xi' \circ \tau_\xi^d \circ L(x, t - x/c_0), \end{aligned} \quad (51)$$

which has the form $v^+(t - x/c_0)$, with $v^+ = \xi' \circ \tau_\xi^d \circ L$, that concludes the proof. \square

C. PROOF OF THE PROPERTY 2

Proof. Let us prove by induction the following property

$$\mathcal{C}_n : \quad \epsilon_n = P_n(\xi(t), \xi'(t), \dots, \xi^{(n-1)}(t)),$$

where P_n is a homogeneous multivariate polynomial of degree n .

• *Step 1: \mathcal{C}_2 is true*, since $\epsilon_2 = \xi(t).\xi'(t)$, which is a polynomial of degree 2.

• *Step 2: If \mathcal{C}_n is true, then \mathcal{C}_{n+1} is true.*

Let $\epsilon_n = P_n(\xi, \xi', \dots, \xi^{(n-1)})$. Then

$$\epsilon_{n+1} = (n+1) \cdot \sum_{k=1}^n \xi^{(k)} \cdot B_{n,k}(\xi(t), \dots, P_{n-k+1}(\xi, \xi', \dots, \xi^{(n-k)})),$$

where the Bell polynomials are developed

$$\begin{aligned} B_{n,k}(\xi, \dots, P_{n-k+1}(\xi, \xi', \dots, \xi^{(n-k)})) &= \sum_{j_1, \dots, j_{n-k} \in \Pi_{n-1,k}} \left(\frac{\xi(t)}{1!} \right)^{j_1} \times \dots \\ &\times \left(\frac{P_{n-k+1}(\xi, \dots, \xi^{(n-k)})}{(n-k+1)!} \right)^{j_{n-k+1}}. \end{aligned}$$

and $\Pi_{n,k} = \begin{cases} j_1 + j_2 + \dots + j_{n-k+1} = k \\ j_1 + 2.j_2 + \dots + (n-k+1).j_{n-k+1} = n. \end{cases}$

$B_{n,k}$ is a sum over products of polynomials, then it is a polynomial. Moreover the degree of each term of the sum over $\Pi_{n,k}$ is

$$1.j_1 + 2.j_2 + \dots + j_{n-k+1} + (n-k+1) = n,$$

from the definition of $\Pi_{n,k}$. Therefore the degree of all nonzero terms of ϵ_{n+1} is $n+1$, that concludes the proof. \square

LATENT FORCE MODELS FOR SOUND: LEARNING MODAL SYNTHESIS PARAMETERS AND EXCITATION FUNCTIONS FROM AUDIO RECORDINGS

William J. Wilkinson, Joshua D. Reiss and Dan Stowell

Centre for Digital Music
 Queen Mary University of London
 London, UK
 w.j.wilkinson@qmul.ac.uk

ABSTRACT

Latent force models are a Bayesian learning technique that combine physical knowledge with dimensionality reduction — sets of coupled differential equations are modelled via shared dependence on a low-dimensional latent space. Analogously, modal sound synthesis is a technique that links physical knowledge about the vibration of objects to acoustic phenomena that can be observed in data. We apply latent force modelling to sinusoidal models of audio recordings, simultaneously inferring modal synthesis parameters (stiffness and damping) and the excitation or contact force required to reproduce the behaviour of the observed vibrational modes. Exposing this latent excitation function to the user constitutes a controllable synthesis method that runs in real time and enables sound morphing through interpolation of learnt parameters.

1. INTRODUCTION

Modal synthesis aims to reproduce the behaviour of the vibrational modes of a sounding object, through consideration of its physical properties [1]. If all the required physical properties are known, then the frequency and amplitude of the modes can be calculated. Alternatively, by taking the Fourier transform of a recording of the sounding object, we can observe these same features empirically. Hence we have a clear link between the physics of vibrating objects and observable acoustic behaviour. This has often been exploited to construct models for sound synthesis that provide users with both physical and phenomenological control [2, 3, 4].

In [4], modal synthesis parameters were learnt automatically from recordings of impact sounds by assuming the excitation force to be an impulse and inferring the modes' mass, stiffness and damping coefficients from data. Others have constructed detailed physical models for source-filter interaction, and set the filter parameters corresponding to observed peaks in the frequency spectrum [2, 5].

Recent work in the machine learning community, namely the development of latent force models (LFM), has shown that it is possible to build a model which incorporates physical knowledge and to fit it to data via an inference procedure [6]. We adopt this approach to formally use learnings from audio recordings to construct a simple mechanistic model for modal synthesis that is generalisable to a large class of sounds.

Our framework for synthesis utilises sinusoidal analysis [7, 8] to track modes over time, and makes assumptions about the behaviour of modes by representing their amplitude with first order ordinary differential equations (ODEs). The introduction of such ODEs into the model prior, a latent force model, allows us to infer both the system parameters and the excitation function required to

reproduce the observed outputs. It does so by coupling the modes' amplitudes through consideration of their common dependence on a low-dimensional latent space, in this case the one-dimensional excitation function. The result is a real-time synthesis model that allows for user control and sound morphing. Interactive sound examples and MATLAB code for latent force modelling of sinusoidal amplitude data are provided.[†]

We formulate our problem in Section 2. In Section 3 we summarise the relevant literature relating to sound synthesis and latent force models. In Section 4 we present our approach to the application of latent force modelling to audio. Section 5 outlines how our approach can be utilised to perform real-time synthesis and sound morphing, and Section 6 presents empirical results and case studies.

2. PROBLEM FORMULATION

Consider M modes of vibration of a sounding object, for which we obtain observation data from sinusoidal analysis of an audio recording. We assume the frequencies f_i of the modes to be fixed and that the amplitudes $x_i(t)$ are modelled by exactly one excitation function $u(t)$ being fed through an idealised physical system:

$$\frac{dx_i(t)}{dt} + D_i g_x(x_i(t)) = S_i g_u(u(t)), \quad i = 1, \dots, M, \quad (1)$$

where coefficients D_i and S_i relate to physical properties of the i^{th} mode, with g_x and g_u being potentially nonlinear functions of outputs x_i and input u respectively.

The task is to fit our data to this model in such a way that we can infer all the system parameters $\{S_i, D_i\}_{i=1}^M$ and predict the behaviour of $u(t)$. Doing so constitutes transformation of the data to a one-dimensional control space. With resynthesis in mind, we must encourage realistic parameters relating to stiffness and damping of the modes to be learnt, and require the predicted behaviour of $u(t)$ to be interpretable as physical energy driving the system.

After the model has been fit, the output audio signal Y can be synthesised through summation of sinusoids with the reconstructed amplitudes (and initial phase ϕ_i):

$$Y(t) = \sum_{i=1}^M x_i(t) \sin(2\pi f_i t + \phi_i). \quad (2)$$

[†]<http://c4dm.eecs.qmul.ac.uk/audioengineering/latent-force-synthesis>

3. BACKGROUND

3.1. Sound Synthesis

Physics-based approaches to sound synthesis vary from detailed numerical simulation of the sound production mechanism represented by differential equations [9, 10], to standard digital filtering techniques informed by those same differential equations [5, 11]. These approaches require significant knowledge regarding the complex interactions that produce sound, and as such are limited to systems for which much of the pertinent physics are known.

Modal synthesis is a more generalisable, physically-inspired approach which typically represents the vibrational modes of a sounding object as a set of decoupled second-order differential equations, also known as mass-spring-damper systems [1, 2]. The forced mass-spring-damper corresponding to the i^{th} mode has coefficients relating to mass m_i , springiness (or stiffness) k_i and damping b_i :

$$m_i \frac{d^2 X_i(t)}{dt^2} + b_i \frac{dX_i(t)}{dt} + k_i X_i(t) = u(t), \quad (3)$$

where $u(t)$ is the forcing function that excites the system. The exact sound production mechanism is not modelled in full detail. Instead it is assumed that sound is produced through the vibration of an object or column of air, and that the frequency and relative amplitude of these vibrations can be predicted based on mass, stiffness and damping parameters determined by the physical properties of the object.

The solution to these mass-spring-damper systems is a bank of modes,

$$X_i(t) = x_i(t) \sin(2\pi f_i t + \phi_i), \quad (4)$$

with time-varying amplitude $x_i(t)$, frequency f_i and initial phase ϕ_i , referred to as damped sinusoids, or oscillators. In traditional modal synthesis $u(t)$ is assumed to be an impulse, and we obtain the solution $x_i(t) = \alpha_i e^{-\beta_i t}$ where α_i and β_i are the amplitude and damping of the mode respectively. If we allow $u(t)$ to be unconstrained, then no analytical solution for the amplitude exists. In the present work we will constrain $u(t)$ by placing a Bayesian prior on its possible values (Section 3.2).

Sinusoidal modelling [7, 8] is an analysis-synthesis technique that compartmentalises a sound into its deterministic and stochastic components, and models the deterministic part as a sum of sinusoids such as those in equation (4). Energy is tracked through sequential frames of the Short Time Fourier Transform to create "partials" — sinusoids with frequency and amplitude that can vary over time.

Links between physical models and statistical behaviour have been exploited in the past to design hybrid synthesis frameworks that learn sound characteristics from data whilst enabling control through spectral transformation [4] or by learning a mapping between computed audio descriptors and a performed control space [12]. Our approach is to view sinusoidal data as the output of a series of digital filters representing the amplitudes $x_i(t)$ of the physical modes. This motivates the introduction of such filters (in ODE form) into the prior assumptions for a machine learning algorithm looking to infer knowledge from audio recordings.

3.2. Latent Force Models

Latent force models [6] are a probabilistic approach to modelling data which assumes that M observed output functions are produced by some $R < M$ unobserved (latent) functions being forced

through a set of differential equations. If this set of differential equations represents some physical behaviour present in the system we are modelling, even if only in a simplistic manner, then such a technique can improve our ability to perform inference from data [13, 14]. This is achieved by placing a Gaussian process prior [15] over the R latent functions, calculating the cross-covariances by solving the ODEs, and performing regression.

Standard latent force modelling involves batch processing of data using prediction equations that involve inversion of large covariance matrices. This motivates the reformulation of the system into its state space representation which allows for inference on sequential time points [16]. This also gives us an intuitive form with which to perform resynthesis (Section 4.3).

The aim here is to construct a joint model which incorporates all of our ODE parameters and our assumptions about the input. From this point onwards we assume $R = 1$, since we are attempting to model a one-dimensional excitation force. The introduction of additional forces is straightforward, but not explored here.

Suppose we can describe the i^{th} output x_i by this linear first-order ODE:

$$\frac{dx_i(t)}{dt} + D_i x_i(t) = S_i u(t). \quad (5)$$

We must now assume that $u(t)$ can be modelled by a linear time invariant (LTI) stochastic differential equation (SDE) of the form

$$\frac{d^p u(t)}{dt^p} + a_{p-1} \frac{d^{p-1} u(t)}{dt^{p-1}} + \dots + a_1 \frac{du(t)}{dt} + a_0 u(t) = w(t), \quad (6)$$

where p is the model order and $w(t)$ is a white noise process. If the covariance function chosen as part of the Gaussian process assumption cannot be written in this form with finite p , then approximations must be used. Here we choose $p = 3$, which is sufficient to represent the Matérn covariance function [15].

The joint state space model is constructed by inserting the coefficients of (5) and (6) into the transition matrix for a stable Markov process driven by $w(t)$:

$$\frac{dx(t)}{dt} = Fx(t) + Lw(t), \quad (7)$$

where, if \dot{u} represents the first differential of u w.r.t t ,

$$x = \begin{bmatrix} x_1 \\ \vdots \\ x_M \\ u \\ \dot{u} \\ \ddot{u} \end{bmatrix}, F = \begin{bmatrix} -D_1 & 0 & 0 & S_1 & 0 & 0 \\ 0 & \ddots & 0 & \vdots & 0 & 0 \\ 0 & 0 & -D_M & S_M & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & -a_0 & -a_1 & -a_2 \end{bmatrix},$$

$$L = \begin{bmatrix} 0 \\ \vdots \\ 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}.$$

This state space model includes all the necessary parameters, and we discretise it using standard techniques involving calculation of the matrix exponential. Its discrete form is

$$x[t_k] = \hat{F}[\Delta t_k]x[t_{k-1}] + q[t_{k-1}], \quad q[t_{k-1}] \sim N(\mathbf{0}, Q[\Delta t_k]), \quad (8)$$

where k is the time index, \hat{F} is the transition matrix calculated using the matrix exponential of F , and Q is the process noise matrix

calculated using the spectral density of $w(t)$. Δt_k is the discrete time step size. Our output measurement model now becomes

$$\mathbf{y}[t_k] = H\mathbf{x}[t_k] + \epsilon[t_k], \quad \epsilon[t_k] \sim N(0, \sigma^2), \quad (9)$$

where H is the measurement matrix that simply selects the outputs from the joint model.

This form allows us to calculate the filtered (i.e. backwards-looking) posterior distribution $p(\mathbf{x}[t_k] | \mathbf{y}[t_{1:k}], \theta)$ of the state $\mathbf{x}[t_k]$ given observations $\mathbf{y}[t_{1:k}]$ and hyperparameters θ , for $k = 1, \dots, T$, through application of Kalman filtering using the standard Kalman update equations [17]. Furthermore, we can also calculate the smoothing (i.e. backwards- and forwards-looking) posterior $p(\mathbf{x}[t_k] | \mathbf{y}[t_{1:T}], \theta)$ using the Rauch-Tung-Streibel smoother. The implementation of these combined sequential techniques is equivalent to Gaussian process regression [14, 18].

Kalman filtering therefore provides us with a method for sequentially estimating the state of the outputs *and* the latent inputs at each point in time given our data and the hyperparameters θ , which now include the ODE parameters. This sequential method provides a large efficiency gain over standard batch processing, and the Kalman filter equations also provide the necessary components to calculate the marginal data likelihood,

$$p(\mathbf{y}[t_{1:n}] | \theta) = \prod_{i=1}^n p(\mathbf{y}[t_i] | \mathbf{y}[t_{1:i-1}], \theta). \quad (10)$$

The usual approach to inference is to iteratively optimise θ by maximising this equation with gradient-based methods.

3.2.1. Nonlinear latent force models

During the prediction stage of Kalman filtering, we calculate the required cross-covariances between the outputs and the latent function by solving the necessary differential equations. However, these calculations are only tractable if our model is linear.

Consider the ODE presented in our problem formulation (1), in which nonlinear functions act on both x_i and $u(t)$. We can similarly construct the LTI SDE form of this model by again constructing a joint state vector $\mathbf{x}(t)$ such that

$$\frac{d\mathbf{x}(t)}{dt} = \mathbf{g}(\mathbf{x}(t), t) + L(\mathbf{x}(t), t)w(t). \quad (11)$$

However, exact calculation of the Kalman prediction equations in this case is not possible. Instead, the filtering and smoothing distributions are approximated with Gaussian distributions and numerically computed with cubature integration methods [19].

4. LATENT FORCE MODELS FOR SOUND

The Spear software [8] is used to obtain the sinusoidal partials from an audio recording. We then apply the above latent force modelling techniques to map the high-dimensional sinusoidal data to a controllable, one-dimensional latent function. In order for synthesis to be intuitively controllable, parameters must be physically meaningful and the learnt latent function must also be interpretable in a physical sense.

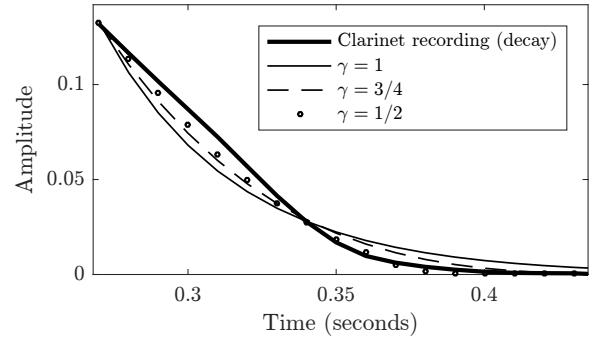


Figure 1: Comparison of amplitude model choice: $\gamma = 1$ represents the standard model for the amplitude of a sinusoid. Selecting $\gamma < 1$ alters the decay behaviour to more closely represent the real data obtained from the decay section of the second harmonic of a recording of a clarinet.

4.1. Modelling the Amplitude Data

Our approach is to consider M vibrational modes of a resonating object, modelled as in equation (4), assuming the modes have fixed frequencies. Given this assumption the problem becomes how to model the amplitude of the modes, $x_i(t), i = 1, \dots, M$.

The analytical solution when $u(t)$ is an impulse is $x_i(t) = \alpha_i e^{-\beta_i t}$. This inverse exponential equation can be modelled with a linear first-order ODE obtained by removing the second-order term from the mass-spring-damper system (3). By doing so we obtain equation (5), where $D_i = k_i/b_i$ and $S_i = 1/b_i$ are physically relevant parameters related to damping and stiffness of the system.

In practice, when observing real amplitude data (for which $u(t)$ will never truly be an impulse), we found that partials tend to decrease in a more linear fashion than can be described by equation (5). Therefore we propose an alternative model containing a parameter γ which alters the "linearity" of the decay of the signal,

$$\frac{dx_i(t)}{dt} + D_i x_i^\gamma(t) = S_i u(t). \quad (12)$$

We found that a suitable range of values for representing real audio data was $\gamma \in [\frac{1}{2}, 1]$, where a reduction in γ increases the linearity of the decay. $\gamma < 1/2$ represents an almost straight line, whilst $\gamma > 1$ would mean the data may never reduce to zero. No formal method for selecting γ is presented here, instead we visually inspect the amplitude data and select an appropriate value based on the decay behaviour. Figure 1 shows the comparison between different choices of γ .

Since predicted values of x_i can go negative, raising our x_i term to the power of $\gamma < 1$ can give unwanted complex results. Therefore in practice we take the real part of the x_i term. This compromises the smoothness of the model, but inference is still possible with the nonlinear filtering approach outlined in Section 3.2.1, numerically approximating the solutions to these equations rather than solving them analytically.

We aim to learn meaningful parameters representing damped modes which reduce to zero in the absence of input. As such it is advantageous for us to enforce a positivity constraint on input $u(t)$ via a function g . This has two major benefits. Firstly, the new excitation force $g(u(t))$ becomes interpretable as a physical entity; positive energy driving the system. Secondly, it encourages

the optimiser to learn damping coefficients D_i that are more physically realistic (i.e. larger / more damped), since they must enable the system to reduce to zero when $g(u(t)) = 0$, whereas in the unconstrained case this could be achieved via negative inputs rather than damping.

A reliable positivity constraint that ensures smoothness is the "softplus" rectification function,

$$g(u(t)) = \log(1 + e^{u(t)}). \quad (13)$$

Introducing this nonlinearity gives us our final model for the amplitude of the i^{th} damped vibrational mode of a sounding object:

$$\frac{dx_i(t)}{dt} + D_i \operatorname{Re}\{x_i^\gamma(t)\} = S_i g(u(t)), \quad (14)$$

which is the target system formulated in (1) with $g_x(x_i) = \operatorname{Re}\{x_i^\gamma\}$ and $g_u(u) = g(u)$.

4.2. Selecting the Modes

Optimising our parameters in the latent force model framework is a high-dimensional problem, since we have parameters D_i and S_i (and the initial conditions) to estimate for all M outputs, in addition to the hyperparameters of the Gaussian process kernel for the latent input (we use the Matérn covariance function). As such it is common for optimisation to get stuck in local minima, and choice of initial parameter settings can significantly affect the optimality of our outcome.

Furthermore, we assume our outputs (the modes) to be strongly correlated, such that a mapping to a low-dimensional space that maintains much of their behaviour exists. The introduction of partials that don't represent vibrational modes could compromise this assumption, in turn compromising the model's ability to represent the system.

We must therefore identify which partials in the sinusoidal model are representative of the vibrational modes. If our analysis signal has strong harmonic content (musical instruments, for example), then picking the modes / harmonics is straightforward. For inharmonic sounds (such as a hammer striking a metal plate), energy is distributed across the sinusoidal model, and there may be a strong noise component. In this case, selecting the modes is not as simple as selecting the largest M partials. In Figure 2, we analyse the frequency spectrum of the signal, designing a filter based on the shape of the spectrum. We invert the filter to flatten the data, allowing us to pick the modes of vibration from the peaks of the filtered spectrum.

Once we have selected our M modes, we scale the observed amplitude data to normalise their weighting prior to inference. Note that it is possible to assign importance to particular modes by altering the observation noise assumptions for a particular dimension of the Kalman filter. We calculate the median frequency value for each partial, and treat their frequency as fixed from this point onwards. Inference on the amplitude data is now performed using the techniques outlined in Section 3.2 with the model in equation (14).

4.3. Resynthesis with the State Space Model

After inference is performed, we obtain an optimised set of parameters θ , and a posterior distribution over the outputs and the latent input. We apply an inverse scaling operation to obtain the original magnitude weightings. The posterior distribution provides us with

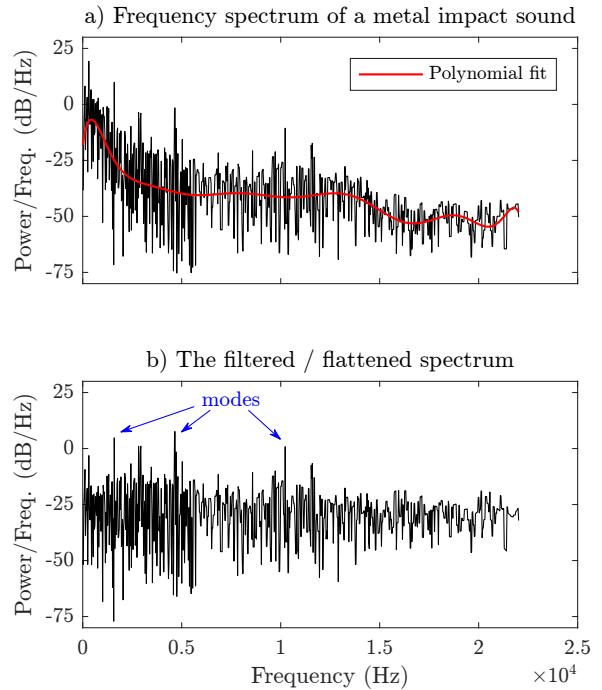


Figure 2: A filter is designed by fitting a polynomial to the shape of the frequency spectrum. The filter is inverted and applied to the signal to flatten the spectrum. Peaks in the flattened spectrum are then used to pick the vibrational modes of the signal.

information about the uncertainty of the prediction, and we can compare the posterior mean of the outputs to the analysis data to evaluate how much of the amplitude behaviour has been encoded.

Drawing samples from the distribution over the latent excitation function and passing them through the model constitutes resynthesis. Alternatively, to reproduce outputs faithful to the analysis data, we can pass the posterior mean through the model. To do so, we discretise equation (14) and restate it in state space form, solving it using the Euler method. The i^{th} output is therefore given by the discrete model

$$\begin{bmatrix} x_i[t_k] \\ \dot{x}_i[t_k] \end{bmatrix} = \begin{bmatrix} 1 & \Delta t_k \\ 0 & 0 \end{bmatrix} \begin{bmatrix} x_i[t_{k-1}] \\ \dot{x}_i[t_{k-1}] \end{bmatrix} + \begin{bmatrix} 0 \\ -D_i \end{bmatrix} x_i^\gamma[t_{k-1}] + \begin{bmatrix} 0 \\ S_i \end{bmatrix} g[u[t_k]], \quad (15)$$

where Δt_k is the time step size, chosen to be identical to the analysis step size in equation (8).

5. EXPRESSIVE REAL TIME SYNTHESIS AND SOUND MORPHING

An advantage of using a relatively simple state space model such as the one in equation (15) is its flexibility with regards to parameter control and time step size. We now illustrate how we can utilise these features to run our model in real time with user control, and to interpolate between parameter values to manipulate the sound timbre.

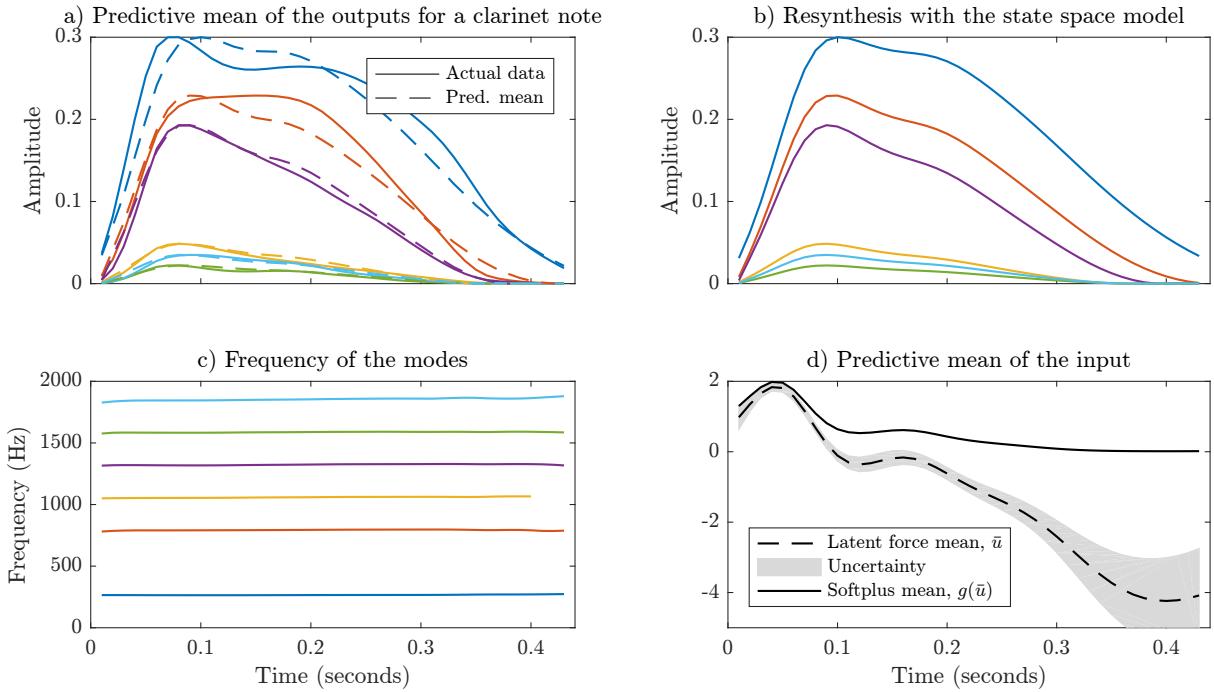


Figure 3: *Latent force modelling of a clarinet note.* 6 modes are picked based on their amplitude, and the predictive mean of the output distribution is compared to the real data (top left). The frequency data (bottom left) shows the modes are, in order of magnitude, the 1st, 3rd, 5th, 4th, 7th and 6th harmonics. The mean, \bar{u} , and 95% confidence interval (uncertainty) of the latent input u is shown (bottom right). $g(\bar{u})$ is fed through the state space model to resynthesise the output (top right). Low uncertainty results in resynthesis very similar to the predictive mean.

5.1. Real Time Synthesis

In the previous section Δt_k was fixed at the analysis time step size, corresponding to framewise modelling. During synthesis we can set the step size to be as large or small as required. Based on our desired sampling frequency, we modify Δt_k such that the model calculates sample-rate data and runs in real time.

This modification allows us to handle audio-rate input, which may be crucial for a synthesis model that requires expressive user control. As mentioned in Section 4.3, resynthesis can be performed by sampling from the posterior distribution over the latent excitation function and passing the sample through the model. However, with the aim of user-controllable synthesis in mind, and given that the excitation function is interpreted as physical energy forcing the system, it is possible to replace the mean of the latent distribution with a new function dependent on some user input.

We control the synthesis model with user input data corresponding to the pressure applied to a MIDI CC button or a force-sensing-resistor, scaling the data appropriately such that it has similar properties to the learnt latent input. Alternatively, we provide the user with a modifiable plot of the excitation function, which they can re-draw and modify to create new sounds.

5.2. Sound Morphing

Our linear time-invariant synthesis model has fixed stiffness and damping parameters corresponding to each mode. Adjusting these parameters has an impact on perceptual characteristics relating to timbre such as attack time, decay time and the modes' amplitudes

relative to one another. Individual modification of these parameters is possible, but not desirable if we wish to maintain coherence across dimensions. Instead, we interpolate parameters between models to create new sound timbres not present in the original recordings.

Prior to parameter interpolation we match the modes between models by ranking them in order of frequency. We also normalise the magnitude of the excitation functions, adjusting the stiffness parameters accordingly. For sounds without definable harmonic structure, pairing the modes is straightforward and simply based on their rank position. For harmonic sounds we must be careful to match the n^{th} harmonic in model A to the n^{th} harmonic in model B. If we fail to do so, interpolation of the frequency value will compromise the harmonic structure of the sound.

Once modes have been paired we perform linear interpolation of physical parameters S_i , D_i and the initial conditions, and logarithmic interpolation of the frequency. Synthesis in this manner negates the need for time-domain modification (such as time-stretching) usually associated with morphing [20].

6. RESULTS

In order to show the versatility of our approach we consider two case studies: musical instruments, demonstrated here by a short clarinet note, and impact sounds, demonstrated by the sound of metal being struck by a solid object. We then measure the accuracy of our reconstructed data for a number of recordings, and show the output produced by morphing between two different sounds.

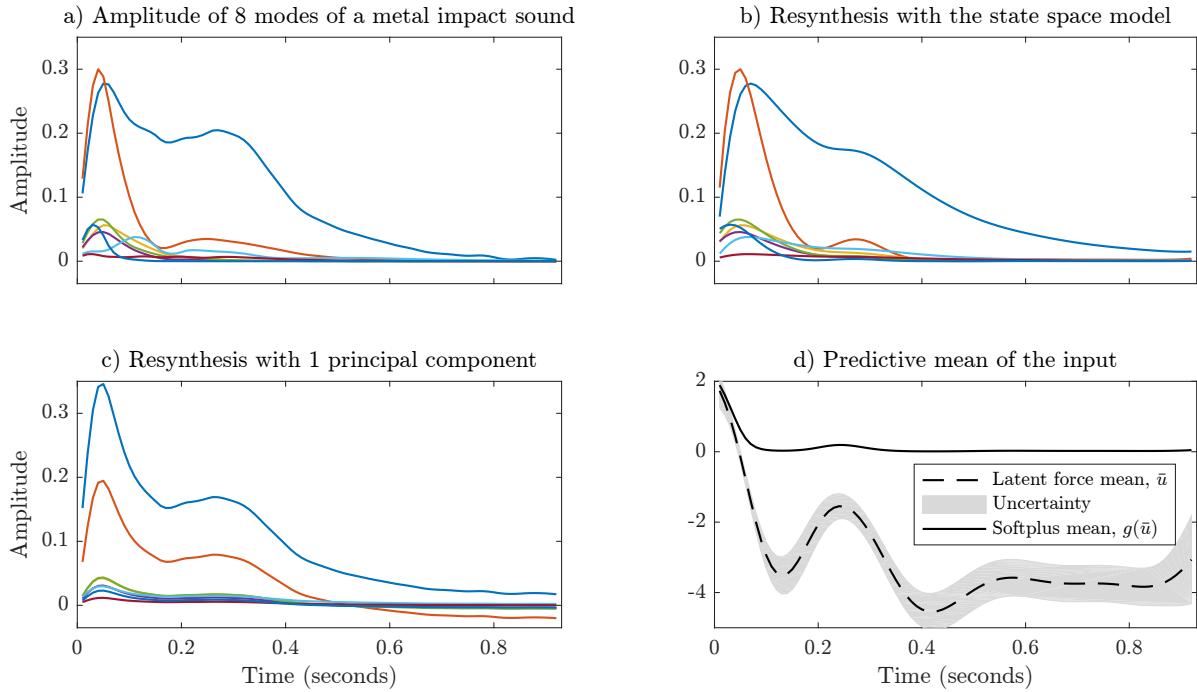


Figure 4: *Latent force modelling of a metal impact sound. The real analysis data shows some variation in behaviour between modes (top left). An increase in uncertainty in the posterior distribution after 0.1s reflects this fact (bottom right). The posterior mean of the latent distribution, \bar{u} , is fed through the state space model, and the result shows that much of the variable behaviour was captured (top right). PCA results are shown as a comparison, and we can see that the variable damping rates have not been reproduced (bottom left).*

6.1. Musical Instruments

Most musical instruments have strong harmonic structure, and the majority of the signal energy tends to be contained within relatively few sinusoids representing these harmonics. By inspecting the data and experimentally testing the results for various values for γ , we found that musical instruments tend to have a relatively linear decay, and a choice of $\gamma = 1/2$ fits the data best.

Figure 3 shows the results of latent force modelling of a clarinet note. The first 6 modes are considered, and on viewing the mean of the distribution of the outputs (Figure 3a), we can see that much of the behaviour has been captured in the model. The attack of the largest mode is partially altered to fit the shape of the other modes, since the simple mechanistic model struggles to encode peaks that are out of phase with each other. However, the variable damping rates have successfully been learnt, with the largest mode reducing to zero at a much slower rate than the smallest modes.

We plot the 95% confidence interval for the latent input (Figure 3d), and observe that the uncertainty in the learnt model increases towards the end of the signal, as some partials reduce to zero and their behaviour no longer correlates with the non-zero partials. The resynthesised outputs (Figure 3b) are almost identical to the predictive mean of the outputs when passing the mean of the latent input, $\bar{u}(t)$, through the model (14). This suggests that the observed degree of uncertainty is acceptable.

6.2. Impact Sounds

Impact sounds often lack clear harmonic structure, and energy is distributed across the frequency spectrum. In selecting just a small

number of modes, we risk losing much of the audio content. However, our selected modes are capable of reproducing much of the deterministic character of the signal. The remainder is treated as a residual, and not addressed here. We found that for impact sounds a model choice of $\gamma = 3/4$ was more appropriate since the decay rate varies as the amplitude decreases (in Figure 4a, the partials' gradient flatten out over time).

Figure 4a shows that for a metal impact sound large variation of behaviour occurred between modes. To account for this, a large variation of stiffness and damping parameters were learnt, enabling much of the behaviour to be captured. Comparing the synthesised outputs for the two largest modes in Figure 4b, we see that whilst they have a similar attack, encoded by the stiffness or sensitivity measure S_i , they have a very different decay, encoded by the damping measure D_i .

Uncertainty in the metal impact model (Figure 4d) increased more quickly than in the clarinet model, reflecting the fact that behaviour is less consistent across these vibrational modes than across the harmonics of the clarinet. In particular we observe an increase in uncertainty after the initial attack, when the modes' behaviour begins to diverge from one another.

6.3. Model Accuracy and Comparison with PCA

To evaluate our results we calculated the root-mean-square (RMS) error between the actual data and our synthesised outputs. This gives us a measure of our ability to reproduce the analysed sinusoidal partials. Readers are also invited to listen to the sound examples provided.

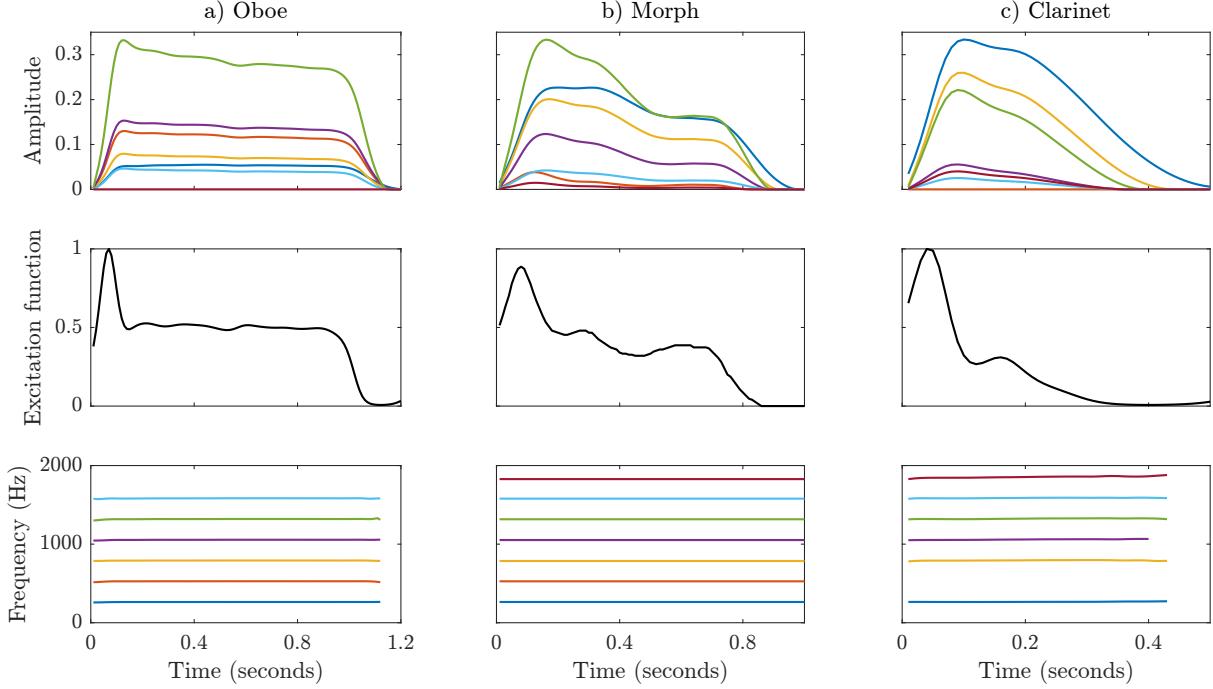


Figure 5: Sound morphing between an oboe and a clarinet. The modes of an oboe (left) are matched with the modes of a clarinet (right) and colour-coded based on their pairings. Since the modes represent harmonics, it is important to maintain the harmonic structure, so the 2nd mode of the oboe does not have a match. Similarly, the 6th mode of the clarinet is not matched. Stiffness and damping parameters are interpolated, and a user-drawn excitation function of arbitrary length is used to produce the morphed output (middle).

As a comparison, we run principal component analysis (PCA) on our amplitude data. PCA is another dimensionality reduction technique that similarly maps high-dimensional data to a lower-dimensional space through an input-output process (a simple scalar weighting), providing us with a set of orthogonal variables, called principal components, ranked in order of how much of the data's variance they describe.

Latent force modelling has many benefits over PCA, such as physical interpretability, model memory (PCA is an instantaneous mapping), the ability to introduce nonlinear mappings between inputs and outputs, and a probabilistic framework for calculating uncertainty and resampling new data (although probabilistic PCA techniques also exist). Regardless, PCA is a worthwhile comparison due to its simplicity and reliability.

Figure 4c shows the results of PCA on a metal impact sound. Using just one principal component to reproduce the 8-dimensional output fails to capture much of the behaviour, most notably the variable damping rates. With the one-dimensional LFM we are able to capture much more of the behaviour (Figure 4b). Note that it is possible to introduce more principal components, and also possible to run latent force modelling with more than one latent dimension, but this violates our assumption that the modes are produced by a common excitation function.

Table 1 compares the RMS error for latent force modelling and PCA for a number of audio recordings. The data is normalised to give equal weighting to each dimension of the model. When disparate behaviour occurs across dimensions, latent force modelling is more accurate than reconstruction with one principal component. For recordings in which the dimensions have high correla-

tion, such as the oboe, even one principal component sometimes outperformed the latent force model. This poor performance of the LFM for the oboe could be due to the optimisation procedure converging on a sub-optimal local minimum, or due to the fact that the oboe partials reduce to zero at an almost linear rate, and their behaviour was not fully captured by our choice of $\gamma = 1/2$, i.e. a more optimal choice of γ exists.

Audio recording	RMS error	
	LFM	PCA
Clarinet	0.0325	0.0593
Oboe	0.0189	0.0156
Piano	0.0441	0.0520
Metal impact	0.0377	0.0609
Wooden impact	0.0139	0.0291

Table 1: Root-mean-square (RMS) error between modal amplitude data and outputs of latent force modelling (LFM) and principal component analysis (PCA). The LFM outperforms PCA when disparate behaviour across dimensions is observed.

6.4. Morphing

Figure 5 shows the results of sound morphing between recordings of an oboe and a clarinet. A user-drawn excitation function is used as input to the morphed model (Figure 5b) and we observe the expected change in relative amplitudes. The modes of the oboe have

much faster decay times than the clarinet, and visual inspection of the morphed sound confirms that decay rates in between these two extremes are achieved.

7. CONCLUSIONS

The aim of this work was to demonstrate our ability to learn about the physical behaviour of sound from recordings. Such an approach will aid those looking to design and build synthesis models that are faithful to the real-world sounds we hear around us, whilst also providing opportunities for control and expression.

We utilised knowledge about the way in which objects vibrate to produce sound to construct a simple mechanistic model for the behaviour of sinusoidal modes. Although this model does not describe all the physical interactions that create sound, its simplicity enables the application of nonlinear latent force modelling techniques to infer physically relevant parameters from audio recordings, in addition to the excitation required to produce meaningful output.

After the learning process was complete, we demonstrated how to perform synthesis in this framework, adapting the model to run in real time with user control. We then provided a way to manipulate sound characteristics through parameter morphing. We showed how the model often outperforms PCA when attempting to map sinusoidal data to a one-dimensional control space, but noted how higher accuracy is not guaranteed since we rely on a high-dimensional optimisation procedure to find suitable parameter values.

As future work, the inference process would benefit greatly from intelligent selection of initial conditions to aid optimisation in finding appropriate solutions. Automatic identification of linearity measure γ , or inclusion of γ as a parameter to be optimised during inference, would also be highly beneficial. The introduction of additional latent functions would allow us to model more complex systems with multiple control inputs.

Subjective evaluation of our ability to reproduce the quality of a given audio recording was not presented here, but is necessary to further assess the suitability of our approach. Complex amplitude modulation is difficult to model if the modes' peaks are out of phase with each other, and a system that allows for variable frequency would greatly improve its applicability. Finally, consideration of the residual component of the signal is crucial for further development of these techniques.

8. REFERENCES

- [1] Jean Marie Adrien and Eric Ducasse, “Dynamic modeling of vibrating structures for sound synthesis, modal synthesis,” in *Audio Engineering Society 7th International Conference: Audio in Digital Times*, 1989.
- [2] Perry R. Cook, “Physically informed sonic modeling (PhISM): Synthesis of percussive sounds,” *Computer Music Journal*, vol. 21, no. 3, pp. 38–49, 1997.
- [3] Gerhard Eckel, Francisco Iovino, and René Caussé, “Sound synthesis by physical modelling with Modalys,” in *Proc. International Symposium on Musical Acoustics*, 1995, pp. 479–482.
- [4] Zhimin Ren, Hengchin Yeh, and Ming C. Lin, “Example-guided physically based modal sound synthesis,” *ACM Transactions on Graphics (TOG)*, vol. 32, no. 1, pp. 1, 2013.
- [5] Julius O. Smith, *Physical audio signal processing: For virtual musical instruments and audio effects*, W3K Publishing, 2010.
- [6] Mauricio A. Alvarez, David Luengo, and Neil D Lawrence, “Latent force models.,” in *International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2009, vol. 12, pp. 9–16.
- [7] Robert McAulay and Thomas Quatieri, “Speech analysis/synthesis based on a sinusoidal representation,” *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 34, no. 4, pp. 744–754, 1986.
- [8] Michael Klingbeil, “Software for spectral analysis, editing, and synthesis.,” in *International Computer Music Conference (ICMC)*, 2005.
- [9] Julien Bensa, Stefan Bilbao, Richard Kronland-Martinet, and Julius O. Smith, “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.
- [10] Lutz Trautmann and Rudolf Rabenstein, “Digital sound synthesis based on transfer function models,” in *IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*, 1999, pp. 83–86.
- [11] Perry R. Cook, *Real sound synthesis for interactive applications*, CRC Press, 2002.
- [12] Alfonso Pérez Carrillo, Jordi Bonada, Esteban Maestre, Enric Guaus, and Merlijn Blaauw, “Performance control driven violin timbre model based on neural networks,” *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 20, no. 3, pp. 1007–1021, 2012.
- [13] Mauricio A. Alvarez, David Luengo, and Neil D. Lawrence, “Linear latent force models using gaussian processes,” *IEEE Transactions on pattern analysis and machine intelligence*, vol. 35, no. 11, pp. 2693–2705, 2013.
- [14] Jouni Hartikainen and Simo Särkkä, “Sequential inference for latent force models,” in *Twenty-Seventh Annual Conference on Uncertainty in Artificial Intelligence (UAI-11)*, 2011, pp. 311–318.
- [15] Carl Edward Rasmussen and Christopher K. I. Williams, *Gaussian processes for machine learning*, MIT Press, 2006.
- [16] Jouni Hartikainen and Simo Särkkä, “Kalman filtering and smoothing solutions to temporal gaussian process regression models,” in *IEEE International Workshop on Machine Learning for Signal Processing (MLSP)*, 2010, pp. 379–384.
- [17] Mohinder S. Grewal, *Kalman filtering*, Springer, 2011.
- [18] Simo Särkkä, *Bayesian filtering and smoothing*, vol. 3, Cambridge University Press, 2013.
- [19] Jouni Hartikainen, Mari Seppanen, and Simo Sarkka, “State-space inference for non-linear latent force models with application to satellite orbit prediction,” in *29th International Conference on Machine Learning (ICML)*, 2012, pp. 903–910.
- [20] Marcelo Caetano and Xavier Rodet, “Musical instrument sound morphing guided by perceptually motivated features,” *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 21, no. 8, pp. 1666–1675, 2013.

ENERGY SHAPING OF A SOFTENING DUFFING OSCILLATOR USING THE FORMALISM OF PORT-HAMILTONIAN SYSTEMS

Marguerite Jossic

S3AM Team
 IRCAM - Institut d'Alembert
 Université Pierre et Marie Curie
 Paris, France
 marguerite.jossic@ircam.fr

Baptiste Chomette

Institut d'Alembert - Université Pierre et Marie Curie
 Paris, France

*David Roze, Thomas Hélie **

S3AM Team
 IRCAM - Université Pierre et Marie Curie
 Paris, France
 david.roze@ircam.fr

Adrien Mamou-Mani

IRCAM - Université Pierre et Marie Curie
 Paris, France

ABSTRACT

This work takes place in the context of the development of an active control of instruments with geometrical nonlinearities. The study focuses on Chinese opera gongs that display a characteristic pitch glide in normal playing conditions. In the case of the *xiaolu* gong, the fundamental mode of the instrument presents a softening behaviour (frequency glides upward when the amplitude decreases). Controlling the pitch glide requires a nonlinear model of the structure, which can be partially identified with experimental techniques that rely on the formalism of nonlinear normal modes. The fundamental nonlinear mode has been previously experimentally identified as a softening Duffing oscillator. This paper aims at performing a simulation of the control of the oscillator's pitch glide. For this purpose, the study focuses on a single-degree-of-freedom nonlinear mode described by a softening Duffing equation. This Duffing oscillator energy proves to be ill-posed - in particular, the energy becomes negative for large amplitudes of vibration, which is physically inconsistent. Then, the first step of the present study consists in redefining a new energetically well-posed model. In a second part, guaranteed-passive simulations using port-Hamiltonian formalism confirm that the new system is physically and energetically correct compared to the Duffing model. Third, the model is used for control issues in order to modify the softening or hardening behaviour of the fundamental pitch glide. Results are presented and prove the method to be relevant. Perspectives for experimental applications are finally exposed in the last section of the paper.

1. INTRODUCTION: PROBLEM STATEMENT

The Duffing equation $\alpha\ddot{x} + \kappa x + \Gamma x^3 = 0$ is commonly used as the simplest nonlinear system that models geometrical nonlinearities. However, the softening Duffing equation ($\Gamma < 0$) leads to an ill-posed problem since the energy is negative for large amplitudes of vibration. In this study, we propose to redefine a well-posed energy to overcome this issue.

Besides, the softening Duffing oscillator is quite interesting for the study of Chinese opera gongs[1] which can present either

* The contribution of this author has been done at laboratory STMS, Paris, within the context of the French National Research Agency sponsored project INFIDHEM. Further information is available at <http://www.lagep.cpe.fr/wwwlagep7/anr-dfg-infidhem-fev-2017-jan-2020/>

hardening or softening behaviour in standard playing conditions. Numerous studies detailed the nonlinear dynamical phenomena that occur in these instruments (internal resonances, chaos, pitch glide, harmonic distortions, etc.)[2][3][4] and their modelization (e.g. Von Karman plate model and nonlinear normal modes[5][6]). These works showed that most of these nonlinear features are the result of nonlinear interactions between vibration modes and require models with a high number of degree of freedom[7][8]. However, in the case of the pitch glide, the uni-modal approximation might be interesting: a single nonlinear mode is able to describe the dependence between the frequency and the amplitude of vibration[5]. Nonlinear normal modes are defined as invariant manifolds in phase space [6]. They are deduced from normal form theory which allows to compute an analytical nonlinear change of variables, from modal coordinates (X_p, \dot{X}_p) to new normal coordinates (R_p, \dot{R}_p) , by cancelling all the terms that are not dynamically important in the equations of motion [9]. The dynamics onto the p -th nonlinear normal mode is governed by the new normal coordinates (R_p, \dot{R}_p) and is written in free vibration regime:

$$\ddot{R}_p + \omega_p^2 R_p + (A_p + C_p)R_p^3 + B_p R_p \dot{R}_p^2 = 0 \quad (1)$$

where R_p and \dot{R}_p are the nonlinear mode displacement and velocity respectively, ω_p is the modal pulsation associated with the p -th mode, and A_p , C_p and B_p are coefficients that take into account the influence of other linear modes in the nonlinear mode dynamics. A first-order perturbative development of this equation [10] leads to the nonlinear relationship between the angular frequency of nonlinear free oscillations ω_{NL} and the amplitude a of the nonlinear mode's response at frequency ω_{NL} :

$$\omega_{NL} = \omega_p(1 + T_p a^2)$$

where the coefficient T_p is $T_p = \frac{3(A_p + C_p) + \omega_p^2 B_p}{8\omega_p^2}$. In practice, an experimental identification of T_p can be performed [11], but afterwards it is no longer possible to identify separately the coefficients A_p , C_p and B_p . However, in the case of the *xiaolu*, it can be shown that the fundamental nonlinear mode described in (1) is equivalent (first-order of perturbation method) to a softening Duffing equation with a negative cubic coefficient Γ_p :

$$\ddot{R}_p + \omega_p^2 R_p + \Gamma_p R_p^3 = 0 \quad (2)$$

Indeed, the T_p coefficient in this case is directly related to the Γ_p by $T_p = \frac{3\Gamma_p}{8\omega_p^2}$. Then, provided that:

$$\Gamma_p = A_p + C_p + \frac{\omega_p^2 B_p}{3}$$

the equation (1) is equivalent to (2). Consequently, the coefficient Γ_p and therefore the nonlinear mode can be experimentally identified with the measurement of T_p .

Finally, the softening Duffing model is assumed for two reasons: first, it provides a convenient basis to experimentally identify isolated nonlinear modes in the case of gongs; second, it gives the opportunity to define a single parameter well-posed energy that can be manipulated through energy shaping control in order to change its softening or hardening behaviour.

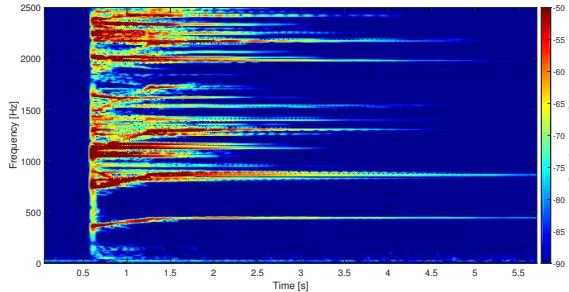


Figure 1: Spectrogram of the sound of a xiaolu after being struck by a mallet. The fundamental mode (~ 449 Hz) displays a softening behaviour.

This study aims at controlling the softening behaviour of the xiaolu gong's nonlinear fundamental mode, that we assume to be a in the form of the softening Duffing equation described by Eq. (2). The control process relies on guaranteed-passive simulations that use port-Hamiltonian approach. Port-Hamiltonian systems (PHS) are an extension of Hamiltonian systems, which represent passive physical systems as an interconnection of conservative, dissipative and sources components. They provide a unified mathematical framework for the description of various physical systems. In our case, the PHS formalism allows for the writing of an energy-preserving numerical scheme [12] in order to simulate and control the Duffing equation - note that other and more precise guaranteed-passive numerical schemes [13] are available but not used in this work. The first observation when tackling the control problem is that the softening Duffing equation defined by Eq. (2) is energetically ill-defined (Section 2). For large amplitudes of vibration, the total system energy, written with the PHS approach, becomes negative and thus, physically inconsistent. The first step of this study consists then to redefine the energy for the fundamental nonlinear mode. This new energy must be (i) as close as possible of the energy of the Duffing equation described in (2) and (ii) physically consistent. Secondly (Section 3), the Duffing energy and the new well-posed energy are both simulated using a guaranteed-passive numerical scheme that relies on the energy discrete gradient. Simulation results confirm the relevance of using the new energy for the control design. Thirdly (Section 4), control simulation of the fundamental mode's pitch glide is realized by shaping the system's new energy. The simulation results confirm the ability to modify the softening behaviour of the fundamental

mode thanks to the new energy defined in Section 2. Finally, conclusion and perspectives for further research offered by this study are discussed in Section 5.

2. PHYSICAL MODEL

2.1. Original Duffing model

2.1.1. Equation of motion

As explained before, the nonlinear normal mode associated with the fundamental mode is modelled by a softening Duffing oscillator, expressed as in Eq. (2) with an added viscous modal damping:

$$\ddot{x}(t) + 2\xi\omega_0\dot{x}(t) + \omega_0^2x(t) - \Gamma x^3(t) = f(t) \quad (3)$$

where x is the amplitude response of the nonlinear normal mode, ξ is the modal damping factor, ω_0 is the modal pulsation, Γ is the nonlinear cubic coefficient ($\Gamma > 0$) and f is the input acceleration. These parameters have been experimentally identified, however the description of the identification methods are beyond the scope of the paper. We assume then the following parameters values:

$$\begin{aligned} \xi &= 1.4 \cdot 10^{-3} \\ \omega_0 &= 2\pi \times 449 \text{ rad/s} \\ \Gamma &= 6, 7 \cdot 10^6 \text{ S.I} \end{aligned}$$

2.1.2. Dimensionless problem

For more convenience, equation (3) is written with dimensionless amplitude \tilde{x} and time \tilde{t} , defined such as $x = X_0\tilde{x}$ and $t = \tau\tilde{t}$. The Duffing equation (3) becomes:

$$\frac{X_0}{\tau^2}\ddot{\tilde{x}}(\tilde{t}) + 2\xi\omega_0^2\frac{X_0}{\tau}\dot{\tilde{x}}(\tilde{t}) + \omega_0^2X_0\tilde{x}(\tilde{t}) - \Gamma X_0^3\tilde{x}^3(\tilde{t}) = f(\tau\tilde{t})$$

that is:

$$\ddot{\tilde{x}}(\tilde{t}) + 2\xi\omega_0^2\tau\dot{\tilde{x}}(\tilde{t}) + \omega_0^2\tau^2\tilde{x}(\tilde{t}) - \Gamma\tau^2X_0^2\tilde{x}^3(\tilde{t}) = \frac{f(\tau\tilde{t})\tau^2}{X_0}$$

Choosing τ and X_0 such that $\tau = \frac{1}{\omega_0}$ and $X_0 = \sqrt{\frac{1}{\tau^2\Gamma}}$ leads to the following Duffing equation:

$$\ddot{\tilde{x}}(\tilde{t}) + \mu\dot{\tilde{x}}(\tilde{t}) + \tilde{x}(\tilde{t}) - \tilde{x}^3(\tilde{t}) = \tilde{f}(\tilde{t}) \quad (4)$$

where $\mu = 2\xi$ and $\tilde{f}(\tilde{t}) = \frac{f(\tau\tilde{t})\sqrt{\Gamma}}{\omega_0^3 m}$.

For sake of legibility, tilde will be omitted in the following.

2.2. Port-Hamiltonian approach

This section introduces some recalls on port-Hamiltonian systems in finite dimensions. The calculation of the Hamiltonian H of the Duffing system demonstrates that its potential energy H_1 is negative for some displacement values. A new equivalent positive definite potential energy H_{1*} is then defined for the control design.

2.2.1. General formulation

A port-Hamiltonian system of state $\mathbf{x}(t)$, input $\mathbf{u}(t)$ and output $\mathbf{y}(t)$ can be represented by the following differential equations [14]:

$$\begin{aligned}\dot{\mathbf{x}} &= (\mathbf{J}(\mathbf{x}) - \mathbf{R}(\mathbf{x}))\nabla_{\mathbf{x}}H(\mathbf{x}) + \mathbf{G}(\mathbf{x})\mathbf{u} \\ \mathbf{y} &= \mathbf{G}(\mathbf{x})^T\nabla_{\mathbf{x}}H(\mathbf{x})\end{aligned}$$

where $\dot{\mathbf{x}}$ is the time derivative of state \mathbf{x} , $\nabla_{\mathbf{x}}$ denotes the gradient with respect to state \mathbf{x} , $H(t)$ is a positive definite function that represents the total energy of the system, matrix \mathbf{J} is skew-symmetric and \mathbf{R} is positive definite ($\mathbf{R} \geq 0$). The power balance of the system can be expressed by the temporal energy variation of the system $\dot{H}(\mathbf{x}(t)) = \nabla_{\mathbf{x}}H(\mathbf{x}(t))^T\dot{\mathbf{x}}(t)$, that is:

$$\dot{H} = \underbrace{\nabla_{\mathbf{x}}H^T\mathbf{J}\nabla_{\mathbf{x}}H}_{=0 \ (\mathbf{J}=-\mathbf{J}^T)} - \underbrace{\nabla_{\mathbf{x}}H^T\mathbf{R}\nabla_{\mathbf{x}}H}_{\text{Dissipated power } \mathcal{P}_d > 0} + \underbrace{\mathbf{y}^T\mathbf{u}}_{\text{Entering power } \mathcal{P}_e}.$$

This power balance equation guarantees the system passivity. The variation of the system energy is expressed as the sum of elementary power functions corresponding to the storage, the dissipation and the exchanges of the system with the external environment. The dissipation term \mathcal{P}_d is positive because \mathbf{R} is positive definite. The power term \mathcal{P}_e denotes the energy provided to the system by the ports $\mathbf{u}(t)$ and $\mathbf{y}(t)$ (external sources).

The formulation $\dot{H}(\mathbf{x}) = \nabla_{\mathbf{x}}H(\mathbf{x})^T\dot{\mathbf{x}}$ underlines the fact that each power function can be expressed as the product of a flux ($[\nabla_{\mathbf{x}}H(\mathbf{x})^T]_i$ or $[\mathbf{x}]_i$) with its associated efforts ($[\dot{\mathbf{x}}]_i$ or $[\nabla_{\mathbf{x}}H(\mathbf{x})^T]_i$). A concrete example is given below with the Duffing oscillator described by Eq. (4).

2.2.2. Softening Duffing oscillator energy

The port-Hamiltonian system corresponding to the Duffing equation (4) can be defined as follow:

- State: $\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} l \\ p \end{bmatrix}$

where l and p are the string elongation and the mass momentum, respectively.

- Dissipation: $\mathcal{P}_D = \mu p^2 > 0$.

- Source: input $u = f$ and output $-y = p$.

where p is the velocity of the nonlinear normal mode. The total energy of the system H is the sum of the energy of the spring H_1 and the energy of the mass H_2 :

$$H(x_1, x_2) = H_1(x_1) + H_2(x_2) = \frac{1}{2}x_1^2 - \frac{1}{4}x_1^4 + \frac{1}{2}x_2^2$$

The flux and efforts associated with the energies H_1 and H_2 are given in Table 1.

	Spring	Mass
Energy	$H_1(x_1) = \frac{1}{2}x_1^2 - \frac{1}{4}x_1^4$	$H_2(x_2) = \frac{1}{2}x_2^2$
Effort	$\frac{dH_1(x_1)}{dl} = x_1 - x_1^3$	$\frac{dx_2}{dt}$
Flux	$\frac{dx_1}{dt}$	$\frac{dH_2(x_2)}{dx_2} = x_2$

Table 1: Energies and associated efforts and flux.

The port-Hamiltonian formulation of Eq. (4) can be deduced:

$$\begin{aligned}\dot{\mathbf{x}} &= (\mathbf{J} - \mathbf{R})\nabla_{\mathbf{x}}H(\mathbf{x}) + \mathbf{Gu} \\ \begin{pmatrix} \dot{x}_1 \\ \dot{x}_2 \end{pmatrix} &= \left[\begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix} - \begin{pmatrix} 0 & 0 \\ 0 & \mu \end{pmatrix} \right] \nabla_{\mathbf{x}}H(x_1, x_2) + \begin{pmatrix} 0 \\ 1 \end{pmatrix} u\end{aligned}$$

The physical interpretation of a system is often analyzed through the derivative of the potential energy (or forces), which is written in our case:

$$H'_1(x_1) = x_1 - x_1^3$$

The derivative H'_1 is plotted on Figure 2(a) with the potential energy derivative of the underlying linear system for comparison. One can see that looking at H'_1 does not give any information about the physical existence of the softening Duffing system. It is only by plotting the potential energy H_1 (see Figure 2(b)) that the softening system proves not to be physically defined for some displacement values x_1 : if $|x_1| > \sqrt{2}$, $H_1 < 0$ and H can be negative. Moreover, the equilibrium points $x_1 = 1$ and $x_1 = -1$ are saddle points, which means that the physical problem is restricted to $|x_1| < 1$.

The softening Duffing system (4) is then not energetically defined, and a new well-posed energy needs to be sought for the control design.

2.2.3. Well-posed problem

The aim of this paper is to seek functions H_{1*} such that:

- $\forall x \in \mathbb{R}, H_{1*}(x) \geq 0$
- H_{1*} increases on \mathbb{R}^+
- H_{1*} decreases on \mathbb{R}^-
- H''_{1*} is equivalent at order 2 to the dynamical stiffness of the softening Duffing $H''_1(x) = 1 - 3x^2$

$H''_1(x)$ corresponds to the first terms of the Taylor expansion of $x \rightarrow \exp(-3x^2)$. Then, one simple choice for H''_{1*} is:

$$\forall x \in \mathbb{R} \quad H''_{1*}(x) = \exp(-3x^2) = \sum_{n=0}^{+\infty} \frac{(-3)^n}{n!} x^{2n}$$

If we assume the conditions $H'_{1*}(0) = 0$ and $H_{1*}(0) = 0$, the simple and double integration of H''_{1*} give, for all $x \in \mathbb{R}$:

$$H'_{1*}(x) = \sum_{n=0}^{+\infty} \frac{(-3)^n}{n!(2n+1)} x^{2n+1} = x - x^3 + O(x^5)$$

$$H_{1*}(x) = \sum_{n=0}^{+\infty} \frac{(-3)^n}{(2n+2)(2n+1)!} x^{2n+2} = \frac{x^2}{2} - \frac{x^4}{4} + O(x^6)$$

and H_{1*} meets the requirements (5). Note that H'_{1*} can be expressed with the help of the error function erf which is defined for $x \in \mathbb{R}$ by:

$$\text{erf}(x) = \frac{2}{\sqrt{\pi}} \int_0^x e^{-\lambda^2} d\lambda = \frac{2}{\sqrt{\pi}} \sum_{n=0}^{\infty} (-1)^n \frac{x^{2n+1}}{n!(2n+1)}$$

leading to:

$$H'_{1*}(x) = \frac{\sqrt{\pi}}{2\sqrt{3}} \text{erf}(\sqrt{3}x)$$

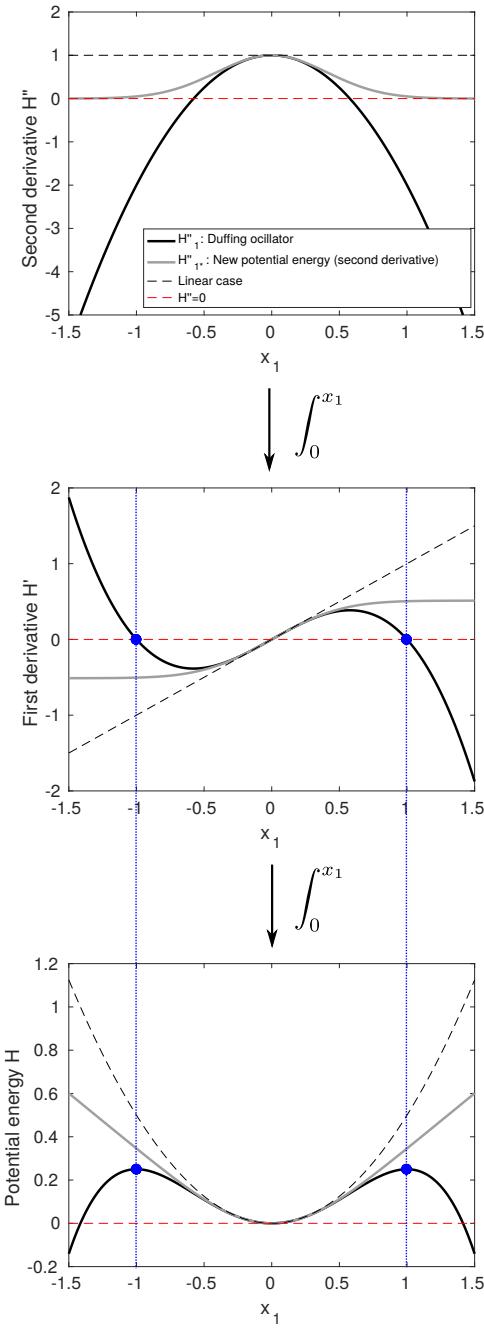


Figure 2: Second derivative, first derivative of potential energy and potential energy obtained by successive integrations with respect to the displacement x_1 , represented in the case of the Duffing oscillator, the new well-posed energy, and the linear case. Red dot line represents the null energy level, and blue dot line indicates the coincidence between the turning points of the potential energy and the zeros of its first derivative.

Finally, the new potential energy is:

$$\begin{aligned} H_{1*}(x) &= \sum_{n=0}^{+\infty} \frac{(-3)^n}{n!(2n+1)(2n+2)} x^{2n+2} \\ &= \frac{\sqrt{\pi}}{6} p(\sqrt{3}x) - \frac{1}{6} \end{aligned}$$

where $p(x) = x \times \text{erf}(x) + \frac{e^{-x^2}}{\sqrt{\pi}}$ is a primitive of the erf function. The potential energy H_{1*} and its derivative are represented in Figure 2 along with H_1 and the linear system potential energy for comparison. Note that H_{1*} is positive and equals the Duffing potential energy H_1 for small amplitudes x_1 .

3. SIMULATION

This section describes the MATLAB guaranteed-passive structure simulation relying on the discrete energy gradient. The simulation of the systems defined by (i) the Duffing potential energy H_1 and (ii) the well-posed problem defined by the new potential energy H_{1*} are performed and compared.

3.1. Discretization of the equations

The discrete-time equations to be solved for the port-Hamiltonian system are:

$$\begin{cases} \frac{\delta \mathbf{x}}{\delta t} = (\mathbf{J}(\mathbf{x}) - \mathbf{R}(\mathbf{x})) \nabla_d H(\mathbf{x}, \delta \mathbf{x}) + \mathbf{G}(\mathbf{x}) \mathbf{u} \\ \mathbf{y} = \mathbf{G}(\mathbf{x})^T \nabla_d H(\mathbf{x}, \delta \mathbf{x}) \end{cases} \quad (6)$$

where $\delta \mathbf{x} = [\delta x_1 \delta x_2]^T$ and δt ($\delta t = 1/f_s$ where $f_s = 44100$ Hz is the sampling frequency) are the discrete space and time step, respectively, and ∇_d denotes the discrete gradient defined by:

$$\begin{aligned} [\nabla_d H(\mathbf{x}, \delta \mathbf{x})]_n &= \frac{H_n(x_n + \delta x_n) - H_n(x_n)}{\delta x_n} \text{ if } \delta x_n \neq 0 \\ &= H'_n(x_n) \text{ else.} \end{aligned}$$

Matrices \mathbf{J} , \mathbf{R} and \mathbf{G} are defined as:

$$\begin{aligned} \mathbf{J} &= \begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix} \\ \mathbf{R} &= \begin{pmatrix} 0 & 0 \\ 0 & \mu \end{pmatrix} \\ \mathbf{G} &= \begin{pmatrix} 0 \\ 1 \end{pmatrix} \end{aligned}$$

Equation (6) is implicit and requires an iterative algorithm to be solved. In this work we use the Newton-Raphson method, which is written for time step k :

$$\delta \mathbf{x}^{(k+1)} = \delta \mathbf{x}^{(k)} - \mathbf{J}_F^{-1}(\delta \mathbf{x}^{(k)}) \mathbf{F}(\delta \mathbf{x}^{(k)}), \quad k \in \mathbb{N}$$

where \mathbf{J}_F is the Jacobian matrix of function \mathbf{F} defined by $\mathbf{F}(\delta \mathbf{x}) = \mathbf{0}$ that is:

$$\begin{aligned} \mathbf{F} &= \begin{pmatrix} F_1(\delta x_1, \delta x_2) \\ F_2(\delta x_1, \delta x_2) \end{pmatrix} \\ &= \frac{1}{\delta t} \begin{pmatrix} \delta x_1 \\ \delta x_2 \end{pmatrix} - \begin{pmatrix} 0 & 1 \\ -1 & -\mu \end{pmatrix} \begin{pmatrix} \nabla_d H_1(x_1, \delta x_1) \\ \nabla_d H_2(x_2, \delta x_2) \end{pmatrix} - \begin{pmatrix} 0 \\ 1 \end{pmatrix} u \end{aligned}$$

3.2. Duffing case

In the Duffing oscillator case, the discrete gradient is:

$$\begin{aligned} \nabla_d H_1(x_1, \delta x_1) &= \frac{H_1(x_1 + \delta x_1) - H_1(x_1)}{\delta x_1} \\ &= \frac{1}{2} (2x_1 + \delta x_1) - \frac{1}{4} (4x_1^3 + 6x_1^2 \delta x_1 + 4x_1 \delta x_1^2 + \delta x_1^3) \quad (7) \end{aligned}$$

and

$$\begin{aligned}\nabla_d H_2(x_2, \delta x_2) &= \frac{H_2(x_2 + \delta x_2) - H_2(x_2)}{\delta x_2} \\ &= x_2 + \frac{\delta x_2}{2}\end{aligned}$$

Then we have:

$$\begin{aligned}F_1(\delta x_1, \delta x_2) &= \frac{\delta x_1}{\delta t} - \nabla_d H_2(x_2, \delta x_2) \\ F_2(\delta x_1, \delta x_2) &= \frac{\delta x_2}{\delta t} + \nabla_d H_1(x_1, \delta x_1) + \mu \nabla_d H_2(x_2, \delta x_2) - u\end{aligned}$$

and the Jacobian matrix is:

$$J_F = \begin{pmatrix} \frac{1}{2} - \frac{3}{2}x_1^2 - 2x_1\delta x_1 - \frac{3}{4}\delta x_1^2 & \frac{-1}{2} \\ \frac{1}{\delta t} + \frac{\mu}{2} & \end{pmatrix}$$

The simulation of the Duffing oscillator is performed with an excitation force $f(t) = f_0 \cdot g(t)$ where g is an impulse. The potential energy H_1 versus the simulated displacement x_1 , for the limit excitation $f_0 = f_{max} = 9.5 \cdot 10^7$, is plotted in Figure 3. The spectrogram of the oscillator response x_1 is also shown in Figure 4 and highlights the softening behaviour of the oscillator. If the value of f_0 exceed f_{max} , the simulation fails since $|x_1| > 1$ (see Section 2). We will see in the next section that this difficulty can be overcome with the definition of a new potential energy.

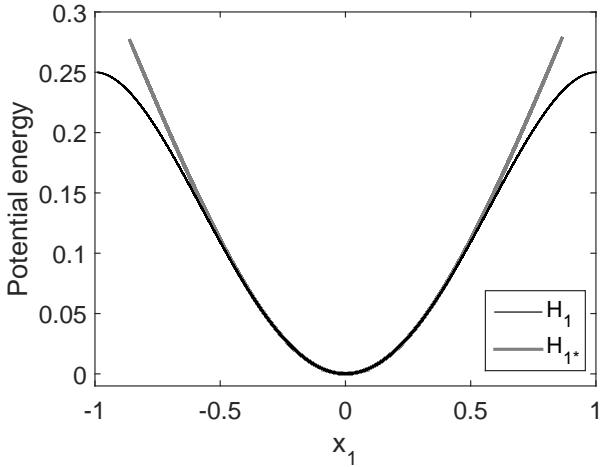


Figure 3: Potential energies as a function of the displacement resulting from simulations of the Duffing oscillator (black) and the new well-posed model (grey). The excitation force is $f_{max} = 9.5 \cdot 10^7$. In the case of the Duffing oscillator, increasing the input force makes the computation fail because $|x_1| > 1$ (see Fig 2).

3.3. Well-posed problem

In the case of the new problem defined by H_{1*} the discrete gradient is

$$\begin{aligned}\nabla_d H_{1*}(x_1, \delta x_1) &= \frac{H_{1*}(x_1 + \delta x_1) - H_{1*}(x_1)}{\delta x_1} \\ &= \frac{\sqrt{\pi}}{6} \frac{p(\sqrt{3}(x_1 + \delta x_1)) - p(\sqrt{3}x_1)}{\delta x_1} \quad (8)\end{aligned}$$

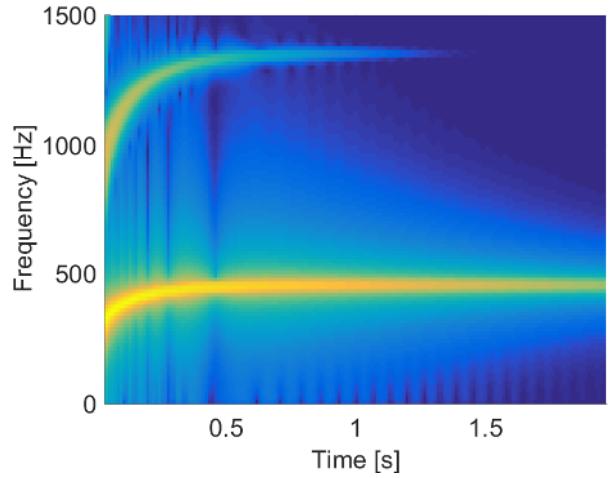


Figure 4: Spectrogram of the simulated Duffing system response x_1 for an input force $f_{max} = 9.5 \cdot 10^7$.

$\nabla_d H_2(x_2, \delta x_2)$ is the same as in the Duffing case. We can then deduce the Jacobian matrix:

$$J_F = \begin{pmatrix} \frac{1}{\delta t} & \frac{-1}{2} \\ \frac{\sqrt{\pi}}{6} \frac{\sqrt{3}p'(\sqrt{3}(x_1 + s_1))s_1 - p(\sqrt{3}(x_1 + s_1)) + p(\sqrt{3}x_1)}{s_1^2} & \frac{1}{\delta t} + \frac{\mu}{2} \end{pmatrix}$$

It is possible to simulate the system associated with the new energy H_{1*} for an excitation force $f_0 = f_{max}$, as in section 3.2. The potential energy H_{1*} versus the simulated displacement is plotted in Figure 3 and can be compared with the potential energy issued from the Duffing simulation performed in section 3.2. However, contrary to the Duffing simulation, the input force f_0 can now be increased without failing the computation. This is demonstrated by running simulations with an input force $f_0 = 2 \cdot 10^8 > f_{max}$. The resulting potential energy H_{1*} is plotted in Figure 5 and the spectrogram of the temporal displacement x_1 is represented in Figure 6: the softening behaviour of the system has been increased.

4. CONTROL DESIGN

In this section, we present the nonlinear mode's pitch glide control design. The former Duffing model presented in Section 2 is abandoned and replaced by the model associated with the new potential energy H_{1*} defined in Section 3. The control of the pitch glide is realized by shaping the energy H_{1*} . The principles of energy shaping are recalled in the first section, and the pitch glide control simulations are presented in the second section.

4.1. Energy reshaping

Let H_{1*}^ϵ be the potential energy parameterized by $\epsilon \neq 0$ such that:

$$H_{1*}^\epsilon(x) = \frac{x^2}{2} - \epsilon \frac{x^4}{4} + O(x^6) \quad (9)$$

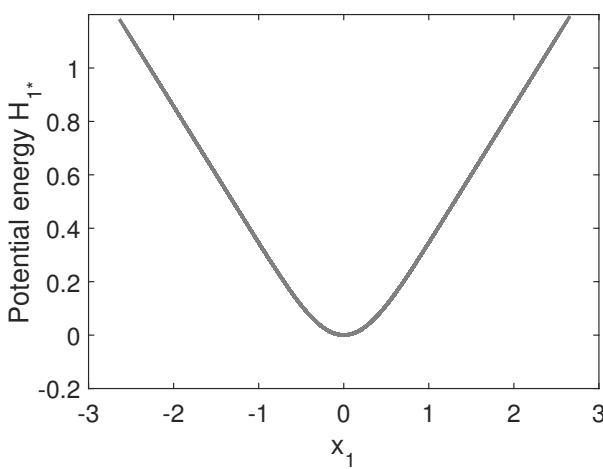


Figure 5: Potential energy H_{1*} as a function of the displacement x_1 resulting from simulations of the new well-posed system for an input force $f_0 = 2 \cdot 10^8 > f_{max} = 9.5 \cdot 10^7$.

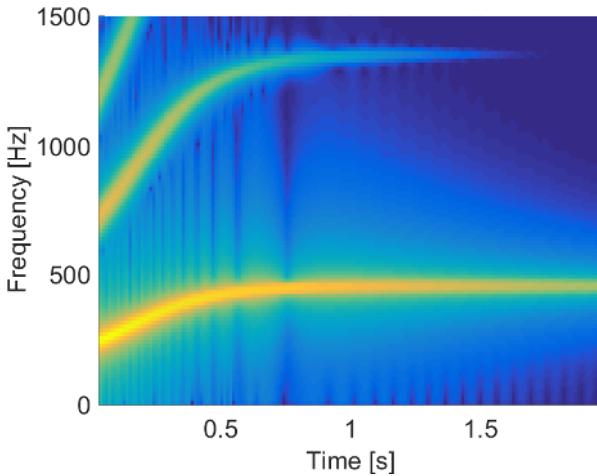


Figure 6: Spectrogram of the simulated well-posed system response x_1 for an input force $f_0 = 2 \cdot 10^8 > f_{max} = 9.5 \cdot 10^7$.

This potential energy can be easily calculated using the same arguments than in section 2.2.3:

$$\begin{aligned} H_{1*}^\epsilon(x) &= \frac{\sqrt{\pi}}{6\epsilon} p(\sqrt{3\epsilon}x) - \frac{1}{6\epsilon} \quad \text{for } \epsilon > 0 \quad (\text{softening}) \\ H_{1*}^\epsilon(x) &= \frac{\sqrt{\pi}}{6\epsilon} p_i(\sqrt{3\epsilon}x) + \frac{1}{6\epsilon} \quad \text{for } \epsilon < 0 \quad (\text{hardening}) \end{aligned}$$

where $p_i(x) = x \times \text{erf}i(x) - \frac{e^{-x^2}}{\sqrt{\pi}}$ is a primitive of the imaginary error function $\text{erf}i$.

The energy shaping control principle is as follows: if a system is defined by the energy $H_{1*}^{\epsilon_1}$, energy shaping consists in changing the system potential energy from $H_{1*}^{\epsilon_1}$ to $H_{1*}^{\epsilon_2}$ ($\epsilon_1 \neq \epsilon_2$) by replacing at each time step t the input force $f(t)$ by:

$$f_1(t) = f(t) + (\nabla_d H_1^{\epsilon_1} - \nabla_d H_{1*}^{\epsilon_2})(x_1(t))$$

The gradient term $+\nabla_d H_1^{\epsilon_1}$ aims at "cancelling" the original system defined by $H_{1*}^{\epsilon_1}$ whereas the gradient term $-\nabla_d H_{1*}^{\epsilon_2}$ introduces the new target system defined by $H_{1*}^{\epsilon_2}$. In our case, note that the original system is defined by $\nabla_d H_{1*} = \nabla_d H_{1*}^{\epsilon_1=1}$.

4.2. Control simulations

Control simulations using energy shaping principle are performed. The simulation parameters are:

- initial (uncontrolled) energy: $H_{1*}^{\epsilon_1} = H_{1*}$
- target energy: $H_{1*}^{\epsilon_2}$, with $\epsilon_2 \in \{0.0746, 0.746, 1.79, -2\}$
- input force: $f_0 = 2 \cdot 10^8$

Note that $\epsilon_2 > 0$ and $\epsilon_2 < 0$ leads to a softening and hardening behaviour, respectively.

Figure 7 presents the potential energy $H_{1*}^{\epsilon_2}$ computed from the simulated responses for the different values of ϵ_2 . Theoretical quadratic energy of the underlying linear system is also plotted to distinguish softening from hardening behaviour. The results show that positive control parameter ϵ_2 leads to a softening behaviour (which increases with the value of ϵ_2), whereas negative value of ϵ_2 results in a hardening behaviour, as expected. This is confirmed by looking at the pitch glide variation of the system response, in Figure 8 to 11.

These results underline the benefits of the definition of the new energy H_{1*} , i.e. the ability to compute systems dynamics with important pitch glide (downward and upward) caused by both large input forces and nonlinear coefficients.

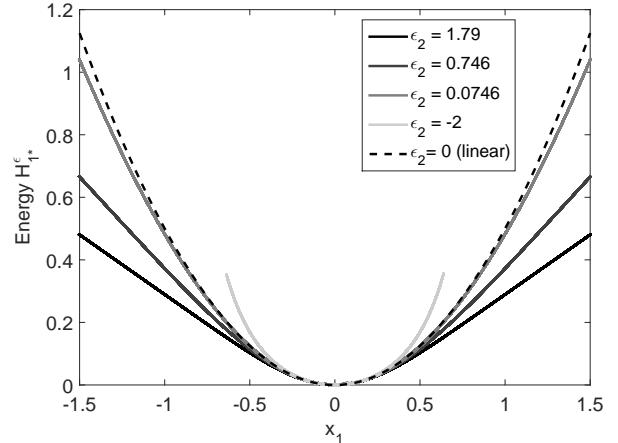


Figure 7: H_{1*}^ϵ energies of the controlled system for different values of control parameter ϵ_2 .

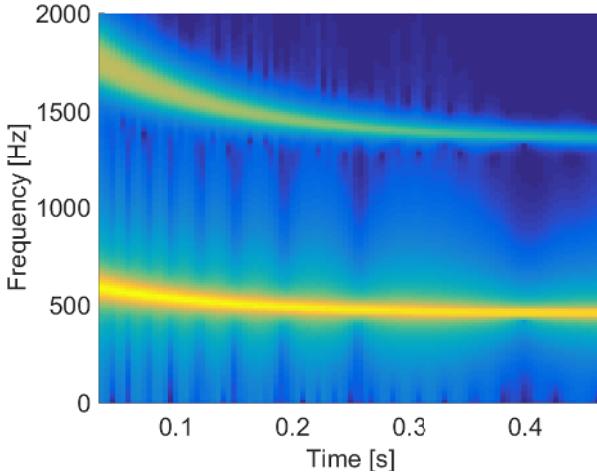


Figure 8: Spectrogram of the system response x_1 for a control parameter value $\epsilon_2 = -2$.

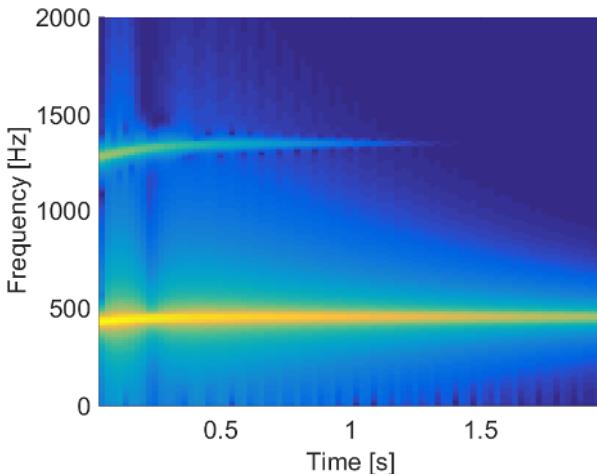


Figure 9: Spectrogram of the system response x_1 for a control parameter value $\epsilon_2 = 0.0746$

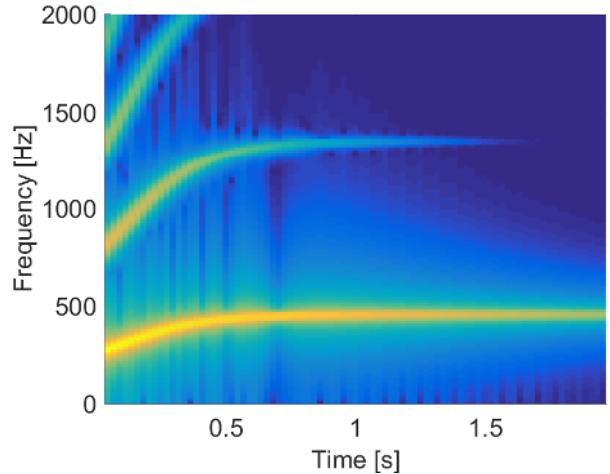


Figure 10: Spectrogram of the system response x_1 for a control parameter value $\epsilon_2 = 0.746$

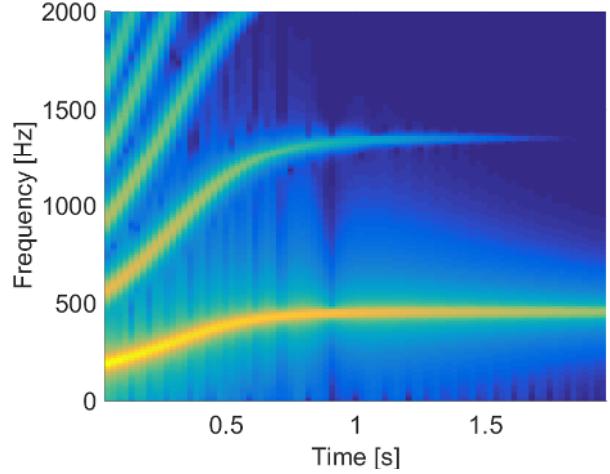


Figure 11: Spectrogram of the system response x_1 for a control parameter value $\epsilon_2 = 1.791$

5. CONCLUSION

This paper has introduced a port-Hamiltonian formulation of a *xi-aoluo* gong's fundamental nonlinear mode described by a softening Duffing oscillator. First, the calculation of the Duffing energy highlighted an inconsistent potential energy that has led to the re-definition of a well-posed potential energy. Guaranteed-passive simulations of the system associated to this new energy prove to overcome the stability problem encountered with the ill-posed Duffing modelling. The new energy formulation has then been used in successful energy shaping control simulations, in order to modify the system's nonlinear behaviour in a more hardening or more softening way.

This work represents the first step toward the development

of an experimental control of a real *xiaolu* gong. However, the various nonlinear phenomena encountered in gong's dynamics, in particular internal resonances (energy exchanges between modes), underline the limitation of a single nonlinear mode modelisation. The control of the instrument pitch glide may require the identification of a MDOF model with interconnected port-Hamiltonian systems.

6. ACKNOWLEDGMENTS

This work was funded by the PhD grant of Marguerite Jossic (Université Pierre et Marie Curie, France) and the French National Research Agency sponsored project INFIDHEM.

7. REFERENCES

- [1] N.H. Fletcher, “Nonlinear frequency shifts in quasispherical-cap shells: Pitch glide in chinese gongs,” *The Journal of the Acoustical Society of America*, vol. 78, no. 6, pp. 2069–2073, 1985.
- [2] Cyril Touzé M. Ducceschi and S. Bilbao, “Nonlinear vibrations of rectangular plates: Investigation of modal interaction and coupling rules,” *Acta Mechanica*, pp. 213–232, 2014.
- [3] Thomas D Rossing and NH Fletcher, “Nonlinear vibrations in plates and gongs,” *The Journal of the Acoustical Society of America*, vol. 73, no. 1, pp. 345–351, 1983.
- [4] N.H. Fletcher and T.D. Rossing, *The Physics of musical instruments*, Springer-Verlag, 1998.
- [5] A. Chaigne C. Touzé O. Thomas, “Hardening/softening behaviour in non-linear oscillations of structural systems using non-linear normal modes,” *Journal of Sound and Vibration*, pp. 77–101, 2004.
- [6] G. Kerschen, M. Peeters, J.C. Golinval, and A.F. Vakakis, “Nonlinear normal modes, part i: A useful framework for the structural dynamicist,” *Mechanical Systems and Signal Processing*, vol. 23, no. 1, pp. 170–194, 2009.
- [7] C. Touzé M. Ducceschi, O. Cadot and S. Bilbao, “Dynamics of the wave turbulence spectrum in vibrating plates: a numerical investigation using a conservative finite difference scheme,” *Physica D*, pp. 73–85, 2014.
- [8] S. Bilbao, “A family of conservative finite difference schemes for the dynamical von karman plate equations,” *Numerical Methods for Partial Differential Equations*, pp. 193–216, 2008.
- [9] Cyril Touzé and M Amabili, “Nonlinear normal modes for damped geometrically nonlinear systems: application to reduced-order modelling of harmonically forced structures,” *Journal of sound and vibration*, vol. 298, no. 4, pp. 958–981, 2006.
- [10] Ali Hasan Nayfeh and Dean T. Mook, *Nonlinear oscillations*, J. Wiley, 1995.
- [11] Simon Peter, Robin Riethmüller, and Remco I. Leine, *Tracking of Backbone Curves of Nonlinear Systems Using Phase-Locked-Loops*, pp. 107–120, Springer International Publishing, Cham, 2016.
- [12] A. Falaize and T. Hélie, “Passive guaranteed simulation of analog audio circuits: A port-hamiltonian approach,” *Applied Sciences (Balcan Society of Geometers)*, vol. 6, pp. 273 – 273, 2016.
- [13] N. Lopes, T. Hélie, and A. Falaize, “Explicit second-order accurate method for the passive guaranteed simulation of port-hamiltonian systems,” *IFAC-PapersOnLine*, vol. 48, no. 13, pp. 223 – 228, 2015.
- [14] V. Duindam, A. Macchelli, S. Stramigioli, and H. Bruyninckx, *Modeling and Control of Complex Physical Systems: The Port-Hamiltonian Approach*, Springer, 2009.
- [15] S. Benacchio, B. Chomette, A. Mamou-Mani, and V. Finel, “Mode tuning of a simplified string instrument using time-dimensionless state-derivative control,” *Journal of Sound and Vibration*, vol. 334, pp. 178–189, 2015.
- [16] Antoine Chaigne, Cyril Touzé, and Olivier Thomas, “Nonlinear vibrations and chaos in gongs and cymbals,” *Acoustical science and technology*, vol. 26, no. 5, pp. 403–409, 2005.
- [17] 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.
- [18] M. Jossic, A. Mamou-Mani, B. Chomette, D. Roze, F. Ollivier, and C Josserand, “Modal active control of chinese gongs,” *Journal of the Acoustical Society of America*, 2017, Submitted.
- [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] T. Meurisse, A. Mamou-Mani, S. Benacchio, B. Chomette, V. Finel, D. B. Sharp, and R. Caussé, “Experimental demonstration of the modification of the resonances of a simplified self-sustained wind instrument through modal active control,” *Acta Acustica*, vol. 101, no. 3, pp. 581–593(13), 2015.
- [21] J.P. Noel and G. Kerschen, “Nonlinear system identification in structural dynamics: 10 more years of progress,” *Mechanical Systems and Signal Processing*, vol. 83, pp. 2–35, 2017.
- [22] Andre Preumont, *Vibration control of active structures: an introduction*, vol. 50, Springer Science & Business Media, 2012.

ON ITERATIVE SOLUTIONS FOR NUMERICAL COLLISION MODELS

Vasileios Chatzioannou

Department of Music Acoustics,
University of Music and
Performing Arts Vienna, Austria
chatzioannou@mdw.ac.at

Sebastian Schmutzhard

Department of Music Acoustics,
University of Music and
Performing Arts Vienna, Austria
schmutzhard@mdw.ac.at

Stefan Bilbao

Acoustics and Audio Group,
University of Edinburgh
Edinburgh, UK
sbilbao@staffmail.ed.ac.uk

ABSTRACT

Nonlinear interactions between different parts of musical instruments present several challenges regarding the formulation of reliable and efficient numerical sound synthesis models. This paper focuses on a numerical collision model that incorporates impact damping. The proposed energy-based approach involves an iterative solver for the solution of the nonlinear system equations. In order to ensure the efficiency of the presented algorithm a bound is derived for the maximum number of iterations required for convergence. Numerical results demonstrate energy conservation as well as convergence within a small number of iterations, which is usually much lower than the predicted bound. Finally, an application to music acoustics, involving a clarinet simulation, shows that including a loss mechanism during collisions may have a significant effect on sound production.

1. INTRODUCTION

Collisions inherently take place during sound production from musical instruments [1]. Recent attempts have been therefore made in order to incorporate numerical collision models to physics-based sound synthesis algorithms [2, 3, 4]. The prevalent approach towards efficient schemes that may cover the whole audio range is to use time-stepping algorithms, such as the finite difference method [5], digital waveguides [6] and modal-based approaches [7, 8]. This requires discrete-time modelling of nonlinear interactions, a problem that gives rise to several challenges, such as existence and uniqueness of solutions to the underlying nonlinear equations, as well as the guarantee of numerical stability.

Although impact losses are often neglected in music acoustics applications, it has been proposed to include this effect by using the Hunt-Crossley impact model [3, 9, 10, 11]. Such a practise may result in minor, yet acoustically significant alterations of the synthesised sounds, as will be shown in Section 4. The accuracy of numerical solutions to this model equations has been the subject of a recent study [12], where a correction-based method was proposed to accurately approximate the velocity of impacting objects, based on enforcing numerical energy consistency. In the present work a solver is employed, that is shown to provide approximations of high accuracy, without the need of a post-processing step. Furthermore, since iterative solvers are employed in order to numerically solve the (nonlinear) model equations, special attention is devoted to the convergence speed of the algorithm, by calculating the maximum number of required iterations. Convergence within a given number of iterations is particularly useful in applications where an efficient solver is sought after, as for example real-time sound synthesis.

Section 2 presents the Hunt-Crossley impact model, along with an energy-based numerical formulation. Section 3 incorporates

this model to the simulation of a damped harmonic oscillator, with or without the presence of external forces. An iterative solution is carried out, for which a bound on the number of required iterations is calculated. Section 4 presents an application of the formulated model to musical instruments, in terms of a clarinet tone simulation and Section 5 discusses the findings of the current study in the context of acoustics research.

2. IMPACT MODELING

Consider a mass approaching a rigid barrier from below, with contact occurring at $y = 0$. The Hunt-Crossley repelling force can be defined as

$$f_c = -k_c |y|^\alpha - \lambda_c |y|^\alpha \frac{dy}{dt}, \quad (1)$$

where $|y|^\alpha = h(y)y^\alpha$, $\alpha \geq 1$ is a power-law exponent and $h(y)$ denotes the Heaviside step function. This non-negative term represents the compression of the mass while in contact with the barrier [2, 10, 13], a model which has been shown to be in agreement with experimental measurements in musical instruments [14, 15, 16].¹ The constant k_c represents stiffness and λ_c is a damping constant. The negative sign indicates that this force is acting against the motion of the mass ‘through’ the barrier. Newton’s second law can be used to derive the equation of motion of the system

$$m \frac{d^2y}{dt^2} = f_c, \quad (2)$$

where m represents mass. For this system of lumped contact Pappeti et al. [12] derived an analytic expression for the energy H as a function of the velocity v , which reads

$$H(v) = \frac{m}{2} v^2 - \frac{m}{r} (v - v_{im}) + \frac{m}{r^2} \ln \left| \frac{1 + rv}{1 + rv_{im}} \right|, \quad (3)$$

where v_{im} is the velocity with which the mass hits the barrier (impact velocity) and $r = \lambda_c/k_c$ is a damping factor. This formula may be used to compare numerically obtained results with an analytical solution.

An energy-based formulation of the system may be derived by defining the collision potential

$$V_c = \frac{k_c}{\alpha + 1} |y|^{\alpha+1}. \quad (4)$$

The collision force can then be written as [13]

$$f_c = -\frac{\partial V_c}{\partial y} - r \frac{\partial V_c}{\partial t}. \quad (5)$$

¹Note that in cases where the collision is assumed to be rigid (see, e.g. [4, 8]), this term corresponds to an artificial penalisation.

Following [10] (2) is cast into Hamiltonian form as

$$\frac{dy}{dt} = \frac{\partial T}{\partial p} \quad (6a)$$

$$\frac{dp}{dt} = -\frac{\partial V_c}{\partial y} - r \frac{\partial V_c}{\partial t}, \quad (6b)$$

where $T = p^2/(2m)$ represents the kinetic energy, p being the conjugate momentum. Similar formulations, including losses within an energy balanced framework, have been recently derived using a port-Hamiltonian formulation (see, e.g. [17]).

2.1. Numerical formulation

System (6) can be discretised by employing mid-point derivative approximations for all terms (see, e.g. [2, 3]). The approximation to the continuous variable $y(t)$ at time $n\Delta t$, where Δt is the sampling interval, is denoted by y^n . Then (6) is discretised as

$$\frac{y^{n+1} - y^n}{\Delta t} = \frac{T(p^{n+1}) - T(p^n)}{p^{n+1} - p^n} \quad (7a)$$

$$\frac{p^{n+1} - p^n}{\Delta t} = -\frac{V_c(y^{n+1}) - V_c(y^n)}{y^{n+1} - y^n} - r \frac{V_c(y^{n+1}) - V_c(y^n)}{\Delta t}. \quad (7b)$$

Defining the normalised momentum $q^n = p^n \Delta t / (2m)$ yields

$$y^{n+1} - y^n = q^{n+1} + q^n \quad (8a)$$

$$q^{n+1} - q^n = -\frac{\Delta t^2}{2m} \frac{V_c(y^{n+1}) - V_c(y^n)}{y^{n+1} - y^n} - r \frac{\Delta t}{2m} (V_c(y^{n+1}) - V_c(y^n)). \quad (8b)$$

Using the auxiliary variable $x = y^{n+1} - y^n$ leads to the following nonlinear equation in x

$$F(x) = x - 2q^n + \frac{\Delta t^2}{2m} \frac{V_c(y^n + x) - V_c(y^n)}{x} + r \frac{\Delta t}{2m} (V_c(y^n + x) - V_c(y^n)) = 0. \quad (9)$$

Note that

$$\lim_{x \rightarrow 0} F(x) = \frac{\Delta t^2}{2m} V'_c(y^n) - 2q^n, \quad (10)$$

where V'_c signifies taking the derivative of V_c with respect to position. This can be used to avoid singularities in $F(x)$. Equation (9) can be solved using, e.g. the Newton-Raphson or the bisection method. A bound on the required number of iterations for these methods can be obtained as shown in Section 3.3. Existence and uniqueness of solutions for (9) can be proven, as explained in [2], using the convexity of V_c and the positivity of V'_c which imply that $F' \geq 1$ and $F'' \geq 0$. Displacement and momentum can be subsequently updated using

$$\begin{aligned} y^{n+1} &= x + y^n \\ q^{n+1} &= x - q^n, \end{aligned} \quad (11)$$

whence p^{n+1} is also obtained. For energy conserving (Hamiltonian) systems this method can be shown to conserve the numerical energy within machine precision in implementations on digital

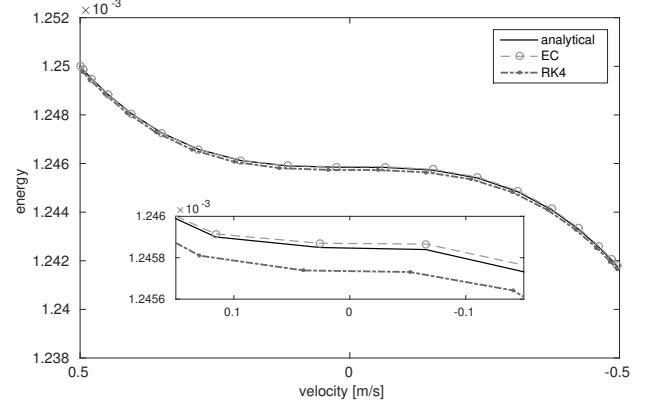


Figure 1: Simulation of a mass colliding with a barrier, using the Hunt-Crossley model. The system energy of the presented method and the fourth order Runge-Kutta method during the impact is compared to the continuous (analytical) solution from (3). A zoomed area shows a more clearer comparison of the approximation methods.

processors [13] and is hence labelled EC for the remainder of this text.

Following [12] the presented method is compared to the analytical solution (3) and to a higher order approximation given by a fourth order Runge-Kutta method (RK4). Figure 1 shows the energy during the impact as a function of the compression velocity. The parameters' values used for the simulations (listed in Table 1) are taken from [12]. It can be observed that the EC method accurately reproduces the analytical result, outperforming the higher-order Runge-Kutta method. This is a result of the exact energy-conserving nature of this algorithm, in the case of Hamiltonian systems, that is projected here to a dissipative case.

Table 1: Parameters used in the impact model.

mass	$m = 0.01 \text{ kg}$
stiffness constant	$k_c = 10^7 \text{ N/m}^\alpha$
damping factor	$r = 0.01 \text{ s/m}$
exponent	$\alpha = 1.3$
impact velocity	$v_{im} = 0.5 \text{ m/s}$
sampling rate	$f_s = 44100 \text{ Hz}$

3. DAMPED OSCILLATOR WITH CONTACT

The above impact model can be readily incorporated to the simulation of a damped oscillator. Consider a mass-spring system with stiffness $k = m(2\pi f_0)^2$, f_0 being the resonance frequency of the oscillator. The potential energy is now given by $V = V_s + V_c$, where $V_s = ky^2/2$ and V_c is the collision potential of the previous section. A damping term is also included in the equation of motion of the system (see Figure 2), which reads

$$m \frac{d^2y}{dt^2} + m\gamma \frac{dy}{dt} + ky + k_c |y|^\alpha \left(1 + r \frac{dy}{dt} \right) = 0, \quad (12)$$

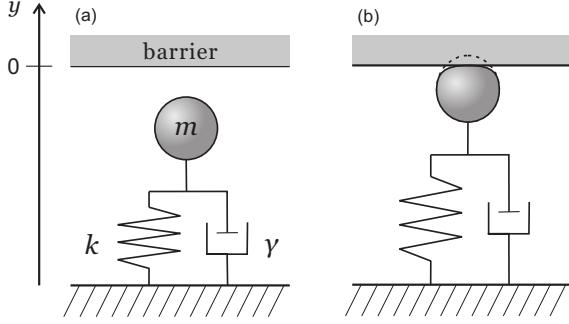


Figure 2: Sketch of a damped harmonic oscillator (a) before and (b) during contact with the barrier.

γ being a damping factor. This system can be written in Hamiltonian form as [10]

$$\frac{dy}{dt} = \frac{\partial T}{\partial p} \quad (13a)$$

$$\frac{dp}{dt} = -\frac{\partial V}{\partial y} - r \frac{dV_c}{dt} - \gamma p. \quad (13b)$$

For the evolution of the system energy $H = T + V$ the following expression can be derived

$$\begin{aligned} \frac{dH}{dt} &= \frac{\partial H}{\partial y} \frac{dy}{dt} + \frac{\partial H}{\partial p} \frac{dp}{dt} = -\frac{\gamma p^2}{m} - \frac{rp}{m} \frac{dV_c}{dt} \\ &= -\frac{p^2}{m^2} (m\gamma + rk_c[y]^\alpha) \leq 0, \end{aligned} \quad (14)$$

which is in accordance with both loss mechanisms, with the energy decreasing due to frictional forces. In search of an invariant quantity, the following conservation law can thus be derived [18]

$$H + \int \frac{\gamma p^2}{m} dt + \int \frac{rp}{m} dV_c = \text{const.} \quad (15)$$

The numerical formulation of (13), using the EC discretisation method, is

$$\frac{y^{n+1} - y^n}{\Delta t} = \frac{T(p^{n+1}) - T(p^n)}{p^{n+1} - p^n} \quad (16a)$$

$$\begin{aligned} \frac{p^{n+1} - p^n}{\Delta t} &= -\frac{V(y^{n+1}) - V(y^n)}{y^{n+1} - y^n} \\ &\quad - r \frac{V_c(y^{n+1}) - V_c(y^n)}{\Delta t} - \gamma \mu_{t+} p^n, \end{aligned} \quad (16b)$$

where $\mu_{t+} p^n = (p^{n+1} + p^n)/2$. This leads to the solution of a nonlinear equation in x (again for $x = y^{n+1} - y^n$)

$$\begin{aligned} F(x) &= (1 + \gamma \frac{\Delta t}{2})x - 2q^n + \frac{\Delta t^2}{4m} k(x + 2y^n) \\ &\quad + \frac{\Delta t^2}{2m} \frac{k_c}{\alpha + 1} \frac{|y^n + x|^{\alpha+1} - |y^n|^{1+\alpha}}{x} \\ &\quad + r \frac{\Delta t}{2m} \frac{k_c}{\alpha + 1} (|y^n + x|^{\alpha+1} - |y^n|^{1+\alpha}) = 0. \end{aligned} \quad (17)$$

Note that proving existence and uniqueness of solutions, as well as avoiding singularities can be shown in a similar fashion as for (9).

3.1. Energy balance

In musical instrument simulation, such lumped oscillators, representing the vibroacoustical behaviour of an instrument, are often driven by external forces due to interacting objects or acoustic pressure. Given such an external force f_{ex} , equation (13b) transforms into

$$\frac{dp}{dt} = -\frac{\partial V}{\partial y} - r \frac{dV_c}{dt} - \gamma p + f_{\text{ex}}, \quad (18)$$

which causes (17) to transform into

$$F(x) - \frac{\Delta t^2}{2m} \mu_{t+} f_{\text{ex}}^n = 0. \quad (19)$$

The energy balance accordingly becomes

$$\frac{dH}{dt} = \frac{pf_{\text{ex}}}{m} - \frac{p^2}{m^2} (m\gamma + rk_c[y]^\alpha). \quad (20)$$

Note that the energy is, in general, not monotonically decreasing any more, due to the power supplied by the external force, hence the system is not dissipative. However, in the absence of excitation (when the external force $f_{\text{ex}} = 0$) the energy is continuously decreasing, since $dH/dt \leq 0$ and the system is dissipative. Discretising (20) yields

$$\begin{aligned} \frac{H^{n+1} - H^n}{\Delta t} &= -\frac{\gamma}{m} (\mu_{t+} p^n)^2 - \frac{r}{m} \mu_{t+} p^n \frac{V_c^{n+1} - V_c^n}{\Delta t} \\ &\quad + \frac{\mu_{t+} p^n}{m} \mu_{t+} f_{\text{ex}}^n, \end{aligned} \quad (21)$$

with $H^n = T(p^n) + V(y^n)$. This induces the following discrete conservation law [18]

$$\begin{aligned} H^{n+1} + \sum_{\kappa=0}^n \frac{\mu_{t+} p^\kappa}{m} \left(\gamma \mu_{t+} p^\kappa + r \frac{V_c^{\kappa+1} - V_c^\kappa}{\Delta t} - \mu_{t+} f_{\text{ex}}^\kappa \right) \Delta t \\ = K^n = \text{const.} \end{aligned} \quad (22)$$

3.2. Bounds on the discrete solution

The magnitude of the numerical approximations for q^n and y^n can be bound in regard to the initial energy of the system, the external force and the model parameters. Since

$$\begin{aligned} \mu_{t+} p^n \frac{V_c^{n+1} - V_c^n}{\Delta t} &= \mu_{t+} p^n \frac{V_c^{n+1} - V_c^n}{y^{n+1} - y^n} \frac{y^{n+1} - y^n}{\Delta t} \\ &= \frac{(\mu_{t+} p^n)^2}{m} V'_c(y) \geq 0, \end{aligned} \quad (23)$$

it follows from (22) that

$$H^{n+1} \leq H^n - \Delta t \frac{\gamma}{m} (\mu_{t+} p^n)^2 + \Delta t \frac{\mu_{t+} p^n}{m} \mu_{t+} f_{\text{ex}}^n. \quad (24)$$

The parabola $-\gamma z^2 + z \mu_{t+} f_{\text{ex}}^n$ attains its maximum at $z = \mu_{t+} f_{\text{ex}}^n/(2\gamma)$, hence we obtain the estimate

$$H^{n+1} \leq H^n + \frac{\Delta t}{m} \frac{(\mu_{t+} f_{\text{ex}}^n)^2}{4\gamma}. \quad (25)$$

Using the facts that $0 \leq \frac{p^2}{2m}, \frac{ky^2}{2} \leq H$, $q = p\frac{\Delta t}{2m}$ and $x = q^{n+1} + q^n$, the following local estimate for the solution of $F(x) = 0$ is obtained

$$|q^{n+1}| \leq \frac{\Delta t}{2m} \sqrt{2mH^n + \frac{\Delta t}{2\gamma} (\mu_{t+} f_{\text{ex}}^n)^2}. \quad (26)$$

Assuming an upper bound on the external force, f_{ex}^{\max} , one can also obtain the global estimate for times smaller than a final time $t_{\text{end}} \geq (n+1)\Delta t$

$$H^{n+1} \leq H^0 + \frac{t_{\text{end}}}{m} \frac{(f_{\text{ex}}^{\max})^2}{4\gamma}, \quad (27)$$

which leads, for $\gamma > 0$ to

$$|q^{n+1}| \leq \frac{\Delta t}{2m} \sqrt{2mH^0 + t_{\text{end}} \frac{(f_{\text{ex}}^{\max})^2}{2\gamma}} \quad (28a)$$

$$|x| \leq \frac{\Delta t}{m} \sqrt{2mH^0 + t_{\text{end}} \frac{(f_{\text{ex}}^{\max})^2}{2\gamma}} := B_x \quad (28b)$$

and for the solution y^{n+1} , we obtain for $k > 0$ the following estimate

$$|y^{n+1}| \leq \sqrt{\frac{2H^0}{k} + t_{\text{end}} \frac{(f_{\text{ex}}^{\max})^2}{2mk\gamma}} := B_y. \quad (29)$$

Note that for $\gamma = 0$ and $f_{\text{ex}} = 0$ (as is the case in Figure 4) these bounds can be shown to be equal to $B_x = \frac{\Delta t}{m} \sqrt{2mH^0}$ and $B_y = \sqrt{\frac{2H^0}{k}}$.

3.3. Bound on the number of iterations

One common issue when using iterative methods to solve nonlinear equations is the number of iterations required for the numerical solution to converge. Indeed, for certain parameter choices similar iterative schemes may fail to converge, as reported in [7]. Therefore a formal calculation is presented here for the maximum number of iterations required for the numerical solution of (19) using Newton's method (or alternatively the bisection method).

In the presence of an external force, the uniqueness of the solution of $F(x) = 0$ needs to be analysed separately. Assuming uniqueness, the bisection method halves the interval whose mean is an approximation to the solution of $F(x) = 0$ in each iteration. From the bound (28b) on x therefore it follows that it takes at most

$$k = \log_2 \frac{B_x}{\varepsilon} \quad (30)$$

iterations for the bisection method to converge up to precision ε , when starting within the interval $[-B_x, B_x]$.

Uniqueness can be guaranteed when the external force does not depend on the state (y and y') of the oscillator, hence f_{ex}^{n+1} is a function independent of x . Under this assumption, one can show that

$$F'(x) \geq 1, \quad (31)$$

$$F''(x) \geq 0, \quad (32)$$

and use these facts for the analysis of Newton's method. This leads (see Appendix) to the fact that in order to achieve a given precision

ε in the approximation of the unique solution x^* of $F(x) = 0$, one needs to perform at most k Newton iterations when starting from $x_0 \geq x^*$ and $k+1$ iterations when starting from $x_0 < x^*$, with

$$k = \frac{\log \varepsilon - \log(2B_x)}{\log \left(1 - \frac{1}{F'(B_x, B_y)} \right)}. \quad (33)$$

When f_{ex} depends on (y, y') but a bound on its magnitude is known, then existence and uniqueness can also be guaranteed for Δx being small enough. Note that both bounds (for the bisection and Newton's method) constitute a worst-case-scenario estimation and, as shown in Figures 3 and 4 below, convergence is expected to occur earlier. It is still advisable however to ensure that such bounds exist.

3.4. Numerical results

Figures 3 and 4 show simulation examples for a damped, driven oscillator and an undamped oscillator, both undergoing repeated collisions including impact damping. Figure 3(b) demonstrates the conservation law (22) by plotting the error

$$e^n = \frac{K^n - K^0}{\mathcal{P}_\in(K^0)} \quad (34)$$

for a system with resonance frequency $f_0 = 3000$ Hz, where $\mathcal{P}_\in(K^0) \leq K^0$ is the nearest power of two to K^0 from the left [19]. The external driving force is a sinusoid with a 440 Hz frequency. The other model parameters are the same as in Section 2 and the initial conditions are $y^0 = -0.1$ mm; $p^0 = 0.005$ kg/m/s. The dashed line in Figure 3(a) shows the mass displacement in the absence of the external force. Figure 3(c) shows how many Newton iterations are required at each time step for convergence to machine precision. These are well below the theoretical bound calculated in Section 3.3, which is equal to 12 iterations for Newton's method and 38 iterations for the bisection method; note however that the bisection method requires less operations at each iteration. Figure 4 presents the case where the external force and the linear damping are omitted ($\gamma = 0$), and the impact damping factor r is increased 500 times to exaggerate its effect. It can be observed that in both cases K^n is conserved within machine precision and Newton's method converges quite fast, which can be explained by the presence of a good starting point for x , available from the solution at the previous time step.

Figure 5 shows how the number of iterations may increase when an arbitrary starting point is chosen. This starting point is enforced for all time-steps during a 10 ms long simulation, using the same parameters as in Figure 3. The maximum number of required iterations across all time-steps is plotted for each chosen starting point value. Since $F'(x) \geq 1$ and $F''(x) \geq 0$ only a poor starting point larger than B_x will result in slower convergence rates², as explained in the Appendix.

²In practice, using the solution at the previous time-step as a starting point guarantees that $x_0 \in [-B_x, B_x]$. However, depending on the shape of $F(x)$, x_1 may indeed lie on the right hand side of B_x . In that case one should set $x_1 = B_x$, since the solution x^* is expected to lie in $[-B_x, B_x]$. When generating Figure 5, starting from an arbitrary x_0 , this substitution was not carried out, in order to demonstrate the possibility of slow convergence rates in the absence of a bound on the discrete solution.

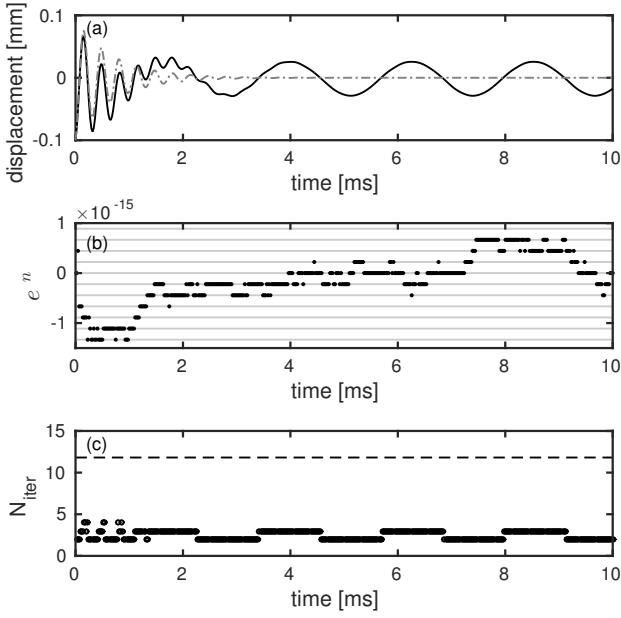


Figure 3: Simulation of a damped, driven oscillator involving multiple impacts modeled using the Hunt-Crossley approach ($\gamma = 3000 \text{ s}^{-1}$, $r = 0.01 \text{ s/m}$). (a): mass displacement (the dashed line shows the displacement in the absence of the external force). (b): error in the conservation of K^n . Horizontal lines indicate multiples of single bit variation. (c): Number of iterations required for Newton’s method to converge (the dashed line indicates the theoretical bound on the number of iterations).

4. APPLICATION TO SOUND SYNTHESIS

An application of the above damping model is demonstrated in this section using a problem from music acoustics. In particular, the motion of a clarinet reed is simulated and the resulting sound is synthesised. The clarinet reed is driven by the pressure difference across it $p_\Delta = p_m - p_{in}$, where p_m is the blowing pressure and p_{in} is the pressure inside the clarinet mouthpiece. Hence the force applied to the reed due to the pressure difference is $f_\Delta = S_r p_\Delta$, where S_r is the effective reed area [20]. Thus the motion of the reed is governed by [3]

$$m \frac{d^2y}{dt^2} + m\gamma \frac{dy}{dt} + ky + k_c |(y - y_c)|^\alpha \left(1 + r \frac{dy}{dt}\right) = f_\Delta. \quad (35)$$

Two distinct nonlinearities take place here. The first one is due to the collision of the reed with the mouthpiece lay and is modelled using the collision potential defined in Section 2. This nonlinear reed-lay interaction becomes effective after the reed displacement y exceeds a certain value y_c [21, 22] and hence an offset is required inside the ‘beating bracket’ defined under equation (1). The second nonlinearity stems from the relationship between mouthpiece pressure p_{in} and mouthpiece flow u_{in} . The flow is built up from two components [3, 22], the Bernoulli flow u_f and the flow u_r induced

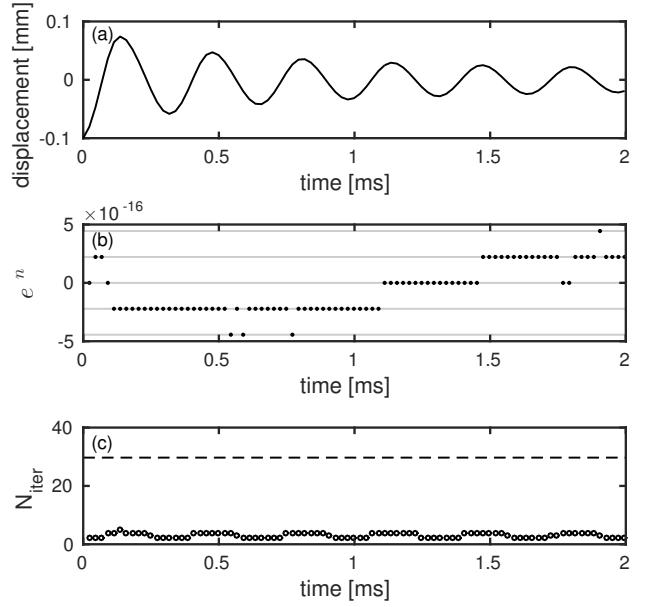


Figure 4: Simulation of an undamped oscillator involving multiple impacts modeled using the Hunt-Crossley approach ($\gamma = 0$, $r = 5 \text{ s/m}$). (a): mass displacement. (b): error in the conservation of K^n . Horizontal lines indicate multiples of single bit variation. (c): Number of iterations required for Newton’s method to converge (the dashed line indicates the theoretical bound on the number of iterations).

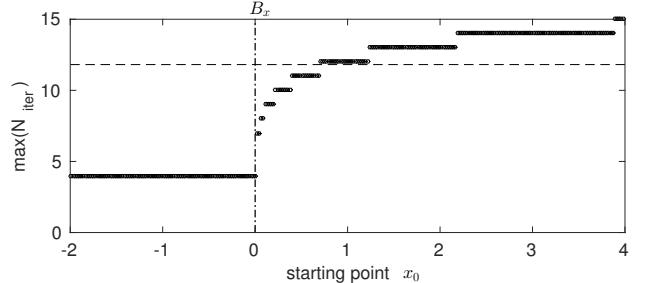


Figure 5: The maximum number of required iterations for different choices of the starting point x_0 for the Newton solver. The horizontal dashed line indicates the theoretical bound on the number of iterations and the vertical one shows the limit that should be enforced on x_1 in order for the iteration bound to be valid (see Appendix; in this case $B_x = 4.42 \cdot 10^{-5}$). Evidently such a limit was not enforced in this numerical experiment.

by the motion of the reed, with

$$u_{in} = u_f + u_r \quad (36a)$$

$$u_f = \sigma w h \sqrt{\frac{2|p_\Delta|}{\rho}} \quad (36b)$$

$$u_r = S_r \frac{dy}{dt}, \quad (36c)$$

where $\sigma = \text{sign}(p_\Delta)$, ρ is the air density, w the width of the reed

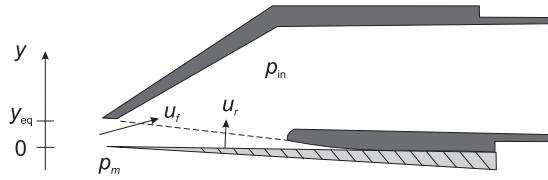


Figure 6: Sketch of the single-reed-mouthpiece system. y_{eq} is the equilibrium position of the reed, u_f and u_r the Bernoulli and reed flow respectively and p_m and p_{in} the mouth pressure and mouthpiece pressure.

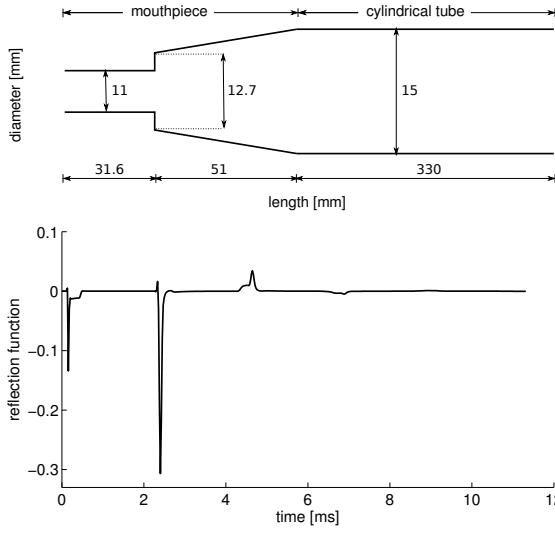


Figure 7: Top: schematic profile of a simplified clarinet bore (not to scale). Bottom: the simulated reflection function.

and $h = y_{\text{eq}} - y$ the reed opening, y_{eq} being the equilibrium opening of the reed after the player positions his lip (see [23]). These parameters, related to the single-reed excitation mechanism, are visualized in Figure 6. Note that in principle h is allowed to become negative, something avoided in the simulations presented here, due to the effect of the collision force. Nevertheless, it is safer to define $h = \lfloor y_{\text{eq}} - y \rfloor$, in order to allow an arbitrary variation of model parameters that might affect the reed opening.

The mouthpiece pressure can be obtained using convolution with the reflection function of the tube [22]. The geometry of the tube (including the mouthpiece) used in the numerical simulations is shown in Figure 7. Its input impedance was calculated using the Acoustics Research Tool [24], including viscothermal losses at the walls and radiation losses at the open end. This can be converted to the reflection function of the tube (also plotted in Figure 7) following the procedure described in [25]. An energy balance for such a coupled system has been explored in [3] where the air column is also discretised using the finite difference method.

The effect of including the impact damping in the single-reed model is visualized in Figure 8 where the spectrogram of the mouthpiece pressure p_{in} is compared to that of the same simulation but with the impact damping omitted. The model parameters used in the simulations are given in Table 2. It can be observed that taking impact damping into account results in the dissipation of higher

Table 2: Physical model parameters used in the clarinet simulation.

reed surface	$S_r = 9.856 \cdot 10^{-5} \text{ m}^2$
stiffness/area	$k/S_r = 1.792 \cdot 10^7 \text{ Pa/m}$
equilibrium	$y_{\text{eq}} = 4.09 \cdot 10^{-4} \text{ m}$
blowing pressure	$p_m = 3637 \text{ Pa}$
reed width	$\lambda = 0.012 \text{ m}$
reed mass/area	$m/S_r = 0.0332 \text{ kg/m}^2$
damping	$\gamma = 3000 \text{ 1/s}$
impact stiffness/area	$k_c/S_r = 2 \cdot 10^{10} \text{ Pa/m}^\alpha$
impact damping	$r = 1 \text{ s/m}$
impact exponent	$\alpha = 2$

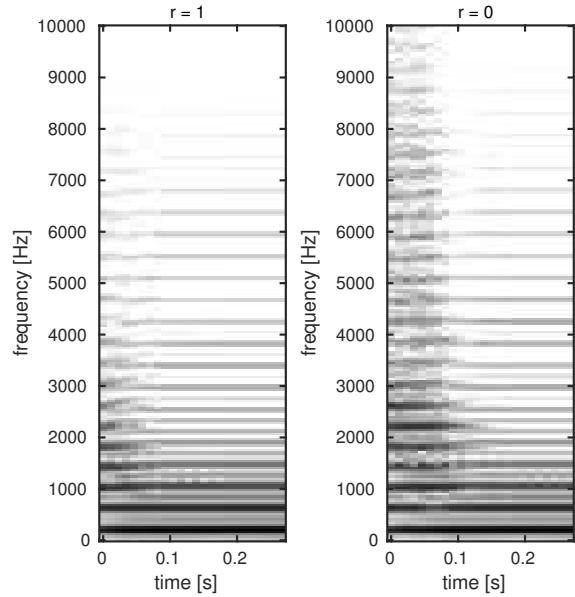


Figure 8: Spectrogram of the simulated mouthpiece pressure p_{in} with (left) and without impact damping (right).

harmonics, especially during the transient. The parameters related to this loss mechanism are here chosen arbitrarily; they are expected to vary depending on the type of reed used (plastic reeds have seen increased use lately) and the material properties and exact geometry of the mouthpiece lay. The latter is often specified by musicians as having a significant effect on the response of the instrument.

5. DISCUSSION

A power balance model for impact damping has been presented. This leads to the numerical solution of a nonlinear equation, with Newton's method being a suitable solver. The fact that such equations need to be solved iteratively led to an efficiency analysis in terms of the maximum number of iterations required for convergence. Note that such a limit represents a worst-case scenario. Convergence is usually achieved earlier, due to the presence of a good starting point for the solver, which is given by the solution at the previous time step. Nevertheless, the presented analysis pro-

vides a guarantee that simulation algorithms will always converge within a given number of iterations steps.

Including impact damping when modelling object collisions, a quantity that is often neglected in sound synthesis applications, appears to have a significant effect on certain systems. This is illustrated by a simulation of a clarinet tone, in which case the influence of impact damping on the simulated tone is apparent on the calculated spectrum. The necessity of such a model for simulating other types of instruments (or different acoustic systems) remains to be investigated using both a numerical and a perceptual approach.

Including two types of damping in this study (parameterised using γ and r) provides a framework for the treatment of a wide range of lumped systems involving nonlinear interactions. An interesting extension of the presented convergence study would be to analyse distributed systems, such as a string interacting with a barrier (see, e.g. [7, 8, 10, 13]). For such systems the nonlinear equation to be solved is a vector equation of the form

$$\mathbf{F}(\mathbf{x}) = \mathbf{0}. \quad (37)$$

In this case a direct analysis of Newton's method is more involved. However (37) could be interpreted as a fixed-point iteration problem $\mathbf{x} = \mathbf{T}(\mathbf{x})$, where the Lipschitz constant of \mathbf{T} [26] relates to the number of required iterations until convergence to the solution \mathbf{x}^* is achieved.

6. ACKNOWLEDGMENTS

This research is supported by the Austrian Science Fund (FWF): P28655-N32. S. Bilbao was supported by the European Research Council, under grant 2011-StG-279068-NESS.

7. REFERENCES

- [1] A. Chaigne and J. Kergomard, *Acoustics of Musical Instruments*, Springer, New York, 2016.
- [2] V. Chatzioannou and M. van Walstijn, “An energy conserving finite difference scheme for simulation of collisions,” in *Sound and Music Computing (SMAC-SMC 2013)*, Stockholm, 2013, pp. 584–591.
- [3] S. Bilbao, A. Torin, and V. Chatzioannou, “Numerical modeling of collisions in musical instruments,” *Acta Acustica united with Acustica*, vol. 101, no. 1, pp. 155–173, 2015.
- [4] M. Ducceschi, S. Bilbao, and C. Desvages, “Modelling collisions of nonlinear strings against rigid barriers: Conservative finite difference schemes with application to sound synthesis,” in *International Congress on Acoustics*, Buenos Aires, 2016.
- [5] S. Bilbao, *Numerical Sound Synthesis*, Wiley & Sons, Chichester, UK, 2009.
- [6] G. Evangelista and F. Eckerholm, “Player-instrument interaction models for digital waveguide synthesis of guitar: Touch and collisions,” *IEEE Transactions on Audio, Speech and Language Processing*, vol. 18, no. 4, pp. 822–832, 2010.
- [7] M. van Walstijn and J. Bridges, “Simulation of distributed contact in string instruments: a modal expansion approach,” in *Europ. Sig. Proc Conf (EUSIPCO2016)*, 2016, pp. 1023–1027.
- [8] C. Issanchou, S. Bilbao, J. Le Carrou, C. Touzé, and O. Doaré, “A modal-based approach to the nonlinear vibration of strings against a unilateral obstacle: Simulations and experiments in the pointwise case,” *Journal of Sound and Vibration*, vol. 393, pp. 229–251, 2017.
- [9] K. Hunt and F. Crossley, “Coefficient of restitution interpreted as damping in vibroimpact,” *Journal of Applied Mechanics*, vol. 42, pp. 440–445, 1975.
- [10] M. van Walstijn and V. Chatzioannou, “Numerical simulation of tanpura string vibrations,” in *Proc. International Symposium on Musical Acoustics, Le Mans*, 2014, pp. 609–614.
- [11] C. Desvages and S. Bilbao, “Two-polarisation physical model of bowed strings with nonlinear contact and friction forces, and application to gesture-based sound synthesis,” *Applied Sciences*, vol. 6, no. 5, 2016.
- [12] S. Papetti, F. Avanzini, and D. Rocchesso, “Numerical methods for a nonlinear impact model: A Comparative study with closed-form corrections,” *IEEE Transactions on Audio, Speech and Language Processing*, vol. 19, no. 7, pp. 2146–2158, 2011.
- [13] V. Chatzioannou and M. van Walstijn, “Energy conserving schemes for the simulation of musical instrument contact dynamics,” *Journal of Sound and Vibration*, vol. 339, pp. 262–279, 2015.
- [14] D.E. Hall, “Piano string excitation. VI: Nonlinear modeling,” *Journal of the Acoustical Society of America*, vol. 92, no. 1, pp. 95–105, 1992.
- [15] A. Chaigne and V. Doutaut, “Numerical simulation of xylophones. I. Time-domain modeling of the vibrating bars,” *Journal of the Acoustical Society of America*, vol. 101, no. 1, pp. 539–557, 1997.
- [16] T. Taguti, “Dynamics of simple string subject to unilateral constraint: A model analysis of sawari mechanism,” *Acoustical science and technology*, vol. 29, no. 3, pp. 203–214, 2008.
- [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] V. Chatzioannou and M. Van Walstijn, “Discrete-time conserved quantities for damped oscillators,” in *Proc. Third Vienna Talk on Music Acoustics*, 2015, pp. 135–139.
- [19] A. Torin, *Percussion Instrument Modelling In 3D: Sound Synthesis Through Time Domain Numerical Simulation*, Ph.D. thesis, The University of Edinburgh, 2016.
- [20] M. van Walstijn and F. Avanzini, “Modelling the mechanical response of the reed-mouthpiece-lip system of a clarinet. Part II. A lumped model approximation,” *Acustica*, vol. 93, no. 1, pp. 435–446, 2007.
- [21] J.P. Dalmont, J. Gilbert, and S. Ollivier, “Nonlinear characteristics of single-reed instruments: Quasistatic volume flow and reed opening measurements,” *Journal of the Acoustical Society of America*, vol. 114, no. 4, pp. 2253–2262, 2003.
- [22] V. Chatzioannou and M. van Walstijn, “Estimation of clarinet reed parameters by inverse modelling,” *Acta Acustica united with Acustica*, vol. 98, no. 4, pp. 629–639, 2012.

- [23] E. Ducasse, “A physical model of a single-reed wind instrument, including actions of the player,” *Computer Music Journal*, vol. 27, no. 1, pp. 59–70, 2003.
- [24] A. Braden, D. Chadefaux, V. Chatzioannou, S. Siddiq, C. Geyer, S. Balasubramanian, and W. Kausel, “Acoustic Research Tool (ART),” <http://sourceforge.net/projects/artool>, 2006–2017.
- [25] B. Gazengel, J. Gilbert, and N. Amir, “Time domain simulation of single reed wind instrument. From the measured input impedance to the synthesis signal. Where are the traps?”, *Acta Acustica*, vol. 3, pp. 445–472, 1995.
- [26] J. Ortega and W. Rheinboldt, *Iterative Solution of Nonlinear Equations in Several Variables*, vol. 30, SIAM, New York, 1970.

APPENDIX: BOUND ON NEWTON INTERATIONS

Starting from a user chosen x_0 , one iteratively gets approximations to the solution of $F(x) = 0$ by

$$x_{k+1} = x_k - \frac{F(x_k)}{F'(x_k)}, \quad k = 0, 1, \dots \quad (38)$$

Let x^* denote the unique solution of $F(x) = 0$. Due to (31), we know that

$$F(x) > 0, \text{ for } x > x^*, \quad (39)$$

$$F(x) < 0, \text{ for } x < x^*. \quad (40)$$

First we assume that $x_0 < x^*$, hence $F(x_0) < 0$. It follows from (32), that

$$-F(x_0) = \int_{x_0}^{x^*} F'(x) dx \geq (x^* - x_0) F'(x_0), \quad (41)$$

hence

$$x_1 = x_0 - F(x_0)/F'(x_0) \geq x^*, \quad (42)$$

so after the first Newton step, we end up to the right of the solution x^* . On the other hand, starting from $x_0 > x^*$, we obtain

$$F(x_0) = \int_{x^*}^{x_0} F'(x) dx \leq (x_0 - x^*) F'(x_0), \quad (43)$$

hence

$$x^* \leq x_0 - F(x_0)/F'(x_0) = x_1. \quad (44)$$

In summary, starting Newton’s method with an $x_0 > x^*$ yields a sequence $x_k, k = 0, 1, \dots$ with $x_k > x^*$, whereas when starting with an $x_0 < x^*$ then $x_k > x^*$ for $k = 1, 2, \dots$

Furthermore, for $x_0 > x^*$, we observe that

$$F(x_0) = \int_{x^*}^{x_0} F'(x) dx \geq (x_0 - x^*) F'(x^*). \quad (45)$$

Therefore

$$0 \leq x_1 - x^* = (x_0 - x^*) - \frac{F(x_0)}{F'(x_0)} \quad (46)$$

$$\leq (x_0 - x^*) - \frac{(x_0 - x^*) F'(x^*)}{F'(x_0)} \quad (47)$$

$$= (x_0 - x^*) \left(1 - \frac{F'(x^*)}{F'(x_0)} \right) \quad (48)$$

$$\leq (x_0 - x^*) \left(1 - \frac{1}{F'(x_0)} \right) \quad (49)$$

We can now estimate the error in the k th step, $x_k - x^*$ by the initial error, using the fact that F' is an increasing function, and the sequence $x_k, k = 0, 1, \dots$ is decreasing.

$$0 \leq x_k - x^* \leq (x_{k-1} - x^*) \left(1 - \frac{1}{F'(x_{k-1})} \right) \quad (50)$$

$$\leq (x_{k-1} - x^*) \left(1 - \frac{1}{F'(x_0)} \right) \quad (51)$$

$$\leq (x_0 - x^*) \left(1 - \frac{1}{F'(x_0)} \right)^k. \quad (52)$$

To finalise a priori estimates on the error $x_k - x^*$, we observe from (28b) that x^* is in the interval $[-B_x, B_x]$, hence $x_0 - x^* \leq 2B_x$, provided the starting point x_0 is taken from the same interval. Finally, we estimate

$$1 \leq F'(x_0) \leq F'(B_x). \quad (53)$$

The function F' depends on y^n as well,

$$\begin{aligned} F'(x, y^n) &= \left(1 + \frac{\gamma \Delta_t}{2} \right) + \frac{\Delta_t^2}{2m} \left(\frac{V'(y^n + x)}{x} \right) \\ &\quad - \frac{\Delta_t^2}{2m} \left(\frac{V(y^n + x) - V(y^n)}{x^2} \right) \\ &\quad + \frac{r \Delta_t}{2m} (V'_c(y^n + x)). \end{aligned} \quad (54)$$

In order to get a bound independent of y^n , we observe that

$$\begin{aligned} \frac{\partial}{\partial y^n} F'(x, y^n) &= \frac{\Delta_t^2}{2m} \frac{V''(y^n + x)}{x} \\ &\quad - \frac{\Delta_t^2}{2m} \frac{V'(y^n + x) - V'(y^n)}{x^2} \\ &\quad + \frac{r \Delta_t}{2m} V''_c(y^n + x) \geq 0 \end{aligned} \quad (55)$$

Using the fact that V' is convex, it can be shown that

$$F'(B_x, y^n) \leq F'(B_x, B_y), \quad (56)$$

where B_y is given by (29), hence for $x_0 \geq x^*$,

$$0 \leq x_k - x^* \leq 2B_x \left(1 - \frac{1}{F'(B_x, B_y)} \right)^k. \quad (57)$$

To achieve a given precision ε , one needs to perform at most

$$k = \frac{\log \varepsilon - \log(2B_x)}{\log \left(1 - \frac{1}{F'(B_x, B_y)} \right)} \quad (58)$$

Newton steps. For $x_0 \leq x^*$, one needs to perform at most $k + 1$ steps, since $x_1 \geq x^*$, and if x_1 exceeds B_x , one should set $x_1 = B_x$.

A NUMERICAL SCHEME FOR VARIOUS NONLINEAR FORCES, INCLUDING COLLISIONS, WHICH DOES NOT REQUIRE AN ITERATIVE ROOT FINDER

Michele Ducceschi

Acoustics and Audio Group,

University of Edinburgh

Edinburgh, UK

michele.ducceschi@ed.ac.uk

ABSTRACT

Nonlinear forces are ubiquitous in physical systems, and of prominent importance in musical acoustics. Though many models exist to describe such forces, in most cases the associated numerical schemes rely on iterative root finding methods, such as Newton-Raphson or gradient descent, which are computationally expensive and which therefore could represent a computational bottleneck. In this paper, a model for a large class of nonlinear forces is presented, and a novel family of energy-conserving finite difference schemes given. The schemes only require the evaluation of the roots of a quadratic function. A few applications in the lumped case are shown, and the robustness and accuracy of the scheme tested.

1. INTRODUCTION

In many musical instruments, collisions and contact forces are involved at various levels in the mechanism of sound production [1]. The interaction of strings with a bow, a membrane (like in the snare drum), a mallet, a finger or a fretboard are all examples of contact forces. Collisions of the reed in wind instruments have a major impact on the perceived tonal quality. Prepared piano string (coupled to rattling elements) are yet another example of such collisions. Outside of musical acoustics, collisions represent an important field of study in robotics [2], and of course computer graphics [3, 4]. The models employed to describe all the above forces are necessarily nonlinear, and hence they represent a challenge in terms of numerical simulation. Although many methods have been used to simulate some specific examples of collisions (including digital waveguides [5], modal methods [6, 7], and time stepping methods [8]), a fairly recent general framework was proposed in order to simulate a large class of collisions and contact forces [1]. In this framework, the forces are generated by a potential which takes the form of some kind of power law, depending on one stiffness coefficient, and one exponent. Such framework allows to simulate collisions of two lumped objects (like a mass and a spring), of one lumped and one distributed object (like a mallet and a string), and even of two distributed systems (like a string and a membrane). For very stiff collisions, the forces are generated by a spurious interpenetration, which can be made as small as possible by increasing the stiffness coefficient. Though extremely versatile, the associated energy-conserving numerical schemes rely on iterative root finding algorithms, such as Newton-Raphson, which in most cases represent a computational bottleneck. In this paper, a novel family of finite difference schemes is proposed for the lumped case, which do not require an iterative root finding method. It should be mentioned that previous works (see for example [9, 10]) do present numerical schemes able to

simulate collisions without iterative root finding algorithms. However, this was showed only in the case of linear response of the barrier with the spurious interpenetration. In this paper, the proposed schemes work for a class of nonlinear potentials. At each update the nonlinear force can be calculated by simply finding the roots of a quadratic function, which basically involves the evaluation of a single square root. In section 2, the model is presented in the form of an ordinary differential equation, and the forms of the potentials given. Finite difference schemes are presented in section 3, derived from a given Hamiltonian depending on one scalar parameter. Boundedness of the energy will be shown, along with a discussion of the realness of the roots of the quadratic. Tests for accuracy are presented at this point. Finally, section 4 shows a few applications of interest.

2. MODEL EQUATIONS

In the course of this paper, the displacement $u(t)$ of a particle of mass M is described by an ordinary differential equation of the following form

$$M\ddot{u} = -\phi'(u) = -\frac{\dot{\phi}(u)}{\dot{u}}, \quad (1)$$

where the last equality was obtained by means of the chain rule. In the equation, the particle is assumed to be subjected to a force described by the potential $\phi(u)$. By multiplying both sides of the equation by \dot{u} the following energy identity is obtained

$$\frac{d}{dt} \left(\frac{M}{2} \dot{u}^2 + \phi(u) \right) \triangleq \frac{d}{dt} \mathfrak{H} = 0 \quad \rightarrow \quad \mathfrak{H} = \mathfrak{H}_0, \quad (2)$$

and hence the energy \mathfrak{H} is non-negative if and only if $\phi(u) \geq 0 \forall u$. Various potentials satisfy such requirement. Specific forms of interest here are the following

$$\phi(u) = \frac{K}{\alpha+1} u^{\alpha+1}, \quad \alpha = 1, 3, 5, \dots \quad (3a)$$

$$\phi(u) = \frac{K}{\alpha+1} [u - h]_+^{\alpha+1}, \quad \alpha \in \mathbb{R}, \alpha > 1 \quad (3b)$$

$$\phi(u) = \frac{K}{\alpha+1} [|u| - h]_+^{\alpha+1}, \quad \alpha \in \mathbb{R}, \alpha > 1 \quad (3c)$$

In the equations, the symbol $[x]_+$ denotes the positive part, i.e. $2[x]_+ \triangleq x + |x|$. The constant K is a stiffness coefficient. The forces originated by such potentials are depicted in Fig. 1. In musical acoustics, though often for distributed systems, these potentials are used in modelling large-amplitude nonlinearities, contact nonlinearities (such as the hammer-string interaction), rattling elements, and other important nonlinear interactions [11].

From (2), one finds immediately a bound on the growth of the solution, i.e.

$$(\dot{u})^2 \leq \frac{2\mathfrak{H}_0}{M}, \quad (4)$$

where \mathfrak{H}_0 (a constant) is the total energy.

3. FINITE DIFFERENCE SCHEMES

Solutions to (1) are now sought in terms of appropriate finite difference schemes, upon the introduction of a sample rate f_s and the associated time step $k = 1/f_s$. The solution is evaluated at the discrete times nk , where $n \in \mathbb{Z}^+$, and is denoted u^n . Finite difference operators are introduced as:

- backward and forward shift operators
 $e_{t-u^n} \triangleq u^{n-1}$, $e_{t+u^n} \triangleq u^{n+1}$
- backward, centered and forward first time derivatives
 $\delta_{t-} \triangleq \frac{1}{k}(1-e_{t-})$, $\delta_t \triangleq \frac{1}{2k}(e_{t+}-e_{t-})$, $\delta_{t+} \triangleq \frac{1}{k}(e_{t+}-1)$
- averaging operators
 $\mu_{t+} \triangleq \frac{1}{2}(1+e_+)$, $\mu_{t-} \triangleq \frac{1}{2}(1+e_-)$, $\mu_{t-}^{(s)} \triangleq s + (1-s)e_{t-}$ ($s \in \mathbb{R}$)
- second time derivative
 $\delta_t \triangleq \delta_{t+}\delta_{t-} = \frac{1}{k^2}(e_{t+}-2+e_{t-})$

In order to derive a finite difference scheme, a general form for the Hamiltonian is given here in terms of the generalised averaging operator defined above, as

$$\mathfrak{h}^{n-1/2} = \frac{M}{2}(\delta_{t-}u^n)^2 + \mu_{t-}^{(s)}\phi(u^n). \quad (5)$$

Notice that the particular choice $s = \frac{1}{2}$ leads to the Hamiltonian considered in [1], whose associated finite difference scheme is second-order accurate, and whose update requires an iterative root finding method such as the Newton-Raphson algorithm.

In this work, the potential energy is Taylor-expanded around the point u^{n-1} up to second order, giving

$$\begin{aligned} \mu_{t-}^{(s)}\phi(u^n) &\approx \phi(u^{n-1}) + s(u^n - u^{n-1})\phi'(u^{n-1}) + \\ &s\frac{(u^n - u^{n-1})^2}{2}\phi''(u^{n-1}) \triangleq P_{n-1,n}^{(s)} \end{aligned} \quad (6)$$

Hence, the Hamiltonian considered in this work, depending on the parameter s , is

$$\mathfrak{h}^{n-1/2} = \frac{M}{2}(\delta_{t-}u^n)^2 + P_{n-1,n}^{(s)} \quad (7)$$

3.1. Boundedness of Potential Energy

The potential energy defined in (6) is a parabola in $u^n - u^{n-1}$. Moreover, for the potentials considered in (3) the following identities hold

$$\phi'(u) = (\alpha + 1)\frac{\phi(u)}{\bar{u}}, \quad \phi''(u) = \alpha(\alpha + 1)\frac{\phi(u)}{\bar{u}^2}, \quad (8)$$

where

$$\bar{u} \triangleq u \quad \text{for (3a)}$$

$$\bar{u} \triangleq u - h \quad \text{for (3b)}$$

$$\bar{u} \triangleq \frac{|u| - h}{\text{sign}(u)} \quad \text{for (3c)}$$

Hence, one has

$$P_{n-1,n}^{(s)} = \phi(u^{n-1}) \left[1 + s(\alpha + 1)x + s(\alpha + 1)\alpha \frac{x^2}{2} \right] \quad (9)$$

where

$$x \triangleq \frac{u^n - u^{n-1}}{\bar{u}^{n-1}}. \quad (10)$$

The potential energy will be non-negative if and only if the discriminant of the quadratic above is less than or equal to zero, i.e.

$$s^2(\alpha + 1)^2 - 2s(\alpha + 1)\alpha \leq 0. \quad (11)$$

This is a parametric inequality that must be evaluated according the sign of s and $(\alpha + 1)$. Notice that, for the potentials in (3), one must check that the solutions are valid $\forall \alpha \geq 1$. This gives

$$0 < s \leq 1. \quad (12)$$

Such values will ensure that $P_{n-1,n}^{(s)}$ is non-negative, and therefore that the discrete Hamiltonian (7) is non-negative, $\forall \alpha \geq 1$.

In this case, boundedness of the potential energy can be achieved for values of s which do not guarantee non-negativity of the potential energy. In fact, given the particular form of the potential energy (9), if the coefficient multiplying x^2 is positive, then the parabola will always have a minimum regardless of the value of u^{n-1} , and hence $\forall n$ (remember that ϕ is non-negative by definition). Such coefficient is $s(\alpha + 1)\alpha$ and, because in this work $\alpha \geq 1$, the potential energy will then be bounded from below $\forall s \geq 0$. The bound depends on the intial conditions, and tends to zero as the sampling rate is increased, see also Fig. 5.

Summarising, in this work

$$s > 0, \quad \alpha \geq 1 \quad (13)$$

with the particular case $0 < s \leq 1$ guaranteeing non-negativity of the potential energy.

3.2. Energy conservation. Finite difference scheme

A finite difference scheme can be derived from the Hamiltonian above by imposing

$$\delta_{t+}\mathfrak{h}^{n-1/2} = 0. \quad (14)$$

Before deriving the scheme, notice that when the potential energy is non-negative, one immediately finds a bound similar to (4), i.e.

$$(\delta_{t-}u^n)^2 \leq \frac{2\mathfrak{H}_0}{M}, \quad (15)$$

When the potential energy is not positive, but bounded from below, such inequality is true up to a correction of the order of k^2 . Upon the introduction of the variable $y \triangleq u^{n+1} - u^{n-1}$, the scheme is

$$Ay + B + \frac{C}{y} = 0, \quad (16)$$

where the coefficients A, B, C depend on previous values, and are given as

$$A = \frac{M}{k^2} + s\phi''(u^n)$$

$$B = -\frac{2M}{k^2}(u^n - u^{n-1}) + 2s\phi'(u^n) + 2s(u^{n-1} - u^n)\phi''(u^n)$$

$$C = 2P_{n,n-1}^{(s)} - 2P_{n-1,n}^{(s)}$$

Under the assumption $y \neq 0$, the scheme can be written as a quadratic in y , i.e.

$$Ay^2 + By + C = 0. \quad (17)$$

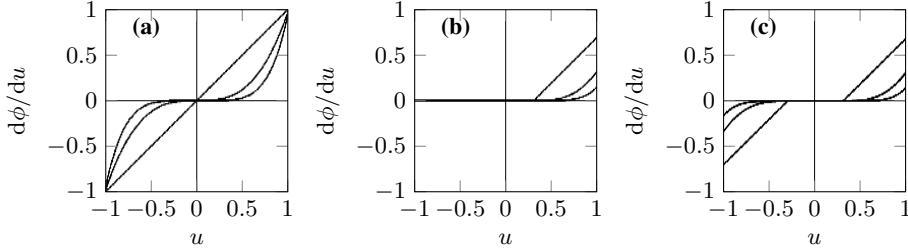


Figure 1: Nonlinear forces. (a): nonlinear power law, as per (3a), for $K = 1, \alpha = 1, 3, 5$. (b): one-sided power law, as per (3b), for $K = 1, \alpha = 1, 1.7, 2.8, h = 0.2$. (c): center limited power law, as per (3c), for $K = 1, \alpha = 1, 1.7, 2.8, h = 0.3$.

3.3. Existence and Uniqueness

Scheme (17) requires the knowledge of the roots of a quadratic function at each update, and is thus very attractive numerically as one may employ the well-known closed-form solution for quadratics instead of a nonlinear root finding algorithm. However, existence of the roots must be checked, and a condition on existence must be given. Also, if the roots exist, they come in pairs, posing a question on uniqueness. Existence is first checked, and the question of uniqueness is discussed later.

3.3.1. Existence

The condition to impose is

$$\Delta^{(\alpha)} \triangleq B^2 - 4AC \geq 0. \quad (18)$$

The discriminant $\Delta^{(\alpha)}$ depends on the particular choice of the exponent α . Existence of the solutions will be checked for $\Delta^{(1)}$, and a discussion for larger values of α will be given later. Also, because potential (3b) and (3c) depend on the positive part of their argument, various cases must be discussed. These are

1. $\phi(u^n) = \phi(u^{n-1}) = 0$. This scenario corresponds to the particle not being in contact with the barrier/spring. In this case $C = 0$ and therefore $\Delta^{(1)} \geq 0$.
2. $\phi(u^{n-1}) > 0, \phi(u^n) = 0$. This scenario corresponds to the particle moving away from the barrier/spring, and this gives $C = -2P_{n-1,n}^{(s)}$ and because $A > 0$ one has $-4AC > 0$ and therefore $\Delta^{(1)} \geq 0$ (remember that $P_{n-1,n}^{(s)}$ under condition (12) is positive-definite).
3. $\phi(u^n) > 0, \phi(u^{n-1}) = 0$. This scenario corresponds to the particle colliding against the barrier/spring, and must be checked.
4. $\phi(u^n) > 0, \phi(u^{n-1}) > 0$. This scenario corresponds to (3a), as well as to (3b), (3c) when the particle and the barrier/spring are in full contact (spurious interpenetration). This case must also be discussed.

Hence, the only cases to discuss are 3 and 4.

Case 3. Upon the definition of $\bar{K} = K/M$ and $u_h = u - h$, this case gives

$$\Delta^{(1)} = \frac{A}{k^2}(u_h^{n-1})^2 + (u_h^n)^2 \left[\frac{1}{k^4} - A\bar{K} + sA\bar{K} \right] - \frac{2}{k^2}Au_h^n u_h^{n-1}.$$

Remembering that for this case $u_h^n > 0, u_h^{n-1} < 0$ one can find a sufficient condition for positiveness by imposing

$$-A\bar{K} + sA\bar{K} \geq 0, \rightarrow s \geq 1.$$

Case 4. Using again the same definition of \bar{K} and u_h , one has (up to a positive constant of proportionality)

$$\begin{aligned} \Delta^{(1)} = & \bar{K}s^2(u_h^{n-1})^2 + \frac{1}{k^2}(\delta_t - u_h^n)^2 \\ & - 2 \left[\frac{s}{k}\bar{K}u_h^{n-1} + Ak \left(\frac{\bar{K}}{2} - s\bar{K} \right) (\mu_t - u_h^n) \right] (\delta_t - u_h^n) \end{aligned}$$

Hence $\Delta^{(1)}$ is a parabola in $(\delta_t - u_h^n)$. Also, for $s \geq \frac{1}{2}$, $\Delta^{(1)}$ could take on negative values only if $u_h^{n-1} > u_h^n$. Under such assumptions, the discriminant of the parabola is calculated, and again the requirement is that such discriminant be negative. Using the fact that $u_h^{n-1} > u_h^n$ one finds a sufficient condition on the time step to be

$$\begin{aligned} k^2 \leq & \frac{s + \frac{1}{2}}{\bar{K}s(s - \frac{1}{2})} \quad \text{for } s > \frac{1}{2} \\ \text{unconditionally positive} & \quad \text{for } s = \frac{1}{2}. \end{aligned}$$

Summarising

- for potential (3a), $\Delta^{(1)}$ will be unconditionally positive for $s = \frac{1}{2}$, otherwise a sufficient condition can be given as: $s > \frac{1}{2}, k^2 \leq \frac{s + \frac{1}{2}}{\bar{K}s(s - \frac{1}{2})}$
- for potentials (3b), (3c), a sufficient condition can be given as: $s \geq 1, k^2 \leq \frac{s + \frac{1}{2}}{\bar{K}s(s - \frac{1}{2})}$

3.3.2. Uniqueness

Assuming realness of the roots of (17), at each time step one has to choose either y_+ or y_- , defined as

$$y_{\pm} = \frac{-B \pm \sqrt{\Delta^{(a)}}}{2A}. \quad (19)$$

The choice is made according to the following rule

- if $B \geq 0$ choose y_-
- if $B < 0$ choose y_+

To understand why this rule holds, consider the case of a free particle, i.e. for which the potential is zero at all times. In this case, scheme (17) reduces to

$$Ay^2 + By = 0, \quad (20)$$

with solutions

$$y_{\pm} = \frac{-B \pm \sqrt{B^2}}{2A} = \frac{-B \pm |B|}{2A}, \quad (21)$$

but because the solution $y = 0$ is ruled out, one recovers the rule above.

3.3.3. Comments on existence, $\Delta^{(\alpha>1)}$ and bounds

In this subsection, some comments are given regarding the realness of the roots of (17), for $\alpha > 1$. A discussion on the sign of $\Delta^{(\alpha>1)}$ is somewhat complicated by the fact that u_h^n, u_h^{n-1} appear nonlinearly with rational exponents, or with high-order powers, and hence it is difficult to carry on a study of the sign of $\Delta^{(\alpha>1)}$ along the same lines as $\Delta^{(1)}$. A possible procedure is to consider a parametric study of the function $d\Delta^{(\alpha)}/d\alpha$, and hence find conditions on maxima and minima of $\Delta^{(\alpha)}$ for the various signs of u_h^n, u_h^{n-1} . This rigorous approach, though desirable, is somewhat lengthy and perhaps beyond the scope of the current work. A less rigorous, though revealing approach is to make use of brute force, i.e. to launch many simulations testing out large portions of the parameter space, and to empirically verify the robustness of the algorithm.

In Fig. 2, the scheme is checked for potential (3a), for $s = \frac{1}{2}, 1, 3$. The particle has mass $M = 1$ kg, and the spring has stiffness $K = 10^3$. Each case presents two subcases, i.e. standard and very high initial velocities (1 m/s, 20 m/s). The figures report the minimum of $\Delta^{(\alpha)}$, for $\alpha \in [1, 3, 5, 7, 9, 11, 13]$. Each colour is associated with a different time step. Missing points correspond to simulations returning complex roots. The time steps are chosen as $k_i = \frac{2^{i-2}}{\sqrt{K}}$, for $i = 1, 2, 3, 4$. Notice that k_3 is the limit of stability of the classic second-order accurate scheme for the simple harmonic oscillator, (22). For $s = \frac{1}{2}$, $\Delta^{(1)}$ is always positive, in accordance with the previous observation that for such value of s , $\Delta^{(1)}$ is unconditionally positive. However, the scheme is quite poorly behaved for higher values of α , especially under extreme initial conditions. Things look much better for $s = 1$, where the scheme always returns real roots for $v = 1$ m/s, as well as for $v = 20$ m/s when $\alpha = 1, 3$, $k = k_1, k_2, k_3$. When $s = 3$, the scheme always returns real roots, for both $v = 1$ and $v = 20$ m/s. In fact, in this case the minimum of $\Delta^{(\alpha)}$ seems to have reached an asymptote. Notice that the values of α selected for the figures are unreasonably high for applications in musical acoustics (in practice, one always chooses $1 \leq \alpha \leq 3$). However, it is remarkable that the scheme still works under such extreme conditions, at no extra computational cost.

A discussion for potentials (3b) and (3c) is not reported here, but the same conclusions apply.

Similar plots suggest that computability is increased as the parameter s is increased.

Summarising, empirical observations suggest that scheme (17) gives real roots in the following cases

- conditional realness for $1 \leq \alpha \leq 3$ if $s = 1$, the condition being (at worst) $k \leq \frac{2}{\sqrt{K}}$
- unconditional realness $\forall \alpha \geq 1$, if $s > 2$

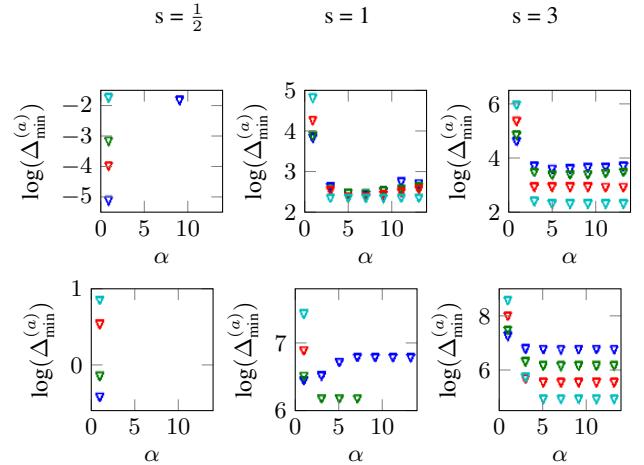


Figure 2: Computability of proposed scheme: values of $\Delta^{(\alpha)}$ for the selected values of the free parameter s . Missing points correspond to complex roots. For the simulations, $M = 1$ kg, $K = 10^3$, $\alpha \in [1, 3, 5, 7, 9, 11, 13]$. Time steps chosen as $k_1 = \sqrt{\frac{M}{4K}}$ (dark blue), $k_2 = \sqrt{\frac{M}{K}}$ (green), $k_3 = \sqrt{\frac{4M}{K}}$ (red), $k_4 = \sqrt{\frac{16M}{K}}$ (light blue). Notice that k_3 is the largest timestep allowed for the classic simple harmonic oscillator scheme, (22). Top row: initial velocity $v_0 = 1$ m/s, initial displacement $u_0 = -1$ mm. Bottom row: initial velocity $v_0 = 20$ m/s, initial displacement $u_0 = -1$ mm.

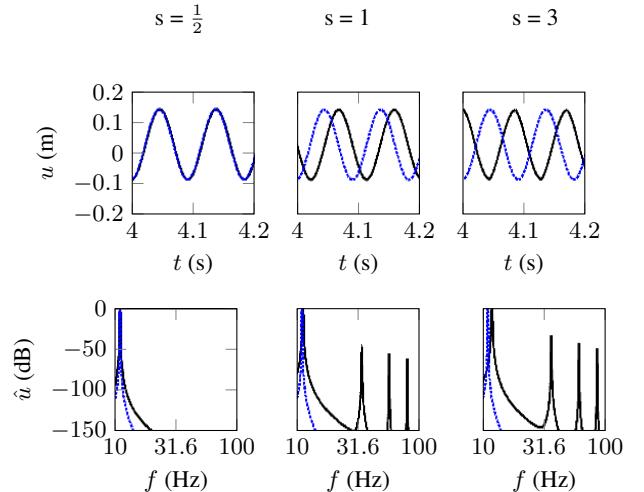


Figure 3: Simple harmonic oscillator. Comparison of current scheme, for potential (3a) with $\alpha = 1$ (solid line), and classic scheme (22) (dashed line). Top row: time domain. Bottom row: frequency domain. Free parameter s chosen as indicated on top. Natural frequency of the oscillator is $f_0 = 11$ Hz. Sampling rate chosen as $f_s = 2300$ Hz. Initial conditions: $v_0 = 0.5$ m/s, $u_0 = -0.1$ m.

4. APPLICATIONS

Scheme (17) is now tested against various benchmark schemes. First, the problem of the simple harmonic oscillator is considered. Then the dynamics of a particle colliding against a stiff barrier is studied, followed by the cubic oscillator.

4.1. Simple Harmonic Oscillator

In order to assess the properties of scheme (17), the simple harmonic oscillator is numerically simulated under various choices of the parameter s , and compared against a second-order accurate benchmark scheme (see [11] for details) given by

$$u^{n+1} = (2 - k^2 \bar{K})u^n - u^{n-1}, \quad k \leq \frac{2}{\sqrt{\bar{K}}}. \quad (22)$$

Fig. 3 presents a few such comparisons, for $s = \frac{1}{2}, 1, 3$. The question of accuracy for the current scheme is an interesting one. For the case of the simple harmonic oscillator considered here there are two sources of error: the first one is numerical dispersion (in fact, this is known as phase errors for the lossless case); the second one is due to the truncation of the Taylor series (6) to second order. Frequency domain analysis (i.e. z-transform techniques) are in this case out of hand, because for scheme (17) the values of the solution at the times $n+1, n, n-1$ appear nonlinearly even under linear conditions for the potential (3a).

There are some interesting facts about Fig. 3. First of all, the scheme is less and less accurate as s is increased. This observation is somewhat in contrast with the observation on existence of the roots of (17) (see also Fig. 2): there is a trade-off between accuracy and computability. In particular, for $s = 1, 3$ the fundamental frequency is higher than the one obtained with the classic scheme, resulting in the sinusoids shifting apart in the time domain. The second interesting fact is that $\forall s \neq \frac{1}{2}$ the simulated system is *nonlinear*, even though the model equation of the simple harmonic oscillator is completely linear. Nonlinearities appear as odd harmonics in the spectra of the cases for $s = 1, 3$. Even though such peaks are much lower in energy than the fundamental (for $s = 1$ the second harmonic is lower than 60 dB in amplitude), as s is increased they become more and more prominent. However, the amplitude of such peaks is insensitive to the initial conditions (in particular they do not grow when higher initial velocities or displacements are used).

It is of course the case to point out that scheme (17) is probably not very well suited for the problem of the simple harmonic oscillator, at least $\forall s \neq \frac{1}{2}$. This is because in general the scheme does not make a distinction between linear and nonlinear cases, so long as the potential ϕ is positive-definite and therefore the properties of existence of the roots and of positiveness of the discrete Hamiltonian are preserved. In other words, the scheme offers a general way to treat a large class of nonlinear problems, including the linear case as some sort of “sub-case”, but where the scheme remains nonlinear.

4.2. Colliding Mass

In this subsection the dynamics of a colliding mass against a stiff barrier is simulated. Fig. 4 presents the comparison of the current scheme, under various choices of the parameter s , and a bench-

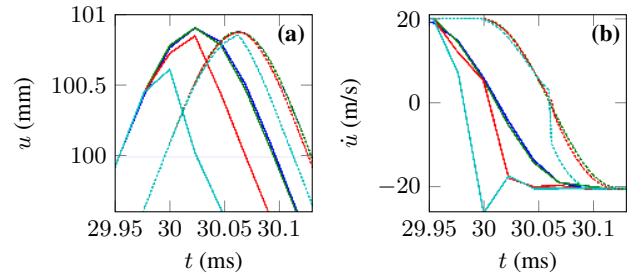


Figure 4: Collision of a fast particle against a stiff barrier. Comparison of benchmark scheme (blue) (23) and proposed scheme, for $s = \frac{1}{2}$ (green), $s = 1$ (red), $s = 3$ (cyan). Particle has mass $M = 1$ kg, and is started at $u_0 = -0.5$ m with initial velocity $v_0 = 20$ m/s against a barrier located at $h = 0.1$ m; barrier parameters are $K = 5 \cdot 10^9$, $\alpha = 1.3$. For both figures, solid lines and dashed lines are obtained using, respectively, $f_s = 44100$ and $f_s = 441000$. (a): Particle displacement during collision (spurious interpenetration). (b): Particle velocity during collision.

mark scheme presented in [1], which reads

$$y + \frac{k^2}{My} [\phi(y + u^{n-1}) - \phi(u^{n-1})] + 2u^{n-1} - 2u^n = 0, \quad (23)$$

where $y \triangleq u^{n+1} - u^{n-1}$. The scheme is second order accurate, and unconditionally stable provided that one is able to calculate y which appears implicitly in the argument of ϕ . In order to solve for such scheme, one must employ a nonlinear root finding algorithm, such as Newton-Raphson. Considering Fig. 4, it is seen that the current scheme departs more and more from the benchmark scheme as s is increased; this observation is consistent with what was already noted for the case of the harmonic oscillator. In particular, for $s = \frac{1}{2}$ the proposed scheme is virtually undistinguishable from the benchmark scheme, whereas for $s = 1, 3$ differences can be noticed. When the sampling rate is increased, such differences are reduced, providing evidence that the benchmark scheme and the proposed scheme display the same dynamics in the limit of infinite sampling rate, and $\forall s$.

From this simulation, it is interesting to plot the energy components for the benchmark scheme and for the proposed scheme. This is done in Fig. 5. In accordance to what was noted previously, for $s = 3$ the potential energy is not non-negative, but remains bounded from below and thus stability is guaranteed. When the sampling rate is increased, the minimum of the potential energy tends to zero.

4.3. Cubic Oscillator

Another interesting system is offered by the cubic oscillator, described by an equation of the type

$$\ddot{u} = -\frac{K}{M}u^3, \quad (24)$$

and for which a second-order accurate, unconditionally stable scheme is offered by (see [11])

$$u^{n+1} = \frac{2}{1 + \frac{K}{2M}k^2(u^n)^2}u^n - u^{n-1}. \quad (25)$$

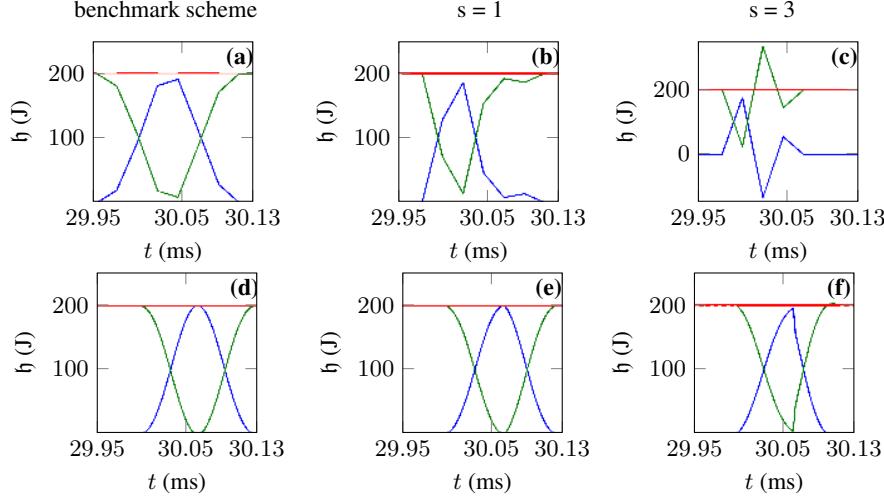


Figure 5: Collision of a fast particle against a stiff barrier, energy components. Comparison of benchmark scheme (23) and proposed scheme, for $s = 1$ and $s = 3$. For all plots, red line is total energy, green is kinetic, and blue is potential. (a)-(c): $f_s = 44100$, (d)-(f): $f_s = 441000$. The energy components are taken from the same simulation as Fig. 4

This benchmark scheme is compared in Fig. 6 to the proposed scheme for various values of s . As in the previous examples, the choice $s = \frac{1}{2}$ gives results that are virtually undistinguishable from the benchmark scheme, with more and more discrepancies as s is increased. As for the case of the simple harmonic oscillator, several spectral peaks appear for $s \neq \frac{1}{2}$ in the higher frequency range, and also the oscillations of the solution in the proposed scheme are a little faster than for the benchmark scheme. With respect to the case of the simple harmonic oscillator, in this case the spectral content of the solution is sensitive to the initial conditions, and in particular the oscillator displays a hardening effect (i.e. shift of the spectrum to higher frequencies for higher initial velocities and displacements). Unlike the case of the simple harmonic oscillator, the spurious spectral peaks appearing in the spectrum for $s \neq \frac{1}{2}$ also display such hardening effect, resulting in some high frequency spectral content which can be quite clearly heard when comparing against the benchmark scheme. Increasing the sampling rate reduces this perceptual effect.

5. CONCLUSIONS

Nonlinear forces, and in particular collisions, are of prime importance for many applications in musical acoustics. In this paper, a novel family of finite difference schemes was presented for collisions in the lumped case, as well as for nonlinear oscillators of any odd power. With respect to previous numerical models, the current scheme requires only the evaluation of a square root at each update, therefore no iterative root finders are needed. The scheme is energy-conserving, and conditions for boundedness of the nonlinear energy can be given in terms a free parameter in the Hamiltonian. The robustness of the algorithm was tested for a large number of cases, showing very good computability properties $\forall \alpha \geq 1$ for a choice of the free parameter $s \geq 1$. The choice of $s = \frac{1}{2}$ gives the most accurate results, however preliminary brute force analysis shows that such case is also the least computable (i.e. the roots of the quadratic are complex in many cases). Although brute force

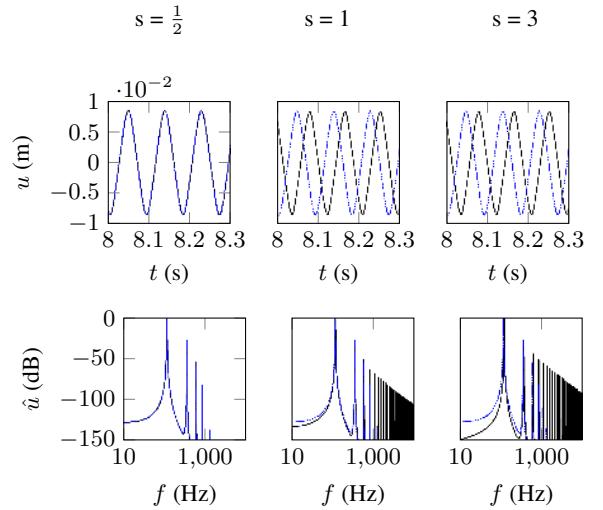


Figure 6: Cubic oscillator. Comparison of current scheme, for potential (3a) with $\alpha = 3$ (solid black line), and benchmark scheme (25) (dashed blue line). Top row: time domain. Bottom row: frequency domain. Free parameter s chosen as indicated on top. Stiffness of the oscillator is $K = 10^{10}$, and mass is $M = 1$ kg. Sampling rate is $f_s = 44100$. Initial conditions: $v_0 = 5$ m/s, $u_0 = 0$ m.

analysis cannot be exhaustive, there is strong evidence that higher values of s do increase the overall robustness of the algorithm, at the expense of accuracy. It is hoped that the current algorithm can be extended to the case of collisions of one lumped and one distributed object (for example, a string and a rattle), for real-time applications.

6. ACKNOWLEDGMENTS

This work was supported by the Royal Society and by the British Academy, through a Newton International Fellowship. Dr Stefan Bilbao is kindly acknowledged for his precious comments on the properties of the scheme.

7. REFERENCES

- [1] S. Bilbao, A. Torin, and V. Chatzioannou, “Numerical modeling of collisions in musical instruments,” *Acta Acustica united with Acustica*, vol. 101, no. 1, pp. 155–173, 2015.
- [2] D. W. Marhefka and D. E. Orin, “A compliant contact model with nonlinear damping for simulation of robotic systems,” *IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans*, vol. 29, no. 6, pp. 566–572, 1999.
- [3] D. Baraff, “Fast contact force computation for nonpenetrating rigid bodies,” in *Proceedings of the 21st Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH)*, 1994.
- [4] P. Wriggers and T. Laursen, *Computational contact mechanics*, Springer-Verlag, Vienna, 2008.
- [5] A. Krishnaswamy and J. O. Smith, “Methods for simulating string collisions with rigid spatial obstacles,” in *2003 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*, 2003.
- [6] C. P. Vyasarayani, S. Birkett, and J. McPhee, “Modeling the dynamics of a vibrating string with a finite distributed unilateral constraint: Application to the sitar,” *The Journal of the Acoustical Society of America*, vol. 125, no. 6, pp. 3673–3682, 2009.
- [7] C. Issanchou, J-L. Le Carrou, S. Bilbao, C. Touzé, and O. Doaré, “A modal approach to the numerical simulation of a string vibrating against an obstacle: applications to sound synthesis,” in *Proceedings of the 19th Conference on Digital Audio Effects (DAFx-16)*, 2016.
- [8] V. Chatzioannou and M. Van Walstijn, “An energy conserving finite difference scheme for simulation of collisions,” in *Proceedings of the Stockholm Musical Acoustics Conference (SMAC)*, 2013.
- [9] M. van Walstijn, J. Bridges, and S. Mehes, “A real-time synthesis oriented tanpura model,” in *Proceedings of the 19th International Conference on Digital Audio Effects (DAFx-16)*, 2016.
- [10] S. Bilbao, “Numerical modeling of string barrier collisions.,” in *Proceedings of the International Symposium on Musical Acoustics (ISMA)*, 2014.
- [11] S. Bilbao, *Numerical Sound Synthesis: Finite Difference Schemes and Simulation in Musical Acoustics*, Wiley, Chichester, UK, 2009.

TRAJECTORY ANTI-ALIASING ON GUARANTEED-PASSIVE SIMULATION OF NONLINEAR PHYSICAL SYSTEMS

Rémy Muller, Thomas Hélie *

S3AM team, IRCAM - CNRS UMR 9912- UPMC
1 place Igor Stravinsky, 75004 Paris, France
remy.muller@ircam.fr

ABSTRACT

This article is concerned with the accurate simulation of passive nonlinear dynamical systems with a particular attention paid on aliasing reduction in the pass-band. The approach is based on the combination of Port-Hamiltonian Systems, continuous-time state-space trajectories reconstruction and exact continuous-time anti-aliasing filter realization. The proposed framework is applied on a nonlinear LC oscillator circuit to study the effectiveness of the method.

1. INTRODUCTION

The need for accurate and passive-guaranteed simulation of nonlinear multi-physical systems is ubiquitous in the modelling of electronic circuits or mechanical systems.

Geometric numerical integration [1] is a very active research field that provides a theoretical framework for structure and invariant preserving integration of dynamical systems. Port-Hamiltonian Systems (PHS) [2] [3] that focus on the energy storage functions and power continuous component interconnections belong to this field and offer a well adapted framework to preserve the system energy (resp. passivity). In the context of nonlinear physical audio systems, it has been applied successfully to the modelling of the wah-wah pedal [4], Fender Rhodes [5], brass instruments [6] and the loudspeaker nonlinearities [7]. Automatic generation of the system equations from a graph of components has been investigated in [8]

However the presence of aliasing errors in the numerical simulation is annoying for three reasons. First it causes audible inharmonic audio artefacts. Second it deteriorates the accuracy of the numerical scheme leading to poor convergence rate. Third it requires the use of significant oversampling. This problem is even more pronounced in the case of systems such as sustained instruments that rely on nonlinearities to achieve auto-oscillation.

Aliasing errors in the context of finite elements simulation and some alternatives have been discussed in [9] (ch 11). Anti-aliased waveform generation without oversampling has been proposed in [10]. Static nonlinearity anti-aliasing has also been proposed in [11] [12] by combining exact anti-derivatives and finite-differences.

Continuous-time input reconstruction has been used in [13] to simulate the frequency response of LTI systems with higher accuracy. It is also central in collocation-based Runge-Kutta methods

that rely on non-uniform polynomial interpolation of the vector field. Splines and in particular uniform B-splines [14] [15] [16], [17] also offer a particularly interesting framework to represent and manipulate piecewise continuous-time signals through their digital representations using the standard tools of linear algebra and digital signal processing.

In this article, we try to combine the geometric and the signal processing viewpoints: we choose a physically informed piecewise smooth polynomial reconstruction model based on a discrete sequence of points generated by a passive-guaranteed simulation method.

The paper is organized as follows. We first recall some results about Port-Hamiltonian systems in Section 3, then we consider passive numerical methods in section 4, we talk about piecewise-continuous trajectory reconstruction in section 5 and continuous-time filtering of piecewise polynomials in section 6. Finally we apply our method to a non linear LC oscillator circuit in section 7.

2. PROBLEM STATEMENT

2.1. Objective

The objective is to simulate nonlinear passive physical audio systems in such a way that:

- (i) The nonlinear dynamics is accurately reproduced,
- (ii) The power balance decomposed into its conservative, dissipative and source parts is satisfied,
- (iii) The observation operator is designed to reduce the aliasing induced by the nonlinearities.

2.2. Approach

To address this problem, the following strategy is adopted.

First, trajectories are approximated in the continuous-time domain by smooth parametric piecewise-defined functions, such that the three following properties are fulfilled:

- (P1) Regularity: functions and junctions are \mathcal{C}^k with $k \in \mathbb{N}$,
- (P2) Accuracy: the approximation has accuracy order p ,
- (P3) Passivity: the power balance is globally satisfied for each frame.

Second, the anti-aliased output is built *a posteriori* in three steps:

1. Observe the output from the approximated dynamics in the continuous-time domain,
2. Apply a continuous-time anti-aliasing filter in order to respect the Shannon-Nyquist sampling theorem,
3. Sample the filtered trajectories to convert them back to discrete-time.

* The contribution of this author has been done at laboratory STMS, Paris, within the context of the French National Research Agency sponsored project INFIDHEM. Further information is available at <http://www.lagep.cpe.fr/wwwlagep7/anr-dfg-infidhem-fev-2017-jan-2020/>

2.3. Methodology

In this article, we restrict ourselves to piece-wise continuous globally C^1 polynomial trajectories of the form

$$\hat{\mathbf{x}}(t) = \sum_{n=-\infty}^{\infty} \hat{\mathbf{x}}_n \left(\frac{t-t_n}{h} \right) \text{rect}_{[0,1]} \left(\frac{t-t_n}{h} \right), \quad t \in \mathbb{R} \quad (1)$$

with $\hat{\mathbf{x}} \in \mathbb{R}^N$, $\hat{\mathbf{x}}_n(\tau)$, $\tau \in [0, 1]$ being a local polynomial model of order r , $t_n = hn$, $n \in \mathbb{Z}$ and h being the time step parameter. The continuity hypothesis (P1) is expressed mathematically by.

$$\hat{\mathbf{x}}_{n+1}^{(\ell)}(\tau) = \hat{\mathbf{x}}_n^{(\ell)}(\tau) \quad \forall n \in \mathbb{Z}, \ell \leq k \quad (2)$$

For property (P2) the local approximation error between the exact solution and its approximation is defined by

$$e(h) = \mathbf{x}(t_0 + h) - \hat{\mathbf{x}}(t_0 + h) \quad (3)$$

provided that $\mathbf{x}(t_0) = \hat{\mathbf{x}}(t_0)$ and it is required that for some p .

$$e(h) = \mathcal{O}(h^{p+1}) \quad (4)$$

Finally to express property (P3) we require the power-balance

$$E'(t) = -\mathcal{P}_d + \mathcal{P}_e \quad (5)$$

where \mathcal{P}_d and \mathcal{P}_e are respectively the dissipated and external power and $E'(t)$ is the instantaneous energy variation of the system.

3. PORT-HAMILTONIAN SYSTEMS

In this article, nonlinear passive physical audio systems are described under their Port-Hamiltonian formulation. The theory of Port-Hamiltonian Systems (PHS) [2] [3] extends the theory of Hamiltonian mechanics to non-autonomous and dissipative open systems. It provides a general framework where the dynamic state-space equations derives directly from an energy storage function and *power-conserving* interconnection of its subsystems.

3.1. Explicit differential form

Consider a system with input $\mathbf{u}(t) \in \mathbb{U} = \mathbb{R}^P$, with state $\mathbf{x}(t) \in \mathbb{X} = \mathbb{R}^N$ and output $\mathbf{y}(t) \in \mathbb{Y} = \mathbb{R}^P$ with the structured state-space equations [2]

$$\begin{cases} \mathbf{x}' = (\mathbf{J}(\mathbf{x}) - \mathbf{R}(\mathbf{x})) \nabla \mathcal{H}(\mathbf{x}) + \mathbf{G}(\mathbf{x})\mathbf{u} = f(\mathbf{x}, \mathbf{u}) \\ \mathbf{y} = \mathbf{G}(\mathbf{x})^T \mathbf{u} \end{cases} \quad (6)$$

where \mathcal{H} gives the stored energy of the system

$$E(t) = (\mathcal{H} \circ \mathbf{x})(t) \quad (7)$$

with $\mathcal{H} \in C^1(\mathbb{X}, \mathbb{R}^+)$, ∇ being the gradient operator, $\mathbf{J} = -\mathbf{J}^T$ a skew-symmetric matrix and $\mathbf{R} = \mathbf{R}^T \succeq 0$ a positive-semidefinite matrix. The energy variation of this system satisfies the power-balance given by the derivative chain rule

$$E'(t) = \nabla \mathcal{H}(\mathbf{x})^T \mathbf{x}' \quad (8)$$

which can be decomposed as

$$E'(t) = \mathcal{P}_c - \mathcal{P}_d + \mathcal{P}_e \quad (9)$$

with.

$$\mathcal{P}_c = \nabla \mathcal{H}(\mathbf{x})^T \mathbf{J}(\mathbf{x}) \nabla \mathcal{H}(\mathbf{x}) = 0 \quad (10)$$

$$\mathcal{P}_d = \nabla \mathcal{H}(\mathbf{x})^T \mathbf{R}(\mathbf{x}) \nabla \mathcal{H}(\mathbf{x}) \geq 0 \quad (11)$$

$$\mathcal{P}_e = \nabla \mathcal{H}(\mathbf{x})^T \mathbf{G}(\mathbf{x}) \mathbf{u} \quad (12)$$

The \mathcal{P}_c term is null because \mathbf{J} is skew-symmetric: it represents conservative power exchange between storage components in the system. The \mathcal{P}_d term is positive because $\mathbf{R} \geq 0$: it represents the dissipated power. Finally the term \mathcal{P}_e represents the power brought to the system by the external ports.

Equation (9) express the system's *passivity property*: with external inputs switched off ($\mathbf{u} = 0$) the energy can either be constant (conservative case $\mathcal{P}_d = 0$) or decaying (dissipative case $\mathcal{P}_d > 0$).

3.2. Component-based approach and semi-explicit DAE form

More generally, PHS can be expressed in Differential Algebraic Equation form. When we consider physical systems containing N energy-storage components, M dissipative components and P external interaction ports described by

\mathcal{P}_c the stored energy level e_n and its variation law defined by $e'_n = \nabla \mathcal{H}_n(x_n)x'_n$ for the state variable x_n .

\mathcal{P}_d the dissipated power $q_m(w) \geq 0$ with the component's flux and effort variables being in algebraic relation of a single variable w .

\mathcal{P}_e the external power $u_p y_p$ brought to the system through this port with u_p being the controllable input of the system and y_p being the observable output.

For a storage component, $e_n = \mathcal{H}_n(x_n)$ gives the physical *energy storage law*. If x'_n is a flux (resp. effort) variable then $\nabla \mathcal{H}_n(x_n)$ is the dual effort (resp. flux) variable.

Similarly, for a dissipative component, the power is $q_m = R_m(w_m)$ so that if w_m is a flux (resp. effort) variable then $z(w_m) = \frac{R_m(w_m)}{w_m}$ is the effort (resp. flux) and gives the *dissipation law*.

We then consider a passive system obtained by interconnection of these components given by

$$\underbrace{\begin{bmatrix} \mathbf{x}' \\ \mathbf{w} \\ -\mathbf{y} \end{bmatrix}}_{\mathbf{b}} = \mathbf{S}(\mathbf{x}, \mathbf{w}) \underbrace{\begin{bmatrix} \nabla \mathcal{H}(\mathbf{x}) \\ z(\mathbf{w}) \\ \mathbf{u} \end{bmatrix}}_{\mathbf{a}} \quad (13)$$

with $\mathbf{S} = -\mathbf{S}^T$ being skew-symmetric, $\mathcal{H}(\mathbf{x}) = \sum_{i=1}^N \mathcal{H}_i(x_i)$ and $z(\mathbf{w}) = [z_1(w_1), \dots, z_m(w_m)]^T$.

The \mathbf{S} matrix represents the power exchange between components: since $\mathbf{S} = -\mathbf{S}^T$ we have $\mathbf{a} \cdot \mathbf{b} = \mathbf{a}^T \mathbf{S} \mathbf{a} = 0$ which again leads to the power balance¹.

$$\underbrace{\nabla \mathcal{H}(\mathbf{x}) \cdot \mathbf{x}'}_{\mathcal{P}_c=E'(t)} + \underbrace{z(\mathbf{w}) \cdot \mathbf{w}}_{\mathcal{P}_d} - \underbrace{\mathbf{u} \cdot \mathbf{y}}_{\mathcal{P}_e} = 0 \quad (14)$$

The explicit form (6) can be found by solving the second row of (13). The \mathbf{S} matrix represents a *Dirac structure* [2] that expresses the power-balance and can be constructed from a component connection graph [8] [18].

¹The minus sign in $-y$ in Eq. (13) is used to restore the receiver convention used for internal components.

4. PASSIVE NUMERICAL INTEGRATION

Whereas most numerical schemes concentrate their efforts on the temporal derivative or the numerical integration quadrature, discrete gradient methods preserve the energy (resp. passivity) given by the power-balance (9), (14) in discrete-time by providing a discrete equivalent of the chain rule derivation property $E'(t) = \nabla\mathcal{H}(\mathbf{x})^T \dot{\mathbf{x}}$. A discrete gradient [19] $\bar{\nabla}\mathcal{H}$ is required to satisfy the following conditions.

$$\mathcal{H}(\mathbf{x} + \delta\mathbf{x}) - \mathcal{H}(\mathbf{x}) = \bar{\nabla}\mathcal{H}(\mathbf{x}, \delta\mathbf{x})^T \delta\mathbf{x} \quad (15)$$

$$\bar{\nabla}\mathcal{H}(\mathbf{x}, 0) = \nabla\mathcal{H}(\mathbf{x}) \quad (16)$$

In this article, we will focus on the average vector field [20].

4.1. Average Vector Field

In the general case, the AVF method is defined by.

$$\frac{\delta\mathbf{x}_n}{\delta t} = \int_0^1 f(\mathbf{x}_n + \tau\delta\mathbf{x}_n) d\tau, \quad \mathbf{x}_{n+1} = \mathbf{x}_n + \delta\mathbf{x}_n \quad (17)$$

When the matrices $\mathbf{J}(\mathbf{x}), \mathbf{R}(\mathbf{x}), \mathbf{G}(\mathbf{x})$ are approximated by constant matrices $\bar{\mathbf{J}}, \bar{\mathbf{R}}, \bar{\mathbf{G}}$, we obtain the separable structure-preserving approximation of (17)

$$\frac{\delta\mathbf{x}_n}{\delta t} = (\bar{\mathbf{J}} - \bar{\mathbf{R}})\bar{\nabla}\mathcal{H}(\mathbf{x}_n, \delta\mathbf{x}_n) + \bar{\mathbf{G}}\bar{\mathbf{u}}_n \quad (18)$$

with the discrete gradient being defined by

$$\bar{\nabla}\mathcal{H}(\mathbf{x}, \delta\mathbf{x}) = \int_0^1 \nabla\mathcal{H}(\mathbf{x} + \tau\delta\mathbf{x}) d\tau \quad (19)$$

and it satisfies the *discrete power balance*

$$\begin{aligned} \delta E &= \bar{\nabla}\mathcal{H}^T \frac{\delta\mathbf{x}}{\delta t} = \bar{\nabla}\mathcal{H}^T (\bar{\mathbf{J}} - \bar{\mathbf{R}})\bar{\nabla}\mathcal{H} + \bar{\nabla}\mathcal{H}^T \bar{\mathbf{G}}\bar{\mathbf{u}} \\ &= 0 - \mathcal{P}_d + \mathcal{P}_e \end{aligned}$$

Then, by the fundamental theorem of calculus, for mono-variant components, i.e. separable Hamiltonians of the form $\mathcal{H}(\mathbf{x}) = \sum_{i=1}^N \mathcal{H}_i(x_i)$, we have for each coordinate:

$$\bar{\nabla}\mathcal{H}_i(x_i, \delta x_i) = \begin{cases} \frac{\mathcal{H}_i(x_i + \delta x_i) - \mathcal{H}_i(x_i)}{\delta x_i} & \delta x_i \neq 0 \\ \nabla\mathcal{H}(x_i) & \delta x_i = 0 \end{cases} \quad (20)$$

which satisfies the discrete gradient conditions (15)-(16). For non-separable Hamiltonians, a discrete-gradient can also be uniquely defined, see [21] for more details.

To summarize, this method relies on two complimentary approximations: the differential operator $\frac{d\mathbf{x}}{dt} \rightarrow \frac{\delta\mathbf{x}}{\delta t}$ and the vector field $f \rightarrow \bar{f}$ to achieve energy (resp. passivity) conservation. The discrete PHS equivalent of (6) is given by the numerical scheme.

$$\begin{cases} \frac{\delta\mathbf{x}_n}{\delta t} &= (\bar{\mathbf{J}} - \bar{\mathbf{R}})\bar{\nabla}\mathcal{H}(\mathbf{x}_n, \delta\mathbf{x}_n) + \bar{\mathbf{G}}\bar{\mathbf{u}}_n \\ \mathbf{y}_n &= \bar{\mathbf{G}}^T \bar{\nabla}\mathcal{H}(\mathbf{x}_n, \delta\mathbf{x}_n) \\ \mathbf{x}_{n+1} &= \mathbf{x}_n + \delta\mathbf{x}_n \end{cases} \quad (21)$$

4.2. Accuracy order

As shown in [22], the AVF has accuracy order $p = 2$, it is a B-series method, is affine-covariant and self-adjoint. When approximated as in Eq (19) by evaluating matrices $\mathbf{J}, \mathbf{R}, \mathbf{G}$ for $\mathbf{x}^* = \mathbf{x}_n$ the accuracy is only of order 1. Order 2 is achieved when either $\mathbf{J}, \mathbf{R}, \mathbf{G}$ are independent of \mathbf{x} or when evaluated at the mid-point $\mathbf{x}^* = \mathbf{x}_n + \frac{\delta\mathbf{x}_n}{2}$ in the conservative case. It is also possible to restore the accuracy order $p = 2$ in the general case using a Runge-Kutta refinement [21].

4.3. Implicit resolution

The discrete system is implicit on $\delta\mathbf{x}_n$ and admits a unique solution when \mathcal{H} is convex. In the general case, an iterative solver is required (typically a fixed-point or Newton iteration), but when the Hamiltonian is quadratic we can avoid the need for an iterative resolution. Furthermore, when the Hamiltonian is convex the method can also be made non-iterative by quadratization of the Hamiltonian [21].

Proof. When the Hamiltonian is quadratic of the form $\mathcal{H}(\mathbf{x}) = \frac{1}{2}\mathbf{x}^T \mathbf{Q} \mathbf{x}$, the discrete gradient reduces to the mid-point rule

$$\bar{\nabla}\mathcal{H}(\mathbf{x}, \delta\mathbf{x}) = \int_0^1 \mathbf{Q}(\mathbf{x}_n + \delta\mathbf{x}_n \tau) d\tau = \mathbf{Q} \left(\mathbf{x}_n + \frac{1}{2} \delta\mathbf{x}_n \right)$$

the implicit dependency on $\delta\mathbf{x}$ can thus be solved by matrix inversion

$$\delta\mathbf{x}_n = \delta t \left(I - \frac{\delta t}{2} \mathbf{A} \right)^{-1} (\mathbf{A}\mathbf{x}_n + \bar{\mathbf{G}}\bar{\mathbf{u}}_n) \quad (22)$$

with $\mathbf{A} = (\bar{\mathbf{J}} - \bar{\mathbf{R}})\mathbf{Q}$ □

5. PIECEWISE-CONTINUOUS TRAJECTORIES

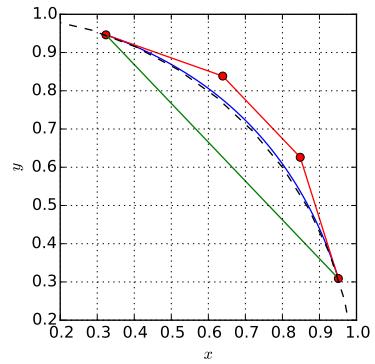


Figure 1: Example of a cubic trajectory with conservative endpoints. The affine trajectory used to compute the average vector field is shown (in green), the associated cubic interpolated approximation (in blue), its control polygon (in red), and the exact manifold (in dashed black).

Given the sequence of points $\{\mathbf{x}_n\}$ obtained by a passive-guaranteed method, we would like to reconstruct piece-wise \mathcal{C}^k -continuous polynomial trajectories informed by the system dynamics.

The idea is to exploit the dynamic equation at each junction point \mathbf{x}_n where the approximation is known to be $\mathcal{O}(h^{p+1})$.

Indeed, if we had the samples of the exact trajectory, by the Weierstrass approximation theorem, arbitrarily close polynomial approximations converging uniformly to the exact solution could be obtained by computing its derivatives to any desired order.

Since we only have an approximation of order $p = 2$, we restrict ourselves to a regularity $k = 1$. This gives four constraints

$$\hat{\mathbf{x}}(0) = \mathbf{x}_n, \quad \hat{\mathbf{x}}(1) = \mathbf{x}_{n+1}, \quad \hat{\mathbf{x}}'(0) = f(\mathbf{x}_n), \quad \hat{\mathbf{x}}'(1) = f(\mathbf{x}_{n+1})$$

that can be satisfied by a cubic polynomial ($r = 3$). We choose to represent it using the Bézier form,

$$\hat{\mathbf{x}}(\tau) = \sum_{i=0}^3 \mathbf{X}_i B_i^3(\tau), \quad B_i^n(t) = \binom{n}{i} (1-t)^{n-i} t^i \quad (23)$$

with $\{\mathbf{X}_i\}$ being its control polygon and $B_i^n(t)$ being the Bernstein polynomial basis functions, because they have important geometric and finite differences interpretations [23].

This choice immediately leads to the following equations,

$$\mathbf{X}_0 = \mathbf{x}_n \quad \mathbf{X}_1 = \mathbf{x}_n + \frac{1}{3} f(\mathbf{x}_n) \quad (24)$$

$$\mathbf{X}_3 = \mathbf{x}_{n+1} \quad \mathbf{X}_2 = \mathbf{x}_{n+1} - \frac{1}{3} f(\mathbf{x}_{n+1}) \quad (25)$$

where the internal control points $\mathbf{X}_1, \mathbf{X}_2$ are computed from the end points $\mathbf{x}_n, \mathbf{x}_{n+1}$ by first order forward / backward prediction using the derivative rule.

$$\hat{\mathbf{x}}'(t) = \sum_{i=0}^{n-1} \mathbf{D}_i B_i^{n-1}(t), \quad \mathbf{D}_i = n(\mathbf{X}_{i+1} - \mathbf{X}_i) \quad (26)$$

An example trajectory is shown in Figure 1.

6. ANTI-ALIASED OBSERVATION

Given an observed signal $\tilde{\mathbf{u}}(t) = \mathbf{y}(t)$ belonging to the class of piecewise polynomials, in order to reject the non-band-limited part of the spectrum, we would like to apply an antialiasing filter operator given by its continuous-time ARMA transfer function $H(s)$, then sample its output $\tilde{y}(t)$ to get back to the digital domain.

Since our anti-aliasing filter will be LTI, we will make use of *exact exponential integration* and decompose its output on a *custom basis* of exponential polynomial functions.

Without loss of generality we only consider single-input single-output filters (SISO) since we can always filter each observed output independently.

6.1. State-space ARMA filtering of polynomial input

We want to filter the trajectory by an ARMA filter given by its Laplace transfer function

$$H(s) = \frac{Y(s)}{U(s)} = \frac{b_0 s^N + b_1 s^{N-1} + \dots + b_N}{s^N + a_1 s^{N-1} + \dots + a_N} \quad (27)$$

This filter can be realized in state-space form as

$$\tilde{\mathbf{x}}' = \mathbf{A}\tilde{\mathbf{x}} + \mathbf{B}\tilde{u} \quad (28)$$

$$\tilde{y} = \mathbf{C}\tilde{\mathbf{x}} + \mathbf{D}\tilde{u} \quad (29)$$

Common choices are the observable and controllable state-space forms.

Furthermore when the denominator can be factored with distinct roots, it is possible to rewrite the transfer function using partial fraction expansion as.

$$H(s) = c_0 + \frac{c_1}{s - \lambda_1} + \dots + \frac{c_N}{s - \lambda_N} \quad (30)$$

which leads to the canonical diagonal form

$$\mathbf{A} = \begin{bmatrix} \lambda_1 & & \\ & \ddots & \\ & & \lambda_N \end{bmatrix} \quad \mathbf{B} = \begin{bmatrix} 1 \\ \vdots \\ 1 \end{bmatrix} \quad (31)$$

$$\mathbf{C} = [c_1 \quad \dots \quad c_N] \quad \mathbf{D} = [c_0] \quad (32)$$

6.2. Exact exponential integration

The exact state trajectory is given by the integral

$$\tilde{\mathbf{x}}(t) = \tilde{\mathbf{x}}_h(t) + \tilde{\mathbf{x}}_e(t) = e^{\mathbf{A}t} \tilde{\mathbf{x}}_0 + \int_0^t e^{\mathbf{A}(t-\tau)} \mathbf{B} \tilde{u}(\tau) d\tau \quad (33)$$

as the sum of the homogeneous solution to the initial conditions $\tilde{\mathbf{x}}_h$ and the forced state-response with zero initial conditions $\tilde{\mathbf{x}}_e$ given by the convolution of the input with the kernel $e^{\mathbf{A}t}$.

Furthermore when \mathbf{A} is diagonal we have

$$e^{\mathbf{A}t} = \begin{bmatrix} e^{\lambda_1 t} & & \\ & \ddots & \\ & & e^{\lambda_N t} \end{bmatrix} \quad (34)$$

which greatly simplifies the computation of the exponential map. In that case (33) can be evaluated component-wise as

$$\tilde{x}^i(t) = e^{\lambda_i t} \tilde{x}_0^i + \int_0^t e^{\lambda_i(t-\tau)} \tilde{u}(\tau) d\tau \quad i \in \{1 \dots N\} \quad (35)$$

where we used the notation x^i to denote the i -th coordinate of the vector \mathbf{x}

6.2.1. Polynomial input

With $\tilde{u}(t)$ being a polynomial of degree K in monomial² form and coefficients \tilde{u}_k

$$\tilde{u}(t) = \sum_{k=0}^K \tilde{u}_k \frac{t^k}{k!} \quad (36)$$

we can expand the forced response $\tilde{\mathbf{x}}_e$ in (35) as a weighted sum

$$\int_0^t e^{\lambda_i(t-\tau)} \left(\sum_{k=0}^K \tilde{u}_k \frac{t^k}{k!} \right) d\tau = \sum_{k=0}^K \tilde{u}_k \varphi_{k+1}(\lambda_i, t) \quad (37)$$

with the basis functions $\{\varphi_k\}$ being defined by the convolution

$$\varphi_k(\lambda, t) = \int_0^t e^{\lambda(t-\tau)} \frac{\tau^{k-1}}{(k-1)!} d\tau \quad k \geq 1 \quad (38)$$

One of the main advantages of using a polynomial input (rather than a more general model) lies in the fact that these basis functions can be integrated exactly, avoiding the need of a quadrature

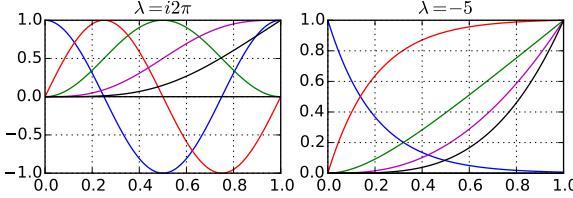


Figure 2: Normalized φ -functions for $k \in \{0 \dots 4\}$. The real parts of the impulse (blue), step (red), ramp (green), quadratic (magenta) and cubic (black) responses are shown for a complex pole $\lambda = i2\pi$ (left plot) and a real pole $\lambda = -5$ (right plot) over the unit interval $t \in [0, 1]$.

approximation formula. See Appendix 12 for a detailed derivation and a recursive formula, and Figure 2 for their temporal shapes.

Using those we can decompose the local state trajectories as.

$$\tilde{x}^i(t) = \tilde{x}_0^i \varphi_0(\lambda_i, t) + \sum_{k=0}^K \tilde{u}_k \varphi_{k+1}(\lambda_i, t) \quad (39)$$

We note that the initial condition is equivalent to an impulsive input $\tilde{x}_0^i \delta(t)$. This filtering scheme can thus be generalized to non polynomial impulsive inputs.

6.2.2. Numerical update scheme

Since we only wish to sample the trajectory on a fixed grid $t_n \in \mathbb{Z}$, we just need to evaluate the local state trajectory $\mathbf{x}(t)$ and the output $y(t)$ at $t = 1$ to finally get the following numerical scheme

$$\tilde{x}_{n+1}^i = \tilde{x}_n^i \varphi_0(\lambda_i) + \sum_{k=0}^K \tilde{u}_{k,n} \varphi_{k+1}(\lambda_i, 1) \quad (40)$$

$$\tilde{y}_{n+1} = \sum_{i=1}^N c_i \tilde{x}_{n+1}^i + c_0 \tilde{u}_n(1) \quad (41)$$

where the coefficients $\varphi_k(\lambda_i, 1)$ can be pre-computed and the components \tilde{x}_{n+1}^i evaluated in parallel.

6.3. Filter examples

6.3.1. Low-pass filter of order 1

We consider a first order low-pass filter with transfer function $H(s) = \frac{a}{s+a}$. The temporal response to a piecewise polynomial input $\{t^2, 1-t, 0, 1\}$ is shown in Figure 3 for $a \in \{1, 3, 6, 10\}$.

6.3.2. Butterworth Filter of order 3

To further illustrate the non-band-limited representation capacity of piece-wise polynomials, and the effectiveness of the filtering scheme, we have shown in Figure 4 the response of a third-order Butterworth filter with cutoff $\omega_c = \pi$ to a triangular input signal. Its Laplace transfer function for a normalized pulsation $\omega_c = 1$ is given by $H(s) = \frac{1}{(s^2+s+1)(s+1)}$ with poles $\lambda_1 = \frac{-1-i\sqrt{3}}{2}$, $\lambda_2 = \frac{-1+i\sqrt{3}}{2}$, $\lambda_3 = -1$ and coefficients $c_0 = 0$, $c_1 = \frac{-3+i\sqrt{3}}{6}$, $c_2 = \frac{-3-i\sqrt{3}}{6}$, $c_3 = 1$.

²We use the monomial form here instead of Bernstein polynomials because this is the one that leads to the most straightforward and meaningful derivation.

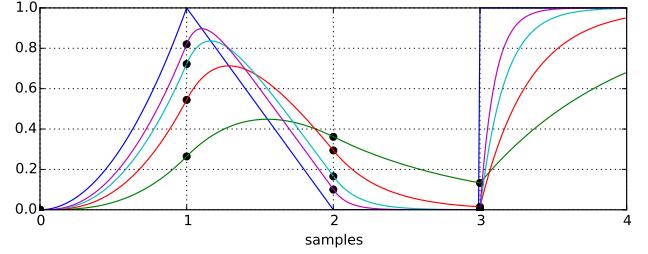


Figure 3: Exact continuous-time responses of a first order low-pass filter to a polynomial input (in blue).

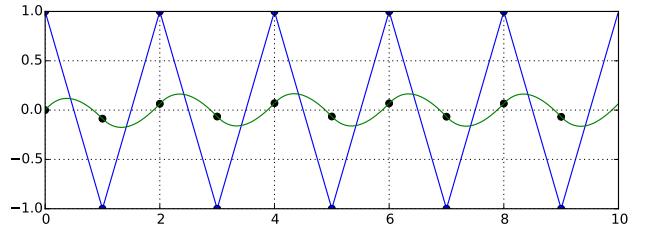


Figure 4: Exact continuous-time response of the order 3 Butterworth filter with cutoff pulsation $\omega_c = \pi$ to a triangle input at the Nyquist frequency.

7. APPLICATION: NONLINEAR LC OSCILLATOR

In order to illustrate the proposed method, we consider the simplest example having non linear dynamics. For that purpose, we use a parallel autonomous LC circuit with a linear inductor and a saturating capacitor with the Hamiltonian energy storage function given by

$$\mathcal{H}(q, \phi) = \frac{\ln(\cosh(q))}{C_0} + \frac{\phi}{2L} \quad (42)$$

where the state q is the charge of the capacitor and ϕ the flux in the inductor. Its circuit's schematic is shown in figure 5 and its energy storage law are displayed in 6

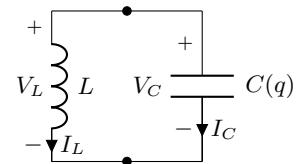


Figure 5: A nonlinear LC oscillator circuit

By partial differentiation of the Hamiltonian function \mathcal{H} by respectively q and ϕ we get the capacitor's voltage and the inductor's current, while applying the temporal derivative on q , ϕ gives the capacitor's current and inductor's voltage.

$$V_C = \partial_q \mathcal{H} = \frac{\tanh(q)}{C_0} \quad I_C = q' \quad (43)$$

$$I_L = \partial_\phi \mathcal{H} = \frac{\phi'}{L} \quad V_L = \phi' \quad (44)$$

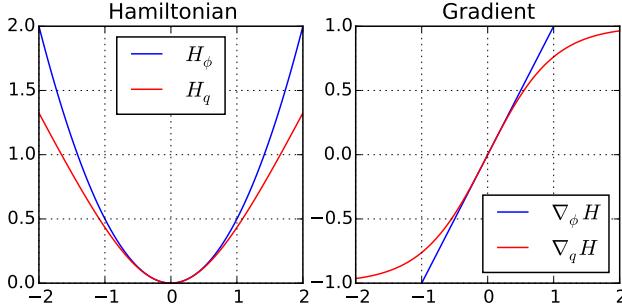


Figure 6: Respective energy storage functions (left plot) and their gradients (right plot), of the nonlinear capacitor (in red) and linear inductor (in blue), for $C = 1, L = 1$.

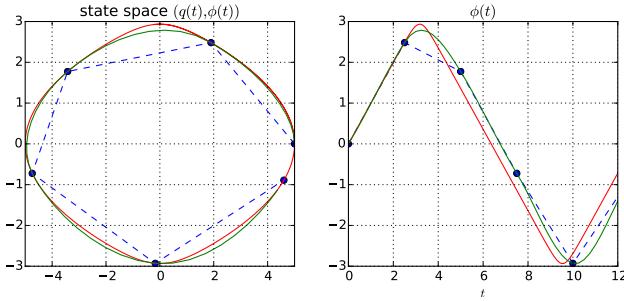


Figure 7: Comparison of simulated orbits with discrete points (in blue) computed using the AVF method, reconstructed cubic trajectory (in green) and reference trajectory computed at 10x sampling rate (in red).

This gives the Branch Component Equations.

Applying Kirchhoff Current and Voltage Laws gives the constraints $I_C = -I_L, V_C = V_L$. We can summarize the previous equations with the conservative autonomous Hamiltonian system.

$$\mathbf{x}' = \mathbf{J} \nabla \mathcal{H}(\mathbf{x}) \quad (45)$$

with.

$$\mathbf{x} = \begin{bmatrix} q \\ \phi \end{bmatrix}, \quad \mathbf{J} = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix}, \quad \nabla \mathcal{H} = \begin{bmatrix} \partial_q \mathcal{H} \\ \partial_\phi \mathcal{H} \end{bmatrix} \quad (46)$$

Its state space and temporal trajectories are shown in Figure 7. We can see that the numerical scheme preserves the energy since the discrete points lie exactly on the orbit of the reference trajectory. The reconstructed state-space trajectory also shows a good match with the reference for most of the interpolated segments, except around transition regions at the bottom and top.

The spectrum of the flux ϕ is shown in Figure 8. One can see that the reference spectrum contains harmonics above twice the representable bandwidth where they pass below -90 dB.

The ZOH and FOH spectrums contains spectral images of the non bandlimited spectrum that decay respectively at -6dB/oct and -12dB/oct. Their aliased components in the audio bandwidth start around -80 dB at the Nyquist frequency and decay slowly toward approximately -100 dB at low frequencies.

Contrary, our method, informed by the dynamic, exhibits both reduced aliasing in the audio bandwidth and sharpened spectrum

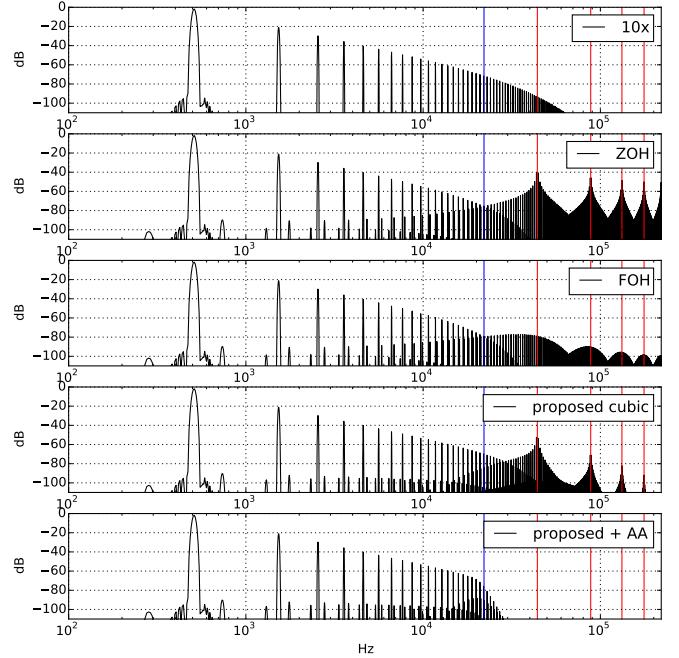


Figure 8: Continuous-time spectrum of the nonlinear LC circuit flux ϕ for a fundamental frequency of 500 Hz and a sampling frequency of 44.1 kHz. The 10x oversampled reference is compared to the AVF method's discrete output with zero-order hold (ZOH), first-order hold (FOH), the proposed method (proposed cubic) and its 12th order Butterworth filtered spectrum (proposed + AA). The Nyquist frequency is materialized in blue and the multiples of the sampling rate in red.

around the Nyquist frequency. It also has a higher spectral images decay rate thanks to its C^1 regularity. Its aliased components start at -85 dB at the Nyquist frequency and decay much faster to reach -100 dB at about 14 kHz where they reach a kind of aliasing noise floor caused by higher harmonics fold-back.

Finally, as expected, the 12th-order Butterworth half-band low-pass filter removes components above the Nyquist frequency thanks to the piecewise continuous cubic input.

8. DISCUSSION

First, we highlight the fact that the vector field approximation in (17) acts as a first-order antialiasing filter: it is a projection of the vector field on a rectangular kernel. It prevents high-order spectral images from disturbing the low frequency dynamic during the numerical simulation and it is consistent with the underlying piecewise linear approximation model.

Second, the numerical scheme is energy-preserving. From a signal processing perspective, the lowpass filtering effect on the vector field is compensated by the finite difference approximation of the derivative. This is a direct generalization of the mid-point / bilinear methods to nonlinear differential equations.

Third, using the fact that the trajectory approximation has accuracy order $p = 2$ at the junctions, we can re-exploit the differential equation to reconstruct an informed C^1 -continuous cubic trajectory. It exhibits reduced aliasing in the passband and better

high-frequency resolution.

We observe that on the studied example, our method manages to reduce aliased components that are folded once into the audio band. However components caused by multiple folding of the spectrum cannot be removed anymore. This is related to the Papoulis generalized sampling expansion [24] who states that a band-limited function can be perfectly reconstructed from its values and derivatives sampled at half the Nyquist rate.

Some difficulties arise when trying to generalize the above ideas to higher order trajectories and filtering kernels. First, the line-integral (17) is no longer computable in closed form when the trajectory model is non-affine. Second, higher order kernels have longer temporal support which can lead to non-causal integrals.

9. CONCLUSION AND PERSPECTIVES

Our main contribution is an approach based on smooth piecewise defined trajectories coupled with a guaranteed-passive simulation. The method proceeds in three steps: 1) an energy-preserving passive numerical scheme is applied, 2) C^k -continuous trajectories are reconstructed, 3) Exact continuous time lowpass filtering and sampling is performed. We have proposed a first instance of this method using the class of piecewise polynomials with regularity $k = 1$ and accuracy order $p = 2$ that exhibits reduced aliasing.

Further work will concern increasing the regularity k and accuracy order p , merging the numerical scheme and the interpolation steps by considering energy-preserving methods with a built-in regular continuous model and considering other classes of models such as rational and exponential functions.

In this regard, exponential integrators [25] that integrate the linear part of the dynamic exactly (as we have done in section 6) and rely on approximations for the nonlinear part are of great interest.

Finally we would like to further investigate the link between multi-stages / multi-derivatives general linear methods, their accuracy orders, numerical dispersion and internal bandwidth, and to analyze their behavior and representation capabilities within the framework of Reproducing Kernels Hilbert Spaces and generalized sampling theory [26] [27] [28].

10. ACKNOWLEDGMENTS

This work has been done at laboratory STMS, Paris, within the context of the French National Research Agency sponsored project INFIDHEM. Further information is available on the project web page.

11. REFERENCES

- [1] E. Hairer, C. Lubich, and G. Wanner, *Geometric Numerical Integration: Structure-Preserving Algorithms for Ordinary Differential Equations; 2nd ed.*, Springer, Dordrecht, 2006.
- [2] 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.
- [3] 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.
- [4] 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. –.
- [5] A. Falaize and T. Hélie, “Passive simulation of the nonlinear port-Hamiltonian modeling of a Rhodes Piano,” *Journal of Sound and Vibration*, vol. 390, pp. 289–309, Mar. 2017.
- [6] 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.
- [7] A. Falaize and T. Hélie, “Passive simulation of electro-dynamic loudspeakers for guitar amplifiers: a port- Hamiltonian approach,” in *International Symposium on Musical Acoustics*, Le Mans, France, July 2014, pp. 1–5.
- [8] A. Falaize and T. Hélie, “Passive guaranteed simulation of analog audio circuits: A port-hamiltonian approach,” *Applied Sciences*, vol. 6, no. 10, 2016.
- [9] J. P. Boyd, *Chebyshev and Fourier Spectral Methods*, Dover Books on Mathematics. Dover Publications, Mineola, NY, second edition, 2001.
- [10] T. S. Stilson, *Efficiently-variable Non-oversampled Algorithms in Virtual-analog Music Synthesis: A Root-locus Perspective*, Ph.D. thesis, 2006.
- [11] V. Zavalishin J. D. Parker and E. Le Bivic, “Reducing the aliasing of nonlinear waveshaping using continuous-time convolution,” in *Proc. Digital Audio Effects (DAFx-16)*.
- [12] S. Bilbao, F. Esqueda J. D. Parker, and V. Valimaki, “Antiderivative antialiasing for memoryless nonlinearities,” in *IEEE Signal Processing Letters*, Nov. 2016.
- [13] S. Sarkka and A. Huovilainen, “Accurate discretization of analog audio filters with application to parametric equalizer design,” *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 19, no. 8, pp. 2486–2493, Nov 2011.
- [14] M. Unser, “Think analog, act digital,” in *Seventh Biennial Conference, 2004 International Conference on Signal Processing and Communications (SPCOM’04)*, Bangalore, India, December 11-14, 2004.
- [15] M. Unser, A. Aldroubi, and M. Eden, “B-Spline signal processing: Part I—Theory,” *IEEE Transactions on Signal Processing*, vol. 41, no. 2, pp. 821–833, February 1993.
- [16] M. Unser, A. Aldroubi, and M. Eden, “B-Spline signal processing: Part II—Efficient design and applications,” *IEEE Transactions on Signal Processing*, vol. 41, no. 2, pp. 834–848, February 1993.
- [17] M. Unser, “Cardinal exponential splines: Part II—Think analog, act digital,” *IEEE Transactions on Signal Processing*, vol. 53, no. 4, pp. 1439–1449, April 2005.
- [18] A. Falaize, N. Lopes, T. Hélie, D. Matignon, and B. Maschke, “Energy-balanced models for acoustic and audio systems: a port-Hamiltonian approach,” in *Unfold Mechanics for Sounds and Music*, Paris, France, Sept. 2014.
- [19] E. L. Mansfield and G. R. W. Quispel, “On the construction of discrete gradients,” 2009.
- [20] E. Celledoni, V. Grimm, R.I. McLachlan, D.I. McLaren, D. O’Neale, B. Owren, and G.R.W. 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.
- [21] N. Lopes, T. Hélie, and A. Falaize, "Explicit second-order accurate method for the passive guaranteed simulation of port-Hamiltonian systems," in *5th IFAC Work- shop on Lagrangian and Hamiltonian Methods for Non Linear Control*, Lyon, France, July 2015, IFAC.
 - [22] E. Celledoni, R. I. McLachlan, D. I. McLaren, B. Owren, G. R. W. Quispel, and W. M. Wright, "Energy-preserving runge-kutta methods," *ESAIM: Mathematical Modelling and Numerical Analysis*.
 - [23] R. T. Farouki, "The bernstein polynomial basis: A centennial retrospective," *Comput. Aided Geom. Des.*, vol. 29, no. 6, pp. 379–419, Aug. 2012.
 - [24] A. Papoulis, "Generalized sampling expansion," *IEEE Transactions on Circuits and Systems*, vol. 24, no. 11, pp. 652–654, Nov 1977.
 - [25] M. Hochbruck and A. Ostermann, "Exponential integrators," *Acta Numerica*, vol. 19, pp. 209–286, 2010.
 - [26] D. Nehab and H. Hoppe, "A fresh look at generalized sampling," *Foundations and Trends in Computer Graphics and Vision*, vol. 8, no. 1, pp. 1–84, 2014.
 - [27] P.L. Dragotti, M. Vetterli, and T. Blu, "Sampling moments and reconstructing signals of finite rate of innovation: Shannon meets strang-fix," *IEEE Transactions on Signal Processing*, vol. 55, no. 5, pp. 1741–1757, May 2007.
 - [28] M. Unser, "Sampling-50 years after shannon," *Proceedings of the IEEE*, vol. 88, no. 4, pp. 569–587, April 2000.

12. APPENDIX: φ -FUNCTIONS

The φ -functions, that appear when doing exact integration of an LTI system with polynomial input given in monomial form, are defined by the convolution integral

$$\varphi_k(\lambda, t) = \int_0^t e^{\lambda(t-\tau)} \frac{\tau^{k-1}}{(k-1)!} d\tau \quad k \geq 1 \quad (47)$$

and by definition

$$\varphi_0(\lambda, t) := e^{\lambda t} \quad (48)$$

For $\lambda = 0$ it is immediate that

$$\varphi_k(\lambda = 0, t) = \frac{t^k}{k!} \quad (49)$$

12.1. Recurrence relation

We first prove that they satisfy the recurrence formula

$$\varphi_{k+1}(\lambda, t) = \frac{\varphi_k(\lambda, t) - \varphi_k(0, t)}{\lambda} \quad \lambda \neq 0 \quad (50)$$

Proof. Using integration by parts

$$\int_a^b u(\tau)v'(\tau)d\tau = [uv]_a^b - \int_a^b u'(\tau)v(\tau)d\tau$$

with $[a, b] = [0, t]$, $u(\tau) = e^{\lambda(t-\tau)}$, $v'(\tau) = \frac{\tau^{k-1}}{(k-1)!}$ and its primitive $v(\tau) = \frac{\tau^k}{k!}$ gives

$$\begin{aligned} \varphi_k(\lambda, t) &= \left[e^{\lambda(t-\tau)} \frac{\tau^k}{k!} \right]_0^t + \lambda \int_0^t e^{\lambda(t-\tau)} \frac{\tau^k}{k!} d\tau \\ &= \frac{t^k}{k!} + \lambda \varphi_{k+1}(\lambda, t) \end{aligned}$$

which after using (49) and identification gives

$$\varphi_{k+1}(\lambda, t) = \frac{\varphi_k(\lambda, t) - \varphi_k(0, t)}{\lambda}$$

□

12.2. Explicit form

Using (50) recursively for $\lambda \neq 0$, the first basis functions are given by

$$\varphi_0(\lambda, t) = e^{\lambda t} \quad (51)$$

$$\varphi_1(\lambda, t) = \frac{e^{\lambda t} - 1}{\lambda} \quad (52)$$

$$\varphi_2(\lambda, t) = \frac{e^{\lambda t} - (1 + \lambda t)}{\lambda^2} \quad (53)$$

$$\varphi_3(\lambda, t) = \frac{e^{\lambda t} - (1 + \lambda t + \frac{(\lambda t)^2}{2!})}{\lambda^3} \quad (54)$$

$$\varphi_4(\lambda, t) = \frac{e^{\lambda t} - (1 + \lambda t + \frac{(\lambda t)^2}{2!} + \frac{(\lambda t)^3}{3!})}{\lambda^4} \quad (55)$$

this suggests the following explicit form

$$\varphi_k(\lambda, t) = \frac{1}{\lambda^k} \left(e^{\lambda t} - \sum_{n=0}^{k-1} \frac{(\lambda t)^n}{n!} \right), \quad \lambda \neq 0 \quad (56)$$

Proof. It is immediate to verify that (56) is satisfied for $k = 0$. Then assuming that (56) is true for some $k \in \mathbb{N}$ and using the recurrence (50) we prove

$$\begin{aligned} \varphi_{k+1}(\lambda, t) &= \frac{\varphi_k(\lambda, t) - \varphi_k(0, t)}{\lambda} \\ &= \frac{1}{\lambda^{k+1}} \left(e^{\lambda t} - \sum_{n=0}^{k-1} \frac{(\lambda t)^n}{n!} \right) - \frac{1}{\lambda} \frac{t^k}{k!} \\ &= \frac{1}{\lambda^{k+1}} \left(e^{\lambda t} - \sum_{n=0}^k \frac{(\lambda t)^n}{n!} \right) \end{aligned}$$

that (56) is also true for $k + 1$. By induction (56) is thus satisfied for all $k \in \mathbb{N}$. □

The φ -functions represent thus the tail of the truncated taylor series expansion of $e^{\lambda t}$ up to a scaling factor. This is clear when rewriting (56) as

$$e^{\lambda t} = \sum_{n=0}^{k-1} \frac{(\lambda t)^n}{n!} + \lambda^k \varphi_k(\lambda, t) \quad (57)$$

THE QUEST FOR THE BEST GRAPHIC EQUALIZER

Juho Liski* and Vesa Välimäki

Aalto University
Acoustics Lab, Dept. of Signal Processing and Acoustics
Espoo, Finland
juho.liski@aalto.fi

ABSTRACT

The design of graphic equalizers has been investigated for decades, but only recently fitting the magnitude response closely enough to the control points has become possible. This paper reviews the development of graphic equalizer design and discusses how to define the target response. Furthermore, it investigates how to find the hardest target gain settings, the definition of the bandwidth of band filters, the estimation of the interaction between the bands, and how the number of iterations improves the design. The main focus is on a recent design principle for the cascade graphic equalizer. This paper extends the design method for the case of third-octave bands, showing how to choose the parameters to obtain good accuracy. The main advantages of the proposed approach are that it keeps the approximation error below 1 dB using only a single second-order IIR filter section per band, and that its design is fast. The remaining challenge is to simplify the design phase so that sufficient accuracy can be obtained without iterations.

1. INTRODUCTION

The design of graphic equalizers (EQ) is surprisingly difficult, and for this reason it has been investigated for decades [1]. The first application of graphic EQs was the enhancement of audio quality in movie systems in the 1950s [2, 1]. In the early years, graphic EQs were analog, but since the 1980s digital designs have been proposed [2, 3, 4, 5]. The graphic EQ has become one of the standard tools in music production [2, 6] and in audio systems [7, 8, 9, 10].

This paper reviews the development of graphic EQs, investigates the graphic EQ design problem, and discusses the recent efforts to improve and simplify the design of digital graphic EQs. Section 2 of this paper reviews the development of graphic equalizers. In Section 3, we first tackle the definition of target response and how to evaluate the accuracy of the design. Section 4 elaborates further on a recently proposed cascade octave graphic EQ design [11], trying to understand whether there are ways to improve it. Furthermore, in Section 5, we expand the proposed cascade design for the third-octave case, which is another popular and useful configuration. Section 6 concludes this paper and shows avenues for future research in this topic.

2. HISTORICAL DEVELOPMENTS

The graphic EQ design problem is simple: to fit the magnitude response of a digital filter through control points, which define the gain at several predefined frequency points. These are often called

the command gains. It was quickly understood that fitting the magnitude response of an EQ through the command points, which are usually spaced logarithmically in frequency is very difficult. Only recent digital design methods have achieved sufficient accuracy for hi-fi audio [12, 13, 1].

One straightforward method is to fit the response of a finite impulse response (FIR) filter to the command point data using interpolation and to apply the inverse discrete Fourier transform to obtain the FIR filter coefficients [14, 15]. However, the frequency division in graphic EQs is usually logarithmic, such as octaves, and the use of an FIR filter leads to complications at low frequencies: the impulse response associated with a sharp change in a low-frequency band becomes very long [15]. One idea to reduce this complication is using a multirate system to optimally downsample the long filters at low frequencies [16, 17]. Alternatively, frequency warping can be used to shorten the filter at low frequencies, but still, FIR graphic EQs are currently more costly to implement than infinite impulse response (IIR) graphic EQs [18].

During the analog era, most graphic EQs were based on the parallel structure, which refers to a bank of bandpass filters all of which receive the same input signal [19, 5, 20, 21, 12, 1]. The outputs of the bandpass filters were amplified according to the command gain of that band, and then combined (summed). Such structures suffered from complications due to the interference of the magnitude and phase responses of the neighboring bandpass filters. These often led to notches in the transition region from one band to another, or accumulation of gain in some bands due to leakage from neighboring bands. Early digital graphic EQs inherited the parallel structure from the analog world [5]. A few years ago, Rämö *et al.* showed how a graphic EQ can be designed accurately using a parallel filter, which is based on least-squares (LS) optimization of a bank of IIR filters having fixed poles [12].

As an alternative, the cascade structure has been considered for graphic EQs [19, 22, 23, 24, 1]. Traditional parametric EQ filters can be used as building blocks of a cascade graphic EQ [22, 25, 13]. It is well known in digital signal processing that the same transfer functions can be implemented using either a parallel or a cascade filter [26]. However, the design of the filter parameter values for these structures can be very different. The main reason for this is that considering the phase response of the band filters is unnecessary in the case of the cascade structure [1, 11]. However, in the case of the parallel structure, the phase response of each band filter is critical, as in the end the output signals of all band filters are combined.

Cascade graphic EQs suffer from similar interference from neighboring band filters as parallel EQs [1]. Some solutions to this problem include variable-Q designs, which change the bandwidth of the band filter according to the gain [27], and higher-order band filters, which improve the summation of the neighboring bands

* J. Liski is supported by the Aalto ELEC Doctoral School. The work was partially funded by Nokia Technologies (Aalto project number 410642).

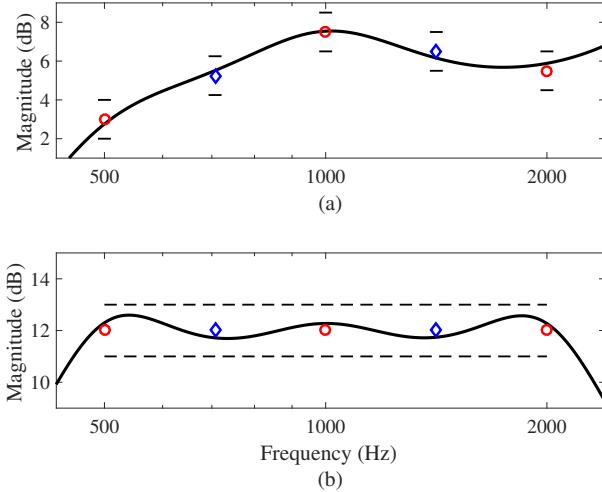


Figure 1: Definition of the ± 1 -dB error limits in the case of (a) different and (b) same neighboring command gains (red circles). In both figures, the blue diamonds indicate the extra points midway between each command point.

at the cost of increased computational complexity [28, 23, 24]. Our recent work showed that in the case of the octave cascade graphic EQ, a sufficiently accurate design can be obtained using a fairly simple design, which involves an unusual definition of filter bandwidth and the LS optimization with one iteration [11]. Additionally, the new cascade filter can be implemented with fewer second-order sections than the best parallel graphic EQ, which requires twice as many biquad filter sections as there are command points. It now seems clear that the best graphic EQ design must be based on the cascading of second-order band filters.

3. DEFINING THE BEST OCTAVE GRAPHIC EQ

There are multiple ways to compare different EQ designs. Commonly, the maximum error, the computational cost of the implementation, and the complexity of the design are used as criteria [12]. This section discusses ways to define the target response and how to test graphic EQ designs. Both aspects affect the evaluation of the maximum error.

3.1. Target Response

Obviously, the magnitude response of a graphic EQ should match the command gains at the control frequencies very closely. In hi-fi audio, a 1-dB accuracy requirement is typical at those points [23, 12, 1]. The error function used is usually the magnitude frequency response error in dB.

However, how to define the target response between the command points is less obvious. Smooth and monotonous transitions from one command point to another are usually desirable, since in practice a large increase or decrease in the filter gain between the command points is not what an audio engineer expects from a graphic EQ. Various interpolation methods have been suggested for producing a virtually continuous target magnitude response from command gain data [12, 29]. Some design methods require such a high-resolution target response.

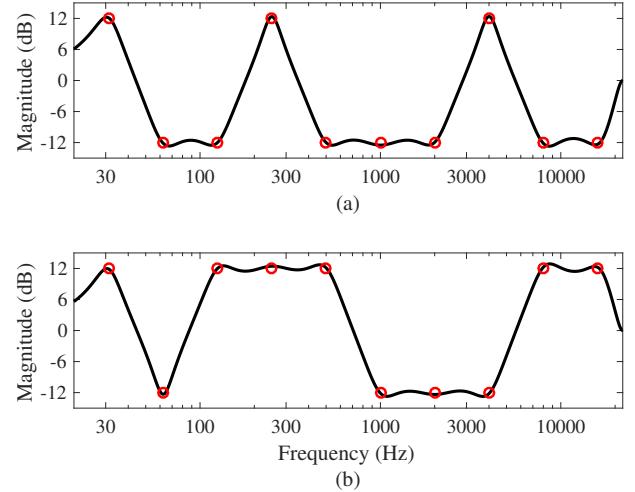


Figure 2: Worst-case command gain (red circles) settings (a) reported by Oliver and Jot for their design [13] and (b) observed in this work for the new cascade design [11]. The black curve is the magnitude response obtained with the proposed design.

We have observed that for a cascade octave graphic EQ having low-order band filters, such as second-order IIR filters, interpolating a high-resolution target response is unnecessary, since the band filter's gain cannot make large deviations between command points. To guarantee a monotonous transition between command points, one option is to simply define one intermediate point between each command point, where the gain error is evaluated, as suggested in [11]. The gain at the intermediate point is determined as the average of the adjacent command gains in dB. The frequency of the intermediate point must be the geometric mean of the neighboring command points. Figure 1 visualizes this error definition. Figure 1(a) shows the ± 1 -dB error limits at three command points and at two intermediate points between them.

Figure 1(b) presents a common special case in which two or more neighboring command gains are equal. In this case, it is meaningful to require the magnitude response to stay within 1 dB from the command gain also at all frequencies between these points.

3.2. Hardest Gain Settings

In a graphic EQ, the gains can usually be adjusted in the range of ± 12 dB [1]. When trying to find the hardest command gain combinations, testing the settings where the gains are set either to $+12$ or -12 dB in order to produce the largest deviations between the bands is natural. Since there are ten bands in an octave EQ and we have two alternatives, we obtain $2^{10} = 1024$ different combinations, or binary settings. These combinations can then easily be tested to find the one producing the largest error.

Figure 2 shows two examples where the gains are selected from the binary cases. Oliver and Jot found that for their proposed design the worst-case gain setting was the one shown in Fig. 2(a) [13]. On the other hand, our previous work found the settings seen in Fig. 2(b) to cause the largest error. Clearly, these two examples have similarities: there are steep transitions between ± 12 dB, but also plateaus, where the EQ has to produce the same gain in multiple adjacent bands. In both cases, the largest error was actually produced at such a plateau, at approximately 1 kHz [13] and

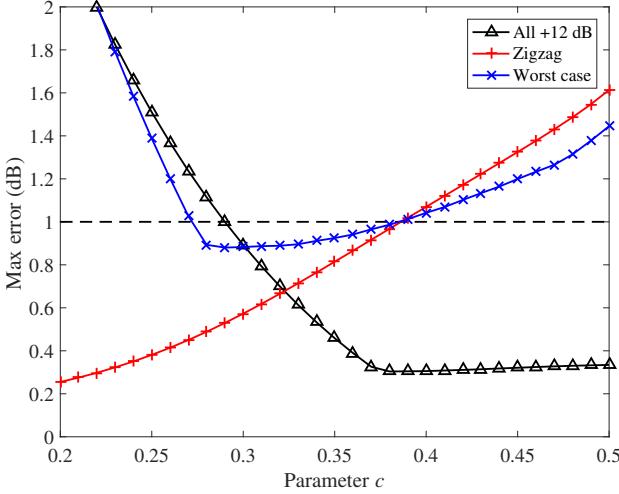


Figure 3: Effect of the value of c on the maximum error in three different command gain settings. The worst case is the one shown in Fig. 2(b). The dashed line indicates the 1-dB error that should not be exceeded.

at 9 kHz [11].

Based on the two cases reviewed above and our own comprehensive testing, we propose that the 1024 hard binary cases should be utilized to test graphic octave EQs in order to reveal the largest approximation error. Note, however, that for the third-octave EQ, there are about 30 command gains, so exhaustive testing of all combinations may not be viable.

4. OCTAVE GRAPHIC EQ DESIGN

Here, a summary of the new cascade graphic EQ design proposed in [11] is presented. The method is based on designs proposed by Abel and Berners [30] and Oliver and Jot [13]: one filter per octave band is used whose interaction with its two neighboring filters at their center frequency is exactly controlled.

The method uses as band filters the following second-order IIR peak/notch filter given by Orfanidis in which the reference gain at dc is set to 1 [31]:

$$H(z) = \frac{1 + G\beta - 2 \cos(\omega_c)z^{-1} + (1 - G\beta)z^{-2}}{1 + \beta - 2 \cos(\omega_c)z^{-1} + (1 - \beta)z^{-2}}, \quad (1)$$

where G is the linear peak gain, $\omega_c = 2\pi f_c/f_s$ is the normalized center frequency in radians (the ten standard octave frequencies 31.25 Hz, 62.5 Hz, 125 Hz etc. are used), f_s is the sampling frequency, β is defined as

$$\beta = \begin{cases} \tan(B/2), & \text{when } G = 1, \\ \sqrt{\frac{|G_B^2 - 1|}{|G^2 - G_B^2|}} \tan\left(\frac{B}{2}\right) & \text{otherwise,} \end{cases} \quad (2)$$

and G_B is the linear gain at bandwidth $B = 2\pi f_B/f_s$.

In the IIR section defined by (1) and (2), the bandwidth can be selected such that for the m th band filter, a specified dB gain $g_{B,m} = cg_m$ is reached at the neighboring center frequencies. We

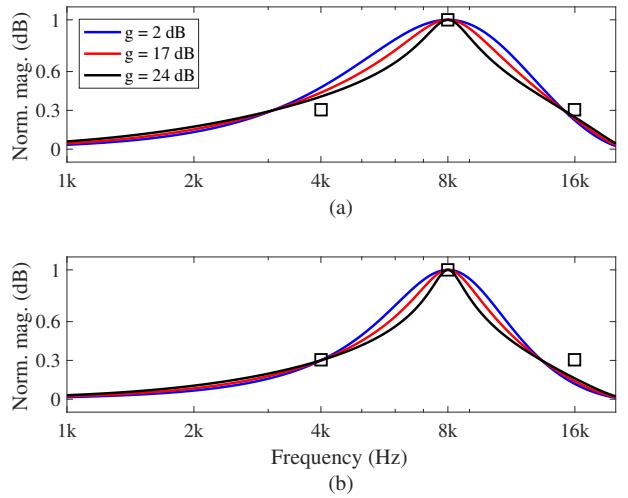


Figure 4: Normalized amplitude responses of the 8-kHz band filter for three different peak gain values: (a) the original bandwidth, which leads to too wide a response (the responses do not cross inside the boxes) and (b) the adjusted bandwidth, which makes the responses cross at the lower neighboring center frequency. The squares indicate the points where the responses should meet.

found that the choice $c = 0.3$ leads to a successful octave graphic EQ design [11]. This is illustrated in Fig. 3, which shows the effect of the parameter c with three different command gain settings. As is seen, the desired accuracy is achieved when c has values between 0.28 and 0.38, and thus $c = 0.3$ can be used. This value of the parameter c leads to an unusual definition of the bandwidth, since traditionally the bandwidth of a resonance is determined as the difference of -3-dB points on each side of a peak, which refers to 0.7 times the linear gain. However, this non-traditional choice appears to be crucial for accurate automatic design.

The bandwidths is selected as $B_m = 1.5\omega_{c,m}$, which equals the difference between the neighboring upper and lower center frequencies. This way, the behavior of each band filter can be exactly controlled at the center frequencies of both its neighbors. However, at high center frequencies, the bandwidth needs to be adjusted in another way, because the filter response becomes asymmetric.

Figure 4(a) shows examples of the normalized magnitude response of the band filter at 8 kHz for three different gains. The magnitude responses have been normalized on the dB scale by dividing them by their respective dB gain, as suggested in [30]. Without the bandwidth adjustment, the filter responses do not cross at the desired points, i.e., at their center frequency and the two neighboring center frequencies, as seen in Fig. 4(a). This anomaly leads to difficulties in predicting the interaction between neighboring band filters. In the octave design, the asymmetry concerns the three band filters with the highest center frequencies, 4 kHz, 8 kHz, and 16 kHz. Their bandwidth is set so that the 0.3 g_m point occurs at the lower neighboring center frequency (but not at the higher one), as shown in Fig. 4(b). This leads to bandwidth values $f_{B,8} = 5580$ Hz, $f_{B,9} = 9360$ Hz, and $f_{B,10} = 12160$ Hz instead of 6000 Hz, 12000 Hz, and 24000 Hz, respectively.

The resulting filter-response shapes for all band filters of the cascade octave graphic EQ are shown in Fig. 5, where the responses are seen to meet at all the desired frequency points (the

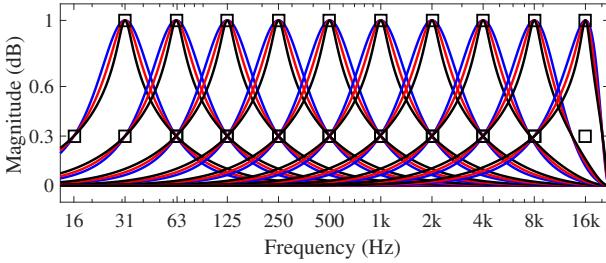


Figure 5: Normalized amplitude responses of all the band filters, which have gains of 0.3 times its peak gain in dB at the neighboring center frequencies, with different peak gain values: 2 dB (blue), 17 dB (red), and 24 dB (black).

squares) except for the three highest filters whose responses do not cross at their higher neighboring frequency. Additionally, Fig. 5 demonstrates the self-similarity of the band filters. Three responses are plotted at each band with different peak gains. Due to the similar shape of each normalized response, the samples taken from the dB amplitude responses can be used as a basis function in order to control the interaction of the band filters at the selected frequencies [30, 13, 11]. The normalized dB amplitude responses of the filters are stored in an interaction matrix \mathbf{B} .

The octave design uses 19 design points [11] instead of ten, as suggested by Oliver and Jot [13]. These 19 points include the filter center frequencies and the geometric mean of these values between them. This decreases the error and thus improves the behavior of the EQ between the command points. With ten octave bands and 19 design frequencies we obtain a 19-by-10 interaction matrix, which is visualized in Fig. 6. The filters for the interaction matrix are designed with (1) and (2) using a prototype dB-gain g_p , which in this case is 17 dB. The values inserted into the interaction matrix represent the magnitude response divided by g_p at the command points and the intermediate points. As is seen in Fig. 6, the diagonal values of the interaction matrix are 1 dB, representing the filter center frequencies, and the other stems indicate the relative effect of each band filter at the other design points.

The interaction matrix is utilized to determine the optimal dB gain for each filter in the LS sense by solving its inverse matrix [32]. However, since the method uses a non-square matrix, the pseudoinverse of the interaction matrix \mathbf{B}^+ is necessary to obtain the optimal solution [32]. The filter gains \mathbf{g} are then obtained as

$$\mathbf{g} = \mathbf{B}^+ \mathbf{t}_1 = (\mathbf{B}^T \mathbf{B})^{-1} \mathbf{B}^T \mathbf{t}_1, \quad (3)$$

where \mathbf{t}_1 is a vector with 19 elements containing the original target dB-gain values in odd rows and their linearly interpolated intermediate values in even rows.

Finally, to achieve the desired accuracy of 1 dB, one iteration step is required, because the shape of the basis functions vary slightly with the filter gains [11]. A new interaction matrix \mathbf{B}_1 is formed with the gains \mathbf{g} obtained from (3) instead of the prototype gain g_p , and new filter gains are calculated similarly to (3). The effect of iteration steps is shown in Fig. 7. Figure 7(a) shows the largest filter gain change in dB as a function of the number of iteration steps. The hardest command gain setting shown in Fig. 2(b) has been used as the target. The first iteration step is seen to cause gain changes up to approximately 0.4 dB when compared to the non-iterative version, whereas the second step has an effect of approximately 0.05 dB or less. After the third iteration step the gains

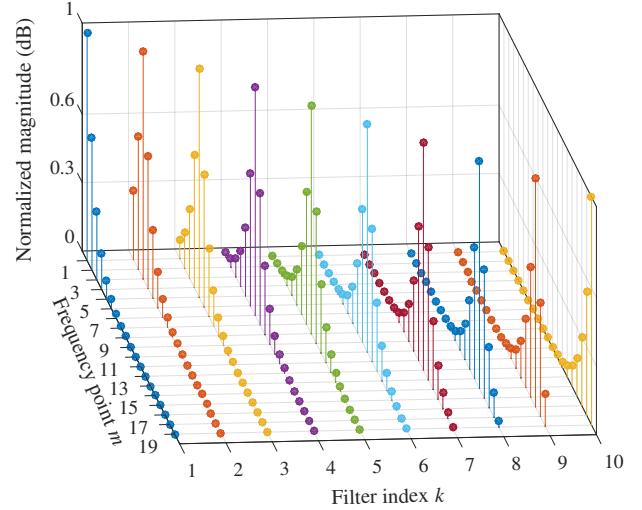


Figure 6: A 19-by-10 interaction matrix containing the normalized leakage of each band filter to the other frequency points.

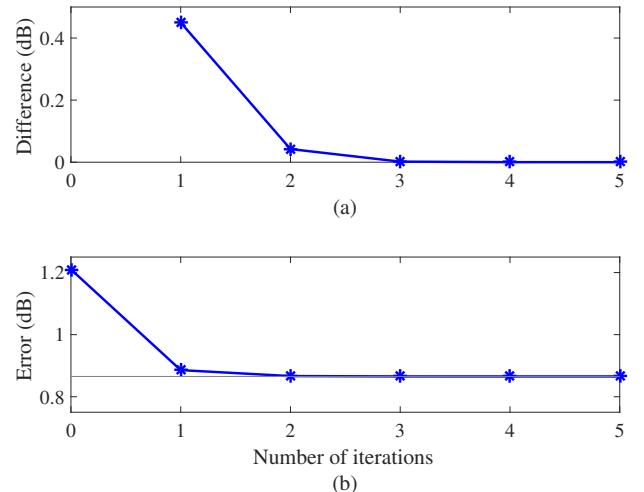


Figure 7: (a) Maximum difference in filter gains between the iteration rounds, and (b) maximum error after each iteration for the hardest command gain setting of the octave graphic EQ, cf. 2(b).

practically do not change at all.

On the other hand, Fig. 7(b) shows the maximum approximation error in each case. As is seen, the acceptable error of less than 1 dB is achieved with one iteration, and the second iteration decreases the error very slightly. After that, the error saturates at 0.87 dB and further iteration steps have practically no effect. Similar error behavior is observed with other tested target responses. Thus, it is safe to assume that iterating the interaction matrix once suffices, and that further iterations are superfluous.

5. THIRD-OCTAVE DESIGN

In this section we devise a third-octave cascade graphic EQ design based on the octave version. One second-order peak/notch filter of the form (1) is used for each band. Also, an interaction matrix

Table 1: Center frequencies and bandwidths for the 31 filters of the third-octave graphic equalizer. The adjusted bandwidths of the six highest band filters are shown in italics.

	1	2	3	4	5	6	7	8	9	10	11
f_c (Hz)	19.69	24.80	31.25	39.37	49.61	62.50	78.75	99.21	125.0	157.5	198.4
f_B (Hz)	9.178	11.56	14.57	18.36	23.13	29.14	36.71	46.25	58.28	73.43	92.51
	12	13	14	15	16	17	18	19	20	21	22
f_c (Hz)	250.0	315.0	396.9	500.0	630.0	793.7	1000	1260	1587	2000	2520
f_B (Hz)	116.6	146.9	185.0	233.1	293.7	370.0	466.2	587.4	740.1	932.4	1175
	23	24	25	26	27	28	29	30	31		
f_c (Hz)	3175	4000	5040	6350	8000	10080	12700	16000	20160		
f_B (Hz)	1480	1865	2350	2846	3502	4253	5038	5689	5573		

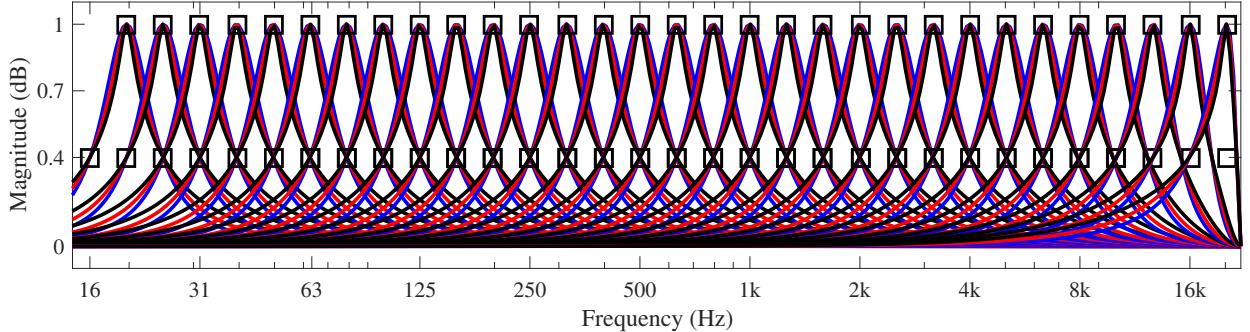


Figure 8: Normalized amplitude responses of all third-octave band filters with three different peak gains, as in Fig. 5. The filter gain at the neighboring center frequencies, indicated with squares, is set to be 0.4 times the peak dB-gain.

with extra frequencies is used with one iteration step to optimize the filter gains. Differences between the two designs are caused by the unequal number of bands and different bandwidths, and therefore some of the parameters must be reselected.

The third-octave design has 31 bands, whose center frequencies are given in Table 1. We use the center frequencies as well as the geometric mean frequencies between them as design frequencies, which leads to 61 design points. Thus, the size of the interaction matrix is now 61-by-31, and since it is non-square, its pseudoinverse is used in the LS design. The initial interaction matrix is designed using the same prototype gain value as the octave design, $g_p = 17$ dB.

The bandwidths for the third-octave design are defined in the same way as in the octave design by selecting a specified dB value that is achieved at the neighboring center frequencies. When using the same principle as in the octave design for calculating the bandwidths (i.e., the difference of center frequencies above and below a filter), we obtain $B_m = (\sqrt[3]{2} - 1/\sqrt[3]{2})\omega_{c,m} \approx 0.4662\omega_{c,m}$. Due to the filter asymmetry, the bandwidths of the six uppermost filters are tuned by hand, resulting in the values presented in Table 1. The effect of the manual adjustments are also seen in Fig. 8, where the left sides of the six last filters now cross the boxes at the lower neighboring center frequencies.

The largest difference between the octave and the third-octave design is found in the selection of the parameter c . Initially the value $c = 0.3$ was tested, but this resulted in too narrow filters and large approximation errors. The EQ is unable to create a flat response when the filters are too narrow, since the overall response

droops between the command points. This is shown in Fig. 9(a): when all command gains are set to +12 dB, the error exceeds 1 dB at high frequencies. Figure 9(c) shows that the same can happen in the middle frequencies with certain command gain settings.

To improve the behavior of the third-octave EQ, different values of c were tested. Additionally, a minor modification needed to be applied to the error criterion with respect to the octave design. For the third-octave design, the error is not evaluated at the intermediate points between the center frequencies, although those points are accounted for in the design. Even though the magnitude response of the EQ varies smoothly from one command point to another the approximation error exceeds 1 dB in the narrow and steep transition bands. As the transition bands are very narrow, such minor undulations are not expected to be perceivable.

Figure 10 shows the effect of parameter c on the maximum error in three different command gain settings. A suitable c parameter range for the third-octave design, where the maximum error of all three test cases remains below 1 dB, is observed to be 0.38–0.4. In this work, $c = 0.4$ is used. The improvements achieved by adjusting c to 0.4 are shown in Figs. 9(b) and in 9(d), where the approximation errors is now smaller than 1 dB and the desired accuracy is thus achieved.

Finally, a non-extreme third-octave EQ design example, taken from [12], is presented in Fig. 11(a). This command gain setting leads to a varied target curve, which the proposed design matches well, confirmed by the error curve shown in Fig. 11(b). In the plateaus, the error has been evaluated at 16 frequency points between each neighboring command points.

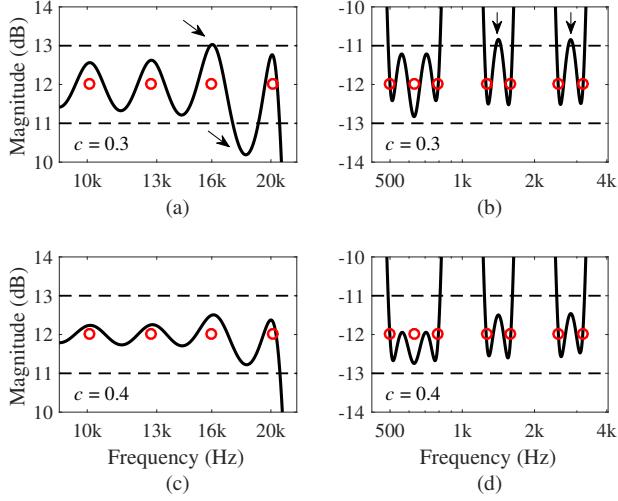


Figure 9: Effect of the value of c on the magnitude response. For (a) and (c), all the command gains are at 12 dB, and for (b) and (d), the command gains are at ± 12 dB: the target seen in Fig. 2(a) repeats itself until the 31 target points are filled. The horizontal dashed lines indicate the ± 1 -dB error tolerances and the arrows show the points where the error exceeds the 1-dB limit. The error is reduced, when the value of c is changed from 0.3 to 0.4.

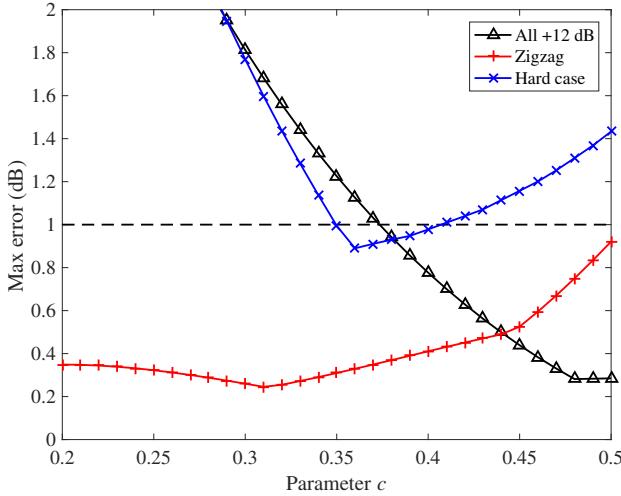


Figure 10: Effect of the parameter c on the maximum error using three different command gain settings. The hard setting is shown in Fig. 12(a).

5.1. Comparison With a Previous Accurate Method

In this section, the proposed third-octave design is compared with another graphic EQ, which, to our knowledge, is currently the most accurate, i.e., the high-precision parallel graphic EQ (PGE) [12]. It comprises twice as many second-order filter sections as there are bands and has a maximum approximation error of less than 1 dB at all tested command gain configurations. However, it is difficult to design, since a high-resolution interpolation of the target magnitude response is required as well also a phase response estimation [29].

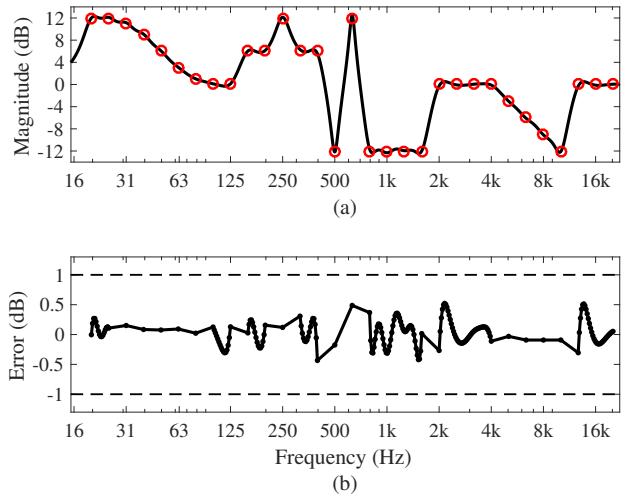


Figure 11: Third-octave EQ example showing varying command gains: (a) the complete response of the EQ and (b) the error, as defined for the third-octave design in Sec. 5. The dots illustrate the frequency points where the error has been evaluated.

Table 2 compares the accuracy of the two EQs with different command gain settings. We are interested in the maximum error that is determined by the guidelines shown in Fig. 1 apart from using the intermediate frequency points as explained in the previous section. The first test case is a zigzag setting in which the command gains alternate between ± 12 dB that reveals the EQ’s ability to create steep transitions. As is seen in Fig. 12(a) and (b), the proposed design and the PGE produce very similar responses everywhere except at very low frequencies below the first command point, and they both stay within ± 1 dB of the targets. However, when looking at the maximum error, we see that the proposed method is slightly better with an approximately 0.2-dB smaller error.

The responses of the second test case are shown in Figs. 12(c) and (d). Here too the command gains vary between ± 12 dB, but there are also flat regions between the steep transitions. The gain setting is inspired by the ones seen in Fig. 2. However, in the case of third-octave filters we could not test all the hard binary settings as in the octave version, because of the huge number of combinations (2^{31} compared to 2^{10}). Instead, some combinations that we thought were hard were tested, and we ended up using the following demanding command gain configuration: $t = [12 -12 12 12 -12 12 -12 -12 -12 12 -12 12 12 12 -12 12 -12 12 12 12 -12 12 12 12 -12 -12 12 12 12 12 12]^\top$. The two methods again produce similar responses, as seen by comparing Figs. 12(c) and (d). When investigating the maximum error, we see that the PGE is slightly better, but that both methods stay within ± 1 dB of the target.

Table 3 lists the number of operations during real-time filter-

Table 2: Maximum error of two third-octave graphic EQs in two test target settings. The best result in each case is highlighted.

Case	PGE	Proposed
Zigzag	0.65 dB	0.41 dB
Hard case	0.66 dB	0.98 dB

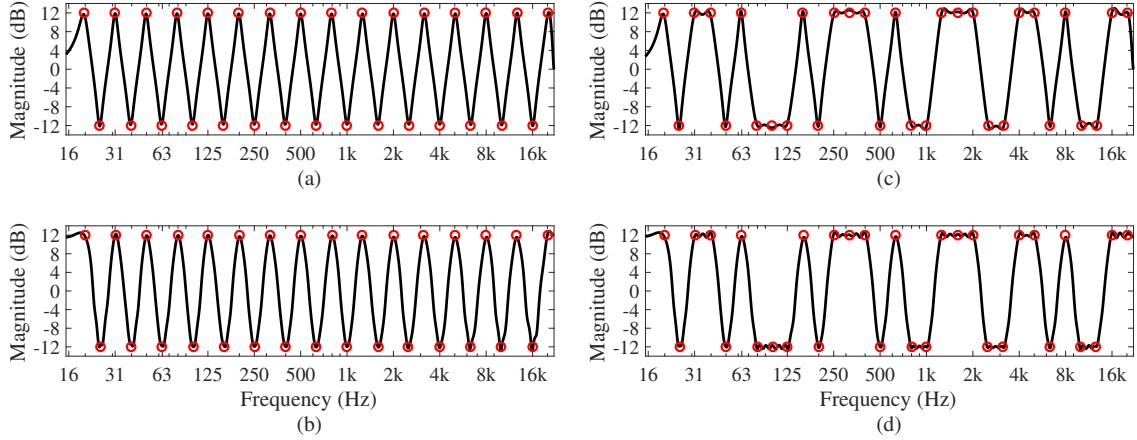


Figure 12: *Magnitude response of (a) the proposed design and (b) the PGE with zigzag (± 12 dB) command settings and (c) the proposed design and (d) the PGE with hard command gain settings (± 12 dB) inspired by the targets in Fig. 2.*

ing for the PGE and the proposed methods. As is seen, the proposed method is approximately 44% more efficient. This advantage mainly comes from using one biquad filter per band rather than the two per band in the PGE. Even though the PGE uses an optimized structure by having one fewer numerator coefficient compared to the traditional second-order filter [12], the larger number of filter sections negates that advantage.

Additionally, we compared another computational aspect of the graphic EQ design, namely the parameter computing time when a command gain is changed. The update times were calculated in MATLAB as an average of 1000 updates using random values between ± 12 dB as command gains. The Internet connection and all other programs were shut down so as not to affect the computation.

The average time for the command gain update was 24 ms for the PGE method and 6.0 ms for the proposed method. This implies that the proposed method is 75% faster than the PGE in updating its parameters. The proposed method requires linear interpolation between the command gains, a matrix inversion, and large matrix multiplications, which increase its update time. However, due to the computation of the high-resolution target magnitude and phase responses and a pseudoinverse of a large matrix, the PGE requires much more time to update its parameters.

In summary, the proposed method is approximately equally accurate as the PGE, but requires fewer operations per output sample and is faster in command gain updating, making it the superior design.

6. CONCLUSION AND FUTURE WORK

This paper reviewed the methods and the target response definition for graphic EQ design, proposed a methodology for testing graphic

Table 3: *Comparison of the operation count in third-octave EQs.*

Operations	PGE	Proposed
Additions	248	124
Multiplications	248	155
Total	496	279

EQs, and expanded a previously proposed accurate graphic EQ to the third-octave equalization problem. In the case of a cascade octave graphic EQ with low-order band filters, one alternative is to evaluate the error at intermediate points between the command points in addition to the command points themselves. This is enough to guarantee a monotonous transition between the bands, since large deviations between bands are impossible using second-order IIR filters having a restricted bandwidth. In addition, some hard command gain settings were presented that can be used to test graphic EQ designs. The largest errors were observed when all command gains were at the extreme values, usually at ± 12 dB.

Finally, a previously proposed accurate graphic EQ design was expanded to the third-octave case. The design method uses one second-order IIR filter per band. The interaction between the different band filters is optimized at the band center frequencies and at one extra point between each center frequency with the help of an interaction matrix. With one iteration step in the interaction matrix design, the method achieves 1-dB accuracy and thus is applicable to high-quality audio.

The new third-octave design was compared with a previously proposed parallel graphic EQ, which, to our knowledge, was the state-of-the-art graphic EQ prior to this work. The new method achieves approximately the same accuracy but requires fewer operations per output sample and is faster to design. The proposed method is thus currently the best graphic EQ design. The relevant MATLAB code is available online [33].

A remaining research challenge is to simplify the EQ design so that sufficient accuracy is achieved without an iteration step. Possible approaches to this end are the determination of data- or frequency-dependent c and g_p parameters. This would lead to computational benefits, since the interaction matrix and its pseudoinverse would not have to be calculated with each command gain change. Furthermore, the cascade EQ could be converted into a parallel form in order to reap benefits in the filter implementation.

7. ACKNOWLEDGMENT

The authors would like to thank Luis Costa for proofreading this paper.

8. REFERENCES

- [1] V. Välimäki and J. D. Reiss, “All about audio equalization: Solutions and frontiers,” *Appl. Sci.*, vol. 6, no. 129/5, pp. 1–46, May 2016.
- [2] D. A. Bohn, “Operator adjustable equalizers: An overview,” in *Proc. Audio Eng. Soc. 6th Int. Conf. Sound Reinforcement*, Nashville, TN, USA, May 1988, pp. 369–381.
- [3] Y. Hirata, “Digitalization of conventional analog filters for recording use,” *J. Audio Eng. Soc.*, vol. 29, no. 5, pp. 333–337, May 1981.
- [4] S. Takahashi, H. Kameda, Y. Tanaka, H. Miyazaki, T. Chikashige, and M. Furukawa, “Graphic equalizer with microprocessor,” *J. Audio Eng. Soc.*, vol. 31, no. 1/2, pp. 25–28, Jan.-Feb. 1983.
- [5] Motorola Inc., “Digital stereo 10-band graphic equalizer using the DSP56001,” 1988, Application note.
- [6] S. Stasis, R. Stables, and J. Hockman, “Semantically controlled adaptive equalisation in reduced dimensionality parameter space,” *Appl. Sci.*, vol. 6, no. 116/4, pp. 1–19, Apr. 2016.
- [7] J. Rämö, V. Välimäki, and M. Tikander, “Perceptual headphone equalization for mitigation of ambient noise,” in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process. (ICASSP)*, Vancouver, BC, Canada, May 2013, pp. 724–728.
- [8] F. Vidal Wagner and V. Välimäki, “Automatic calibration and equalization of a line array system,” in *Proc. Int. Conf. Digital Audio Effects (DAFx-15)*, Trondheim, Norway, Nov. 2015, pp. 123–130.
- [9] D. Griesinger, “Accurate timbre and frontal localization without head tracking through individual eardrum equalization of headphones,” in *Proc. Audio Eng. Soc. 141st Conv.*, Los Angeles, CA, USA, Sept. 2016.
- [10] J. Liski, V. Välimäki, S. Vesa, and R. Väänänen, “Real-time adaptive equalization for headphone listening,” in *Proc. 25th European Signal Process. Conf. (EUSIPCO-17)*, Kos, Greece, Aug. 2017.
- [11] V. Välimäki and J. Liski, “Accurate cascade graphic equalizer,” *IEEE Signal Process. Lett.*, vol. 24, no. 2, pp. 176–180, Feb. 2017.
- [12] J. Rämö, V. Välimäki, and B. Bank, “High-precision parallel graphic equalizer,” *IEEE/ACM Trans. Audio Speech Lang. Process.*, vol. 22, no. 12, pp. 1894–1904, Dec. 2014.
- [13] R. J. Oliver and J.-M. Jot, “Efficient multi-band digital audio graphic equalizer with accurate frequency response control,” in *Proc. Audio Eng. Soc. 139th Conv.*, New York, NY, USA, Oct. 2015.
- [14] A. T. Johnson, Jr., “Magnitude equalization using digital filters,” *IEEE Trans. Circ. Theory*, vol. 20, no. 3, pp. 308–311, May 1973.
- [15] M. Waters, M. Sandler, and A. C. Davies, “Low-order FIR filters for audio equalization,” in *Proc. Audio Eng. Soc. 91st Conv.*, New York, NY, USA, Oct. 1991.
- [16] J. A. Henriquez, T. E. Riemer, and R. E. Trahan, Jr., “A phase-linear audio equalizer: Design and implementation,” *J. Audio Eng. Soc.*, vol. 38, no. 9, pp. 653–666, Sept. 1990.
- [17] R. Väänänen and J. Hiipakka, “Efficient audio equalization using multirate processing,” *J. Audio Eng. Soc.*, vol. 56, no. 4, pp. 255–266, Apr. 2008.
- [18] J. Siiskonen, “Graphic equalization using frequency-warped digital filters,” M.Sc. thesis, Aalto University, Espoo, Finland, Aug. 2016.
- [19] R. A. Greiner and M. Schoessow, “Design aspects of graphic equalizers,” *J. Audio Eng. Soc.*, vol. 31, no. 6, pp. 394–407, June 1983.
- [20] S. Tassart, “Graphical equalization using interpolated filter banks,” *J. Audio Eng. Soc.*, vol. 61, no. 5, pp. 263–279, May 2013.
- [21] Z. Chen, G. S. Geng, F. L. Yin, and J. Hao, “A pre-distortion based design method for digital audio graphic equalizer,” *Digital Signal Process.*, vol. 25, pp. 296–302, Feb. 2014.
- [22] P. A. Regalia and S. K. Mitra, “Tunable digital frequency response equalization filters,” *IEEE Trans. Acoust. Speech Signal Process.*, vol. 35, no. 1, pp. 118–120, Jan. 1987.
- [23] M. Holters and U. Zölzer, “Graphic equalizer design using higher-order recursive filters,” in *Proc. Int. Conf. Digital Audio Effects*, Montreal, Canada, Sept. 2006, pp. 37–40.
- [24] J. Rämö and V. Välimäki, “Optimizing a high-order graphic equalizer for audio processing,” *IEEE Signal Process. Lett.*, vol. 21, no. 3, pp. 301–305, Mar. 2014.
- [25] R. Bristow-Johnson, “The equivalence of various methods of computing biquad coefficients for audio parametric equalizers,” in *Proc. Audio Eng. Soc. 97th Conv.*, San Francisco, CA, USA, Nov. 1994.
- [26] W. Chen, “Performance of cascade and parallel IIR filters,” *J. Audio Eng. Soc.*, vol. 44, no. 3, pp. 148–158, Mar. 1996.
- [27] R. Miller, “Equalization methods with true response using discrete filters,” in *Proc. Audio Eng. Soc. 116th Conv.*, Berlin, Germany, May 2004.
- [28] D. McGrath, J. Baird, and B. Jackson, “Raised cosine equalization utilizing log scale filter synthesis,” in *Proc. Audio Eng. Soc. 117th Conv.*, San Francisco, CA, USA, Oct. 2004.
- [29] J. A. Belloch and V. Välimäki, “Efficient target response interpolation for a graphic equalizer,” in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process. (ICASSP)*, Shanghai, China, Mar. 2016, pp. 564–568.
- [30] J. S. Abel and D. P. Berners, “Filter design using second-order peaking and shelving sections,” in *Proc. Int. Computer Music Conf.*, Miami, FL, USA, Nov. 2004.
- [31] S. J. Orfanidis, *Introduction to Signal Processing*, Rutgers Univ., Piscataway, NJ, USA, 2010.
- [32] L. B. Jackson, *Digital Filters and Signal Processing*, Kluwer, Norwell, MA, USA, 2nd edition, 1989.
- [33] J. Liski and V. Välimäki, “Companion page to The Quest for the Best Graphic Equalizer [Online],” <http://research.spa.aalto.fi/publications/papers/dafx17-geq/>, 2017, [accessed Jun. 21, 2017].

AUDIO PROCESSING CHAIN RECOMMENDATION

Spyridon Stasis

Digital Media Technology Lab,
Birmingham City University
Birmingham, UK
spyridon.stasis@bcu.ac.uk

Nicholas Jillings

Digital Media Technology Lab,
Birmingham City University
Birmingham, UK
nicholas.jillings@bcu.ac.uk

Sean Enderby

Digital Media Technology Lab,
Birmingham City University
Birmingham, UK
sean.enderby@mail.bcu.ac.uk

Ryan Stables

Digital Media Technology Lab,
Birmingham City University
Birmingham, UK
ryan.stables@bcu.ac.uk

ABSTRACT

In sound production, engineers cascade processing modules at various points in a mix to apply audio effects to channels and busses. Previous studies have investigated the automation of parameter settings based on external semantic cues. In this study, we provide an analysis of the ways in which participants apply full processing chains to musical audio. We identify trends in audio effect usage as a function of instrument type and descriptive terms, and show that processing chain usage acts as an effective way of organising timbral adjectives in low-dimensional space. Finally, we present a model for full processing chain recommendation using a Markov Chain and show that the system's outputs are highly correlated with a dataset of user-generated processing chains.

1. INTRODUCTION

Mixing audio involves a range of complex processes that include balancing source amplitudes, applying audio effects, and positioning a sound source in a perceived space. These tasks can be time consuming and are often carried out for either corrective or creative reasons. Corrective tasks are straightforward but time consuming operations, including noise removal, temporal alignment, and level correction using equalisation and dynamics processing. Creative tasks on the other hand require artistic interpretation, and can involve perceptually mapping ideas and descriptions of audio to processing parameters.

During production, audio effect modules are often cascaded to create processing chains for each channel and bus in the mix, including the master. This allows combinations of linear and nonlinear systems to be able to apply processing to the audio signal at various points in the workflow, typically using a Digital Audio workstation (DAW). For corrective purposes, the effects included in each of the processing chains are selected by the engineer as a reaction to audible cues, such as artefacts in the mix. For creative purposes, they can be selected with a view to make a source more or less prominent in the mix, or to achieve a given aesthetic.

1.1. Intelligent Music Production

Intelligent Music Production aims to provide interfaces and algorithms to automate and facilitate the music production process [1].

This should effectively reduce the time spent by producers and engineers on time consuming, menial tasks, and allow them to focus on the creative aspects of music production. Previously, these systems have been developed for automatic mixing, whereby aspects of the production workflow such as balance [2, 3] and panning [4], can be optimised according to psychoacoustic principles.

Studies in the area of Intelligent Music Production have explored methods for the automation of parameter settings based on external semantic cues, such as descriptive language [5]. This has been applied to a range of audio effects such as equalisation [6, 7, 8], distortion [9], compression [10] and reverberation [11]. In each of these cases, parameter automation is applied within the context of a single effect, meaning the user of the system needs to be aware of the type of processing required to achieve the desired aesthetic.

The use of descriptive terms in music production can be shown to represent changes in musical timbre, which often requires the application of multiple audio effects [12, 13]. This suggests that in order to provide users with a flexible interface, combinations of effects need to be explored. This process is nontrivial, as individual audio effects are complex, multidimensional processing units [6], and the combination of linear and nonlinear systems are noncommutable, resulting in a large number of possible combinations. For example, Dynamic range compression placed before an equaliser (EQ) will provide a potentially different outcome than an EQ placed after the compressor [14], even when the settings of the two audio effects are retained and only the order of them is altered. This is confounded by additional contextual conventions, such as effects being used for specific instrument classes (e.g. drum compression or vocal equalisation).

1.2. Objectives

In this paper, we present a method for full processing chain recommendation, based on a dataset of empirically captured user data. We provide a comparative analysis of audio processing tools, based firstly on the frequency of individual effect usage within a chain, then on combinations thereof. Using this information, we can then identify commonalities and patterns in audio effects usage, with respect to contextual attributes such as timbral descriptions, audio effects and genre. We conclude by presenting a system which is able to recommend a number of processing modules based on the likelihood of an effect's position in a processing chain, weighted

by external factors. This method for processing chain recommendation can help lower the barrier to entry of novice engineers, and can reduce the time required for expert users, when incorporated into an intelligent mixing environment [15].

2. METHODOLOGY

In order to investigate the ways in which processing chains are constructed by audio engineers, we conducted an experiment in which participants were asked to apply audio processing to a number of predefined audio samples, to achieve a specified timbral transformation. Subjects were provided with a range of audio effects, with no restrictions placed on the number of instances of an effect, or the order in which they can be selected.

Audio samples were taken from the *Mixing Secrets* library¹ and were selected to span a range of instruments and genres. The instruments, selected for their popularity and availability in the dataset were *acoustic guitar*, *bass guitar*, *drums* (mixed), *electric guitar*, *piano*, *saxophone*, *violin*, and *vocals*. In order to evaluate the effects of the channel type, *mixed* signals were also used. These covered 5 genres: *Reggae*, *Folk*, *Hip Hop*, *Rock* and *Jazz*. From the multi-track recordings 30 seconds long excerpts were selected, in which all the instruments were active.

To describe the transformations requested from the users, a range of timbral descriptors were obtained from the SAFE Project [12]. Firstly, the ten most frequent terms were chosen based on the number of entries into the dataset, and then the ten terms with the highest generality across audio effects. This was to ensure that both terms which are associated with a single audio effect and terms that are associated with multiple audio effects were used [12]. The terms were: *air*, *boom*, *bright*, *close*, *cream*, *crisp*, *crunch*, *damp*, *deep*, *fuzz*, *punch*, *room*, *sharp*, *smooth*, *thick*, *thin*, and *warm*. The total number of terms was 18, as two of the most frequently used terms also displayed a high generality score. These were then stored in a relational database, resulting in 450 possible combinations of instrument, genre, and descriptor.

The tests were deployed using a web interface, in which subjects were given a URL² and asked to participate in their home studios. Whilst distributed tests like these contain more ambiguity due to uncontrolled variables such as listening environment and participant experience, we were able to collect larger amounts of data. This is a common practice in audio research [8, 16] provided suitable screening of participants takes place [17]. Participants were asked to follow the test instructions for a predetermined period of time. The duration of the tests was set to 5, 10, 15 or 30 minutes, depending on their availability. During this time, audio samples were presented along with a descriptor and a set of available audio effects.

The four audio effects available to test participants were (1) a parametric equaliser, (2) a dynamic range compressor, (3) a non-linear distortion, and (4) an algorithmic reverberation. These were chosen to reflect the plugins available through the SAFE Project³ [5], and were built using the JSAP web audio plugin framework [18]. Each time a processing chain is submitted, the plugins and their parameters are transmitted to a server along with an extensive set of differential audio features for each node in the chain, using

the JS-Xtract feature extraction library. A full list of the audio features can be found in [19].

3. SINGLE EFFECTS

A total of 178 submissions were made over a two week period by 47 participants. Of the four available plugins, 124 equalisation, 72 compression, 57 reverb and 40 distortion effects were used. 90 of the 178 entries only use a single plugin (50.6%) with a further 64 only using two plugins (36.0%). The longest processing chain created is 4, giving 256 possible permutations. As there were no bounds on the number of plugins in a chain, this suggests that long processing chains are unnecessary for the given task. Tables 1, 2 and 3 show the number of entries per instrument, descriptor and genre respectively, distributed across the audio effect type.

Inst.	#	EQ	Comp.	Dist.	Reverb.
Ac. Gtr	8	4 (0.43)	3 (0.29)	1 (0.00)	4 (0.43)
El. Gtr.	14	9 (0.73)	4 (0.14)	5 (0.36)	5 (0.22)
Saxo.	4	4 (1.00)	2 (0.33)	2 (0.33)	1 (0.00)
Mix	36	22 (0.68)	17 (0.40)	6 (0.15)	15 (0.48)
Bs. Gtr.	31	22 (0.65)	12 (0.36)	7 (0.08)	8 (0.27)
Vocals	21	15 (0.81)	6 (0.16)	3 (0.04)	5 (0.27)
Drums	30	21 (0.57)	13 (0.53)	9 (0.19)	8 (0.27)
Violin	17	12 (0.70)	7 (0.26)	4 (0.27)	8 (0.36)
Piano	17	15 (0.81)	6 (0.53)	3 (0.07)	3 (0.20)

Table 1: Number of entries for each instrument with the number of plugins applied and generality g_i across all descriptors in braces.

Desc.	#	EQ	Comp.	Dist.	Reverb.
air	7	4 (0.60)	3 (0.40)	0 (0.00)	3 (0.40)
boom	7	7 (0.86)	4 (0.30)	0 (0.00)	0 (0.00)
bright	10	10 (0.73)	3 (0.33)	1 (0.00)	3 (0.33)
close	13	11 (0.64)	5 (0.40)	1 (0.00)	5 (0.67)
cream	9	8 (0.81)	4 (0.38)	0 (0.00)	3 (0.17)
crisp	9	9 (0.58)	3 (0.44)	0 (0.00)	1 (0.00)
crunch	15	5 (0.34)	7 (0.61)	14 (0.78)	2 (0.14)
damp	8	4 (0.75)	1 (0.00)	1 (0.00)	9 (0.83)
deep	9	9 (0.82)	3 (0.11)	0 (0.00)	2 (0.17)
dream	9	4 (0.50)	1 (0.00)	1 (0.00)	9 (0.82)
fuzz	11	2 (0.25)	0 (0.00)	11 (0.64)	1 (0.00)
punch	9	7 (0.86)	9 (0.71)	1 (0.00)	0 (0.00)
room	13	4 (0.50)	2 (0.17)	1 (0.00)	13 (0.72)
sharp	7	7 (0.86)	5 (0.53)	1 (0.00)	0 (0.00)
smooth	9	6 (0.67)	5 (0.48)	2 (0.20)	3 (0.40)
thick	9	8 (0.56)	5 (0.20)	3 (0.50)	1 (0.00)
thin	11	11 (0.73)	1 (0.00)	1 (0.00)	2 (0.17)
warm	13	11 (0.82)	5 (0.40)	2 (0.17)	3 (0.11)

Table 2: Number of entries for each descriptor with the number of plugins applied and generality g_d across all descriptors in braces.

3.1. Effect Generality

A plugin or transform can be considered *general* if the likelihood of it occurring is not bound by the measured term (instrument, genre or descriptor). If an effect is only applied to a single instance, it has a low generality score; if an effect is applied to every

¹Available at <http://www.cambridge-mt.com/ms-mtk.htm>

²Available at <http://dmtlab.bcu.ac.uk/nickjillings/safe-AMT/>

³Available at <http://www.semanticaudio.co.uk>

Genre	#	EQ	Comp.	Dist.	Reverb.
Reggae	32	22 (0.61)	13 (0.43)	7 (0.21)	9 (0.33)
Jazz	33	24 (0.66)	20 (0.65)	9 (0.10)	13 (0.49)
Hip Hop	38	25 (0.64)	10 (0.39)	12 (0.21)	12 (0.34)
Folk	22	14 (0.684)	10 (0.31)	2 (0.07)	10 (0.31)
Rock	53	39 (0.60)	19 (0.47)	10 (0.20)	13 (0.20)

Table 3: Number of entries for each genre with the number of plugins applied and generality g_{genre} across all descriptors in braces.

Term	EQ	Comp.	Dist.	Reverb.
Genre	0.786	0.799	0.713	0.904
Instrument	0.692	0.625	0.656	0.640
Descriptor	0.766	0.631	0.291	0.464

Table 4: Generality of plugins against the type of term (genre, instrument, descriptor).

instance, it has a high generality score. Equations 1 and 2 calculate a single generality score from the data for a given contextual term (instrument, genre or descriptor). If a plugin only occurs for a small number of the terms, then the g will be low. These were adapted from [12]. Table 1 gives the generality of each plugin according to the source instrument.

$$g_i(p) = \frac{2}{D-1} \sum_{d=0}^{D-1} \text{dsort}(x(d, p)_i) \quad (1)$$

$$x(d, p)_i = \frac{n_p(d, i)}{\sum_{d=0}^{D-1} n_p(d, i)} \quad (2)$$

Here, $g_i(p)$ is the generality of a plugin p on instrument i . $n_p(d, i)$ is the number of plugin occurrences on descriptor d and instrument i . These measurements refer to the range of plugins are used to process a given instrument.

Table 1 shows that distortion is the least general effect whilst the EQ exhibits a consistently high generality score. The compressor is most general on Piano, Drums and full Mixes, suggesting that in these cases, the effects are applied irrespective of the genre and timbral descriptor. Table 2 measures the generality of descriptors across instrument classes. The genre, like the instrument, has little impact on the choice of plugin with all plugins attaining similar generality scores across the various genres, shown in table 4. Distortion is significantly less general when used on *Folk* samples, indicating that it must only be used in very specific use cases, with only 2 instances when a distortion was used from all 22 responses for *Folk*. The reverb effect followed similarly low generality scores.

Table 4 shows the cumulative generality scores for an effect being used across contexts: *genre*, *instrument* and *descriptor*. A low score here indicates the term has a high impact on whether a plugin appears. This suggests the genre and instrument play a relatively small role in the selection of effects in a processing chain. However the *descriptor* has an impact on whether distortion or reverb is used in the chain, indicating these only appear when a specific descriptor is used. EQ and compressor appear to be universally more general and can appear in any chain.

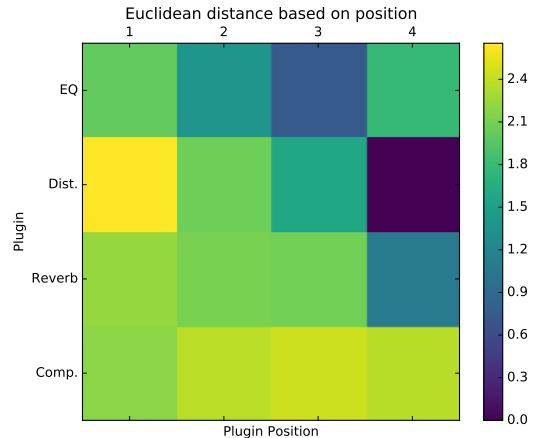


Figure 1: Euclidean distances of the features according to plugin type (SAFE distortion, SAFE equaliser, SAFE compressor and SAFE reverb) and position in the chain

3.2. Effect Salience

Each processor in the chain has a varying number of parameters, each set empirically by the participant. We quantify this by extracting audio features before and after each effect in the chain. Over 30 temporal and spectral features are averaged over each audio sample, extracted using JS-Xtract [19], in line with the features extracted by the SAFE project [5]. The impact of each plugin can then be characterised by the change in feature space before and after processing. We measure this using the Euclidean distance over each feature dimension, defined in Eq. 3.

$$d(\mathbf{p}, \mathbf{q}) = \sqrt{\sum_{i=1}^N (q_i - p_i)^2} \quad (3)$$

Each feature is vector normalised against all other instances of that same feature within a processing chain, thus capturing relative salience in the chain. Fig. 1 shows the feature distance as a function of position in the chain across all entries into the dataset. This indicates the first plugin generally has the greatest impact, irrespective of plugin type. As the plugin index increases, the feature differences decrease. The mean effect chain length is 1.64, where the probability of an effect being selected for a given position in the chain is presented in Table 5.

Effect	1st	2nd	3rd	4th
EQ (E)	0.44	0.43	0.21	0.33
Compressor (C)	0.22	0.28	0.25	0.33
Distortion (D)	0.15	0.10	0.16	0.00
Reverb (R)	0.17	0.18	0.38	0.33

Table 5: Probability of effects per chain position

For the first position, which includes chains of single effects, the EQ is the most popular, appearing in 44.9% of the total instances, followed by the compressor (22%). The first and second positions retain the same structure, and the effects in order of descending popularity are EQ, compressor, reverb and distortion.

This aspect shifts when moving to the third position, were the most popular effect is the reverb (37.5%), followed by the compressor (25%), EQ (20%) and distortion (16%). Finally the fourth position is split equally between EQ, compressor and reverb with the distortion never appearing in that position.

3.3. Plugin Order

We consider each processing chain to be a multi-dimensional vector, where each dimension represents a plugin instance. Each index in the vector is then considered to have a finite state. The likelihood of a plugin appearing at position k , given the state at positions $\{0, \dots, k - 1\}$ can be evaluated using a Markov chain [20, 21]. Eq. 4 and 5 give the state transition matrix for the chain, highlighting the probability of the next plugin type given the previous plugin, defined in equation 6. A fifth state is entered which represents a blank plugin state. The chain must start at this *empty* plugin and the chain terminates once this state is re-entered. The state vector v comprises equalisation (E), compression (C), distortion (D), and reverb(R).

$$v = [\text{'None'}, \text{'E'}, \text{'C'}, \text{'D'}, \text{'R'}] \quad (4)$$

$$P = \begin{bmatrix} 0.000 & 0.645 & 0.555 & 0.675 & 0.544 \\ 0.449 & 0.000 & 0.250 & 0.200 & 0.316 \\ 0.191 & 0.250 & 0.013 & 0.025 & 0.088 \\ 0.124 & 0.056 & 0.111 & 0.000 & 0.053 \\ 0.235 & 0.048 & 0.069 & 0.100 & 0.000 \end{bmatrix} \quad (5)$$

$$\Pr(A_n = p_i | A_{n-1} = p_j) = P_{i,i-1} = P_{i,j} \quad (6)$$

$$\Pr(A_n = p_i | A_{n-1} = p_j, \dots, A_0 = p_0) = \prod_{n=1}^N P(A_n, A_{n-1}) \quad (7)$$

Here, the probability that the k^{th} plugin is the last plugin in the chain is given by the first row, whilst the probability of the first plugin in the chain is given by the first column. The transition matrix can be used to generate all possible outcomes with their probabilities. Eq. 7 provides a formal definition, showing nodes in the chain be represented a probabilistic series of states. Using a Markov Chain, the most likely sequences from the model are: 1) EQ (29.0%), 2) reverb (12.8%), 3) compressor (10.6%), 4) distortion (8.3%), 5) compressor-EQ (6.2%) and 6) EQ-reverb (4.8%).

4. PROCESSING CHAINS

In total, 30 unique processing chains were constructed during the experiment. In order to compare the various combinations, plugin chains that were implemented only once in our dataset were excluded, leaving 19 unique entries. The mean usage of a processing chain is 8.78 times, and the most popular processing chains are EQ (27.5%), reverb (12.5%), compressor-EQ (11.9%), distortion (8.9%), EQ-compressor (8.9%) and EQ-reverb (5.3%). This correlates with effect transitions generated by the Markov Chain in Section 3.3.

To measure processing chain similarity, a matrix of descriptor occurrences per chain is constructed, using a distance measure based on the coexistence of terms in each pair of processing chains. We then compute pairwise distances to perform multidimensional scaling [22], followed by agglomerative clustering to establish a

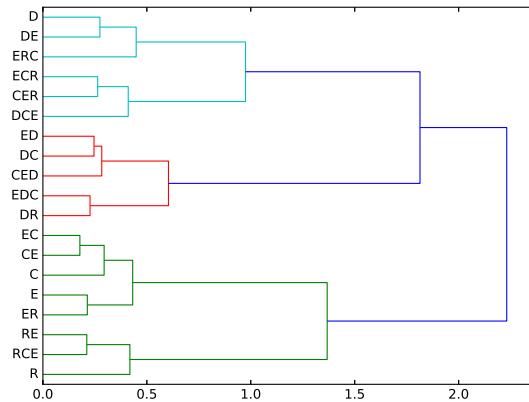


Figure 2: Hierarchical clustering of unique chains based on term usage

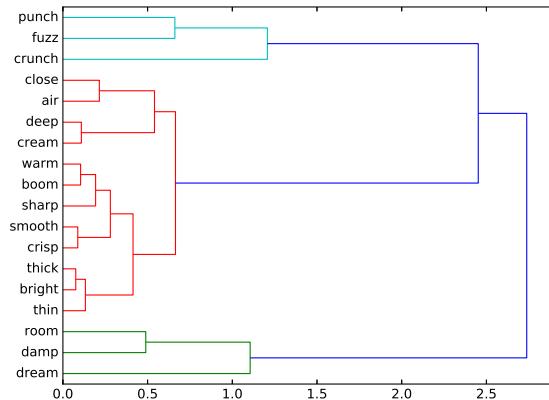


Figure 3: Hierarchical clustering of unique terms based on processing chain usage.

hierarchy of the plugin chains in the dataset, presented in Fig. 2. Plugin chains that are used to achieve similar terms, as is the case with EQ and EQ-compressor, are placed close in the hierarchy, whereas chains that do not share any descriptors, as is expected with distortion and reverb are placed further apart. Similarly, we can identify the relationship between transform descriptions based on the frequency of use across processing chains. In this case we represent the descriptive terms in multidimensional space, where each of the dimensions is the frequency of use for a given processing chain. A matrix D , with dimensions $M \times N$ is constructed, where M is the descriptor and N the unique plugin chain, and entry $D(i, j)$ is the amount of times plugin chain j was used to achieve descriptor i . This process allows us to perform hierarchical clustering, this time on the descriptive terms, presented in Fig. 3. The results show that the terms are organised in three predominant groups: a group that uses mainly distortion (*punch*, *fuzz*, *crunch*), a group that uses mainly reverb (*room*, *dream*, *damp*), and a group with high generality, distributed across a range of plugin

chains. For example *warm* is a descriptor that can be achieved by 8 different unique plugin chains, using 42% of the unique chains in our dataset, while *fuzz* makes use of just 15%.

4.1. Transform Similarity

An audio effect can have a more significant effect on the audio signal than others in the chain, based on its respective parameter space. In order to quantify this, audio feature differences across each effect in the chain are captured using Euclidean distance. This is represented as a matrix P with dimensions $M \times E$, where M represents the number of descriptors and E is the number of base effects (EQ, compressor, distortion, reverb). We can then apply dimensionality reduction to this matrix using PCA and project the audio effect classes into the low-dimensional space, as shown in Fig. 4.

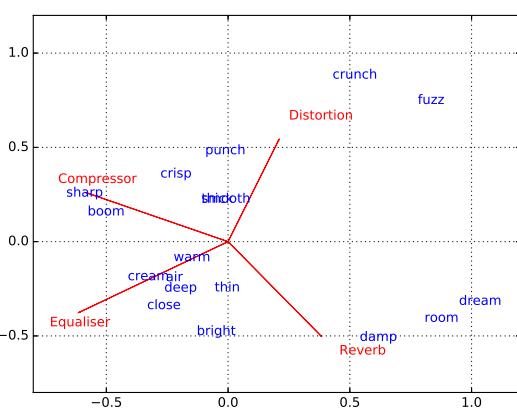


Figure 4: Low-dimensional descriptor mapping with prevalent dimensions

The figure shows that terms associated with specific effects, i.e. those with low generality scores, are highly correlated with the effect dimensions. *Fuzz* and *crunch* for example are correlated with distortion, *damp* and *room* with the reverb, and *sharp* with the compressor. Terms with a more general representation, such as *warm* tend to exhibit lower correlation scores.

5. FULL CHAIN RECOMMENDATION

5.1. Descriptor-based chain recommendation

The Markov chain approach to processing chain generation uses a matrix of conditional probabilities, based solely on the effect in the chain at position $k - 1$. This method will be inherently biased to favour plugins with a high generality (tables 1 and 2). For example, there is a high probability that the matrix P (Eq. 5) will generate a chain consisting of a single effect, given the likelihood of 60.7%. To improve the recommendation, we use a state transition matrix, based on the specific probabilities for each descriptive term P_d . Sequentially generating states using P_d will then produce a set of chains for the specified descriptor d . For instance, *fuzz* will generate a chain of just distortion with a likelihood of 74.38%, and an EQ-distortion chain with a likelihood of 16.53%.

By implementing an independent matrix for each timbral descriptor, we are also able to compare the way in which terms are organised in our generated processing chains, to the way the terms are organised in our dataset. By generating the same number of entries per transform, we construct a similarity matrix of terms, based on their frequency of plugin chain usage. We then apply dimensionality reduction and the resulting mapping is presented in Fig. 5. Here, descriptors that generate similar chains are placed together such as *room* and *damp*, or *sharp* and *punch*. This behaviour adds a level of versatility to the system, given that similar or identical chains, which can be used for achieving neighbouring descriptors, have a high probability of co-occurring.

To demonstrate the similarity of the original and generated descriptor mappings, the two spaces can be assessed using the *trustworthiness* (T_k) and *continuity* (C_k) metrics [23, 24], shown in Eq. 8 and 9.

$$T_k = 1 - \frac{2}{nk(2n - 3k - 1)} \sum_{i=1}^n \sum_{j \in U_i^{(k)}} (r(i, j) - k) \quad (8)$$

$$C_k = 1 - \frac{2}{nk(2n - 3k - 1)} \sum_{i=1}^n \sum_{j \in V_i^{(k)}} (\hat{r}(i, j) - k) \quad (9)$$

Here, the distances of the n entries in two spaces (U and V) are converted to ranks (r and \hat{r}) between points i and j . The measurements then evaluate the distributions of datapoint in the respective spaces over a number of neighbouring datapoint (k).

The low-dimensional space generated by the descriptor-wise state transition matrices (presented in Fig. 5) achieves a *trustworthiness* score of 0.78 for the original structure of unique terms (the space used to generate the clusters in Fig. 3). This suggests that the organisation of terms is preserved when generating processing chains using the Markov Chain approach. Similarly, for *continuity*, the structure of the probability matrix is retained with a score of 0.782.

For terms with low generality, i.e. those that have very specific plugin usage patterns such as *fuzz* and *dream* (see table 2), very specific plugin chains will appear from P_d . However, for entries which consist of more general effects, *warm*, *smooth* and *cream*, more chains can appear which have lower probabilistic scores. This can be interpreted as a low confidence score, thus producing more variance in the results. P_{smooth} generates the following chains: EQ (16.67%), EQ-compressor (13.33%), reverb (11.11%) and distortion (11.11%). These all have low and relatively equal probability of occurring, highlighting how unspecified this matrix is.

This can be improved by weighting the transition matrix probabilities to penalise plugins which are not related to the term. This is done by incorporating weights which indicate the plugin's prevalence in a chain. A weight w_p is found using Eq. 10 to 12.

$$w_p = \frac{\sum_{n=0}^{N-1} \sum_{l=0}^{L-1} f(d, l) g(x(l), p)}{\sum_{n=0}^{N-1} n} \quad (10)$$

$$f(d, l) = \begin{cases} 1, & \text{if } l = \text{argmax}(d) \\ 0, & \text{otherwise} \end{cases} \quad (11)$$

$$g(x, p) = \begin{cases} 1, & \text{if } x = p \\ 0, & \text{otherwise} \end{cases} \quad (12)$$

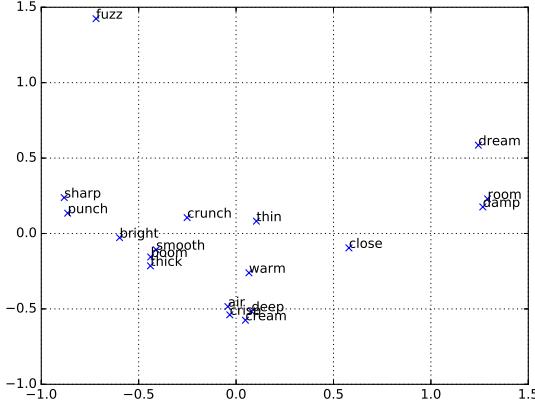


Figure 5:
Low-Dimensional Mapping for unweighted Markov Chains
recommender

Here, N is the number of chains, L the length of chain n , d is a vector of the plugin Euclidean distances (Eq. 3) and x a vector of the plugin codes. Function f (Eq. 11) returns 1 if the plugin at position l is the most prevalent effect and function g (Eq. 12) returns a 1 if the plugin at position l is the same as plugin p . The weights are then multiplied with the corresponding row of the descriptor’s transition matrix.

Applying weights to the probability increases the possibility of the prevalent effect(s) appearing at any position in the chain. In this manner the system gains an additional level of adaptivity, being able to recommend chains that might not exist in the original dataset, but concurrently takes into account the most important plugin in the chain. For example, using the weighted transition matrix, we are able to predict chains for the the *smooth* descriptor with higher accuracy, predicting compressor (28.87%), EQ (21.66%) and EQ-compressor (15.28%).

The distribution of terms using the weighted Markov chain approach are presented in Fig. 6. Using the *trustworthiness* and *continuity* metrics, the original structure of the descriptors is retained at a value of 0.75, and the *continuity* between the transformed space to the original data has increased to 0.86.

5.2. Instrument-based chain recommendation

As the source instrument also proves to be a salient attribute plugin selection, we can apply the same weighting method to the instrument classes P_i . This will allow the effects, which are specific to an instrument to be favoured in the processing chain. For instance, for the *Mix* samples, the Markov chain method generates EQ (19.69%), reverb (17.78%) and compressor (13.73%). However, applying the weighting w_i based on the most prevalent effect does not improve the model. With the weights, the generated processing chains for *Mix* are compressor (30.05%), EQ (25.69%) and reverb (21.44%). Thus, applying the weights reduces the likelihood of complex chains occurring. The weights for the *Mix*, w_{Mix} are 0.250, 0.333, 0.138 and 0.278 for the EQ, compressor, distortion and reverb respectively. Conversely, the weights for the *crunch*, w_{crunch} are 0.066, 0.266, 0.667 and 0. This shows that whilst certain plugins are applied more generally to a given instru-

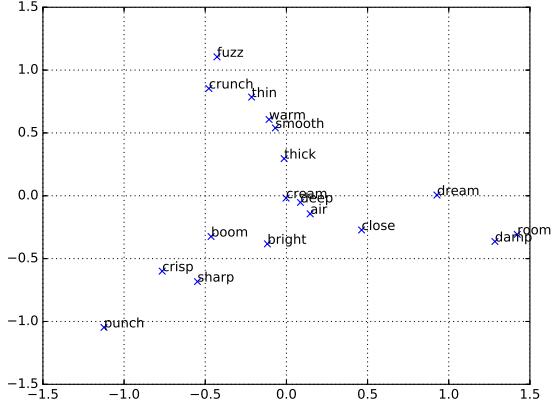


Figure 6: Low-Dimensional Mapping for weighted Markov Chains recommender

ment class, the most prevalent effect in a chain is actually driven by the descriptor. The weights for the *mixed* samples are all relatively similar, except the distortion, which means after scaling and normalising the matrix, a very similar transition matrix is created.

6. CONCLUSION

We have introduced a method for audio processing chain recommendation, based on a dataset of user-inputs. We captured information regarding the instrument, genre and descriptor and used them to weight a state transition matrix. To evaluate the output of the model, we measured the similarity of descriptor mappings in low-dimensional space using *trustworthiness* and *continuity*, and we showed that a descriptor-based Markov chain method achieves a score of $T(k) = .78$, $C(k) = .782$ and the weighted descriptor-based model achieves a score of $T(k) = .75$, $C(k) = .86$.

We provide an evaluation of plugin usage in processing chains and show that the role of genre is negligible in plugin selection, whilst the descriptor heavily influences this decision making process. The EQ and compressor plugins both exhibit high generality, which suggests they are selected for most processing chains irrespective of instrument or descriptor. The distortion and reverb plugins are very specific, which means they are more frequently used when a specific timbral transformation is required.

7. ACKNOWLEDGMENTS

The work of the first author is supported by The Alexander S. Onassis Public Benefit Foundation.

8. REFERENCES

- [1] Joshua Reiss, “Intelligent systems for mixing multichannel audio,” in *17th International Conference on Digital Signal Processing*, July 2011, pp. 1–6.
- [2] Enrique Perez-Gonzalez and Joshua D. Reiss, “Automatic gain and fader control for live mixing,” in *2009 IEEE Work-*

- shop on Applications of Signal Processing to Audio and Acoustics*, Oct 2009, pp. 1–4.
- [3] Stuart Mansbridge, Saorise Finn, and Joshua D. Reiss, “Implementation and evaluation of autonomous multi-track fader control,” in *Audio Engineering Society Convention 132*, 2012.
 - [4] Stuart Mansbridge, Saorise Finn, and Joshua D. Reiss, “An autonomous system for multitrack stereo pan positioning,” in *Audio Engineering Society Convention 133*, 2012.
 - [5] Ryan Stables, Sean Enderby, Brecht De Man, György Fazekas, and Joshua D. Reiss, “SAFE: A system for the extraction and retrieval of semantic audio descriptors,” in *15th International Society for Music Information Retrieval Conference*, 2014.
 - [6] Spyridon Stasis, Ryan Stables, and Jason Hockman, “Semantically controlled adaptive equalisation in reduced dimensionality parameter space,” *Applied Sciences*, vol. 6, no. 4, 2016.
 - [7] Spyridon Stasis, Jason Hockman, and Ryan Stables, “Navigating descriptive sub-representations of musical timbre,” in *Proceedings of the International Conference on New Interfaces for Musical Expression*, Copenhagen, Denmark, 2017.
 - [8] Mark Cartwright and Bryan Pardo, “Social-EQ: Crowdsourcing an equalization descriptor map,” in *Proceedings of the International Society of Music Information Retrieval*, 2013, pp. 395–400.
 - [9] Sean Enderby and Ryan Stables, “A nonlinear method for manipulating warmth and brightness,” in *Proceedings of the 20th International Conference on Digital Audio Effects*, Edinburgh, UK, 2017.
 - [10] Jacob A Maddams, Saorise Finn, and Joshua D Reiss, “An autonomous method for multi-track dynamic range compression,” in *Proceedings of the 15th International Conference on Digital Audio Effects*, 2012.
 - [11] Prem Seetharaman and Bryan Pardo, “Reverbalize: a crowdsourced reverberation controller,” in *Proceedings of the 22nd ACM international conference on Multimedia*, 2014, pp. 739–740.
 - [12] Ryan Stables, Brecht De Man, Sean Enderby, Joshua D. Reiss, György Fazekas, and Thomas Wilmering, “Semantic description of timbral transformations in music production,” in *Proceedings of the 2016 ACM on Multimedia Conference*, 2016, pp. 337–341.
 - [13] Taylor Zheng, Prem Seetharaman, and Bryan Pardo, “SocialFX: Studying a crowdsourced folksonomy of audio effects terms,” in *Proceedings of the 2016 ACM on Multimedia Conference*. ACM, 2016, pp. 182–186.
 - [14] Roey Izhaki, *Mixing audio: concepts, practices and tools*, Taylor & Francis, 2013.
 - [15] Nicholas Jillings and Ryan Stables, “Investigating music production using a semantically powered digital audio workstation in the browser,” in *Audio Engineering Society Conference on Semantic Audio*, Erlangen, Germany, June 2017.
 - [16] Michael H. Birnbaum, “Human research and data collection via the internet,” *Annual Review of Psychology*, vol. 55, pp. 803–832, October 2003.
 - [17] Mark Cartwright, Bryan Pardo, Gautham J. Mysore, and Matt Hoffman, “Fast and easy crowdsourced perceptual audio evaluation,” in *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, March 2016, pp. 619–623.
 - [18] Nicholas Jillings, Yonghao Wang, Joshua D. Reiss, and Ryan Stables, “JSAP: A plugin standard for the web audio api with intelligent functionality,” in *Audio Engineering Society Convention 141*, 2016.
 - [19] Nicholas Jillings, Jamie Bullock, and Ryan Stables, “JS-Xtract: A realtime audio feature extraction library for the web,” in *17th International Society for Music Information Retrieval Conference*, 2016.
 - [20] Andrey Markov, “Extension of the limit theorems of probability theory to a sum of variables connected in a chain,” *Dynamic Probabilistic Systems*, vol. 1, 1971.
 - [21] George Tauchen, “Finite state markov-chain approximations to univariate and vector autoregressions,” *Economics Letters*, vol. 20, no. 2, pp. 177 – 181, 1986.
 - [22] Warren S. Torgerson, *Theory and methods of scaling.*, Wiley, 1958.
 - [23] Jarkko Venna and Samuel Kaski, “Visualizing gene interaction graphs with local multidimensional scaling.,” in *ESANN*, 2006, vol. 6, pp. 557–562.
 - [24] Laurens Van Der Maaten, Eric Postma, and Jaap Van den Herik, “Dimensionality reduction: a comparative,” *J Mach Learn Res*, vol. 10, pp. 66–71, 2009.

INVESTIGATION OF A DRUM CONTROLLED CROSS-ADAPTIVE AUDIO EFFECT FOR LIVE PERFORMANCE

Saurjya Sarkar

Birla Institute of Technology & Science
Pilani, India
saurjya.sarkar@gmail.com

Joshua Reiss

Queen Mary University of London
London, United Kingdom
josh.reiss@qmul.ac.uk

Oeyvind Brandtsegg

Norwegian University of Science and Technology
Trondheim, Norway
oyvind.brandtsegg@ntnu.no

ABSTRACT

Electronic music often uses dynamic and synchronized digital audio effects that cannot easily be recreated in live performances. Cross-adaptive effects provide a simple solution to such problems since they can use multiple feature inputs to control dynamic variables in real time. We propose a generic scheme for cross-adaptive effects where onset detection on a drum track dynamically triggers effects on other tracks. This allows a percussionist to orchestrate effects across multiple instruments during performance. We describe the general structure that includes an onset detection and feature extraction algorithm, envelope and LFO synchronization, and an interface that enables the user to associate different effects to be triggered depending on the cue from the percussionist. Subjective evaluation is performed based on use in live performance. Implications on music composition and performance are also discussed.

Keywords: Cross-adaptive digital audio effects, live processing, real-time control, Csound.

1. INTRODUCTION

Adaptive audio effects are characterized by a time-varying control on a processing parameter. The control is computed by extracting features from the input audio and mapping their scaled versions to relevant control parameters of an effect [1]. Cross-adaptive audio effects are adaptive audio effects where the features of a signal are analysed to control the processing parameters of another signal. This opens a range of possibilities as it presents a new form of communication between musicians. The idea of another performer interfering with the way your instrument sounds is disruptive at first, but if routed carefully it can have profound impact on the way we perform music.

In recent years, we have seen a few implementations of cross-adaptive audio effects. [2] described how such effects can be used to automate the mixing process by utilizing features across multiple tracks to determine the processing applied to each track. Such intelligent mixing systems have been implanted and evaluated for automation of multitrack equalization [3] and of multitrack dynamic range compression [4]. [5-7] presented audio processing plugins with the capability to implement user-defined cross-adaptive effects. [8] developed a genetic algorithm (GA) and artificial neural network (ANN) based cross-adaptive audio effect that can control user-defined parameters to make a source audio file sound as close as possible to a given target audio file.

An advantage of audio post-production is the ability to move recorded segments and synchronize them after recording. Effects are applied during production which are time-aligned with audio events. Replicating such effects live with musicians controlling these effects while playing their instrument is often not feasible. However, it may be possible to use cross-adaptive digital audio effects to synchronize the effect applied to one source to events produced by another source. Using cues from a percussionist to synchronize effects across instruments could be highly beneficial in live performance. It can introduce live instruments to replace

sampled sounds, since acting on such cues can keep effects synchronized while accommodating for human error and reaction times.

In this paper, we explore a cross-adaptive framework that enables a percussionist to orchestrate effects on other instruments in real time. In particular, we evaluate such an approach for several scenarios where musicians try to recreate, in live performance, a synchronized audio effect that otherwise would only be implemented in the studio. The effects presented here are feed-forward cross-adaptive audio effects as the features are extracted from the drum input source(s) and effects are implemented on the synth input [1]. They include short duration effects such as ducking, tremolo and filter sweep effects that can be triggered by drum cues in real time. These effects have been implemented as a live performance tool which can synchronize effects across multiple instruments using cues from the drummer. This enables the drummer to orchestrate effects across multiple instruments during performance.

Though similar effects can be implemented using MIDI triggers and sidechaining, our framework has more flexibility. With MIDI triggers or sidechaining, each drum hit would trigger the cross-adaptive effect. In our framework, the onset detector can be manipulated by the performer to trigger the effect only on louder notes or avoid re-triggering on closely spaced notes. Using an onset detector also enables the use of onsets from physical cymbals and/or drums, and does not require additional hardware.

2. IMPLEMENTATION

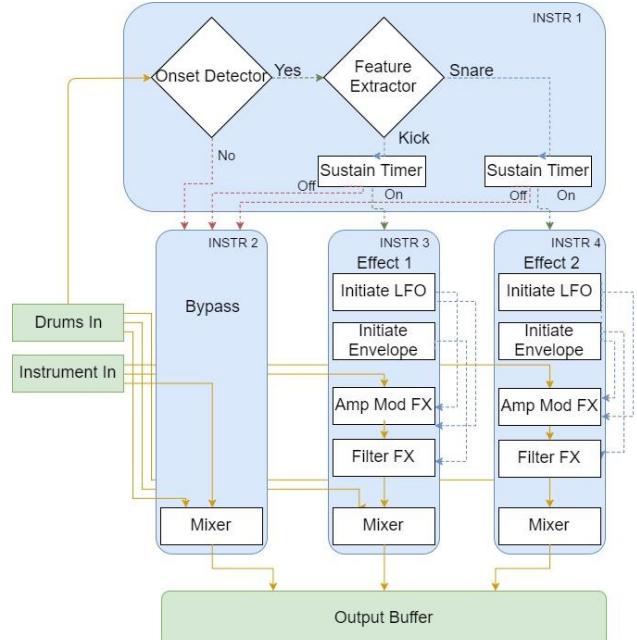


Figure 1: A signal flowchart for the cross-adaptive audio effect framework.

The cross-adaptive effects were implemented as a VST plugin using Csound (code available here: <https://code.soundsoftware.ac.uk/attachments/download/2231/CrossAdaptiveMain.csd>). The effects run within a signal flow as depicted in Fig. 1, where an onset detector runs on percussive input channels. Upon detection of an event, it initializes an envelope profile and an LFO which are sent to the desired effects acting on another instrument.

The VST plugin has three inputs, two control inputs coming from kick and snare drums and one instrument input (in this case the synthesizer) on which the effect was applied. Onset detection was applied on both the control inputs to trigger two different effects on the synth channel.

REAPER was used to run the VST plugin since it supports multitrack plugins. The effects were routed as following.

- Track 1 – Kick Drum Input. The VST is loaded on this track.
- Track 2 – Snare Drum Input. Master Send is switched off, channel 3/4 is sent to Track 1.
- Track 3 – Synth Input. Master Send is switched off, channel 5/6 is sent to Track 1.



Figure 2: Plugin User Interface, including Envelope and LFO controls.

2.1. Onset Detection

An onset detection algorithm [5] was applied to the amplitude envelope of the percussive signal, and transients extracted from it. We used the Csound opcode ‘follow2’ based on the work by Jean-Marc Jot, to extract the amplitude envelope of the percussive signal.

$$g = 0.001^{(1/(f_s \cdot \tau_a))} \quad \text{if } e[n-1] < x[n] \quad (1)$$

$$= 0.001^{(1/(f_s \cdot \tau_r))} \quad \text{else}$$

$$e[n] = (1 - g) \cdot x[n] + g \cdot e[n-1] \quad (2)$$

$x[n]$ is the input drum signal, $e[n]$ is the envelope extracted from signal $x[n]$. τ_a and τ_r are attack and release times, and f_s is the sampling frequency.

We then down-sample $e[n]$ to the frame rate with a frame width of 32 samples. We convert this down-sampled envelope follower to the decibel scale and compare it to a delayed copy of the same to detect transients. For an onset to be detected, all three of the following conditions should be simultaneously true for a given sample.

$$1. \quad e[n] > T_{gate} \quad (3)$$

$$2. \quad M - e[n-d] > T_{decay} \quad (4)$$

$$3. \quad e[n] - e[n-d] > T_{slope} \quad (5)$$

Where M is the local maximum (the last onset trigger), d is the distance between the samples between which the slope (transient) is calculated, T_{gate} is the noise gate threshold, T_{decay} is the minimum

amount by which the signal needs to decay before allowing the next onset to be detected, and T_{slope} is the minimum increase in amplitude required between $e[n]$ and $e[n-delay]$ to detect an onset. Whenever the onset detector detects an event from the drum signal, it initiates an event for the duration set by the envelope control panel.

2.2. Envelope Control

The envelope control panel, shown in Fig. 2, sets the duration and behaviour of the ASR envelope that is mapped to the parameter of the effects. The parameters mean the same as for any conventional dynamic effect. There are individual envelope controls for effects associated to different drums.

The attack time sets the time required for the control value to reach its maximum value. The sustain time sets the duration during which the control value will approach and hold its maximum value. Therefore, if the attack time is greater than sustain time, the control value will never reach its maximum value, and it will start reducing after the sustain time has passed. Thus, the actual duration of the effect is not affected by the attack time.

The release time sets the amount of time the control value takes to drop to zero from its value at the end of the sustain. The sum of sustain time and release time determine the effect duration.

2.3. LFO Control

The LFO is a low frequency oscillator used to produce tremolo and vibrato among other effects. The frequency of the LFO is set by the tempo (in BPM) and count (eg. 1/8th or 1/16th notes) inputs, shown in Fig. 2.

$$f[n] = \text{tempo} \cdot \text{count} / 60 \quad (6)$$

The frequency control button allows the frequency of the LFO to vary according to the control value ‘ $e[n]$ ’ set by the envelope. When turned on, the frequency of the LFO is given by:

$$f[n] = \text{tempo} \cdot \text{count} \cdot e[n] / 60 \quad (7)$$

The floor control button enables the depth of the tremolo effect to be controlled by the envelope control value. The LFO output is then given by:

$$y[n] = 1 - \text{depth} \cdot e[n] \cdot LFO[n] \quad (8)$$

The shape of the LFO can be set to sine, triangle, square bipolar, square unipolar, saw-tooth down, or saw-tooth up.

3. IMPLEMENTED EFFECTS

3.1. Amplitude Modulation Effects

Two types of amplitude modulation effects were implemented.

3.1.1. Ducking

If the amplitude modulation effect is turned on without enabling the LFO, then a simple ducking effect is produced as follows:

$$y[n] = 1 - \text{depth} \cdot e[n] \quad (9)$$

where $y[n]$ is the amplitude envelope of the output signal.

3.1.2. Tremolo

If the amplitude modulation effect is turned on with the LFO enabled, then a tremolo effect is produced. If the LFO control is switched off, then a simple tremolo is produced as per Eq. (6). If the frequency control button is turned on, then the tremolo frequency is modulated by the control envelope as given in Eq. (7). If the floor control is switched on, then a ducking tremolo effect is produced as per Eq. (8).

3.2. Filter Effects

Filter sweep effects were implemented using low pass, high pass and notch filters whose center or cutoff frequencies were dynamically controlled by the ASR envelope value. The filter center/cutoff frequency was updated at the frame rate.

3.3. Effect Performance

Fig. 3 shows synthesizer and drum samples used to show the spectral and temporal changes applied by the effect. The drum sample was composed of a kick followed by a snare drum hit, which were routed to different effects.

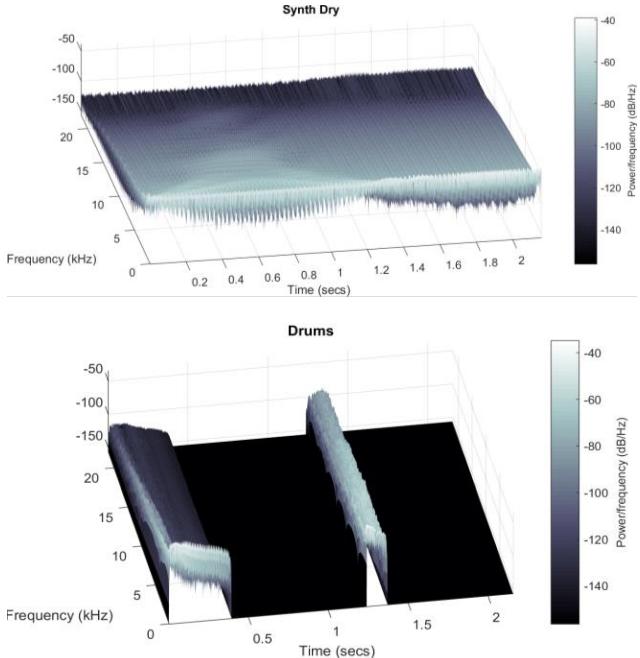


Figure 3: Spectrograms of dry synthesizer sample (top) and drum sample (bottom).

For the filter sweep effect, shown in Fig. 4, the kick trigger was associated to a low pass filter with cutoff frequency varying from 20 Hz to 22 kHz, as per the control envelope, and the snare trigger was associated to a high pass filter with cutoff frequency from 20 Hz to 20 kHz.

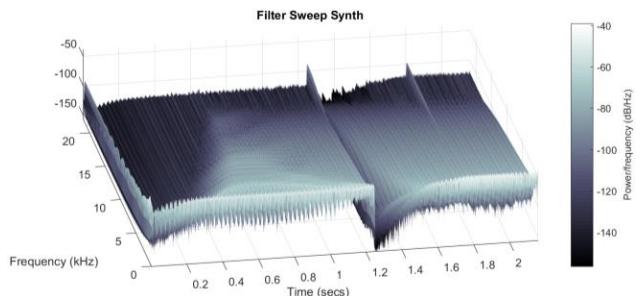


Figure 4: Filter Sweep Effect Spectrogram.

For the tremolo effect, the kick trigger was associated with a sine tremolo at quarter note triplets and the snare trigger was associated with a sawtooth tremolo at quarter notes, shown in Fig. 5.

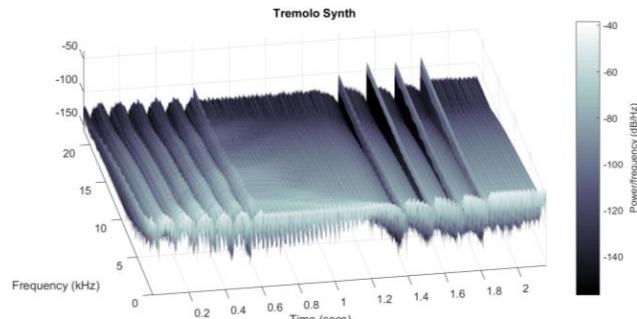


Figure 5: Tremolo Effect Spectrogram

A frame glitch was observed in the spectrogram at turn-off and turn-on time of each effect (seen prominently in Fig. 5 at 1.8 sec) but this was not audible during performance.

4. PERFORMANCE EVALUATION

To evaluate the performance of this effect and its implications on live performance, a performance study was conducted with 5 amateur musicians (2 drummers, 3 keyboardists). Participants were asked to replicate drum-synchronized effects on synthesizers from popular songs, with and without the effect. Their experiences were contrasted and analysed to assess the effectiveness of cross-adaptive audio effects as a live tool, and their applicability in modern music.

4.1. Test Setup

A two microphone setup was used to take input from the drums, one for the snare and one for the kick. Gate thresholds of -10 dB and -18 dB were used for the kick and snare microphones respectively. The high thresholds allowed the drummer to selectively trigger the effect on specific notes using heavily accented notes.

A MOTU Hybrid MKIII sound card was used as an interface to provide input to Reaper. The native MOTU ASIO drivers were used with a buffer of 256 samples and sampling rate of 96kHz. The cross-adaptive effect was applied on a VST synthesizer representing the instrument.

Routing:

- Kick Input → Channel 1 of plugin
- Snare Input → Channel 3 of plugin
- Synth Input → Channel 5 & 6 of plugin

Table 1: List of Songs for Performance Test.

Name	Artist	Start/Stop	Effect
Closer	The Chainsmokers feat. Halsey	1:10/1:31	Level controlled by ASR synchronized with kick and Snare
In the name of love	Martin Garrix feat. Bebe Rexha	0:49/1:03	Kick triggered tremolo and snare triggered mute
All we know	The Chainsmokers feat. Phoebe Ryan	1:14/1:36	Kick triggered fixed duration tremolo

Table 1 lists the three song sections and the respective effects that were selected as test cases. Each song section was performed with 2 groups of musicians with alternating order of performing manually or with the effect. Table 2 shows the order and total time spent practicing and recording for each of the 13 experiments that were conducted. The test conditions and setup were identical for

all sessions and symmetrically alternated between different songs and different approaches.

Table 2: Log of Conducted Performance.

Expt. No.	Song No.	Drummer	Key-boardist	Method	Time Taken
1	1	A	C	Effect	4:41
2	2	A	A	Manual	12:15
3	2	A	A	Effect	4:23
4	3	A	B	Effect	2:17
5	3	A	B	Manual	7:40
6	2	B	B	Manual	6:43
7	2	B	B	Effect	3:35
8	1	B	B	Manual	7:15
9	1	B	B	Effect	4:42
10	3	B	A	Effect	1:54
11	3	B	A	Manual	7:10
12	1	B	A	Manual	5:15
13	1	B	A	Effect	2:40

4.2. Performer Background

Keyboardist A is a classical pianist with 12 years of training and enjoys listening to ethnic music.

Keyboardist B has played piano for 20 years, sang as an Alto in multiple choirs over time and listens to classical music and orchestral movie soundtracks.

Keyboardist C has played piano for 5 years and occasionally plays other instruments including keyboards, guitars, harps, and ukuleles.

Drummer A studied music at GCSE and A-level on clarinet and started playing drums, bass and singing in bands at 16. He likes music that is simple and elegant, like a tight funk groove over a flashy show of chops.

Drummer B played keyboard since childhood and has played drums for the last 8 years. He plays mostly rock-oriented genres like Blues-Rock, Grunge, Indie, Nu Metal, Punk-Rock.

4.3. Test Case Analyses

The plugin setting for each experiment is available at <https://code.soundsoftware.ac.uk/attachments/download/2230/Table%203.docx>.

4.3.1. Song 1 (All we know)

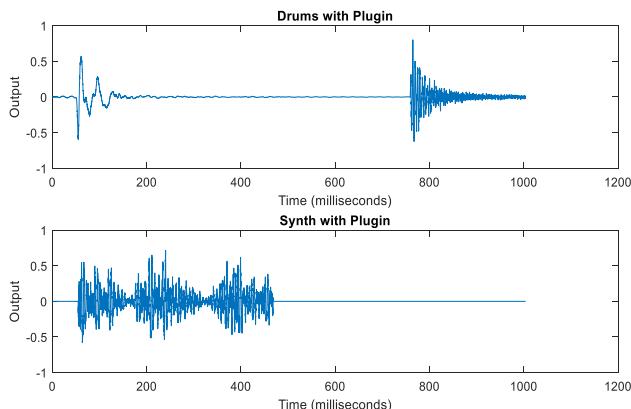


Figure 6: Song 1 Recording with effect (Expt. No. 4).

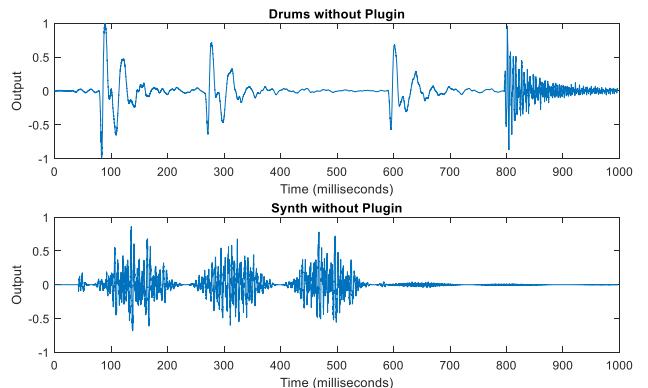


Figure 7: Song 1 Recording without effect (Expt. No. 5).

The effect to be replicated was to trigger 3 cycles of tremolo synchronized with the kick from the drummer. The duration of the synthesizer note was supposed to be exactly 3 cycles.

With Effect:

As seen in the drum track recordings shown in Fig. 6 & 7, the drummer had to modify the groove by removing a few double kicks to make the plugin trigger the effects on the synthesizer in the desired manner. The keyboardist did not need to control the onset time and duration of the chord, but only keep holding the correct chord throughout.

In Fig. 6, we see that each synth note was synchronized to each drum onset with a delay of 2.3ms from the edge of the transient of the kick drum hit (see Fig. 12 & 13). The duration of the note was fixed by the sustain duration. This was set to 3 cycles of the tremolo at quarter notes at 90 BPM.

Without Effect:

In Fig. 7 we see the kind of issues that performers faced. First, onset times of the synth and the drum always had a difference due to human error, ranging from 23ms to 98ms. Second, note durations varied between 478 and 580ms, consistently greater than the desired value of 48 ms. This can be attributed to reaction times since performers depended on the audio cue to complete 3 cycles of tremolo and then release the note.

Moreover, sometimes the phase of the tremolo effect was not perfectly synchronized because of the above-mentioned errors, which can be disorienting for the drummer.

4.3.2. Song 2 (In the name of love)

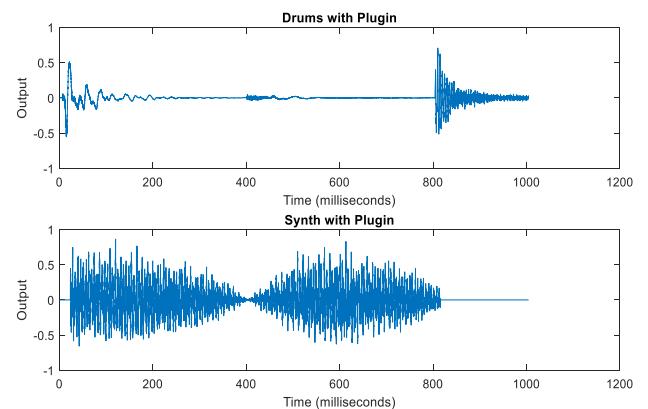


Figure 8: Song 2 Recording with effect (Expt. No. 7).

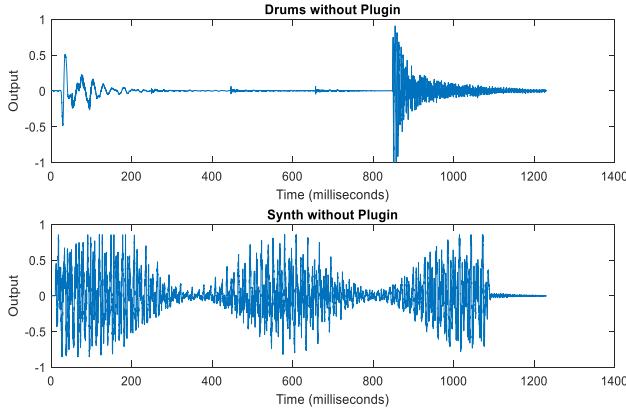


Figure 9: Song 2 Recording without effect (Expt. No. 2).

This song required the synthesizer to start a note with full depth tremolo on each kick drum hit, and stop the note at every snare drum hit. This was implemented by using the kick-based trigger to enable output and start an LFO to control the amplitude modulator for a long duration (longer than the length of one bar), and setting the snare-based trigger to mute the output. The manual performance used a regular note-synchronized tremolo effect and required the keyboardist to control the note onset and release.

With Effect:

This song required the drummer to play normally (as seen in Fig. 8 & 9, the drum tracks are identical), but the keyboardist had to hold the notes slightly before the drummer and hold it longer than the song required. This ensured that the note onset and release were controlled by the drum triggers alone. Since the song was slow, this was not difficult and the effect worked very well, see Fig. 8. A constant delay of 2.3 ms was observed between the peak of each drum transient and the note onset/release. This delay was limited by the latency of the sound card and driver.

Without Effect:

Fig. 9 shows that, when manually trying to replicate this song, there were similar onset differences as in the previous experiment. The slower tremolo of this song caused phase errors to be less than in the previous experiment. But due to the slow tempo and abrupt stop at every snare hit, it was very noticeable when the keyboardist released the note later than the snare hit.

4.3.3. Song 3 (Closer)

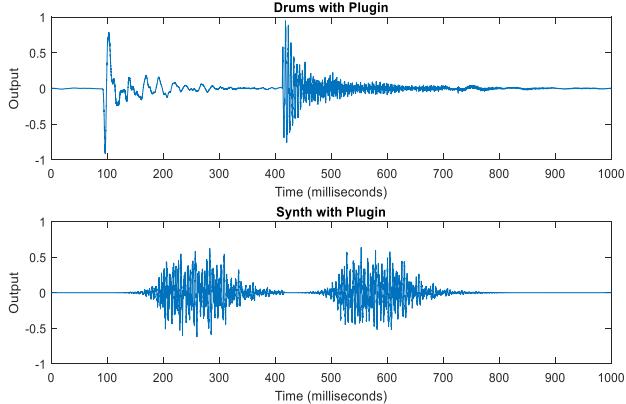


Figure 10: Song 3 Recording with effect (Expt. No. 9).

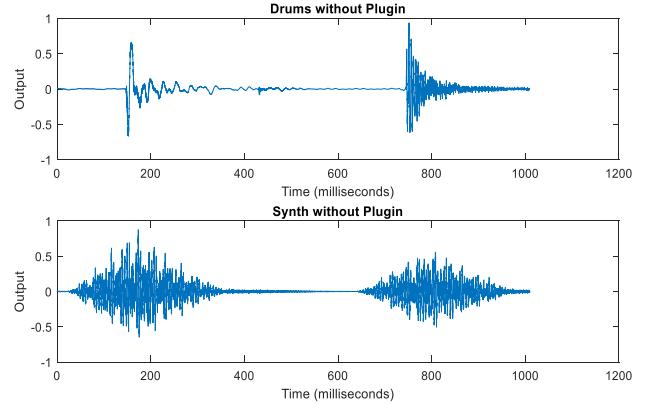


Figure 11: Song 3 Recording without effect (Expt. No. 13).

In this experiment, each synthesizer note had ~ 0.1 second attack and release time, and were played staccato. Each of the synth notes were played only with each kick and snare notes.

With Effect:

The effect had an exponential ASR envelope to control amplitude of the synth output. Due to the exponential nature of the attack curve, the output amplitude grows very slowly initially, staying inaudible for a while. Hence, the perceived onset of the note from the effect was delayed compared to the trigger time of the effect, especially for short attack times (<0.2 seconds). As seen in Fig. 10, performers perceived a lag of about 70 ms while using the effect. This was disruptive for the performers and made it difficult for them to use the effect. We observed that during performance, notes that occurred with the snare hits did not seem to have the distinct lag, although the processing complexity was the same. This was probably because the snare drum itself has longer sustain, thus masking the duration when the effect output is inaudible and slowly blends in as the effect becomes louder and the snare fades away. The song had multiple, fast chord changes. This made the effect more difficult to use since the keyboardist needed to anticipate the drum onset and hit the note prior to that.

Without Effect:

This experiment was trivial as the only variable was the onset times. The difference in onset times in this experiment was greater than the previous experiments since the song was faster and had frequent chord changes. We see in Fig. 11 an instance where the keyboardist's onset occurs significantly before the onset of the drummer.

4.4. Performer Analysis

4.4.1. Drummer A

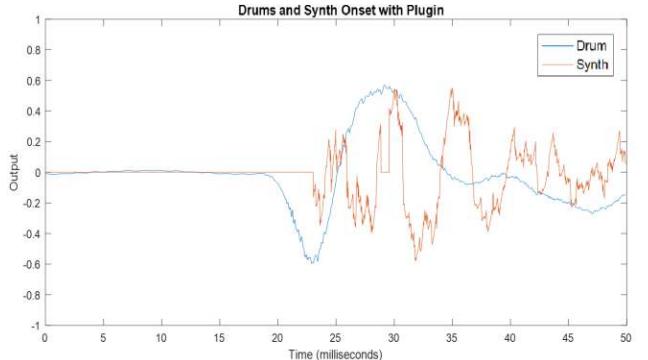


Figure 12: Drummer A onset detection example.

Drummer A used the heel down technique while playing the kick drum. Thus, his kick drum hits were soft and had long rise and short decay times. As seen in Fig. 12, the onset detector was triggered at the end of the first half oscillation of the kick drum diaphragm. Drummer A also hit the snare softly with a rim click, which caused the snare to sustain for a shorter time.

4.4.2. Drummer B

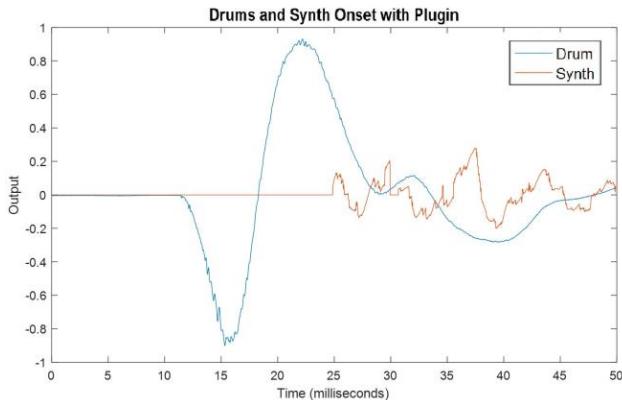


Figure 13: Drummer B onset detection example.

Drummer B used the heel up technique while playing the kick drum, thus producing loud hits with short rise and long decay times. As seen in Fig. 13, the amplitude envelope increases until the second half oscillation of the diaphragm, when the onset detector was triggered. Drummer B also hit the snare drum heavily, causing the snare to rattle for a longer duration. This reduced the perceived lag of the effect in experiments 9 and 12 (as seen in Sec. 4.3.3).

4.5. Performer Feedback

A survey was conducted for the performance test and participant responses and comments were recorded. Questions were designed to be answered on a 1 to 5 scale and comments were taken for each question.

4.5.1. Understanding the effect

All participants responded that they understood how the effect worked and felt that the effect was very intuitive. One comment stated that effects with shorter attack times were easier to anticipate and accommodate for while performing. Some performers reported latency with the song Closer which was disorienting and made it difficult to use the effect (see Sec. 4.3.3).

4.5.2. Ease of use

All drummers believed that the effect did not make their performances easier. This can be attributed to the fact that the drummers are constantly concerned with listening to the output from the keyboards to verify if they are playing correctly to trigger the effect as the keyboardist desires. Another factor may be that they were asked to replicate certain effects as they were in a song. So, they were constantly evaluating whether they are performing as expected for the test.

In all cases except the song Closer, all the keyboardists believed that the effect made their performance a lot easier.

4.5.3. Impact on performance

All performers responded that they had to change their performance technique mildly or moderately to make the plugin work. Drummers also noted that with the kick drum, using dynamics to control the effect was very difficult. Thus, when using the effect, they could not change the kick pattern, and potentially lost some of the groove in a song.

The drummers responded that they had to change their groove for the songs Closer and All We Know, while the keyboardists responded that they had to change their performance only for Closer.

4.5.4. Creative possibilities

We received a mixed response when we asked musicians about the possibility to improvise while using the effect. Some performers who could use the effect comfortably believed there was room for improvisation, especially for the drummer to perform a solo while the keyboardist is passive. Others felt that using the effect successfully required musicians to know in advance what the other musician is playing, thus making improvisation difficult.

Most participants said that the effect is well suited for electronic music genres. One comment stated that it could be applied to a broad variety of genres if used creatively.

Some participants believed that this effect might restrict the way musicians compose their music. One participant noted that this need not be true since the effect is a performance tool rather than a composition tool, and might open up interesting opportunities to bands that may not have included keyboards otherwise.

4.5.5. Musical Expression

One drummer responded that the effect opens new dimensions of expression for a drummer but it also gives the drummer more responsibilities. Another drummer noted that constraints on the ability to change the kick/snare pattern in the groove is limiting. And being in control of effects across multiple instruments is interesting but also frustrating for the other instrumentalists.

One keyboardist responded that the effect is more useable when the song is rhythmic, enhancing tightness while simplifying the keyboardist's job. Another responded that it is not possible for a keyboardist to express oneself using the effect, but the drummer has more freedom and ability to express and make the song richer.

5. LATENCY ANALYSIS

The plugin has very little latency (<2.6 ms). For the live implementation, the latency of the sound card, driver and the buffer size used determine the overall latency of the effect. Fig. 14 shows the time difference between the onset of the gated drum note and the onset of the effect.

Offline rendering gives a latency of 0.72 ms (32 samples) from the peak of the transient (can be seen in Fig. 14 by comparing the Drum and Filter Synth plot at Time = 0), which is the minimum latency due to a 32-sample frame length and detection commencing at the peak of the onset transient.

During live testing, a time difference of <12.3 ms (~550 samples) was observed from the beginning of the transient (excitation due to drum hit) and ~2.5 ms (~110 samples) from the peak (onset). This is because physical drums were observed to have an attack transient of 4-10ms and the sound card used with ASIO driver at 256 samples buffer size has a latency of 2.6 ms. Except the experiments related to song 3, none of the performers reported issues due to latency as the effect latency was <10 ms which shows no significant difference in responsiveness of an instrument [9].

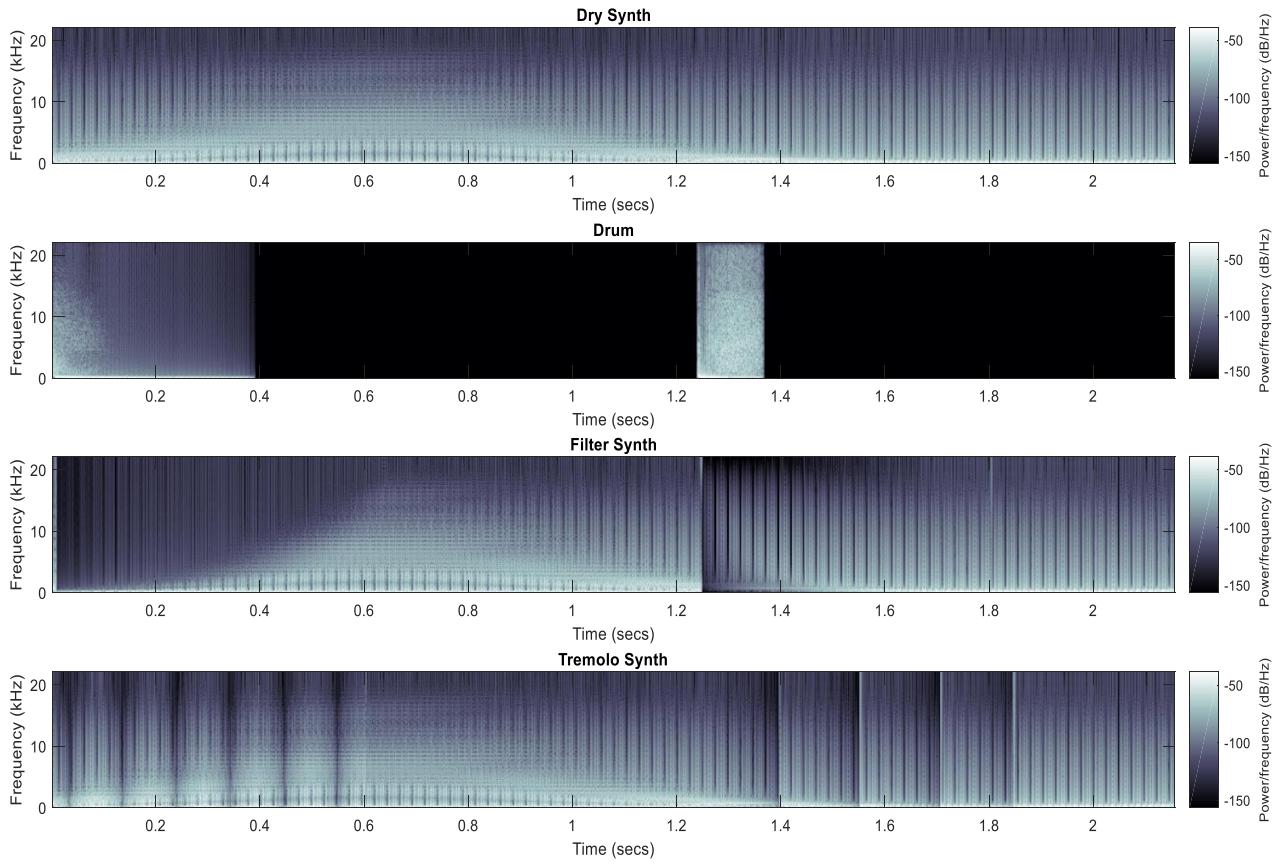


Figure 14: Temporal view of offline test spectrogram.

6. RESULTS

Performance tests showed that the effect works better for LFO synchronization and fast attacks. Effects with longer attack times do not provide immediate feedback to the performers, thus giving a sense of lag. This longer attack time setting was a larger contributor to lag than any signal processing latency. Fig. 15 compares the time taken to achieve final recording for each of the experiments. D-A K-B implies the experiment conducted with Drummer A and Keyboardist B. The time required for performers to achieve desired performance while using the effect was consistently shorter than when performing with manual effects. This might be biased for the selected songs and tasks since the effects in the song were difficult to replicate manually and the effect was particularly useful for the given situations.

Sound samples from the performance tests are available at:
<https://code.soundsoftware.ac.uk/attachments/download/2232/Performance%20Examples.rar>

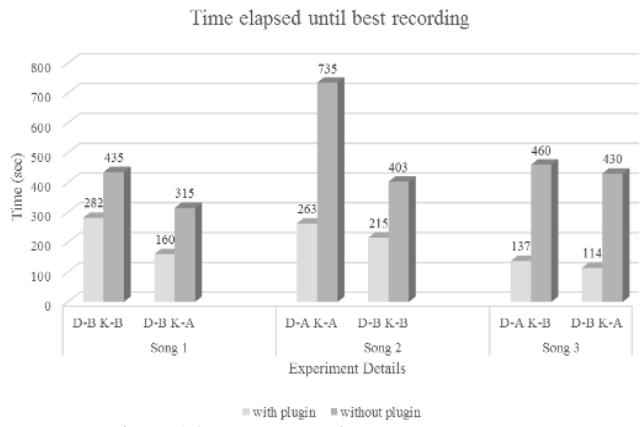


Figure 15: Experiment duration comparison.

7. CONCLUSION

We investigated whether the kind of effects created in the studio for popular music using modern digital audio tools can potentially be recreated live using cross-adaptive architectures. We implemented a cross-adaptive audio effect for live performance that enabled the drummer to orchestrate effects across instruments using drum cues. The idea to have effects synchronised to the drum cues is very intuitive and our experiments have reflected the same. Results showed that the cross-adaptive architecture was successful for achieving tasks in several scenarios based on application of effects in post-production, although such limited evaluation may not uncover the most significant challenges in their use.

The use of cross-adaptive effects not only has a drastic impact on the performance of musicians but also potentially affects the way music is composed when using such effects. Better feature recognition techniques to distinguish higher level drum cues like grooves and rolls would enhance the functionality of these effects, allowing them to be less intrusive and more powerful.

8. REFERENCES

- [1] V. Verfaille, U. Zolzer, and D. 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] E. Perez Gonzalez and J. D. Reiss, Automatic Mixing, *DAFx: Digital Audio Effects*, Second Edition (ed U. Zölzer), John Wiley & Sons, Ltd, Ch. 13, p. 523–550.
- [3] S. Hafezi and J. D. Reiss, ‘Autonomous multitrack equalisation based on masking reduction,’ *Journal of the Audio Engineering Society*, Vol. 63 No. 5, May 2015.
- [4] Z. Ma et al., ‘Intelligent multitrack dynamic range compression,’ *Journal of the Audio Engineering Society*, Vol. 63 No.6, June 2015.
- [5] O. Brandtsegg, ‘A Toolkit for Experimentation with Signal Interaction’, *18th Int. Conference on Digital Audio Effects (DAFx-15)*, Trondheim, Norway 2015.
- [6] M. Wright, A. Freed, and A. Momeni, ‘Opensound control: State of the art 2003,’ *New interfaces for musical expression (NIME)*, pp. 153–160, Singapore 2003.
- [7] O. Campbell et. al., ‘ADEPT: A framework for adaptive digital audio effects’, *2nd AES Workshop on Intelligent Music Production (WIMP)*, London 2016.
- [8] I. Jordal, O. Brandtsegg, G. Tufte, ‘Evolving neural networks for cross-adaptive audio effects’, *2nd AES Workshop on Intelligent Music Production (WIMP)*, London 2016.
- [9] R. H. Jack, A. McPherson, T. Stockman, ‘Effect of latency on performer interaction and subjective quality assessment of a digital musical instrument’ *Proc. Audio Mostly*, Norrköping, Sweden 2016.

REAL-TIME PITCH TRACKING IN AUDIO SIGNALS WITH THE EXTENDED COMPLEX KALMAN FILTER

Orchisama Das, Julius O. Smith III, Chris Chafe

Center for Computer Research in Music and Acoustics,
Stanford University
Stanford, USA
[orchi | jos | cc]@ccrma.stanford.edu

ABSTRACT

The Kalman filter is a well-known tool used extensively in robotics, navigation, speech enhancement and finance. In this paper, we propose a novel pitch follower based on the Extended Complex Kalman Filter (ECKF). An advantage of this pitch follower is that it operates on a sample-by-sample basis, unlike other block-based algorithms that are most commonly used in pitch estimation. Thus, it estimates sample-synchronous fundamental frequency (assumed to be the perceived pitch), which makes it ideal for real-time implementation. Simultaneously, the ECKF also tracks the amplitude envelope of the input audio signal. Finally, we test our ECKF pitch detector on a number of cello and double bass recordings played with various ornaments, such as vibrato, portamento and trill, and compare its result with the well-known YIN estimator, to conclude the effectiveness of our algorithm.

1. INTRODUCTION

Pitch detection in music and speech has been an active area of study. In [1], Gerhard gives a history of pitch recognition techniques. He also establishes in the importance of pitch in carrying much of the semantic information in tonal languages, which makes it useful in the context of speech recognition. In music, its obvious application is in automatic transcription. It is also to be noted that pitch is a perceptual feature [2], whereas most pitch-detectors detect fundamental frequency, which corresponds to perceived pitch for periodic signals.

Algorithms for pitch detection can be classified into three broad categories – *time domain methods*, *frequency domain methods* and *statistical methods*. Time domain methods based on the zero-crossing rate and autocorrelation function are particularly popular. The best example of this is perhaps the YIN [3] estimator, which makes use of a modified autocorrelation function to accurately detect periodicity in signals. Among frequency domain methods, the best known techniques are cepstrum [4], harmonic product spectrum [5] and an optimum comb filter algorithm [6]. Statistical methods include the maximum likelihood pitch estimator [5, 7], more recent neural networks [8] and hidden Markov models [9].

In [10] Cuadra et al. discuss the performance of various pitch detection algorithms in real-time interactive music. They establish the fact that although monophonic pitch detection seems like a well-researched problem with little scope for improvement, that is not true in real-time applications. Some of the most common issues in real time pitch tracking are optimization, latency and accuracy in noisy conditions. The ECKF pitch detector proposed in this paper can be easily implemented on hardware with less computational power, has a maximum latency of 20 ms (a latency of

30–40 ms is tolerable) and has excellent performance in presence of a high amount of noise. It also yields pitch estimates on a fine-grained, sample-by-sample basis, resulting in very accurate pitch tracking. Instruments like the cello and flute which have strong harmonics, pose an additional challenge in pitch detection, which makes testing on cello recordings a reasonable way to check the performance of our algorithm.

The Kalman filter [11] has several applications in power system frequency measurements, and one such brilliant application inspired this work. For example, in [12], the extended Kalman filter was used to track the harmonics of the 60Hz power signal. Several models of the extended Kalman filter exist for tracking the fundamental frequency in power signals, but the one we use here was proposed by Dash et al. [13] The extended complex Kalman filter (ECKF) developed here is ideal for use in real-time, with a high tolerance for noise. The complex multiplications can be carried out on a floating point processor. The ECKF simultaneously tracks fundamental frequency, amplitude and pitch, in presence of harmonics and noise. The assumption is that the strength of the harmonics is less than that of the fundamental. Of course, several modifications need to be made before applying it to audio signals, in which correct pitch detection has to happen within milliseconds and which can have large variations in pitch in a short amount of time.

The rest of this paper is organized as follows – in Section 2 we give details of the model used for ECKF pitch tracking. In Section 3, details of its implementation are given, including calculation of initial estimates for attaining steady state values quickly, resetting the error covariance matrix based on silent frame detection, and an adaptive process noise variance based on the measurement residual. In Section 4, we give the results of testing our algorithm on audio data. In Section 4.1, we note some limitations of our algorithm and delineate scope for improvement. Finally we conclude the paper in Section 5, and talk about the scope for future work.

2. ECKF MODEL AND EQUATIONS

We make use of the sines+noise model for music [14] (that matches the model used in [13]) to derive our state space equations. The non-linear nature of the model calls for an extended Kalman filter [15], which linearizes the function about the current estimate by using its Taylor series expansion. Only the first order term is kept in the Taylor series expansion, and higher order terms are ignored. In vector calculus, the first derivative of a function is found by computing its *Jacobian*.

Let there be an observation y_k at time instant k , which is a sum

of additive sines and a residual noise.

$$y_k = \sum_{i=1}^N a_i \cos(\omega_i t_k + \phi_i) + v_k \quad (1)$$

where a_i , ω_i and ϕ_i are the amplitude, frequency and phase of the i th sinusoid and v_k is a normally distributed Gaussian noise $v \sim N(0, \sigma_v^2)$. Of course in music signals, the residual is never precisely a Gaussian white noise but we make that assumption for this model. σ_v^2 is known as the measurement noise variance. We also assume that the fundamental is considerably stronger than the partials, and 1 reduces to

$$y_k = a_1 \cos(\omega_1 k T_s + \phi_1) + v_k \quad (2)$$

where a_1 , ω_1 and ϕ_1 are the fundamental amplitude, frequency and phase respectively and T_s is the sampling interval. The state vector is constructed as

$$x_k = \begin{bmatrix} \alpha \\ u_k \\ u_k^* \end{bmatrix} \quad (3)$$

where

$$\begin{aligned} \alpha &= \exp(j\omega_1 T_s) \\ u_k &= a_1 \exp(j\omega_1 k T_s + j\phi_1) \\ u_k^* &= a_1 \exp(-j\omega_1 k T_s - j\phi_1) \end{aligned} \quad (4)$$

This particular selection of state vector ensures that we can track all three parameters that defines the fundamental – frequency, amplitude and phase. The relative advantage of choosing this complex state vector has been described in [13]. The state vector estimate update rule x_{k+1} relates to x_k as

$$\begin{aligned} \begin{bmatrix} \alpha \\ u_{k+1} \\ u_{k+1}^* \end{bmatrix} &= \begin{bmatrix} 1 & 0 & 0 \\ 0 & \alpha & 0 \\ 0 & 0 & \frac{1}{\alpha} \end{bmatrix} \begin{bmatrix} \alpha \\ u_k \\ u_k^* \end{bmatrix} \\ x_{k+1} &= f(x_k) \\ f(x_k) &= \left[\alpha \quad \alpha u_k \quad \frac{u_k^*}{\alpha} \right]^T \end{aligned} \quad (5)$$

y_k relates to x_k as

$$\begin{aligned} y_k &= H x_k + v_k \\ H &= [0 \quad 0.5 \quad 0.5] \end{aligned} \quad (6)$$

where H is the observation matrix. We can see that

$$\begin{aligned} H x_k &= \frac{a_1}{2} [\exp(j\omega_1 k T_s + j\phi_1) + \exp(-j\omega_1 k T_s - j\phi_1)] \\ &= a_1 \cos(\omega_1 k T_s + \phi_1) \end{aligned} \quad (7)$$

2.1. Kalman Filter equations

The recursive Kalman filter equations aim to minimize the trace of the error covariance matrix. The EKF equations are given as follows

$$K_k = \hat{P}_{k|k-1} H^{*T} [H \hat{P}_{k|k-1} H^{*T} + 1]^{-1} \quad (8)$$

$$\hat{x}_{k|k} = \hat{x}_{k|k-1} + K_k (y_k - H \hat{x}_{k|k-1}) \quad (9)$$

$$\hat{x}_{k+1|k} = f(\hat{x}_{k|k}) \quad (10)$$

$$\hat{P}_{k|k} = (I - K_k H) \hat{P}_{k|k-1} \quad (11)$$

$$\hat{P}_{k|k+1} = F_k \hat{P}_{k|k} F_k^{*T} + \sigma_w^2 I \quad (12)$$

where F_k is the Jacobian given by

$$F_k = \frac{\partial f(x_k)}{\partial x_k} \Big|_{x_k=\hat{x}_{k|k}} = \begin{bmatrix} 1 & 0 & 0 \\ \hat{x}_{k|k}(2) & \hat{x}_{k|k}(1) & 0 \\ -\frac{\hat{x}_{k|k}(3)}{\hat{x}_{k|k}(1)^2} & 0 & \frac{1}{\hat{x}_{k|k}(1)} \end{bmatrix} \quad (13)$$

- $\hat{x}_{k|k-1}$, $\hat{x}_{k|k}$, $\hat{x}_{k|k+1}$ are the *a priori*, current and *a posteriori* state vector estimates respectively.
- $\hat{P}_{k|k-1}$, $\hat{P}_{k|k}$, $\hat{P}_{k|k+1}$ are the *a priori*, current and *a posteriori* error covariance matrices respectively.
- K_k is the Kalman gain that acts as a weighting factor between the observation y_k and *a priori* prediction $\hat{x}_{k|k-1}$ in determining the current estimate.
- σ_w^2 is modeled as the process noise variance and I is an identity matrix of dimensions 3×3 .
- Initial state vector and error covariance matrix are denoted as $\hat{x}_{1|0}$ and $\hat{P}_{1|0}$ respectively

The fundamental frequency, amplitude and phase estimates at instant k are given as

$$\begin{aligned} f_{1,k} &= \frac{\ln(\hat{x}_{k|k}(1))}{2\pi j T_s} \\ a_{1,k} &= \sqrt{\hat{x}_{k|k}(2) \times \hat{x}_{k|k}(3)} \\ \phi_{1,k} &= \frac{1}{2j} \ln \left(\frac{\hat{x}_{k|k}(2)}{\hat{x}_{k|k}(3)\hat{x}_{k|k}(1)^{2k}} \right) \end{aligned} \quad (14)$$

The state vector multiplied with the observation matrix essentially gives the low passed observation signal with the cutoff frequency of the LPF approximately at the signal's fundamental frequency.

3. IMPLEMENTATION DETAILS

For best performance, our proposed ECKF pitch estimator needs to be modified in many ways. This includes keeping track of silent regions in the signal, resetting the error covariance matrix whenever there is a transition from silence to transient, giving it correct initial estimates for a low settling time and calculating an adaptive process noise variance.

3.1. Detection of Silent Zones

It is important to keep track of note-off events (unpitched moments) in the signal because the estimated frequency for such *silent* regions should be zero. Moreover, whenever there is a transition from note-off to note-on, the Kalman filter error covariance matrix needs to be reset. This is because the filter quickly converges to a steady state value, and as a result the Kalman gain K_k and error covariance matrix $\hat{P}_{k|k}$ settle to very low values. If there is a sudden change in frequency of the signal (which happens at note onset), the filter will not be able to track it unless the covariance matrix is reset.

To keep track of *silent* regions, we divide the signal into frames of 20 ms. One way to determine if a frame is silent or not is to calculate its energy. The energy of the i th frame, E_i is given as the

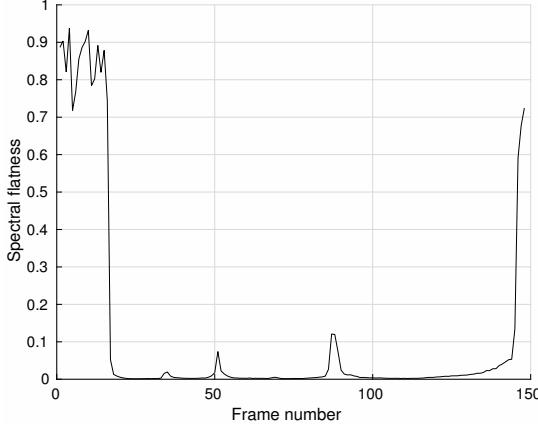


Figure 1: Spectral flatness varying across frames. A high value indicates silent frame.

sum of the square of all the signal samples in that frame. If the energy is below -60dB, then the frame is classified to be silent. If the frame has N samples, then

$$E_i = 20 \log_{10} \sum_{n=0}^{N-1} y(n)^2. \quad (15)$$

However, for noisy input signals, the energy in silent frames is significant. To find silent frames in noisy signals, we make use of the fact that noise has a fairly flat power spectrum. Therefore, for each frame, we calculate the spectral flatness [16]. If spectral flatness ≈ 1 , then the frame is classified to be silent. Figure 1 shows spectral flatness v/s frame number for an audio file containing a single note preceded and followed by some silence.

The power spectral density (PSD) of the observed signal, Φ_{yy} , is given as

$$\Phi_{yy}(e^{j\omega}) = \sum_{n=-\infty}^{\infty} \phi_{yy}(n) e^{-jn\omega} \quad (16)$$

where $\phi_{yy}(n)$ is the autocorrelation function of the input signal y , given as

$$\phi_{yy}(n) = \sum_{m=-\infty}^{\infty} \overline{y(m)} y(n+m). \quad (17)$$

The power spectrum is the DTFT of the autocorrelation function and one way of estimating it is Welch's method [17] which makes use of the periodogram. The power spectral density can be calculated in Matlab with the function `pwelch`. The spectral flatness is defined as the ratio of the geometric mean to the arithmetic mean of the PSD.

$$\text{spf} = \frac{\sqrt[K]{\prod_{k=0}^{K-1} \hat{\Phi}_{yy}(e^{j\omega_k})}}{\frac{1}{K} \sum_{k=0}^{K-1} \hat{\Phi}_{yy}(e^{j\omega_k})} \quad (18)$$

where $\hat{\Phi}_{yy}(e^{j\omega_k})$ is the estimated power spectrum for K frequency bins covering the range $[-\pi, \pi]$.

For white noise $v \sim N(0, \sigma_v^2)$ corrupting the measurement signal, y , the silent frames of y will have pure white noise with the

following properties

$$\begin{aligned} \phi_{yy}(n) &= \sigma_v^2 \delta(n) \\ \Phi_{yy}(e^{j\omega}) &= \sigma_v^2 \forall \omega \in [-\pi, \pi] \\ \text{spf} &= \frac{\sqrt[K]{\sigma_v^{2K}}}{\frac{1}{K} (K \sigma_v^2)} = 1 \end{aligned} \quad (19)$$

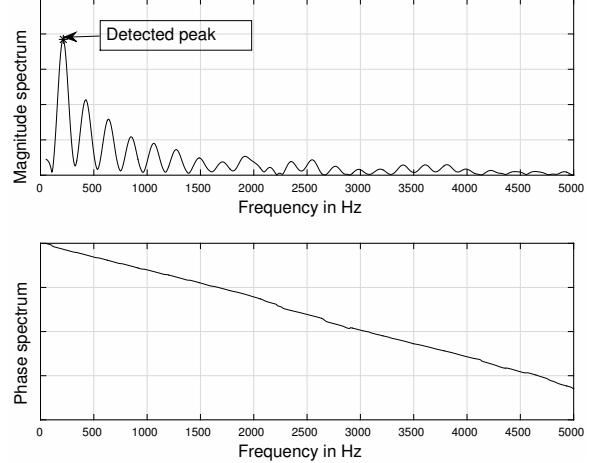


Figure 2: Frequency spectrum used for calculating initial estimates of the state vector.

3.2. Calculating Initial State and Resetting the Error Covariance Matrix

It has already been established that resetting the error covariance matrix is necessary whenever there is a change in the frequency content of the signal (i.e., whenever a new note is played). Along with resetting the covariance matrix, we need to recalculate our initial estimates for the state vector. This is because the rate of convergence of the Kalman filter depends on the accuracy of its initial estimates. Basically, we need to calculate $\hat{x}_{1|0}$ and $\hat{P}_{1|0}$ whenever there is a note onset, i.e., whenever there is a transition from silent frame to non-silent frame.

Depending on how strong the attack of the instrument is, we may need to skip a few audio frames until we reach steady state to accurately estimate initial states. This is because the transient is noisy which makes frequency estimates go haywire, and we must wait for the signal to settle. The number of frames to skip after detecting a transition from silent to non-silent frame can be a user-defined parameter.

To calculate the initial estimates of the state vector $\hat{x}_{1|0}$ we take an FFT of the first non-silent frame following a silent frame, after multiplying it with a *Blackman* window and zero-padding it by a factor of 4. Zero padding increases the FFT size which increases the sampling density along the frequency axis. We calculate the magnitude and frequency of the peaks in the magnitude spectrum, and take $f_{1,0}$ as the minimum of the frequencies corresponding to the largest peaks. $a_{1,0}$ is the magnitude corresponding to $f_{1,0}$ normalized by the mean of the window and number of points in the FFT. $\phi_{1,0}$ is the corresponding phase. Next, we perform parabolic interpolation on $f_{1,0}, a_{1,0}, \phi_{1,0}$ to get more accurate estimates. A typical plot of the spectrum used to calculate initial

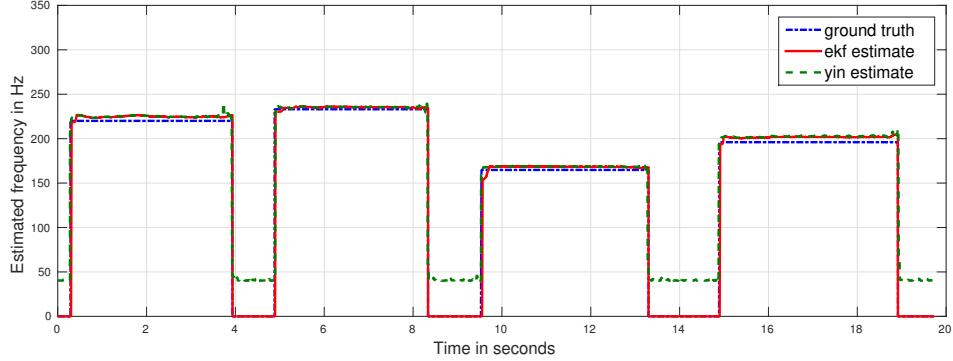


Figure 3: Plot of a) Ground Truth (blue) b) ECKF pitch detector (red) c) YIN estimator (green)

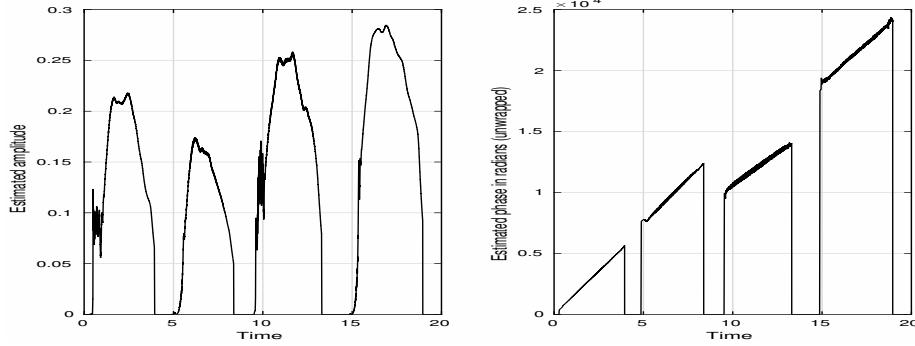


Figure 4: ECKF: Estimated amplitude and unwrapped phase for same input

estimates is given in Figure 2. The initial states are as follows

$$\begin{aligned} \hat{x}_{1|0} &= \mathbb{E}[x_1] \\ &= \begin{bmatrix} e^{(2\pi j f_{1,0} T_s)} \\ a_{1,0} e^{(2\pi j f_{1,0} T_s + j\phi_{1,0})} \\ a_{1,0} e^{(-2\pi j f_{1,0} T_s - j\phi_{1,0})} \end{bmatrix} \\ \hat{P}_{1|0} &= \mathbb{E}[(x_1 - \hat{x}_{1|0})(x_1 - \hat{x}_{1|0})^*]^T \\ &= \mathbf{0}_{(3,3)} \end{aligned} \quad (20)$$

where $\hat{P}_{1|0}$ is a null matrix of order 3×3 .

3.3. Adaptive Process Noise

We only reset the covariance matrix when there is a transition from silent to non-silent frame. However, new notes may be played without any rest in between, and dynamics such as vibrato also need to be captured. To track these changes, an additional term is added to equation 12. σ_w^2 is modeled as the process noise variance given as

$$\log_{10}(\sigma_w^2) = -c + |y_k - H\hat{x}_{k|k}| \quad (21)$$

where $c \in \mathbb{Z}^+$ is a constant.¹ The term $y_k - H\hat{x}_{k|k}$ is known as the *innovation*. It gives the error between our predicted value and the actual data. Whenever the innovation is high, there is a significant discrepancy between the predicted output and the input, which is

¹for this paper, c lies in the range 7–9 but its value can be tuned according to the input

probably caused by a change in the input that the ECKF needs to track. In that case, σ_w^2 increases and there's a term added to the *a posteriori* error covariance matrix $\hat{P}_{k|k+1}$. This increase in the error covariance matrix causes the Kalman gain K_k to increase in the next iteration according to equation 8. As a result, the next state estimate $\hat{x}_{k|k+1}$ depends more on the input, and less on the current predicted state $\hat{x}_{k|k}$. Thus, the innovation reduces in the next iteration, and so does σ_w^2 . In this way, the process noise acts as an error correction term that is adaptive to the variance in input.

4. RESULTS

The ECKF pitch detector was tested on cello notes downloaded from the MIS database, which contains ground truth labels in the form of annotated note names, and on cello and double bass notes played with various ornaments that we recorded ourselves.² The data was summed with white noise normally distributed with 0 mean and 0.01 variance, to test the performance of our algorithm with noisy input.

Figure 3 shows the output when the notes A3-Bb3-E3-G3 were played on the cello one after another with pauses in between. The pitch detected by the ECKF is compared with the ground truth and YIN estimator output. Figure 4 shows the corresponding estimated amplitude and phase plots for the same input given by the ECKF estimator. The mean and standard deviation of absolute er-

²The Matlab implementation can be cloned from <https://github.com/orchidas/Pitch-Tracking>

ror is given in Table 1. A zoomed in plot can be seen in Figure 5 which explains the cause of higher standard deviation of error with the ECKF. Unlike the YIN estimator which yields a single pitch estimate for an entire block of data, the ECKF yields a unique pitch value for every sample of data, and it fluctuates about a mean value. The frequency of these oscillations is approximately equal to the fundamental frequency of the note being played. The oscillations maybe caused due to the resonance of the bridge of the instrument or some artifact introduced by the tracker. However, the amplitude of the oscillations is small, so in reality it is perceptually insignificant, hence we neglect it. It would be interesting to explore the cause of these fluctuations in a future work.

	Mean	Std. Dev
YIN	5.195	9.637
ECKF	5.181	14.778

Table 1: Error statistics for Figure 3

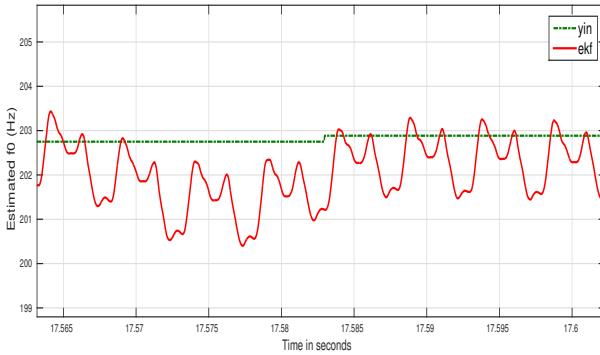
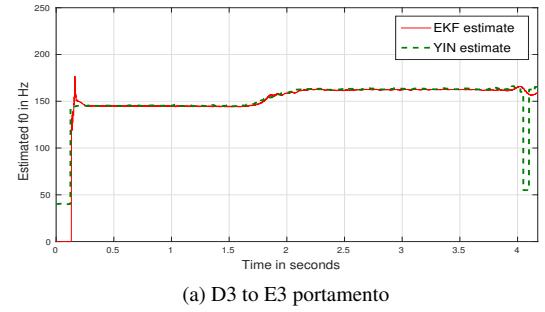


Figure 5: Higher variance of ECKF is caused by rapid fluctuations of estimated pitch about a mean value.

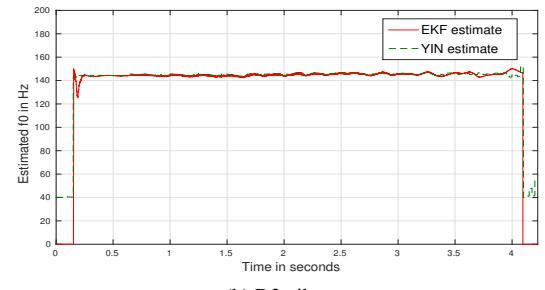
Figure 6a shows the pitch detector output when the input is a portamento played on the note *D3* and glided to *E3*. Figure 6b shows the output when the input is a vibrato on the note *D3*, and Figure 6c shows the output when the input is a vibrato trill on the notes *D3-Eb3*. Figures 6a and 6b were notes played on the cello whereas Figure 6c was a note played on the double bass. All three plots show excellent agreement with what we expect and the output of the YIN estimator. In fact, in Figure 6c the output of the ECKF is much smoother than that of the YIN estimator and shows less drastic fluctuations. Moreover, the YIN estimator gets the pitch wrong in a number of frames where it dips and peaks suddenly. It can be concluded that the ECKF pitch follower is ideal for detecting ornaments and stylistic elements in monophonic music. It could also be used successfully in tracking minute changes in speech formants.

4.1. Limitations

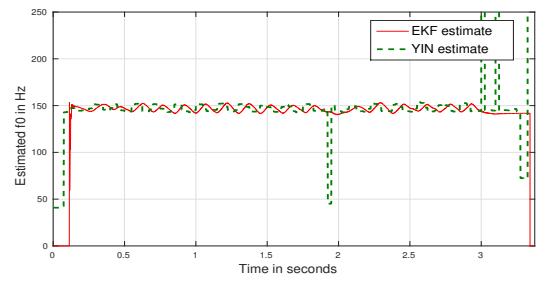
Although our proposed pitch detector performs well in many cases, it has certain drawbacks. Firstly, the additional processing that includes detecting silent frames and estimating initial state makes its implementation more computationally expensive than the one in [13], which is a drawback in real-time processing. This is because estimating the power spectrum with Welch's method requires computing an FFT for each block of data, which is of complexity $\mathcal{O}(N \log_2 N)$. However, depending on the noise environment, a



(a) D3 to E3 portamento



(b) D3 vibrato



(c) D3-Eb3 vibrato trill

Figure 6: ECKF estimated pitch for various ornaments played on the cello and double bass

cheaper algorithm can be used to distinguish between non-silent and silent frames. Calculating initial estimates also requires computing an FFT but we only need to do that whenever there is a transition from silent to non-silent frame, not for every block of data.

Secondly, if the initial states estimated from the FFT are off by 20 Hz or more, then the filter is slow to converge to its steady state value. In Figure 6a, it is observed that the ECKF gets the pitch wrong during the transient, but that is expected since there is no well-defined pitch during the transient. To avoid getting a spurious value of pitch, the method of skipping a few buffers to wait until the steady state can be used as described in Section 3.2. Perhaps the ECKF pitch tracker's biggest limitation is tracking notes played quickly together without any pause, as demonstrated in Figure 7. ECKF is slow in tracking very rapid and large changes. The faster the notes are played, the higher the latency in convergence. A solution to this could be to observe note onsets and estimate initial state and reset the covariance matrix whenever there is an onset

detected. However, we leave this problem open for future work.

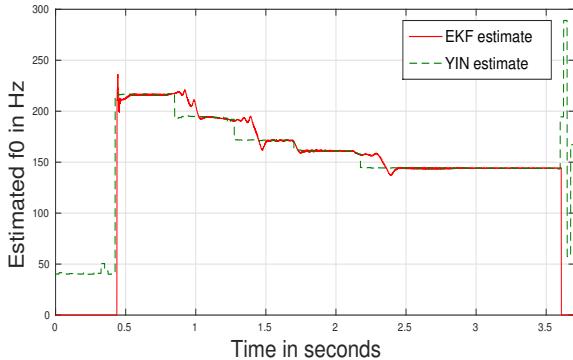


Figure 7: ECKF is slow in tracking fast note changes. Notes played are descending fifths from the A string on the double bass.

5. CONCLUSION

In this paper, we have proposed a novel real-time pitch detector based on the Extended Complex Kalman Filter (ECKF). Several adjustments have been made for optimum tracking. An algorithm based on spectral flatness has been proposed to detect silence in incoming noisy audio signal. The importance of accurate initial state estimates and resetting the error covariance matrix has been explained. A correction factor, σ_w^2 , has been included to track fast, small changes in input signal.

After all these changes have been incorporated into the ECKF pitch detector, the results match those of robust and successful pitch detectors like the YIN estimator. Perhaps its greatest advantage is the fact that it estimates pitch on a sample-by-sample basis, against most methods which are block-based. Moreover, its performance is robust to the presence of noise in the measurement signal. The ECKF has been observed to be well suited for tracking fine changes in playing dynamics and ornamentation, which makes it an excellent candidate for real-time transcription of solo instrument music. Additionally, the ECKF also yields the amplitude envelope and phase of the signal, along with its fundamental frequency.

5.1. Future Work

We hope this work will encourage new uses of the Kalman filter in audio and music. In recent years, the Kalman filter has been used in music for online beat tracking [18] and partial tracking [19, 20]. It has also been used for frequency tracking in speech [21]. Since the Kalman filter is such a powerful tool that can work for any valid model with the right state-space equations, we believe it can have many more potential real-time applications in music. Some of the other topics we wish to explore include partial tracking and real-time onset detection with the Kalman filter. A real-time pitch detector along with an onset detector lays the groundwork for real-time transcription, which remains an exciting and advanced problem.

6. REFERENCES

- [1] David Gerhard, “Pitch extraction and fundamental frequency : History and current techniques,” Tech. Rep., University of Regina, 2003.
- [2] Joseph Carl Robnett Licklider, “A duplex theory of pitch perception,” *The Journal of the Acoustical Society of America*, vol. 23, no. 1, pp. 147–147, 1951.
- [3] 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.
- [4] A. Michael Noll, “Cepstrum pitch detection,” *The Journal of the Acoustical Society of America*, vol. 41, pp. 293–309, 1967.
- [5] A. Michael Noll, “Pitch determination of human speech by the harmonic product spectrum, the harmonic sum spectrum, and a maximum likelihood estimate,” in *Proceedings of the symposium on computer processing communications*, 1969, vol. 779.
- [6] James A Moorer, “On the transcription of musical sound by computer,” *Computer Music Journal*, pp. 32–38, 1977.
- [7] J Wise, J Caprio, and T Parks, “Maximum likelihood pitch estimation,” *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 24, no. 5, pp. 418–423, 1976.
- [8] Etienne Barnard, Ronald A Cole, Mathew P Vea, and Fileno A Alleva, “Pitch detection with a neural-net classifier,” *IEEE Transactions on Signal Processing*, vol. 39, no. 2, pp. 298–307, 1991.
- [9] F Bach and M Jordan, “Discriminative training of hidden markov models for multiple pitch tracking,” in *Proceedings of International Conference on Acoustics, Speech and Signal Processing*. IEEE, 2005, vol. 5.
- [10] Patricio De La Cuadra, Aaron S Master, and Craig Sapp, “Efficient pitch detection techniques for interactive music.,” in *ICMC*, 2001.
- [11] Rudolph E. Kalman et al., “A new approach to linear filtering and prediction problems,” *Journal of Basic Engineering*, vol. 82, no. 1, pp. 35–45, 1960.
- [12] Adly A Girgis, W Bin Chang, and Elham B Makram, “A digital recursive measurement scheme for online tracking of power system harmonics,” *IEEE transactions on Power Delivery*, vol. 6, no. 3, pp. 1153–1160, 1991.
- [13] Pradipta Kishore Dash, G Panda, AK Pradhan, Aurobinda Routray, and B Duttagupta, “An extended complex kalman filter for frequency measurement of distorted signals,” in *Power Engineering Society Winter Meeting, 2000. IEEE*. IEEE, 2000, vol. 3, pp. 1569–1574.
- [14] Xavier Serra and Julius 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.
- [15] Gabriel A Terejanu, “Extended kalman filter tutorial,” Tech. Rep., University of Buffalo, 2008.
- [16] A Gray and J Markel, “A spectral-flatness measure for studying the autocorrelation method of linear prediction of speech analysis,” *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 22, no. 3, pp. 207–217, 1974.

- [17] Peter Welch, “The use of fast fourier transform for the estimation of power spectra: a method based on time averaging over short, modified periodograms,” *IEEE Transactions on Audio and Electroacoustics*, vol. 15, no. 2, pp. 70–73, 1967.
- [18] Yu Shiu, Namgook Cho, Pei-Chen Chang, and C-C Jay Kuo, “Robust on-line beat tracking with kalman filtering and probabilistic data association (kf-pda),” *IEEE transactions on consumer electronics*, vol. 54, no. 3, 2008.
- [19] Andrew Sterian and Gregory H Wakefield, “Model-based approach to partial tracking for musical transcription,” in *SPIE’s International Symposium on Optical Science, Engineering, and Instrumentation*. International Society for Optics and Photonics, 1998, pp. 171–182.
- [20] Hamid Satar-Boroujeni and Bahram Shafai, “Peak extraction and partial tracking of music signals using kalman filtering.,” in *ICMC*, 2005.
- [21] Özgül Salor, Mübellel Demirekler, and Umut Orguner, “Kalman filter approach for pitch determination of speech signals,” *SPECOM*, 2006.

EFFICIENT ANTI-ALIASING OF A COMPLEX POLYGONAL OSCILLATOR

Christoph Hohnerlein

Quality & Usability Lab,
Technische Universität Berlin
Berlin, Germany
mail@chohner.com

Maximilian Rest

E-RM Erfindungsbüro
Berlin, Germany
m.rest@e-rm.de

Julian D. Parker

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

ABSTRACT

Digital oscillators with discontinuities in their time domain signal derivative suffer from an increased noise floor due to the unbound spectrum generated by these discontinuities. Common anti-aliasing schemes that aim to suppress the unwanted fold-back of higher frequencies can become computationally expensive, as they often involve repeated sample rate manipulation and filtering.

In this paper, the authors present an effective approach to applying the four-point polyBLAMP method to the continuous order polygonal oscillator by deriving a closed form expression for the derivative jumps which is only valid at the discontinuities. Compared to the traditional oversampling approach, the resulting SNR improvements of 20 dB correspond to 2–4× oversampling at 25× lower computational complexity, all while offering a higher suppression of aliasing artifacts in the audible range.

1. INTRODUCTION

A novel complex oscillator algorithm was recently proposed [1], which generates waveforms by traversing a two-dimensional polygon over time. Such a polygon may contain any number of vertices, corresponding to a desired order $n > 2$, which expresses the vertices per rotation. The term complex conveniently combines both the internal dependence on the complex plane as well as the resulting complex spectral behaviour. A projection of the path around a shape in the complex plane can be interpreted as a time-domain signal, with the rotational speed corresponding to the fundamental pitch. Such a signal will naturally contain a number of discontinuities in its derivative. These discontinuities produce an unbounded spectrum and therefore introduce aliasing artifacts into the signal. In order to produce high-quality audio output, this aliasing should be minimized.

Anti-aliasing of digital oscillator algorithms is a well developed topic in literature, but to-date was focused primarily on the generation of classical waveforms or on wavetable synthesis. Research into the anti-aliasing of classical analog waveforms began with the invention of the Band Limited Impulse Train (BLIT) method [2], which generates all waveforms by integrating an underlying sequence of bandlimited impulses. The next major advancement was the invention of the Band Limited stEP (BLEP), and derived Band Limited rAMP (BLAMP) methods [3], which can be applied to the anti-aliasing of discontinuities (of any order) in any type of waveform, as long as the position and magnitude of the discontinuity is known. Further research has concentrated on more efficient polynomial approximations of the ideal BLEP, known as polyBLEP [4, 5, 6, 7].

A parallel stream of research has investigated techniques based on pre-integration of the waveform to be generated, followed by digital differentiation. These techniques have been applied to classical waveforms [8, 9] and to wavetable synthesis [10, 11]. Work

has also explored techniques that are a hybrid of these two streams [12, 13]. More recently, both approaches to anti-aliasing have been generalized to apply to the processing of arbitrary input signals with a nonlinear waveshaping function [14, 15, 16, 17, 18].

In the following Section 2, the implementation of the polygon-based oscillator is layed out, which is then anti-aliased in Section 3. The results in terms of SNR and performance are discussed in Section 4, followed by a short summary in Section 5.

2. POLYGON OSCILLATOR

The following is a quick recap of the continuous order polygon waveform synthesis presented in [1]. To create the polygon P of order n , where n denotes the number of vertices after one rotation of the sampling phasor, a corresponding radial amplitude $p(\varphi)$ is generated:

$$p(\varphi) = \frac{\cos\left(\frac{\pi}{n}\right)}{\cos\left[\frac{2\pi}{n} \cdot \text{mod}\left(\frac{\varphi n}{2\pi}, 1\right) - \frac{\pi}{n}\right]}, \quad (1)$$

where φ is a linearly incrementing phase whose slope depends on the desired pitch f_0 . The amplitude $p(\varphi)$ can then used to scale a unit circle, resulting in the polygon P in the complex plane:

$$P(\varphi) = p(\varphi) \cdot e^{j\varphi} \quad (2)$$

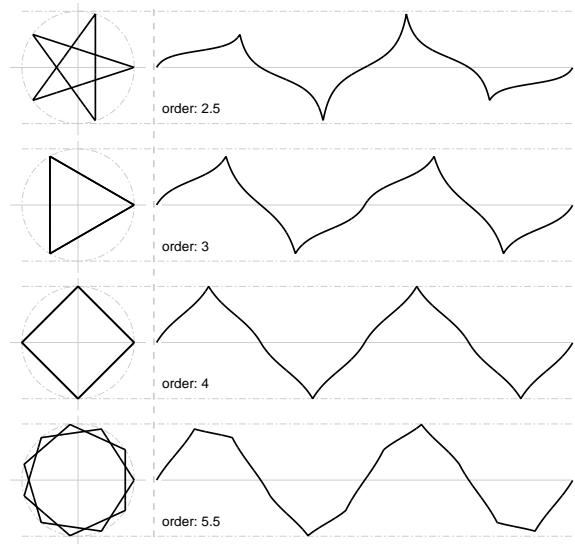


Figure 1: Projections of polygons $P(\phi)$ of different orders n from the 2D space (left) radially sampled into the time domain (right).

Real and imaginary projections x, y then form a 90° phase shifted quadrature output which can be interpreted as oscillator signals in the time domain:

$$f_x(\varphi) = \Re\{P(\varphi)\} = \cos(\varphi) \cdot p(\varphi) \quad (3)$$

$$f_y(\varphi) = \Im\{P(\varphi)\} = \sin(\varphi) \cdot p(\varphi) \quad (4)$$

Several of such polygons $P(\varphi, n)$ and their corresponding time-domain projections $f_x(\varphi)$ are shown in Figure 1. One can see that with increasing order / edge count n , the polygon naturally approaches a unit circle, which corresponds to a pure sine in the time domain when sampled spatially. A more in-depth analysis of the link between the shape of the polygon and the resulting spectrum can be found in [1].

3. ANTI-ALIASING

Anti-aliasing of the polygon oscillator could be achieved via most of the available approaches. However, the BLAMP technique is particularly suited to this problem, as the exact positioning and magnitude of the discontinuities in the derivative of the waveform can be obtained in a very efficient closed-form solution.

To apply a four-point polyBLAMP based on third-order B-spline approximation (as presented in [18]), the jump of the first order derivative has to be evaluated at the points of discontinuities, along with the the fractional delay d between the exact time of the discontinuity and the next sample in discrete, sampled time. Table 1 lists the corresponding polynomials that are to be subtracted from the four samples surrounding the discontinuity.

$[-2T, -T]$	$d^5/120$
$[-T, 0]$	$[-3d^5 + 5d^4 + 10d^3 + 10d^2 + 5d + 1]/120$
$[0, T]$	$[3d^5 - 10d^4 + 40d^3 - 60d + 28]/120$
$[T, 2T]$	$[-d^5 + 5d^4 - 10d^3 + 10d^2 - 5d + 1]/120$

Table 1: Four-point polyBLAMP residual for the four samples surrounding a discontinuity, where d is the fractional delay.

3.1. Fractional delay

In the sampled digital time domain, the samples left and right of the discontinuity can be traced from the continuous form, as the positions of the discontinuities are exactly at $\varphi = 2\pi/n \cdot k, k \in [0, 1, \dots, \infty)$. The fractional delays d are the difference between the ceiled sample time and the exact time:

$$d = \lceil f_s/(nf_0) \cdot k \rceil - f_s/(nf_0) \cdot k, \quad (5)$$

where $\lceil \cdot \rceil$ denotes the ceiling function, f_0 is the fundamental pitch and f_s is the sampling frequency.

Figure 2 shows the discrete time-domain signal $f(\varphi)$ along with its derivative $f'(\varphi)$ while marking the precise positions and amplitude jumps of the derivative at the discontinuities as \times and their quantized position by \circ .

3.2. Derivative jump at discontinuity

To properly scale the residuum of table 1, we still need to find the jump in amplitude of the derivative at the discontinuities $\hat{f}'(\varphi)$.

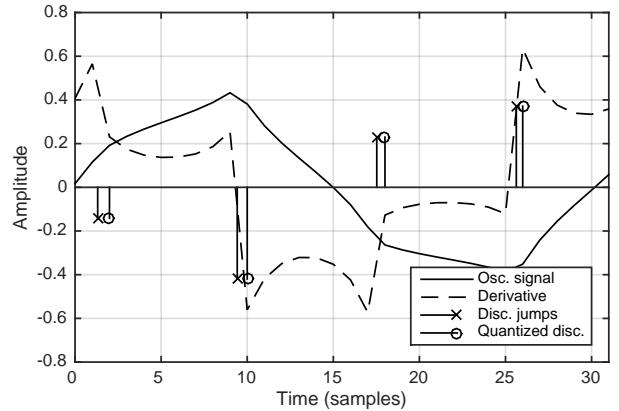


Figure 2: Signal $f(\varphi)$ of order $n = 3.75$ and pitch $f_0 = 1415$ Hz with its derivative $f'(\varphi)$. The discontinuities $\hat{f}'(\varphi)$ are shown at exact and quantized positions.

The closed form derivative of equation (3), $f'_x(\varphi)$ with $a = \frac{\pi}{n}$ is:

$$\frac{df_x(\varphi)}{d\varphi} = \frac{d}{d\varphi} \frac{\cos(\varphi) \cos(a)}{\cos[\mod(\varphi, 2a) - a]}, \quad (6)$$

Treating $\varphi_m = \mod(\varphi, 2a)$ as a special case of φ , which needs to be differentiated but marked, yields:

$$f'_x(\varphi) = \cos(a) \frac{\cos(\varphi) \sin(\varphi_m - a) - \sin(\varphi) \cos(\varphi_m - a)}{\cos(\varphi_m - a)^2} \quad (7)$$

$$f'_y(\varphi) = \cos(a) \frac{\cos(\varphi) \cos(\varphi_m - a) - \sin(\varphi) \sin(\varphi_m - a)}{\cos(\varphi_m - a)^2} \quad (8)$$

3.3. Efficient implementation

We are only interested in the change in amplitude of the derivative at the discontinuities \hat{f}' , which happens when $\varphi_m \in [0 \dots 2a)$ wraps around. Looking from both sides ($\varphi_{m\downarrow} = 0$ and $\varphi_{m\uparrow} = 2a$) yields:

$$\lim_{\varphi_m \downarrow 0} \hat{f}'_x(\varphi) = \hat{f}'_\downarrow(\varphi) = \frac{-\sin(\varphi) \cos(a) + \sin(a) \cos(\varphi)}{\cos(a)} \quad (9)$$

$$\lim_{\varphi_m \uparrow 2a} \hat{f}'_x(\varphi) = \hat{f}'_\uparrow(\varphi) = \frac{-\sin(\varphi) \cos(a) - \sin(a) \cos(\varphi)}{\cos(a)} \quad (10)$$

This leaves us with a simple expression for the change in amplitude at discontinuities:

$$\hat{f}'(\varphi) = \hat{f}'_\downarrow(\varphi) - \hat{f}'_\uparrow(\varphi) = -2 \tan(a) \cos(\varphi) \quad (11)$$

Using Equation (11), we can now correctly scale the residuum of Table 1. Figure 3 shows the original function and its anti-aliased version as well as their difference, with is zero everywhere except the 4 samples surrounding a discontinuity.

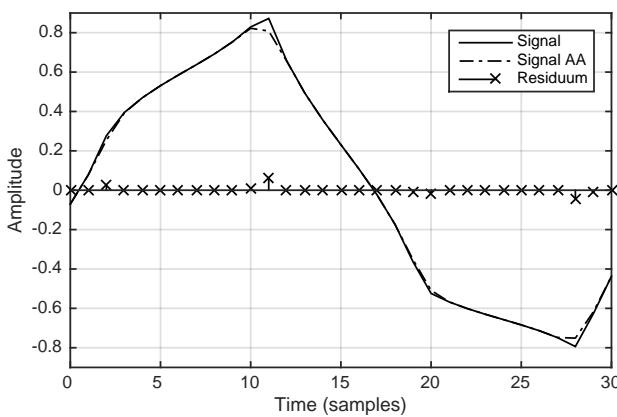


Figure 3: Original and anti-aliased versions of the signal $f(\varphi)$ of order $n = 3.75$ and pitch $f_0 = 1350$ Hz.

4. RESULTS

In the following Section, the presented method is compared to traditional oversampling (2x and 4x) in terms of Signal-to-Noise Ratio (SNR) and computational load.

4.1. Signal-to-Noise Ratio

Sharp discontinuities exhibit unbounded frequency requirements, which results in foldback at $f_{\text{NY}} = f_s/2$ and effectively raises the noise floor of the oscillator. Therefore, the Signal-to-Noise Ratio (SNR), which denotes the ratio between the energy in the fundamental + harmonics and that of the rest of the spectrum is a good metric for measuring and comparing the performance of different anti-aliasing approaches.

Firstly, the harmonic overtones $f_{H,k}$, which depend on order n and the fundamental pitch f_0 , need to be determined. For the continuous order polygonal oscillator, the frequencies of the first K harmonics $f_{H,k}$ can be found to be:

$$f_{H,k}(f_0, n) = f_0 \left(2 \left\lfloor \frac{k}{2} \right\rfloor + 1 + (n-2)(1 + \left\lfloor \frac{k-1}{2} \right\rfloor) \right), \quad (12)$$

where $\lfloor \cdot \rfloor$ denotes the flooring function, n is the order, f_0 the fundamental frequency and $k \in [1 \dots K]$.

The energy of the fundamental and harmonics is extracted directly from the Fourier spectrum, the noise energy is simply the difference of the full and the signal energy:

$$\text{SNR} = \frac{E_{\text{sig}}}{E_{\text{noise}}} = \frac{\sum |f_0 + f_H|^2}{\sum |f_\forall|^2 - \sum |f_0 + f_H|^2} \quad (13)$$

The SNR can then be calculated for both the original signal and various anti-aliased versions. While the measured SNR depends on order n as well as the fundamental frequency f_0 , large improvements of the SNR on the magnitude of 20 dB compared to the original signal were found consistently, as shown in Table 2. It can be seen that the measured improvements of the employed method falls between 2x and 4x oversampling, which was implemented using a 64 / 128 (2x / 4x oversampling) order FIR lowpass

f_0	n	SNR (dB)			
		original	2x OX	4x OX	BLAMP
400 Hz	2.53	57.2	72.5	82	78.7
751 Hz	4.42	66.7	82.6	94	87.8
1350 Hz	3.75	59.4	74	83.6	80.3

Table 2: SNR at different pitches f_0 and orders n of the original signal, 2x oversampling, 4x oversampling and our BLAMP implementation.

filter and Matlab's Polyphase FIR decimator as implemented in the `dsp.FIRDecimator` object.

Figure 4 shows the Fourier spectra of the original (top) and three anti-aliased versions: 2x oversampling (second), 4x oversampling (third) and closed form BLAMPS (bottom), with the overtones of the fundamental marked in each. The chosen settings correspond to Table 2 and are $n = 2.53$, $f_0 = 400$ Hz in Fig. 4a) and $n = 3.75$, $f_0 = 1350$ Hz in Fig. 4b). Although the three anti-aliased versions exhibit slightly different characteristics, the lowered noise floor can be seen clearly in each. In both oversampling cases, a considerable drop in harmonics close to $f_s/2$ can be observed due to the low-pass filtering before decimation, as well as a uniformly suppressed noise floor. On the other hand, the BLAMP implementation exhibits only a small drop in high-frequency harmonics and has a continuously decreasing noise floor.

The stronger aliasing at higher frequencies, although less audible, lowers the overall SNR measurement of the BLAMP approach which uniformly weights all anti-aliasing noise. This is an advantage that is not easy to measure but arguably of large importance - BLAMP suppresses the potentially more disturbing anti-aliasing artifacts (below 10 kHz) stronger compared to other approaches.

4.2. Performance

As shown above, relatively large improvements in SNR can be achieved quite efficiently by precisely analysing the oscillator function at hand. The validity of the derivation is limited to the points at the discontinuities but for a given order, only a single trigonometric function has to be evaluated.

Compared to traditional oversampling, the presented approach falls in between 2x and 4x oversampling in terms of SNR improvement, as shown in Table 2. However, oversampling involves three computationally expensive steps (upsampling - lowpass - downsampling), which on a current machine come with an averaged 25x performance hit when comparing execution times in Matlab.

5. CONCLUSIONS

It was shown that even for non-traditional digital oscillators, elegant and computationally cheap solutions for the polyBLAMP anti-aliasing approach may be found by only considering the necessary points of discontinuity. For an oscillator method that inherently generates high levels of aliasing noise such as the one presented, this can increase the SNR by 20 dB - beating 2x oversampling - at 25x lower computational complexity. The perceptual comparison are even be more favorable, as the polyBLAMP method continuously drives down the noise floor (compared to constant uniform suppression of oversampling methods) which results in lower aliasing artifacts at audible frequencies below 10 kHz.

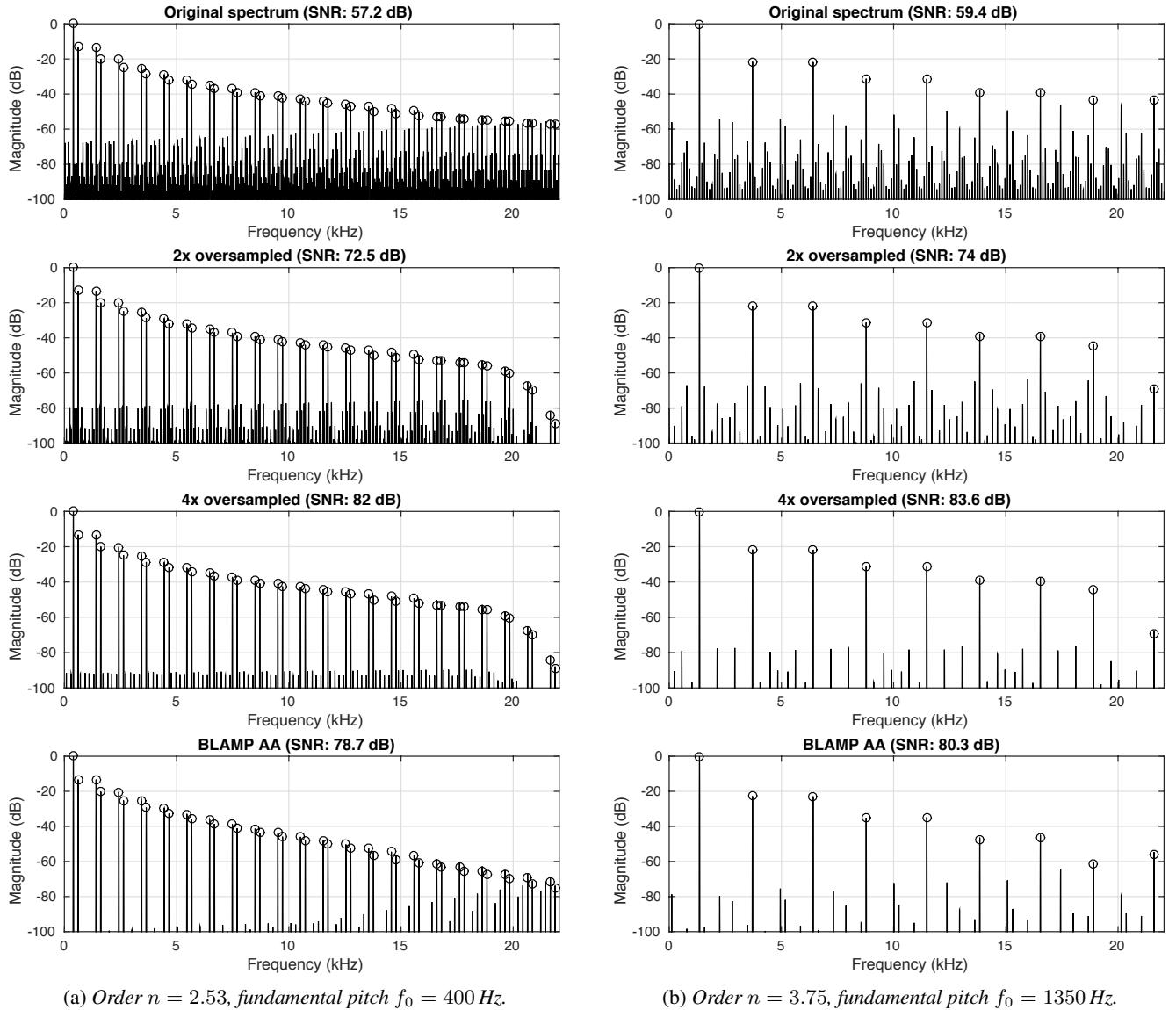


Figure 4: Magnitude spectra of the polygonal oscillator at different settings and different anti-aliasing strategies. Original (top), 2x oversampled (second), 4x oversampled (third) and BLAMP anti-aliased (bottom). Sampling frequency $f_s = 44.1 \text{ kHz}$, fundamental and harmonics used for SNR computation marked with \circ .

6. REFERENCES

- [1] C. Hohnerlein, M. Rest, and J. O. Smith III, “Continuous order polygonal waveform synthesis,” in *Proceedings of the International Computer Music Conference*, Utrecht, Netherlands, Sept. 12–16, 2016, pp. 533–536.
- [2] Tim Stilson and Julius Smith, “Alias-free digital synthesis of classic analog waveforms,” in *Proceedings of the International Computer Music Conference*, Hong Kong, 1996, pp. 332–335.
- [3] E. Brandt, “Hard sync without aliasing,” in *Proc. Int. Computer Music Conf.*, Havana, Cuba, Sept. 2001, pp. 365–368.
- [4] V. Välimäki and A. Huovilainen, “Oscillator and filter algorithms for virtual analog synthesis,” *Computer Music J.*, vol. 30, no. 2, pp. 19–31, 2006.
- [5] V. Välimäki and A. Huovilainen, “Antialiasing oscillators in subtractive synthesis,” *IEEE Signal Process. Mag.*, vol. 24, no. 2, pp. 116–125, 2007.
- [6] J. Pekonen, J. Nam, J. O. Smith, J. Abel, and V. Välimäki, “On minimizing the look-up table size in quasi-bandlimited classical waveform oscillators,” in *Proc. Int. Conf. Digital Audio Effects (DAFx-10)*, Graz, Austria, Sept. 2010, pp. 419–422.
- [7] Vesa Välimäki, Jussi Pekonen, and Juhan Nam, “Perceptually informed synthesis of bandlimited classical waveforms using integrated polynomial interpolation,” *The Journal of the Acoustical Society of America*, vol. 131, no. 1, pp. 974–986, 2012.
- [8] V. Välimäki, “Discrete-time synthesis of the sawtooth waveform with reduced aliasing,” *IEEE Signal Process. Lett.*, vol. 12, no. 3, pp. 214–217, Mar. 2005.
- [9] 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.
- [10] A. Franck and V. Välimäki, “Higher-order integrated wavetable and sampling synthesis,” *J. Audio Eng. Soc.*, vol. 61, no. 9, pp. 624–636, Sept. 2013.
- [11] A. Franck and V. Välimäki, “An ideal integrator for higher-order integrated wavetable synthesis,” in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process. (ICASSP-13)*, Vancouver, BC, Canada, May 2013, pp. 41–45.
- [12] J. Kleimola and V. Välimäki, “Reducing aliasing from synthetic audio signals using polynomial transition regions,” *IEEE Signal Process. Lett.*, vol. 19, no. 2, pp. 67–70, Feb. 2012.
- [13] D. Ambrits and B. Bank, “Improved polynomial transition regions algorithm for alias-suppressed signal synthesis,” in *Proc. 10th Sound and Music Computing Conf. (SMC2013)*, Stockholm, Sweden, Aug. 2013, pp. 561–568.
- [14] J. D. Parker, V. Zavalishin, and E. Le Bivic, “Reducing the aliasing of nonlinear waveshaping using continuous-time convolution,” in *Proc. Int. Conf. Digital Audio Effects (DAFx-16)*, Brno, Czech Republic, Sept. 2016, pp. 137–144.
- [15] S. Bilbao, F. Esqueda, J. Parker, and V. Valimaki, “Antiderivative antialiasing for memoryless nonlinearities,” *IEEE Signal Processing Letters*, vol. PP, no. 99, pp. 1–1, 2017.
- [16] F. Esqueda, V. Välimäki, and S. Bilbao, “Aliasing reduction in soft-clipping algorithms,” in *Proc. European Signal Processing Conf. (EUSIPCO 2015)*, Nice, France, Aug. 2015, pp. 2059–2063.
- [17] F. Esqueda, S. Bilbao, and V. Välimäki, “Aliasing reduction in clipped signals,” *IEEE Trans. Signal Process.*, vol. 60, no. 20, pp. 5255–5267, Oct. 2016.
- [18] F. Esqueda, V. Välimäki, and S. Bilbao, “Rounding corners with BLAMP,” in *Proc. Int. Conf. Digital Audio Effects (DAFx-16)*, Brno, Czech Republic, Sept. 2016, pp. 121–128.

MODELING CIRCUITS WITH OPERATIONAL TRANSCONDUCTANCE AMPLIFIERS USING WAVE DIGITAL FILTERS

Ólafur Bogason

CIRMMT
McGill University
Montreal, Canada
olafur.bogason@mail.mcgill.ca

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

In this paper, we show how to expand the class of audio circuits that can be modeled using Wave Digital Filters (WDFs) to those involving operational transconductance amplifiers (OTAs). Two types of behavioral OTA models are presented and both are shown to be compatible with the WDF approach to circuit modeling. As a case study, an envelope filter guitar effect based around OTAs is modeled using WDFs. The modeling results are shown to be accurate when compared to state of the art circuit simulation methods.

1. INTRODUCTION

A component commonly found in audio circuits is the operational transconductance amplifier (OTA) [1, 2, 3, 4]. Audio gear that contains it includes famous guitar effects pedals [5, 6], voltage controlled amplifiers and oscillators [7, 8, 9]. Despite its wide usage in audio circuits, little research on OTAs in the Virtual Analog context is available [10]. Understanding both idealized and non-idealized models of OTAs and how to apply them in common physical modeling frameworks, such as the state-space model [11, 12] or Wave Digital Filters (WDFs) [13], is paramount if circuits containing them are to be accurately modeled.

WDFs provide an elegant framework for creating digital models of analog reference circuits, or other lumped physical systems [14]. Until recently, the scope of circuits that were tractable using standard WDF techniques was limited to circuits composed of linear components, connected in series and/or parallel and could contain up to one nonlinearity. Although researchers have, in special cases, been able to go beyond these limitations by exploiting topologies of reference circuits [15, 16, 17, 18, 19] or by adding fictitious unit delays to yield computable structures [20, 21], the aforementioned limitations made simulating most audio circuits unfeasible within the WDF framework.

By incorporating Modified Nodal Analysis from Circuit Theory [22], along with grouping nonlinearities into a vector at the root of a WDF tree [16, 23], recent research has provided new theoretical ground which facilitates the modeling of most audio circuits using WDFs. This approach requires that each component in the reference circuit has a suitable Kirchhoff domain model. In this paper we will leverage the recent theoretical advancement to provide two models for the OTA which are suitable for use in WDFs.

This paper is structured as follows: §2 reviews OTAs and presents an ideal model and non-ideal linear macromodel in the Kirchoff domain. §3 discusses recent theoretical advancements in WDF theory and how they have permitted us to derive wave-domain OTA models from Kirchoff-domain models. It further-

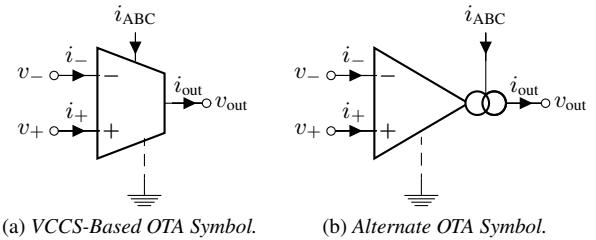


Figure 1: OTA symbols

more outlines how to incorporate the newly derived models to simulate circuits containing OTAs using WDFs. §4 builds upon the derived OTA models and derives WDFs of an envelope filter reference circuit. §§5–6 discusses the accuracy of the proposed model, discusses future work and concludes.

2. OPERATIONAL TRANSCONDUCTANCE AMPLIFIERS

OTAs are active, tunable, high-gain devices, that take differential voltage as input and output current. The external tuning of the gain, termed *transconductance*, make OTAs the perfect building block for audio circuit designers. By modifying the transconductance, OTAs are most commonly used to decouple control from audio circuitry, as is done in the filter of the MS-20 synthesizer [9] or in the envelope filter discussed in §4.

OTAs may be built using bipolar or CMOS transistor technology [4]. CMOS OTAs are widespread in high-frequency applications [24] but are less common in audio circuits where bipolar OTAs are ubiquitous.

On the device level modern OTAs can be quite complex, containing multiple transistors and other nonlinear components in complicated topologies. To simplify circuit design, analysis, and simulation, OTA behavior is often idealized completely or approximated, as is often done with traditional op amps [25]. Such approximations include linear [26] or nonlinear macromodels [27].

The OTA symbol most commonly used by the research community is the VCCS symbol augmented with an additional bias current port as shown in Figure 1a. The OTA symbol widely found in circuit schematics is shown in Figure 1b.

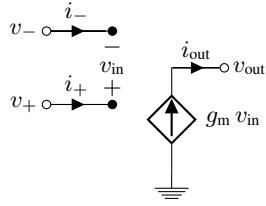


Figure 2: Ideal OTA

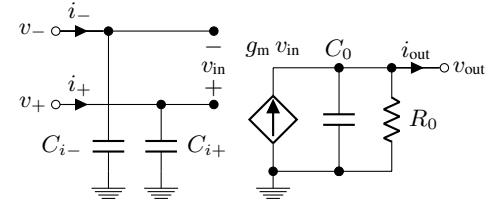


Figure 3: Macromodel OTA

2.1. Ideal VCCS Model of an OTA

An ideal OTA is a voltage dependent current source, with an adjustable gain, called transconductance g_m [4]. The output current i_{out} is equal to the multiplication of the differential voltage input $v_{\text{in}} = v_+ - v_-$ and g_m

$$i_{\text{out}} = g_m v_{\text{in}}. \quad (1)$$

The conductance between the input terminals is assumed zero as is the case with traditional op amps [25]. The output is assumed to be a current source and so the output impedance is large. This is not the case for the standard op amp which exhibits low impedance at its output terminal. Low output impedance is often desirable when designing audio circuits and so modern bipolar OTAs, such as the LM13700 [28], include controlled impedance buffers on device.

The transconductance of a real world device is a multivariate dynamic nonlinear function, dependent on temperature, device geometry, the manufacturing process, etc. [27, 29]. In the ideal case it is a simple function on temperature and a bias current, which the device sinks through a dedicated input terminal. This bias current is often referred to as the Amplifier Bias Current i_{ABC} . For an OTA based on a bipolar transistor differential pair, the output current and transconductance are given by [3, 4]

$$i_{\text{out}} = i_{\text{ABC}} \tanh \frac{v_{\text{in}}}{2V_T} \quad (2)$$

$$g_m = \frac{di_{\text{out}}}{dv_{\text{in}}} = \frac{i_{\text{ABC}}}{2V_T} \operatorname{sech}^2 \frac{v_{\text{in}}}{2V_T} \quad (3)$$

In this equation the transconductance depends instantaneously on the differential input voltage. It is however desirable for circuit designers if the transconductance is independent on the differential voltage input as discussed in §2. Assuming that $|v_{\text{in}}| \ll 2V_T$ the transconductance becomes

$$g_m \approx \frac{i_{\text{ABC}}}{2V_T} \quad (4)$$

V_T , the thermal voltage, is usually on the scale of tens of millivolts and so making this assumption limits the dynamic range of the input. However, modern OTAs, such as the LM13700 [28], include linearizing diodes that increase the input range of the differential voltage input while keeping the transconductance gain linear with respect to i_{ABC} [3].

Methods on how to handle nonlinearities in WDFs have been proposed in the past. Some have involved the introduction of ad-hoc unit delays [19, 18, 30], simplification of nonlinear devices [17, 31], or ad-hoc [21] or systematic [32] global iteration. A systematized method, sidestepping the aforementioned limitations,

was proposed in [23]. To balance accuracy, physical interpretability and complexity we make the simplification that the transconductance is a linear function of i_{ABC} , as in (4).

Modified Nodal Analysis (MNA) is a systematic way to keep track of physical quantities within a circuit [33]. The method becomes automatable by *stamping* each component into a MNA matrix. MNA element stamps will also work with the Nodal DK-method for deriving nonlinear state-space systems [12] and in §3 we will describe how to transfer a populated MNA matrix from the Kirchoff domain to the wave domain [22]. An ideal OTA is shown in Figure 2 while a MNA element stamp is given by (5).

$$\begin{array}{c|c|c} \alpha & \beta & n \\ \hline \gamma & & 1 \\ \delta & & -1 \\ \hline \text{next} & -g_m & g_m \\ & & -1 \end{array} \quad (5)$$

Nodes shown in Figure 2 are $\alpha = v_+$, $\beta = v_-$, $\gamma = v_{\text{out}}$, $\delta = \text{ground}$.

2.2. OTA Macromodel

Real-world OTAs exhibit multiple nonidealities. Some of which may lead to audible effects and need to be taken into account when designing or analyzing audio circuits. Similar to the nonidealities exhibited by standard opamps [25], real-world OTAs have finite input and output conductances and capacitances, input offset voltage, input bias currents, input offset current, differential and common mode gain and also other OTA-specific nonidealities such as frequency dependent transconductance gain [26].

Choosing which effects to include in a macromodel is a trade-off between complexity and accuracy. Complex nonlinear macromodels for CMOS OTAs exist in the literature [27] and can be adapted to bipolar OTAs by tuning the model parameters. In this paper we balance complexity and accuracy by proposing a linear macromodel which models input and output capacitances C_{i+} , C_{i-} and C_o as well as a finite output conductance R_o .

3. WAVE DIGITAL FILTERS

Here, we briefly elaborate on the recent theoretical developments that have allowed us to model circuits containing OTAs using WDFs techniques. For the sake of brevity a detailed discussion on WDF theory is omitted. The interested reader is referred to the classic article by Fettweis [13] and other recent work in the field [17, 34, 35, 36].

The scattering behavior of multiport series and parallel adaptors has been known since the 1970s [37]. The issue with complicated topologies that can not be divided into series and/or parallel adaptors was first recognized by Martens and Meerkötter [38].

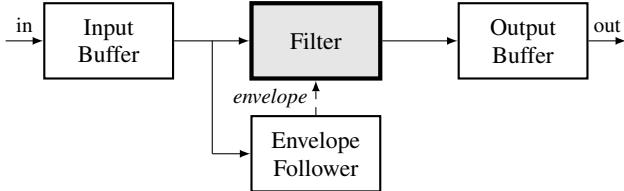


Figure 4: An abstract building block diagram of an envelope filter.

They used a graph-theoretic approach to find the scattering matrix of an adaptor with complicated topology, relying heavily on the orthogonality of the reference circuit for their derivation. Although their method was a step in the right direction, it could not be used to derive scattering matrices for circuits containing common audio circuit devices, such as op amps, OTAs, or controlled sources.

A derivation of the scattering matrix for arbitrary topologies was outlined by Werner *et al.* in [22]. Thévenin sources are placed at each port of a \mathcal{R} -type adaptor and a MNA matrix is populated. By treating all incident and reflected waves simultaneously by grouping them together as vectors \mathbf{b} and \mathbf{a} , a scattering matrix \mathbf{S} describing the relationship between the waves, $\mathbf{b} = \mathbf{S} \mathbf{a}$, is given by

$$\mathbf{S} = \mathbf{I} + 2\mathbf{R}[\mathbf{0} \ \mathbf{I} \ \mathbf{0}] \mathbf{X}^{-1} [\mathbf{0} \ \mathbf{I} \ \mathbf{0}]^\top \quad (6)$$

where \mathbf{R} is a diagonal matrix containing the Thévenin/port resistances and \mathbf{X} is the populated MNA matrix.

An important detail of this approach is that controlled sources may be absorbed into the scattering matrix itself. This is traditionally done with MNA matrices [33] but also in WDF theory by absorbing sources or resistors into adaptors [13]. Absorbing sources into a scattering matrix in this manner is what enables us to model arbitrary circuits containing OTAs.

3.1. OTAs in WDFs

The process of deriving a WDF model of a reference circuit that contains OTAs follows in a similar manner as in [25]. Starting with a reference circuit, replace existing OTAs with an ideal model or macromodel. Following the steps in [39], a WDF structure is found and for each \mathcal{R} -type adaptor, (6) is used to determine its scattering matrix.

The resulting WDF structure can be represented by a SPQR tree that is used to visually indicate how one computation cycle is carried out. For reference circuits that contain OTAs, one additional step must be taken at each cycle as transconductance parameters that reside within \mathcal{R} -type adaptors must be computed and the scattering matrix for each \mathcal{R} -type adaptor updated accordingly.

4. CASE STUDY

In this case study we use the derived OTA models to model an envelope filter guitar effect¹ using WDF techniques. An envelope filter works by tracking the temporal envelope of an input signal. The envelope is used to modulate the critical frequency of a filter, that in turn is used to filter the input. The general structure of the envelope filter can be divided into several abstract building blocks shown in Figure 4.

¹Based on <http://topopiccione.atspace.com/PJ11DODfx25.html>

Table 1: Component values

Component	Value
R_{10}	10 kΩ
$R_{11}, R_{12}, R_{17}, R_{18}$	1 kΩ
$R_{13}, R_{15}, R_{16}, R_{19}, R_{20}$	22 kΩ
R_{14}	100 kΩ
C_6	1 μF
C_7, C_8	10 nF

The derivation of a WDF structure for the entire envelope filter will be presented in detail in an upcoming master thesis [40]. In this paper we will concentrate on the section of the guitar effect that contains OTAs, namely the filter circuit. The circuit schematics of the filter are shown in Figure 5 and the component values are listed in Table 1.

In order to simplify the schematics, we idealize each Darlington transistor pair as an ideal buffer via the circuit theoretic steps detailed in Figure 6. Assuming a transistor operates ideally, it can be replaced by a nullor (step 1) [41, 42]. The junction of two nullators and a norator is equivalent to a single norator (step 2). Finally, a norator in series with any normal one-port is equivalent to just a norator (step 3) [43]. What remains is an ideal (nullor) buffer for each Darlington pair.

This step implies that the voltage drop over the PN-junctions of the Darlington pair is 0 V. As we will see in §5 this assumption does not appear to have a great impact on the output signal. Furthermore, any dc bias that is introduced by this voltage drop would be filtered away in the output buffer stage.

The presence of the nullors also implies that certain electrical components can be dropped from the circuit without affecting its behavior. Specifically, R_{13} and R_{19} and their two series voltage sources v_{EE} end up in parallel with norators coming from the Darlington pairs, an arrangement which is equivalent on a circuit-theoretic level to just a norator [43]. Therefore R_{13} and R_{19} and their series voltage sources v_{EE} are removed from the circuit in preparation for forming the WDF model.

A final transformation to the circuit involves the input stage of the circuit (v_{in} , R_{10} , and C_6). In this stage, R_{10} and C_6 are swapped, as shown in Fig. 7a. This does not affect the dynamics of the circuit, but does ease the implementation as a WDF by allowing R_{10} to be associated directly with v_{in} , forming a resistive voltage source which is suitable for inclusion anywhere in a WDF structure. This transformation does however affect the polarity of the components. An additional inverter, I_1 , must be included in the structure to obtain correct polarization [36, 44].

4.1. Filter Description Assuming Ideal OTA

To gain insight into which kind of filter the circuit is realizing we derive its transfer function. We replace the OTA with our ideal model and assume that i_{ABC} and the range parameter are constants, i.e., we study the system under LTI conditions. We continue to derive the transfer function using MNA (and the newly derived OTA MNA element stamps) [45].

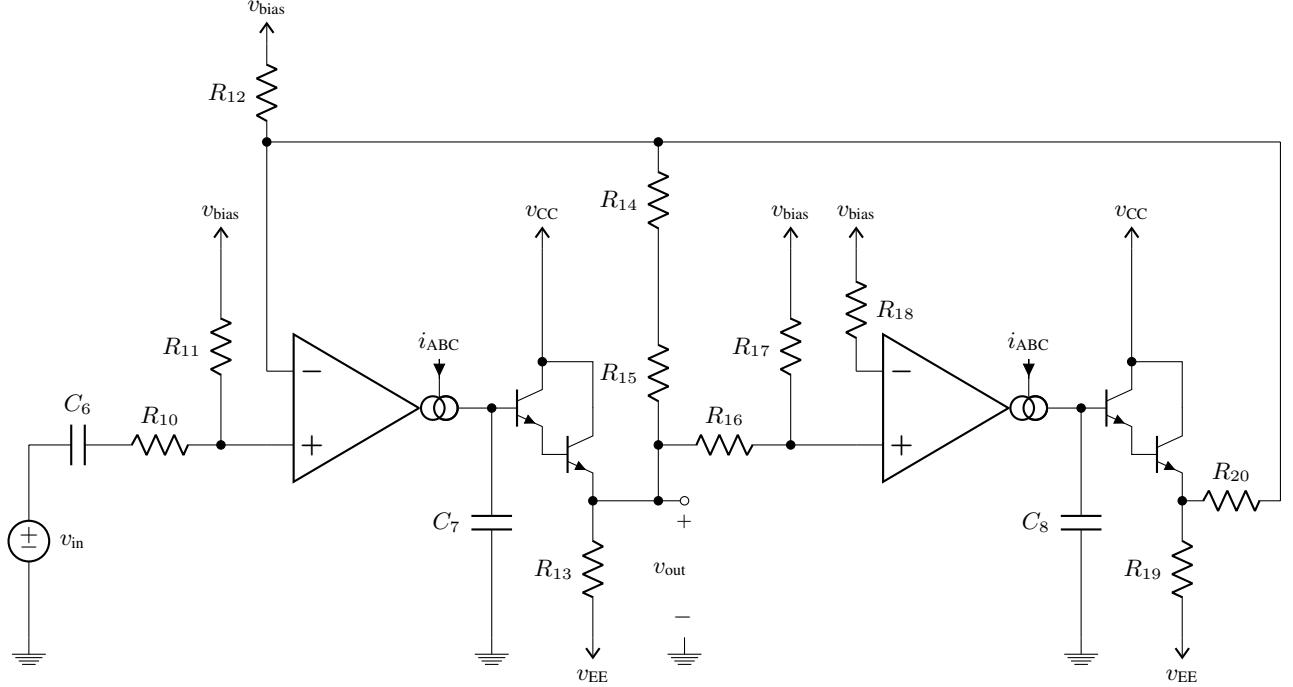


Figure 5: Filter circuit schematic.

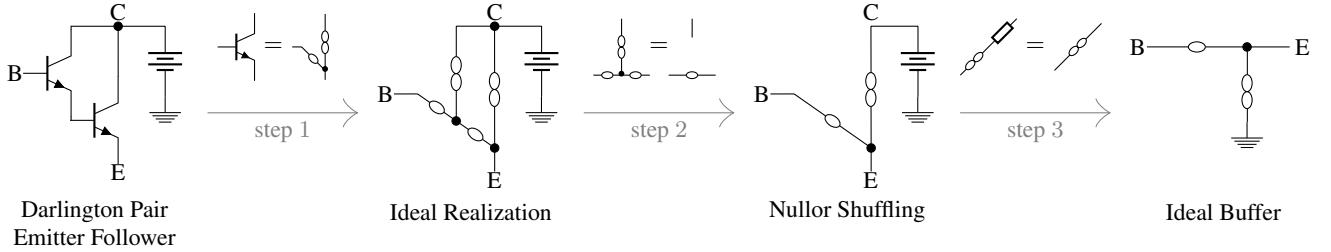


Figure 6: Darlington pair emitter follower to idealized nullor realization.

$$H(s) = H_0 \underbrace{\frac{s}{s + \omega_c}}_{\text{Gain}} \underbrace{\frac{\frac{\omega_0}{Q_\infty} s}{s^2 + \frac{\omega_0}{Q_\infty} s + \omega_0^2}}_{\text{Bandpass filter}} \quad (7)$$

The transfer function this circuit realizes is essentially a 1st order highpass filter, composed of components C₆, R₁₀ and R₁₁, cascaded with a 2nd order bandpass filter [46]. To simplify the expression of the transfer function components with identical values are grouped together, $R_a = R_{10}$, $R_b = R_{11}$, R_{12} , R_{17} , R_{18} , $R_c = R_{15}$, R_{16} , R_{20} , $R_d = R_{14}$, $C_a = C_6$ and $C_b = C_7$, C_8 . We define $R_q = R_c + rR_d$, where $r \in]0, 1]$ determines the range.

$$\omega_c = \frac{1}{C_a (R_a + R_b)} \quad (8)$$

$$H_0 = \frac{\frac{R_b R_c}{R_q} + R_b + R_c}{3 (R_a + R_b)} \quad (9)$$

$$\omega_0^2 = \frac{R_b^2 g_m^2}{C_b^2 (R_b + R_c) (\frac{R_b R_c}{R_q} + R_b + R_c)} \quad (10)$$

$$Q_\infty = \frac{R_q}{R_c} \sqrt{\frac{\frac{R_b R_c}{R_q} + R_b + R_c}{R_b + R_c}}. \quad (11)$$

Interestingly R_q , controlled by the range, influences all parameters of the transfer function except ω_c . The transconductance, g_m , whose highest value can be set by the sensitivity knob in the envelope follower section, only influences the center frequency. That means that the filter is a constant-Q filter with respect to the transconductance, surely a desirable trait when sweeping the frequency spectrum.

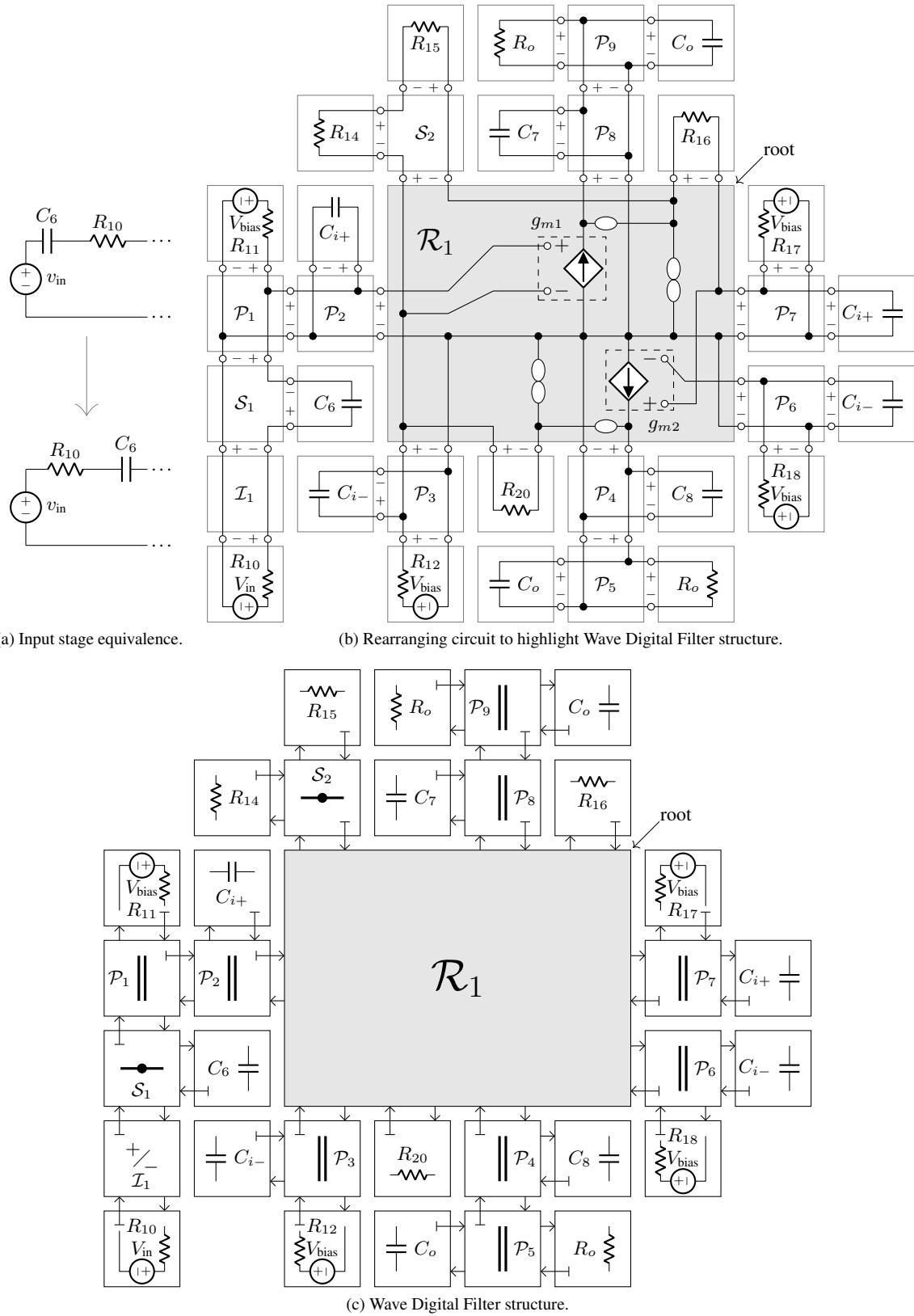


Figure 7: Setting up circuit to highlight topology, and corresponding Wave Digital Filter structure.

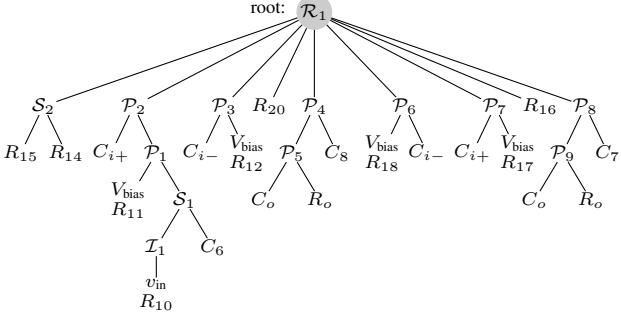


Figure 8: Macromodel OTA—Filter SPQR tree

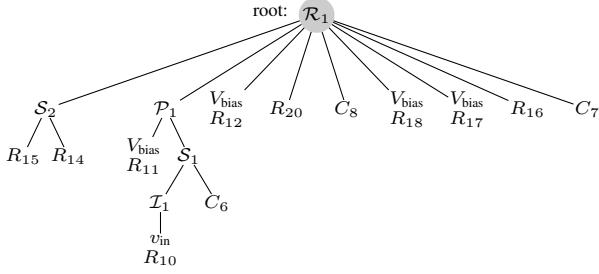


Figure 9: Ideal OTA—Filter SPQR tree.

4.2. WDF Using Macromodel OTA

By first modeling the filter using the macromodel OTA we can show how to derive the complete WDF structure. In §4.3 we simplify this WDF to model the ideal OTA case. We chose the macromodel parameters $C_{i+} = C_{i-} = 5 \text{ pF}$, $C_o = 700 \text{ pF}$ and $R_o = 50 \text{ M}\Omega$ so that a Bode plot of the macromodel-based filter matches a Bode plot of a component-level-based model².

We proceed to model the filter by following the steps described in §3.1. We limit the size of the resulting \mathcal{R} -type adaptor by absorbing resistors into the bias voltage sources V_{bias} . Capacitors are discretized using the bilinear transform. The filter circuit rearranged to highlight the WDF structure is shown in Figure 7b while the corresponding WDF structure is shown in Figure 7c.

4.3. WDF Using Ideal OTA

For the ideal OTA case we simply remove the components belonging to the macromodel (both R_o , both C_o , both C_{i-} , and both C_{i+}) and follow the same procedure as before. Again choosing the \mathcal{R}_1 adaptor as the root, the SPQR tree for the resulting WDF structure is given in Figure 9.

5. RESULTS

Comparison of Bode plots with three amplifier bias currents $i_{ABC} = \{6, 60, 600\} \mu\text{A}$ and four range settings $r = \{0.01, 0.22, 0.6, 1.0\}$ are shown in Figure 10. The input is supplied via the ideal voltage source, v_{in} , and output at taken at v_{out} in Figure 5.

The ideal OTA based circuit shows excellent results when compared to the transfer function. The only visible difference between

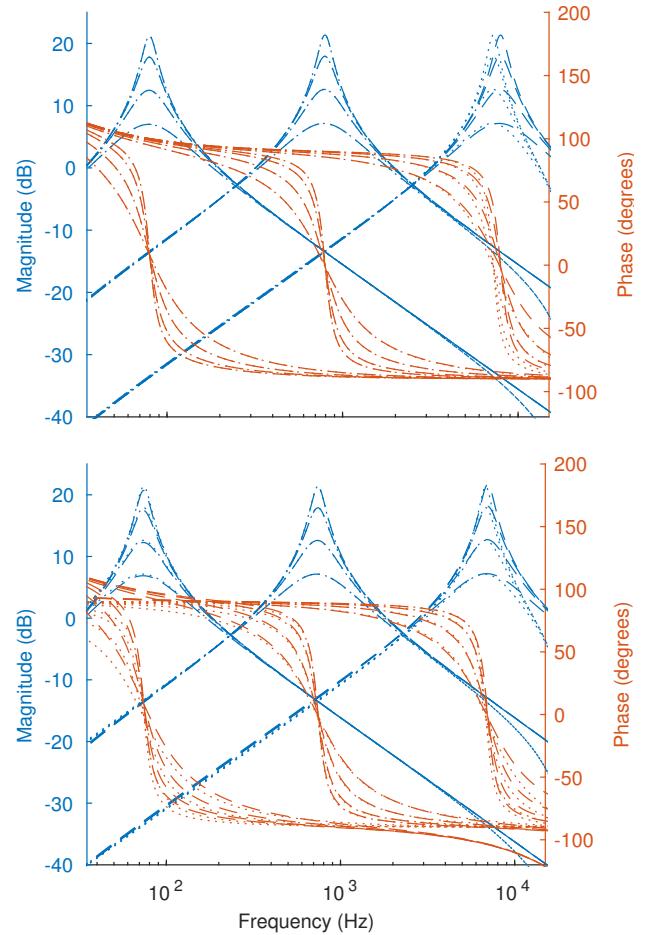


Figure 10: Bode plot comparison of magnitude (blue) and phase (orange) spectrums. Upper plot compares the ideal OTA transfer function (dashed lines) and WDF model (dotted lines). Lower plot compares the component-level SPICE model of the LM13700 (dashed lines) with a macromodel-based model (dotted lines).

to two happens as the frequency approaches the Nyquist frequency where the warping effects from the bilinear transform become noticeable. The plot of the macromodel is in good accordance with a component-level SPICE model where the magnitude spectrum matches almost exactly throughout the range of amplifier bias currents and range controls. There are minor differences in the location of critical frequencies and bandwidth between the two plots in Figure 10.

We briefly study the circuit's behavior under time-varying conditions. The input is a 440 Hz sawtooth, r , the parameter controlling the range, is set to 0.01 and i_{ABC} is increased linearly over time as indicated in the first row of Figure 11. In the second row of the same figure a comparison of a SPICE simulation of the filter circuit composed of an ideal OTA simulated using SPICE is compared to the ideal OTA WDF. The third row compares the ideal and macromodel WDFs to a component-level model of the LM13700 OTA as simulated in SPICE. Despite the assumptions of (4) and idealizing Darlington pairs as ideal buffers, good results are obtained.

²http://www.idea2ic.com/LM13600/SpiceSubcircuit/LM13700_SpiceModel.html

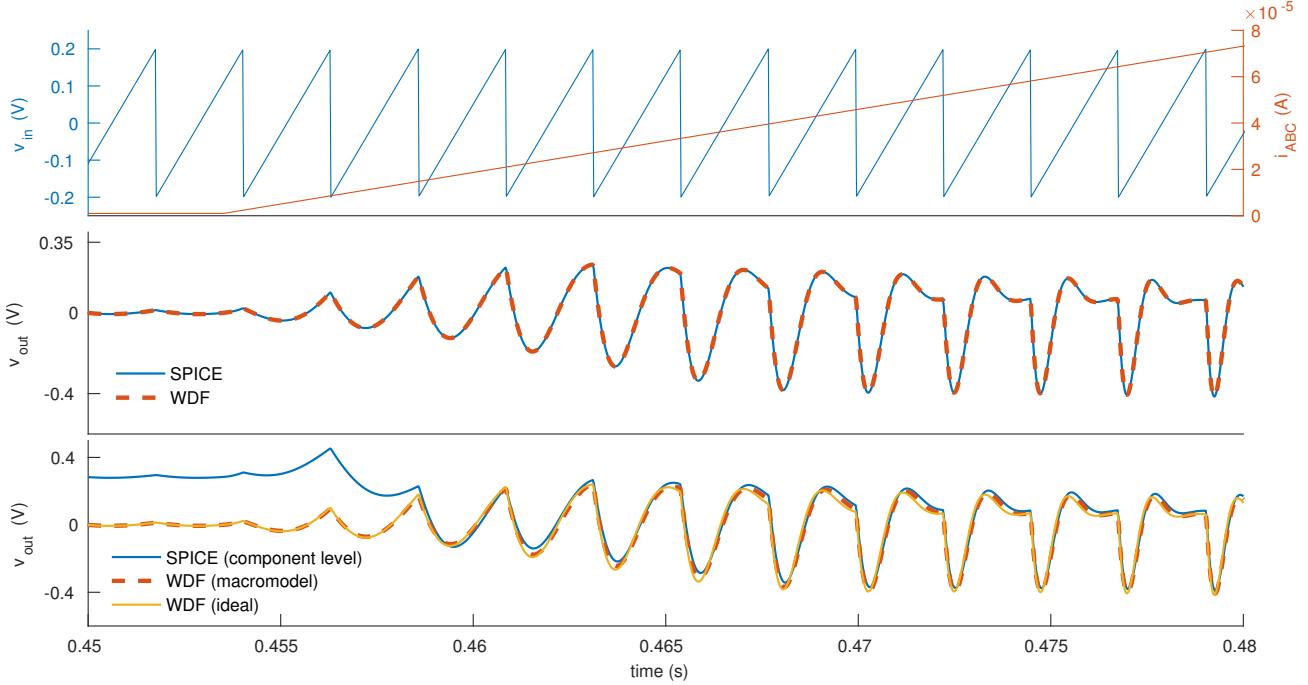


Figure 11: Simulation of filter circuit under time-varying i_{ABC} .

The clipping behavior of the OTA (3) will have a more pronounced effect as the amplitude of the input is increased. This will cause the differences between the simulations from the component-level SPICE model and the WDFs to deviate more at higher amplitudes than lower ones.

6. CONCLUSIONS

Two behavioral models of the operational transconductance amplifier, a commonly found component in audio circuits, were presented. How to incorporate the models into WDF was explained and a WDF model of an envelope filter was derived and simulated. Excellent results in the frequency domain were obtained when compared to an analytically derived transfer function of the filter section of the envelope filter while the results from a macromodel-based WDF performed well when compared to a component-level SPICE model of the LM13700 OTA.

The circuit was briefly studied under time-varying conditions and good results obtained when compared with state-of-the-art circuit simulation software, SPICE. In future work we hope to incorporate the clipping behavior of the OTA into our simulations. We hope also to further elaborate on the time-varying properties of WDFs in particular with respect to choice of s -to- z plane transform and/or numerical method as well as choice of wave variables.

7. REFERENCES

- [1] R. Marston, “Understanding and using OTA op-amp ICs, part 1,” *Nuts and Volts*, pp. 58–62, Apr. 2003.
- [2] R. Marston, “Understanding and using OTA op-amp ICs, part 2,” *Nuts and Volts*, pp. 70–74, May 2003.

- [3] A. Gratz, “Operational transconductance amplifiers,” Tech. Rep., 2008, Online: synth.stromeko.net/diy/OTA.pdf.
- [4] T. Parveen, *Textbook of Operational Transconductance Amplifier and Analog Integrated Circuits*, I.K. International Pvt. Ltd., 2009.
- [5] A. Huovilainen, “Enhanced digital models for analog modulation effects,” in *Proc. Int. Conf. Digital Audio Effects (DAFx-05)*, Madrid, Spain, Sept. 2005.
- [6] J. Pakarinen, V. Välimäki, F. Fontana, V. Lazzarini, and J. S. Abel, “Recent advances in real-time musical effects, synthesis, and virtual analog models,” *EURASIP J. Adv. Signal Process.*, 2011.
- [7] Roland Corporation, “Juno 60 service notes,” Tech. Rep., Apr. 1983.
- [8] Roland Corporation, “SH-101 service notes,” Tech. Rep., Nov. 1982.
- [9] Korg Electronic Laboratory Corporation, “MS-20 service notes,” Tech. Rep., 1978.
- [10] O. Kröning, K. Dempwolf, and U. Zölzer, “Analysis and simulation of an analog guitar compressor,” in *Proc. Int. Conf. Digital Audio Effects (DAFx-11)*, Paris, France, Sept. 19–23, 2011.
- [11] M. Holters and U. Zölzer, “Physical modelling of a wah-wah pedal as a case study for application of the nodal DK method to circuits with variable parts,” in *Proc. Int. Conf. Digital Audio Effects (DAFx-11)*, Paris, France, Sept. 19–23, 2011.
- [12] 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 Trans. Audio, Speech, Language Process.*, vol. 18, no. 4, pp. 728–737, May 2010.
- [13] A. Fettweis, “Wave digital filters: Theory and practice,” in *Proc. IEEE*, 1986, vol. 74, pp. 270–327.
- [14] S. Bilbao, *Wave and Scattering Methods for Numerical Simulation*, John Wiley & Sons, Ltd., 2005.
- [15] A. Sarti and G. De Poli, “Toward nonlinear wave digital filters,” *IEEE Trans. Signal Process.*, vol. 47, no. 6, pp. 1654–1668, 1999.
- [16] S. Petrausch and R. Rabenstein, “Wave digital filters with multiple nonlinearities,” in *Proc. 12th European Signal Process. Conf. (EUSIPCO)*, Sept. 2004, pp. 77–80.
- [17] 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, 2010.
- [18] 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.
- [19] A. Bernardini and A. Sarti, “Dynamic adaptation of instantaneous nonlinear dipoles in wave digital networks,” in *Proc. 24th European Signal Process. Conf. (EUSIPCO)*, Aug. 2016, pp. 1038–1042.
- [20] M. Karjalainen and J. Pakarinen, “Wave digital simulation of a vacuum-tube amplifier,” in *IEEE Int. Conf. Acoust., Speech, Signal Process.*, May 2006.
- [21] S. D’Angelo, J. Pakarinen, and V. Välimäki, “New family of wave-digital triode models,” *IEEE Trans. Audio, Speech, Language Process.*, vol. 21, no. 2, pp. 313–321, 2013.
- [22] K. J. Werner, J. O. Smith III, and J. Abel, “Wave digital filter adaptors for arbitrary topologies and multiport linear elements,” in *Proc. Int. Conf. Digital Audio Effects (DAFx-15)*, Trondheim, Norway, Nov. 2015.
- [23] K. J. Werner, V. Nangia, J. O. Smith III, and J. S. Abel, “Resolving wave digital filters with multiple/multiport nonlinearities,” in *Proc. Int. Conf. Digital Audio Effects (DAFx-15)*, Trondheim, Norway, Nov. 2015.
- [24] A. S. Sedra and K. C. Smith, *Microelectronic Circuits*, Oxford University Press, New York, sixth edition, 2015.
- [25] K. J. Werner, W. R. Dunkel, M. Rest, M. J. Olsen, and J. O. Smith III, “Wave digital filter modeling of circuits with operational amplifiers,” in *Proc. 24th European Signal Process. Conf. (EUSIPCO)*, Budapest, Hungary, Aug. 2016.
- [26] C. Acar, F. Anday, and H. Kuntman, “On the realization of OTA-C filters,” *Int. J. Circuit Theory Appl.*, vol. 21, pp. 331–341, 1993.
- [27] H. Kuntman, “Simple and accurate nonlinear OTA macro-model for simulation of CMOS OTA-C active filters,” *Int. J. Electron.*, vol. 77, no. 6, pp. 993–1006, 1994.
- [28] Texas Instruments, “LM13700 datasheet,” Tech. Rep., Nov. 1999.
- [29] R. L. Geiger and E. Sánchez-Sinencio, “Active filter design using operational transconductance amplifiers: A tutorial,” *IEEE Circuits Devices Mag.*, vol. 1, no. 2, pp. 20–32, Mar. 1985.
- [30] K. Meerkötter and T. Felderhoff, “Simulation of nonlinear transmission lines by wave digital filter principles,” in *Proc. IEEE Int. Symp. Circuits Syst.*, May 1992, vol. 2, pp. 875–878.
- [31] A. Bernardini, K. J. Werner, A. Sarti, and J. O. Smith III, “Modeling nonlinear wave digital elements using the Lambert function,” *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 63, no. 8, pp. 1231–1242, 2016.
- [32] T. Schwerdtfeger and A. Kummert, “A multidimensional approach to wave digital filters with multiple nonlinearities,” in *Proc. 22nd European Signal Process. Conf. (EUSIPCO)*, Lisbon, Portugal, Sept. 2014, pp. 2405–2409.
- [33] C. Ho, A. Ruehli, and P. Brennan, “The modified nodal approach to network analysis,” *IEEE Trans. Circuits Syst.*, vol. 22, no. 6, pp. 504–509, June 1975.
- [34] K. J. Werner, *Virtual Analog Modeling of Audio Circuitry Using Wave Digital Filters*, Ph.D. dissertation, Stanford University, 2016.
- [35] A. Bernardini, K. J. Werner, A. Sarti, and J. O. Smith III, “Modeling a class of multi-port nonlinearities in wave digital structures,” in *Proc. 23rd European Signal Process. Conf. (EUSIPCO)*, 2015.
- [36] 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. Int. Conf. Digital Audio Effects (DAFx-16)*, Brno, Czech Republic, Sept. 2016, pp. 271–278.
- [37] 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.
- [38] G. O. Martens and K. Meerkötter, “On N-port adaptors for wave digital filters with application to a bridged-tee filter,” in *Proc. IEEE Int. Symp. Circuits Syst. (ISCAS)*, Munich, Germany, Apr. 1976, pp. 514–517.
- [39] 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.
- [40] Ó. Bogason, “Modeling audio circuits containing typical nonlinear components with wave digital filters,” M.S. thesis, McGill University, Montreal, Quebec, Canada, 2017.
- [41] G. Martinelli, “On the nullor,” *Proc. IEEE*, vol. 53, no. 3, pp. 332, Mar. 1965.
- [42] B. R. Myers, “Nullor model of the transistor,” *Proc. IEEE*, vol. 53, no. 7, pp. 758–759, July 1965.
- [43] L. T. Bruton, *RC-Active Circuits*, Prentice-Hall Inc, Englewood Cliffs, New Jersey, 1980.
- [44] S. D’Angelo and V. Välimäki, “Wave-digital polarity and current inverters and their application to virtual analog audio processing,” in *IEEE Int. Conf. Acoust., Speech, Signal Process. (ICASSP)*, Mar. 2012, pp. 469–472.
- [45] A. B. Yıldız, “Modified nodal analysis-based determination of transfer functions for multi-inputs multi-outputs linear circuits,” *Automatika*, vol. 51, no. 4, pp. 353–360, 2010.
- [46] U. Zölzer, *Digital Audio Signal Processing*, John Wiley & Sons, Ltd., 2 edition, 2008.

AUTOMATIC DECOMPOSITION OF NON-LINEAR EQUATION SYSTEMS IN AUDIO EFFECT CIRCUIT SIMULATION

Martin Holters, Udo Zölzer

Department of Circuits and Systems,
Helmut Schmidt University
University of the Federal Armed Forces
Hamburg, Germany
martin.holters|udo.zoelzer@hsu-hh.de

ABSTRACT

In the digital simulation of non-linear audio effect circuits, the arising non-linear equation system generally poses the main challenge for a computationally cheap implementation. As the computational complexity grows super-linearly with the number of equations, it is beneficial to decompose the equation system into several smaller systems, if possible. In this paper we therefore develop an approach to determine such a decomposition automatically. We limit ourselves to cases where an exact decomposition is possible, however, and do not consider approximate decompositions.

1. INTRODUCTION

Digital emulation of analog circuits for musical audio processing, like synthesizers, guitar effect pedals, or vintage amplifiers, is an ongoing research topic. Various methods exist to derive a mathematical model for an analog circuit in a systematic hence automatable way, most notably wave digital filters [1, 2, 3], port-Hamiltonian approaches [4, 5], and state-space based approaches [6, 7]. In the following, we will focus on the method of [7], as it has no limitations concerning the circuit topology and is general enough to also handle pathological circuits or element models (e.g. [8]). However, the underlying ideas should be equally applicable to the other approaches.

The downside of such automated approaches, and of [7] in particular, is that they often will lead to one large system of non-linear equations, collected from all the circuit's non-linear elements. If possible, however, it is usually more efficient to solve many small equation systems instead of a single large one. Typically, the convergence of small systems will be better, allowing for a smaller number of iterations in an iterative solver. And more tangible, the complexity of solving the linear equation in e.g. the Newton method scales with the square of the number of involved equations, so that for a system of size N , the asymptotic complexity per iteration is $O(N^2)$, while for N systems of size 1, it is $O(N)$.

We therefore develop a method to decompose a non-linear equation system into smaller subsystems. We limit ourselves to the case where this is possible without resorting to approximations, although this unfortunately precludes the method from being applied to many circuits of practical relevance, especially those with global feedback paths. But while automatically deriving approximate decompositions, e.g. like those of [9, 10, 11], is beyond the scope of this paper, we are confident it is still useful by itself and furthermore, may form the basis for future methods to automatically find such approximate decompositions.

2. MODEL DERIVATION METHOD

We shall first provide a short introduction into the method used to obtain the circuit model and the non-linear equation in particular, focusing on the example of Figure 1 (based on “Der Birdie”¹), as also discussed in [12], while the reader is referred to [7] for details.

First, the equations of the individual circuit elements are rewritten in a uniform way in terms of branch voltages and currents and internal states. For example, a resistor with resistance R is described as

$$v_R + Ri_R = 0, \quad (1)$$

where v_R and i_R are the voltage across and the current through the resistor, respectively. The discrete-time model² of a capacitor with capacitance C is derived using bilinear transform as

$$\begin{pmatrix} C \\ 0 \end{pmatrix} v_C(n) + \begin{pmatrix} 0 \\ 1 \end{pmatrix} i_C(n) - \begin{pmatrix} \frac{1}{2} \\ \frac{1}{T} \end{pmatrix} x_C(n) = \begin{pmatrix} \frac{1}{2} \\ -\frac{1}{T} \end{pmatrix} x_C(n-1), \quad (2)$$

where T denotes the sampling interval and n the current time step, v_C and i_C are again the voltage across and the current through the capacitor, respectively, and the state x_C corresponds to the capacitor’s charge. Voltage sources are easily expressed using a non-zero right-hand side, e.g. as

$$vv_{CC} + 0 \cdot iv_{CC} = 9V \quad (3)$$

for the supply voltage source V_{CC} , with a constant voltage $vv_{CC} = 9V$ across it, driving an arbitrary current iv_{CC} .

The non-linear equation of non-linear elements is expressed in terms of an auxiliary vector \mathbf{q} , which is related to voltages and currents (and potentially also states) through a linear equation. Thus, a diode is expressed using the linear equation

$$\begin{pmatrix} 1 \\ 0 \end{pmatrix} v_D(n) + \begin{pmatrix} 0 \\ 1 \end{pmatrix} i_D(n) + \begin{pmatrix} -1 & 0 \\ 0 & -1 \end{pmatrix} \mathbf{q}_D(n) = \begin{pmatrix} 0 \\ 0 \end{pmatrix}, \quad (4)$$

fixing $q_{D,1} = v_D$ as the voltage across and $q_{D,2} = i_D$ as the current through the diode, and the non-linear Shockley equation, rewritten to implicit form as

$$f_D(\mathbf{q}_D) = I_s \cdot \left(e^{\frac{q_{D,1}}{\eta v_T}} - 1 \right) - q_{D,2} = 0, \quad (5)$$

¹<http://diy.musikding.de/wp-content/uploads/2013/06/birdieschalt.pdf>

²In [7], the equations are first derived in the continuous-time domain and then transformed to discrete time using the bilinear form. Equivalently, the bilinear transform may be applied per element, which we do here for brevity’s sake.

where I_s , η , and v_T denote, respectively, reverse saturation current, emission coefficient (ideality factor), and thermal voltage. Similarly, a transistor is expressed using the linear equation

$$\begin{aligned} \begin{pmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 0 \\ 0 & 0 \end{pmatrix} \mathbf{v}_T(n) + \begin{pmatrix} 0 & 0 \\ 0 & 0 \\ 1 & 0 \\ 1 & 0 \end{pmatrix} \mathbf{i}_T(n) \\ + \begin{pmatrix} -1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & -1 \end{pmatrix} \mathbf{q}_T(n) = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}, \quad (6) \end{aligned}$$

where \mathbf{v}_T and \mathbf{i}_T hold the voltages across and currents through the base-emitter and base-collector branch in that order, respectively, and the non-linear equation is obtained by rewriting the Ebers-Moll equation to implicit form as

$$\begin{aligned} f_T(\mathbf{q}_T) = \\ \begin{pmatrix} I_{sE} \cdot (e^{\frac{qT,1}{\eta_E v_T}} - 1) - \frac{\beta_r}{1+\beta_r} I_{sC} \cdot (e^{\frac{qT,2}{\eta_C v_T}} - 1) - q_{T,3} \\ - \frac{\beta_f}{1+\beta_f} I_{sE} \cdot (e^{\frac{qT,1}{\eta_E v_T}} - 1) + I_{sC} \cdot (e^{\frac{qT,2}{\eta_C v_T}} - 1) - q_{T,4} \end{pmatrix} \\ = \begin{pmatrix} 0 \\ 0 \end{pmatrix}, \quad (7) \end{aligned}$$

where β_f and β_r denote forward and reverse current gain, respectively, and the reverse saturation currents I_{sE} and I_{sC} and the emission coefficients η_E and η_C can differ between base-emitter and base-collector junction.

Once the individual elements are modeled in a suitable form, the circuit topology is taken into consideration by formulating the Kirchhoff voltage law

$$\mathbf{T}_v \mathbf{v} = \mathbf{0} \quad (8)$$

and the Kirchhoff current law

$$\mathbf{T}_i \mathbf{i} = \mathbf{0}, \quad (9)$$

where \mathbf{v} and \mathbf{i} collect all the circuit's branch voltages and currents, using matrices \mathbf{T}_v of independent loop and \mathbf{T}_i of independent node (or cut-set) equations, which can be obtained by well-known methods (see e.g. [13]). For the circuit of Figure 1 one e.g. finds

$$\mathbf{T}_v = \begin{pmatrix} V_{CC} & C_5 & D & V_{in} & R_1 & C_1 & R_2 & R_3 & T_{BE} & T_{BC} & R_4 & R_5 & C_3 & P_{1,1} & P_{1,2} \\ 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & -1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & -1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & -1 & 0 & 0 & -1 & 0 & 0 & -1 & 1 & 1 & 1 & 1 \end{pmatrix} \quad (10)$$

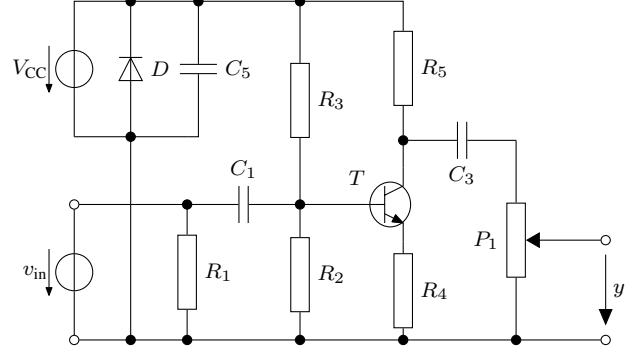


Figure 1: Example treble booster circuit.

Table 1: Component/parameter values for the circuit of Figure 1.

comp.	value	comp.	param.	value
V_{CC}	9	V		
R_1	1	$M\Omega$	D	I_s 350 pA
R_2	43	k Ω	D	η 1.6
R_3	430	k Ω	T	I_{sE} 64.53 fA
R_4	390	Ω	T	I_{sC} 154.1 fA
R_5	10	k Ω	T	η_E 1.06
P_1	100	k Ω	T	η_C 1.10
C_1	2.2	nF	T	β_f 500
C_3	2.2	nF	T	β_r 12
C_5	100	μF		

and

$$\mathbf{T}_i = \begin{pmatrix} V_{CC} & C_5 & D & V_{in} & R_1 & C_1 & R_2 & R_3 & T_{BE} & T_{BC} & R_4 & R_5 & C_3 & P_{1,1} & P_{1,2} \\ 1 & -1 & -1 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & -1 & -1 \\ 0 & 0 & 0 & 0 & 0 & 1 & -1 & -1 & 0 & 0 & -1 & -1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 \end{pmatrix}. \quad (11)$$

Combining the element equations with the topology equations yields the equation system

$$\begin{pmatrix} \mathbf{M}_v & \mathbf{M}_i & \mathbf{M}_x & \mathbf{M}_q \\ \mathbf{T}_v & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{T}_i & \mathbf{0} & \mathbf{0} \end{pmatrix} \begin{pmatrix} \mathbf{v}(n) \\ \mathbf{i}(n) \\ \mathbf{x}(n) \\ \mathbf{q}(n) \end{pmatrix} = \begin{pmatrix} \mathbf{M}_x \\ \mathbf{0} \\ \mathbf{0} \end{pmatrix} \mathbf{x}(n-1) + \begin{pmatrix} \mathbf{M}_u \\ \mathbf{0} \\ \mathbf{0} \end{pmatrix} \mathbf{u}(n) + \begin{pmatrix} \mathbf{u}_0 \\ \mathbf{0} \\ \mathbf{0} \end{pmatrix}. \quad (12)$$

The matrices \mathbf{M}_v , \mathbf{M}_i , \mathbf{M}_x , \mathbf{M}_q , and \mathbf{M}_x , are constructed as block diagonal matrices (with potentially rectangular blocks) from the respective factors in the element equations, and \mathbf{x} and \mathbf{q} are obtained by stacking the respective entries of the individual elements. For the example circuit, the top left parts of the matrices, corresponding

to V_{CC} , C_5 , and D , e.g. are

$$\mathbf{M}_v = \begin{pmatrix} 1 & 0 & 0 & \dots \\ 0 & C_5 & 0 & \dots \\ 0 & 0 & 0 & \dots \\ 0 & 0 & 1 & \dots \\ 0 & 0 & 0 & \dots \\ \vdots & \vdots & \vdots & \ddots \end{pmatrix} \quad \mathbf{M}_i = \begin{pmatrix} 0 & 0 & 0 & \dots \\ 0 & 0 & 0 & \dots \\ 0 & 1 & 0 & \dots \\ 0 & 0 & 0 & \dots \\ 0 & 0 & 1 & \dots \\ \vdots & \vdots & \vdots & \ddots \end{pmatrix} \quad (13)$$

$$\mathbf{M}_x = \begin{pmatrix} 0 & \dots \\ -\frac{1}{2} & \dots \\ -\frac{1}{T} & \dots \\ 0 & \dots \\ 0 & \dots \\ \vdots & \ddots \end{pmatrix} \quad \mathbf{M}_z = \begin{pmatrix} 0 & \dots \\ \frac{1}{2} & \dots \\ -\frac{1}{T} & \dots \\ 0 & \dots \\ 0 & \dots \\ \vdots & \ddots \end{pmatrix} \quad (14)$$

$$\mathbf{M}_q = \begin{pmatrix} 0 & 0 & \dots \\ 0 & 0 & \dots \\ 0 & 0 & \dots \\ -1 & 0 & \dots \\ 0 & -1 & \dots \\ \vdots & \vdots & \ddots \end{pmatrix}. \quad (15)$$

The vector \mathbf{u}_0 holds constant source values, in this case a single non-zero as first entry, namely $V_{CC} = 9$ V, while $\mathbf{u}(n)$ holds all time-varying source values, i.e. the circuit inputs, in this case only the input voltage $v_{in}(n)$, which is mapped to the appropriate row by the matrix \mathbf{M}_q , a one-column matrix in the example with a single 1 in the sixth row.

The matrix on the left-hand side of (12) has dimension 36×39 , hence the equation system is under-determined. However, we can solve for a solution set

$$\begin{pmatrix} \mathbf{v}(n) \\ \mathbf{i}(n) \\ \mathbf{x}(n) \\ \mathbf{q}(n) \end{pmatrix} = \begin{pmatrix} \mathbf{D}_v \\ \mathbf{D}_i \\ \mathbf{E}_v \\ \mathbf{A} \\ \mathbf{D}_q \end{pmatrix} \mathbf{x}(n-1) + \begin{pmatrix} \mathbf{E}_v \\ \mathbf{E}_i \\ \mathbf{B} \\ \mathbf{F}_q \end{pmatrix} \mathbf{u}(n) + \begin{pmatrix} \mathbf{v}_0 \\ \mathbf{i}_0 \\ \mathbf{x}_0 \\ \mathbf{q}_0 \end{pmatrix} + \begin{pmatrix} \mathbf{F}_v \\ \mathbf{F}_i \\ \mathbf{C} \\ \mathbf{F}_q \end{pmatrix} \mathbf{z}(n), \quad (16)$$

where the first three terms form a particular solution, while the last term with a three-dimensional arbitrary vector $\mathbf{z}(n)$ spans the solution set. Note that neither the particular solution nor the basis of the nullspace to span the solution set are unique. In the ACME framework³ that implements this method, an algorithm based on [14] is used that exploits the sparsity of the matrices, and the particular solution is further modified such that \mathbf{D}_q and \mathbf{E}_q are orthogonal to \mathbf{F}_q as motivated in [12].

In the following, the last row of (16) is of particular interest, and we shall drop the time index n , i.e. we write

$$\mathbf{q} = \mathbf{D}_q \mathbf{x} + \mathbf{E}_q \mathbf{u} + \mathbf{q}_0 + \mathbf{F}_q \mathbf{z}. \quad (17)$$

In order to find \mathbf{z} , the non-linear element equations need to be

considered, which are gathered to obtain

$$\begin{aligned} \mathbf{f}(\mathbf{q}) &= \mathbf{f}\left(\begin{pmatrix} \mathbf{q}_D \\ \mathbf{q}_T \end{pmatrix}\right) = \begin{pmatrix} \mathbf{f}_D(\mathbf{q}_D) \\ \mathbf{f}_T(\mathbf{q}_T) \end{pmatrix} \\ &= \begin{pmatrix} I_s \cdot \left(e^{\frac{q_1}{\eta_E v_T}} - 1\right) - q_2 \\ I_{sE} \cdot \left(e^{\frac{q_3}{\eta_E v_T}} - 1\right) - \frac{\beta_r}{1+\beta_r} I_{sC} \cdot \left(e^{\frac{q_4}{\eta_C v_T}} - 1\right) - q_5 \\ -\frac{\beta_f}{1+\beta_f} I_{sE} \cdot \left(e^{\frac{q_3}{\eta_E v_T}} - 1\right) + I_{sC} \cdot \left(e^{\frac{q_4}{\eta_C v_T}} - 1\right) - q_6 \end{pmatrix} = \mathbf{0}. \end{aligned} \quad (18)$$

Note that the number $N_n = 3$ of subequations is smaller than the number $N_q = 6$ of entries in the vector \mathbf{q} . Thus, this implicit non-linear equation confines \mathbf{q} to a solution manifold. On the other hand, \mathbf{q} is restricted to the affine subspace spanned by (17) for arbitrary \mathbf{z} . As \mathbf{z} has exactly $N_n = 3$ entries, there is in general a finite number of permissible \mathbf{z} , and for physically meaningful circuit schematics, there will be a unique solution. Once \mathbf{z} is found, it is used to calculate the circuit's output (by extracting the desired entry of $\mathbf{v}(n)$) and update its states according to (16).

3. DECOMPOSITION METHOD

In general, the non-linear equation system is formed by collecting subequations of the N individual non-linear elements contained in the circuit as

$$\mathbf{f}(\mathbf{q}) = \begin{pmatrix} \mathbf{f}_1(\mathbf{q}_1) \\ \mathbf{f}_2(\mathbf{q}_2) \\ \vdots \\ \mathbf{f}_N(\mathbf{q}_N) \end{pmatrix} = \mathbf{0} \quad (19)$$

where \mathbf{q} is likewise formed from subvectors as

$$\mathbf{q}^T = (\mathbf{q}_1^T \quad \mathbf{q}_2^T \quad \cdots \quad \mathbf{q}_N^T). \quad (20)$$

Unfortunately, this does not mean that the subequations can be solved individually, as we need to solve for \mathbf{z} , not \mathbf{q} or the individual \mathbf{q}_n . In fact, it will only be possible in very benign cases to solve the equations element-by-element. It is more likely that groups of elements can be identified into which the non-linear equation can be decomposed.

3.1. Decomposition assuming known grouping

Assume the non-linear equation system is split into M equation groups $\tilde{\mathbf{f}}_m(\tilde{\mathbf{q}}_m)$, $m = 1, \dots, M$ with $M \leq N$. That is, each $\tilde{\mathbf{q}}_m$ and $\tilde{\mathbf{f}}_m(\tilde{\mathbf{q}}_m)$ is the concatenation of one or more \mathbf{q}_n and $\mathbf{f}_n(\mathbf{q}_n)$, respectively. As the ordering of the elements in $\mathbf{f}(\mathbf{q})$ and \mathbf{q} is arbitrary, we may assume without loss of generality that

$$\mathbf{f}^T(\mathbf{q}) = (\tilde{\mathbf{f}}_1^T(\tilde{\mathbf{q}}_1) \quad \cdots \quad \tilde{\mathbf{f}}_M^T(\tilde{\mathbf{q}}_M)) \quad (21)$$

and

$$\mathbf{q}^T = (\tilde{\mathbf{q}}_1^T \quad \cdots \quad \tilde{\mathbf{q}}_M^T). \quad (22)$$

Let $\mathbf{D}_{q,m}$, $\mathbf{E}_{q,m}$, and $\mathbf{F}_{q,m}$ denote the corresponding rows of the respective matrices and likewise $\mathbf{q}_{0,m}$ the corresponding entries of \mathbf{q}_0 so that

$$\tilde{\mathbf{q}}_m = \mathbf{D}_{q,m} \mathbf{x} + \mathbf{E}_{q,m} \mathbf{u} + \mathbf{q}_{0,m} + \mathbf{F}_{q,m} \mathbf{z}. \quad (23)$$

Further let \mathbf{z} also be decomposed as

$$\mathbf{z}^T = (\mathbf{z}_1^T \quad \mathbf{z}_2^T \quad \cdots \quad \mathbf{z}_M^T) \quad (24)$$

³<https://github.com/HSU-ANT/ACME.jl>

such that \mathbf{z}_m has as many entries as $\tilde{\mathbf{f}}_m(\tilde{\mathbf{q}}_m)$ and let $\mathbf{F}_{q,m,n}$ denote the corresponding columns of $\mathbf{F}_{q,m}$, that is

$$\mathbf{F}_q = \begin{pmatrix} \mathbf{F}_{q,1,1} & \cdots & \mathbf{F}_{q,1,M} \\ \vdots & \ddots & \vdots \\ \mathbf{F}_{q,M,1} & \cdots & \mathbf{F}_{q,M,M} \end{pmatrix} \quad (25)$$

such that

$$\tilde{\mathbf{q}}_m = \mathbf{D}_{q,m}\mathbf{x} + \mathbf{E}_{q,m}\mathbf{u} + \mathbf{q}_{0,m} + \mathbf{F}_{q,m,1}\mathbf{z}_1 + \cdots + \mathbf{F}_{q,m,M}\mathbf{z}_M. \quad (26)$$

Now, let the element groups be chosen such that $\mathbf{F}_{q,m,n} = \mathbf{0}$ for $n > m$. Then

$$\tilde{\mathbf{q}}_1 = \mathbf{D}_{q,1}\mathbf{x} + \mathbf{E}_{q,1}\mathbf{u} + \mathbf{q}_{0,1} + \mathbf{F}_{q,1,1}\mathbf{z}_1 \quad (27)$$

so that \mathbf{z}_1 can be obtained by solving $\tilde{\mathbf{f}}_1(\tilde{\mathbf{q}}_1) = \mathbf{0}$. With \mathbf{z}_1 known, \mathbf{z}_2 can then be obtained by solving $\tilde{\mathbf{f}}_2(\tilde{\mathbf{q}}_2) = \mathbf{0}$ and so forth up to M . Thus, if \mathbf{F}_q is such that a partitioning with $\mathbf{F}_{q,m,n} = \mathbf{0}$ for $n > m$ (and $M > 1$) exists, we can decompose the non-linear equation into individually solvable subequations, where in general, the m -th subequation depends on the solution of the $m - 1$ previous subequations.

Of course, we may rarely be lucky and find \mathbf{F}_q to be suitable for this decomposition. However, [7] leaves some freedom in the exact choice of the coefficient matrices. In particular, only the space spanned by $\mathbf{F}_q\mathbf{z}$ is of importance, which does not change if we substitute $\mathbf{F}_q \leftarrow \mathbf{F}_q\mathbf{R}$ for a regular matrix \mathbf{R} . (This will, of course, change the resulting \mathbf{z} , so \mathbf{F}_V , \mathbf{F}_I , and \mathbf{C} have to be updated in the same way.) Now, finding an \mathbf{R} such that the upper right subblocks of \mathbf{F}_q become zero is similar to bringing \mathbf{F}_q^T into upper triangular form, but with respect to the rectangular subblocks $\mathbf{F}_{q,m,n}$. Thus if the chosen decomposition into subequation groups allows a suitable choice of \mathbf{F}_q , it can be found using standard tools of linear algebra, e.g. Gaussian elimination.

3.2. Identification of a suitable grouping

The remaining question is how to determine a suitable grouping. If \mathbf{f}_q and \mathbf{q} were ordered in correspondence with the yet-to-be-found grouping, i.e. fulfilling (21) and (22), we could just determine \mathbf{R} to eliminate the maximum number of elements in the upper right part of \mathbf{F}_q and examine the zero-structure thus obtained. Unfortunately, the number of possible permutations of the entries in \mathbf{f}_q and \mathbf{q} grows too fast with N to make trying all of them feasible. E.g. for $N = 10$ non-linear elements, we would need to examine $N! = 3\,628\,800$ permutations.

We can do a little better than that by greedily trying to separate a single subgroup by trying all $2^N - 1$ non-empty subsets of $\{1, \dots, N\}$ in order of increasing cardinality. Once the smallest permissible subgroup has been identified, i.e. one for which \mathbf{F}_q can be transformed in a suitable way when that subgroup is placed first in \mathbf{f}_q and \mathbf{q} , the process is repeated for the remaining elements. In the worst case, if the circuit does not allow decomposition, all $2^N - 1$ non-empty subsets have to be tried. Otherwise, the number of trials in the first iteration is lower, but additional trials are needed for the remaining non-linear elements. Nevertheless, it can be seen that the complete procedure never has to try more than $2^N - 1$ subgroupings. This is still an exponential growth with N , but in the realm of audio effect circuits where a number N of non-linear elements in the low two-digit range is already considered quite complex, the needed computational time during the offline

pre-computation step may be well acceptable. E.g. for the $N = 10$ case, at most 1023 trials would be needed.

3.3. Dimensionality reduction of the input vector

After the decomposition, we can obtain \mathbf{z}_m from $\tilde{\mathbf{f}}_m(\tilde{\mathbf{q}}_m) = \mathbf{0}$, which depends on \mathbf{x} , \mathbf{u} , and $\mathbf{z}_1, \dots, \mathbf{z}_{m-1}$, a potentially high number of input values. This would pose a major problem if one would like to tabulate precomputed values in a look-up table. However, the method proposed in [12] can be easily extended to not only treat \mathbf{x} and \mathbf{u} as inputs, but also $\mathbf{z}_1, \dots, \mathbf{z}_{m-1}$, to find an index vector \mathbf{p}_m of minimal dimension that can be used as input.

The idea is to apply a rank factorization to the matrix

$$\begin{aligned} & (\mathbf{D}_{q,m} \quad \mathbf{E}_{q,m} \quad \mathbf{F}_{q,m,1} \quad \cdots \quad \mathbf{F}_{q,m,m-1}) \\ &= \mathbf{Q}_m \cdot (\hat{\mathbf{D}}_{q,m} \quad \hat{\mathbf{E}}_{q,m} \quad \hat{\mathbf{F}}_{q,m,1} \quad \cdots \quad \hat{\mathbf{F}}_{q,m,m-1}) \end{aligned} \quad (28)$$

such that $(\hat{\mathbf{D}}_{q,m} \quad \hat{\mathbf{E}}_{q,m} \quad \hat{\mathbf{F}}_{q,m,1} \quad \cdots \quad \hat{\mathbf{F}}_{q,m,m-1})$ has minimal number of rows. Then

$$\mathbf{p}_m = \hat{\mathbf{D}}_{q,m}\mathbf{x} + \hat{\mathbf{E}}_{q,m}\mathbf{u} + \hat{\mathbf{F}}_{q,m,1}\mathbf{z}_1 + \cdots + \hat{\mathbf{F}}_{q,m,m-1}\mathbf{z}_{m-1} \quad (29)$$

is the index vector of minimal dimension to be used in

$$\tilde{\mathbf{q}}_m = \mathbf{q}_{0,m} + \mathbf{Q}_m\mathbf{p}_m + \mathbf{F}_{q,m,m}\mathbf{z}_m. \quad (30)$$

4. EXAMPLES

4.1. Treble booster

As a first, relatively trivial example, we consider the treble booster circuit of Figure 1 with the component values given in Table 1. The circuit contains two non-linear elements, a diode and a transistor. But note that the formers sole purpose is to protect the circuit against connecting the power supply with wrong polarity. Usually, one would omit the diode from the simulation as it has no influence on the output signal. Here, we include it to verify that we can then eliminate it algorithmically.

We choose to put the diode first, so that \mathbf{q}_1 has two elements and $\mathbf{f}_1(\mathbf{q}_1)$ has one (see (5)), and \mathbf{q}_2 has four elements and $\mathbf{f}_2(\mathbf{q}_2)$ has two (see (7)). Using the ACME implementation of [7], we find

$$\mathbf{F}_q = \left(\begin{array}{c|ccc} 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ \hline 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & -2.917 \times 10^{-4} & 9.705 \times 10^{-5} & 0 \\ 0 & 9.705 \times 10^{-5} & -1.054 \times 10^{-4} & 0 \end{array} \right) \left. \begin{array}{l} \\ \\ \\ \\ \end{array} \right\} D_1 \left. \begin{array}{l} \\ \\ \\ \end{array} \right\} T_1 \quad (31)$$

where the lines demarcate the decomposition according to (25).

Apparently, $F_{q,1,2} = \mathbf{0}$ without any further modifications, and we can determine \mathbf{z}_1 from the diode equation alone. In this particular case, we do not even need \mathbf{z}_1 to determine \mathbf{z}_2 , as also $F_{q,2,1} = \mathbf{0}$, so we could just as well have put the diode second.

Using the method of [12] to determine index vectors \mathbf{p}_m of minimal dimension, we find \mathbf{p}_2 has two entries, while \mathbf{p}_1 has no entries at all. Thus, $\mathbf{f}_1(\mathbf{q}_1)$ and hence also \mathbf{z}_1 do not depend on values changing during simulation and can therefore be pre-computed offline.

To evaluate the performance impact, a guitar signal of 33.6 s duration sampled at 44.1 kHz is processed with the help of ACME

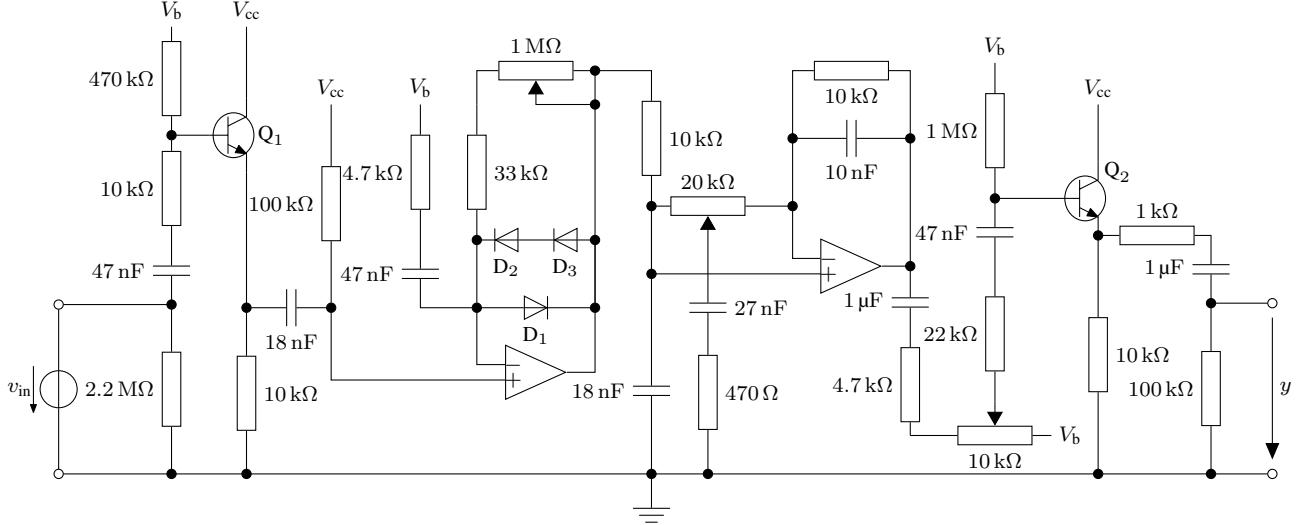


Figure 2: Example overdrive circuit.

v0.4.1 and the required processing time is measured using Julia v0.6.0 on an Intel Xeon E5-1620v2 CPU at 3.7 GHz. The non-linear equation is solved using Newton's method, where the initial solution is determined by extrapolating from the previous time step's solution using linearization (see [12] for details). No further pre-computing/caching of solutions is employed. As the circuit introduces relatively little distortion, only 1.789 iterations are required on average to refine the initial solution. While the original model needs 2.45 s to run, the decomposed one reduces the time to 1.99 s, an improvement by 19 %.

4.2. Overdrive

As a second example, we consider the more complex overdrive circuit of Figure 2 (based on “Der Super Over”⁴). Again, there is a diode anti-parallel to the supply voltage source as shown in Figure 3a which we include in the model. However, we simplify the bias voltage V_b generation; while the original circuit contains a voltage divider and a stabilizing capacitor (see Figure 3b), we enforce a constant bias voltage by directly connecting an ideal voltage source (see Figure 3c). We assume the operational amplifiers ideal, leading to a model with six non-linear elements: the protective diode anti-parallel to the supply voltage, three diodes in the feedback around the first operational amplifier, and two transistors. The non-linear equation system $\mathbf{f}(\mathbf{q}) = \mathbf{0}$ therefore comprises eight equations (one per diode, two per transistor), while vector \mathbf{q} has 16 entries (two per diode, four per transistor).

We again choose to put the protective diode first, then proceed

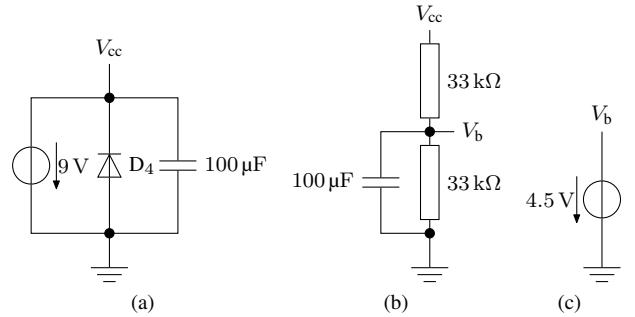


Figure 3: Power supply circuitry omitted from Figure 2 for (a) main supply voltage, (b) bias voltage, (c) simplified bias voltage.

left to right, and the ACME implementation of [7] yields

$$\mathbf{F}_q = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & * & * & 0 & 0 & 0 & 0 & 0 \\ 0 & * & * & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & * & * & * & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & * & * \\ 0 & * & * & * & 0 & 0 & * & * \end{pmatrix} \begin{cases} D_4 \\ Q_1 \\ D_2 \\ D_1 \\ D_3 \\ Q_2 \end{cases} \quad (32)$$

where * denotes non-zero entries whose exact values depend on the potentiometer settings. It can be observed that the first diode

⁴<http://diy.musikding.de/wp-content/uploads/2013/06/superoverschalt.pdf>

can again be extracted without problems. Continuing by just considering the remaining matrix, the first transistor can likewise be extracted without the need to modify \mathbf{F}_q . Now looking at the remaining submatrix

$$\left(\begin{array}{c|c|c|c|c} -1 & -1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ \hline 1 & 0 & 0 & 0 & 0 \\ * & 0 & 1 & 0 & 0 \\ \hline 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ \hline 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & * & * \\ * & 0 & 0 & * & * \end{array} \right) \left\{ \begin{array}{l} D_2 \\ D_1 \\ D_3 \\ Q_2 \end{array} \right\}$$

for the three diodes and the second transistor, none of the three diodes can be extracted by itself. Each of the three row pairs belonging to the three diodes obviously forms a matrix of rank 2, so we cannot possibly find a regular matrix with which to multiply from the right to zero out all but one of the columns. Likewise, the four lowest rows, belonging to the transistor, obviously form a rank-3 matrix, from which we cannot cancel all but two columns. As no single element can be extracted, the next thing to try is to extract pairs of elements. Again, none of the six possible pairs turn out to be extractable. But continuing with groups of three, the three diodes can be extracted as one group, without requiring modifications of \mathbf{F}_q .

We thus arrive at the decomposition

$$\mathbf{F}_q = \left(\begin{array}{c|c|c|c|c|c} 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ \hline 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & * & * & 0 & 0 & 0 \\ 0 & * & * & 0 & 0 & 0 \\ \hline 0 & 0 & 0 & -1 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & * & * & * & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ \hline 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & * \\ 0 & * & * & * & 0 & * \end{array} \right) \left\{ \begin{array}{l} D_4 \\ Q_1 \\ D_2 \\ D_1 \\ D_3 \\ Q_2 \end{array} \right\}, \quad (33)$$

comprising four subsystems of one, two, three, and two equations, respectively. Unlike the first example, the sub-diagonal blocks do contain non-zero entries, so the solution of earlier equations is required for the later ones (with the exception of the protective diode). Looking at the index vectors obtained with the method of [12], \mathbf{p}_1 again has no entries, so the solution to $\tilde{\mathbf{f}}_1(\tilde{\mathbf{q}}_1) = \mathbf{0}$ can be pre-computed off-line. Furthermore, \mathbf{p}_2 and \mathbf{p}_4 both have two entries and \mathbf{p}_3 just one, greatly simplifying the construction of lookup tables compared to a \mathbf{p} with five entries for the original (not decomposed) system.

Performing the same performance evaluation as for the treble booster example above, the processing time of the original model is determined as 6.82 s with 2.965 Newton iterations needed on average per sample. Decomposing the model reduces the number

of iterations needed to 1.460, 2.924 and 1.696 for the three sub-equations, respectively. The needed processing time however is reduced insignificantly to 6.79 s. Nevertheless, the advantage of making lookup tables feasible remains.

Unfortunately, the obtained results depend on the simplification of fixing the bias voltage V_b . With the original circuitry, all nodes connected to V_b influence each other, effectively creating a global feedback path, and the only decomposition possible is the extraction of the protective diode of the main power supply. In terms of the \mathbf{F}_q matrix, this manifests itself in additional entries in the fourth and last column that cannot be canceled and preclude further decomposition.

5. IMPLEMENTATION ASPECTS

The proposed method has been implemented as part of the ACME framework. One of the main challenges encountered is the condition $\mathbf{F}_{q,m,n} = \mathbf{0}$ which, when using floating point arithmetic, will usually only be fulfilled approximately. The obvious approach then is to treat entries with very small absolute value as zero. However, determining an appropriate threshold proves to be anything but trivial, as the scale of the entries in both \mathbf{q} and \mathbf{z} can differ by orders of magnitude from each other. So instead, we employ exact arithmetic, using rationals of arbitrary precision integers. This is possible as the whole model derivation process only needs a bounded number of additions, subtractions, multiplications, and divisions, so that the numerators and denominators may grow very large, but are still bounded. Of course, for the simulation itself, the values are converted to floating point for efficiency.

Unfortunately, the benefits reaped from the decomposition in terms of run-time turn out to be smaller than expected; sometimes, the decomposed model may even run slightly slower than the original one. The reason seems to be constant overhead that may dominate over the asymptotic behavior for small problem sizes. In an earlier implementation, where LAPACK routines were used for linear equation solving, this effect was even more pronounced. The constant calling overhead for the LAPACK routines is significant: Solving a system of size eight takes less than five times as long as solving a system of size one. We hope for a future, optimized implementation to further reduce these overheads, however.

6. CONCLUSION

The proposed method is able to decompose a system of non-linear equations into a number of smaller subsystems if that is possible in an exact way. This may lead to more efficient simulations in terms of computational load (when using iterative solvers) or memory requirements (when using lookup tables). However, the time needed to find this decomposition during model creation grows exponentially with the number of non-linear circuit elements. Fortunately, typically modeled audio effect circuits do not contain enough non-linear elements to make the approach infeasible.

The present paper does not, however, tackle the more challenging problem of automatically finding an approximate decomposition, which may either give a solution of sufficient accuracy for the complete system directly, or could at least be used to find a good initial solution for an iterative solver then applied to the complete system. We hope this paper to be a valuable step in that direction, however. E.g. for the overdrive circuit, numerical analysis could reveal that the bias voltage is almost constant, and that in fact fixing it at a constant does not change the output in a significant way. Based

on that, the proposed method could be applied as exemplified in Sec. 4.2.

7. REFERENCES

- [1] G. De Sanctis and A. Sarti, “Virtual analog modeling in the wave-digital domain,” *IEEE Trans. on Audio, Speech and Language Process.*, vol. 18, no. 4, pp. 715–727, 2010.
- [2] A. Bernardini, K. J. Werner, A. Sarti, and J. O. Smith, “Modeling a class of multi-port nonlinearities in wave digital structures,” in *23rd European Signal Process. Conf. (EUSIPCO)*, Nice, France, 2015, pp. 664–668.
- [3] K. Werner, V. Nangia, J. Smith, and J. Abel, “Resolving wave digital filters with multiple/multiport nonlinearities,” in *Proc. 18th Int. Conf. Digital Audio Effects (DAFx-15)*, Trondheim, Norway, 2015.
- [4] A. Falaize and T. Hélie, “Passive guaranteed simulation of analog audio circuits: A port-hamiltonian approach,” *Applied Sciences*, vol. 6, no. 10, 2016.
- [5] A. Falaize-Skrzak and T. Hélie, “Simulation of an analog circuit of a wah pedal: a port-hamiltonian approach,” in *135th Audio Engineering Society Convention*, New York, 2013.
- [6] 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 Trans. on Audio, Speech and Language Process.*, vol. 18, no. 4, pp. 728–737, 2010.
- [7] M. Holters and U. Zölzer, “A generalized method for the derivation of non-linear state-space models from circuit schematics,” in *23rd European Signal Process. Conf. (EUSIPCO)*, Nice, France, 2015, pp. 1078–1082.
- [8] M. Holters and U. Zölzer, “Circuit simulation with inductors and transformers based on the Jiles-Atherton model of magnetization,” in *Proc. 19th Int. Conf. Digital Audio Effects (DAFx-16)*, Brno, Czech Republic, 2016, pp. 55–60.
- [9] D. T. Yeh, J. S. Abel, and J. O. Smith, “Simplified, physically-informed models of distortion and overdrive guitar effects pedals,” in *Proc. 10th Int. Conf. Digital Audio Effects (DAFx-07)*, Bordeaux, France, 2007.
- [10] J Macak and J Schimmel, “Real-time guitar tube amplifier simulation using an approximation of differential equations,” in *Proc. 13th Int. Conf. Digital Audio Effects (DAFx-10)*, Graz, Austria, 2010.
- [11] J. Macak and J. Schimmel, “Real-time guitar preamp simulation using modified blockwise method and approximations,” *Eurasip J. on Advances in Signal Process.*, vol. 2011, 2011.
- [12] M. Holters and U. Zolzer, “A k-d tree based solution cache for the non-linear equation of circuit simulations,” in *24th European Signal Process. Conf. (EUSIPCO)*, Budapest, Hungary, Aug. 2016, pp. 1028–1032.
- [13] L. O. Chua and P.-M. Lin, *Computer-Aided Analysis of Electric Circuits: Algorithms and Computational Techniques*, Prentice-Hall, Englewood Cliffs, NJ, 1975.
- [14] M. Khorramizadeh and N. Mahdavi-Amiri, “An efficient algorithm for sparse null space basis problem using ABS methods,” *Numerical Algorithms*, vol. 62, no. 3, pp. 469–485, June 2012.

WDF MODELING OF A KORG MS-50 BASED NON-LINEAR DIODE BRIDGE VCF

Maximilian Rest

E-RM Erfindungsbüro
Berlin, Germany
m.rest@e-rm.de

Julian D. Parker

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

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

The voltage-controlled low-pass filter of the Korg MS-50 synthesizer is built around a non-linear diode bridge as the cutoff frequency control element, which greatly contributes to the sound of this vintage synthesizer. In this paper, we introduce the overall filter circuitry and give an in-depth analysis of this diode bridge. It is further shown how to turn the small signal equivalence circuit of the bridge into the necessary two-resistor configuration to uncover the underlying Sallen-Key structure.

In a second step, recent advances in the field of WDFs are used to turn a simplified version of the circuit into a virtual-analog model. This model is then examined both in the small-signal linear domain as well as in the non-linear region with inputs of different amplitudes and frequencies to characterize the behavior of such diode bridges as cutoff frequency control elements.

1. INTRODUCTION

Modeling of electrical circuits used in music and audio devices is a topic of ongoing research in the audio signal processing domain. One particular method for modeling such circuits is the Wave Digital Filter (WDF) approach [1]. Recent work sought to expand the class of circuits that can be modeled using WDFs, focusing mainly on topological aspects of circuits [2, 3], iterative and approximative techniques for non-linearities [4, 5], methods for op-amp-based circuits [6, 7] and diode clippers [8, 9].

These developments enabled modeling a large number of circuits that were previously intractable, including guitar tone stacks [2], transistor-based guitar distortion stages [3], transistor [2] and triode [10] amplifiers, the full Fender Bassman preamp stage [11], and the bass drum from the Roland TR-808 [12].

In this paper, we examine the low-pass filter of the MS-50 synthesizer manufactured by Korg. This synthesizer was released in 1978 and intended as a companion for the more famous MS-20 model. The MS-50's filter is known primarily for exhibiting strongly nonlinear behaviors. These behaviors are due to the usage of a diode ring (similar to that seen in ring-modulators) as the cutoff controlling element in the circuit, which generates a more and more complex spectrum with rising input signal level.

Diode rings in the modulator context have been previously examined as a simplified digital model with static non-linearities [13] and as a WDF [14]. Such configurations within a filter structure have not been studied, and it is that which motivates this work.

The following paper is structured as follows: Section 2 describes the Korg MS-50 voltage-controlled low-pass filter (VCF) circuit [15] and its core element, the diode bridge (Sec. 2.1) which is used to control the cutoff frequency. Based on a small signal linearization of this diode bridge, the Sallen-Key structure [16] of the circuit is pointed out (Sec. 2.2). In Section 3, the original circuit is

reduced to the diode bridge and surrounding filter components and turned into a WDF structure. Section 4 assesses transfer functions and nonlinear behaviors of a WDF implementation in the RT-WDF framework [17] and compares them with a SPICE [18] simulation. It also evaluates the performance of the iterative Newton-Raphson Solver [4] and gives an estimate on the expected real-time requirements. Section 5 concludes the work presented here.

2. KORG MS-50 FILTER CIRCUIT

As mentioned above, in contrast to the MS-20, whose VCF is analyzed in great detail in [19], the MS-50 features a rather non-standard VCF low-pass circuit based on a biased diode bridge as the voltage controllable element to vary the cutoff frequency. This configuration has been used in Korg's 700, 700S, 800DV and 900PS synthesizers before and was covered by patent US 4039980 [20]. A contemporary reinterpretation can be found in the AM-Synths Eurorack module AM8319 Diode Ring VCF [21].

Figure 1 shows the original schematic of the filter [15]. The diode bridge is based on the RCA CA 3019 diode array [22], which features a total of six matched silicon diodes. Left of the diode bridge is the biasing circuitry and input stage. Starting from the very left of the figure, voltages from external cutoff control, manual cutoff control and a temperature compensation are added and conditioned by IC_7 into a symmetric positive and negative biasing voltage, which is fed into the two adjacent ends of the bridge. The input signal is buffered by a unity gain amplifier made of IC_6 and capacitively coupled into the input node of the bridge. Right of the bridge is the actual low-pass filter circuitry. The output signal of the diode bridge is fed into a Sallen-Key filter structure [16] with controllable resonance, high gain and amplitude limiting clipping diodes D_5-D_{12} built around IC_4 . The output of this op-amp is then fed back into two additional taps in the diode bridge, which forms the necessary feedback path. This structure will be analyzed in greater detail in Section 2.2. In the rightmost part of the schematic, the signal is then differentially taken from two points in the output circuitry of the filter and passed through a DC blocking stage to the signal output. The nested configuration of diode bridge non-linearities in the feedback path of the op-amp IC_4 makes the MS-50 filter well suited for a topologically preserving modeling approach with WDFs.

2.1. Diode Bridge

The diode bridge around IC_5 in the filter structure acts as a controllable impedance element for both the input signal and the feedback path. This technique is already pointed out in the component's application note [23], and is consistent with the use of such circuits as a signal multiplier in radio applications.

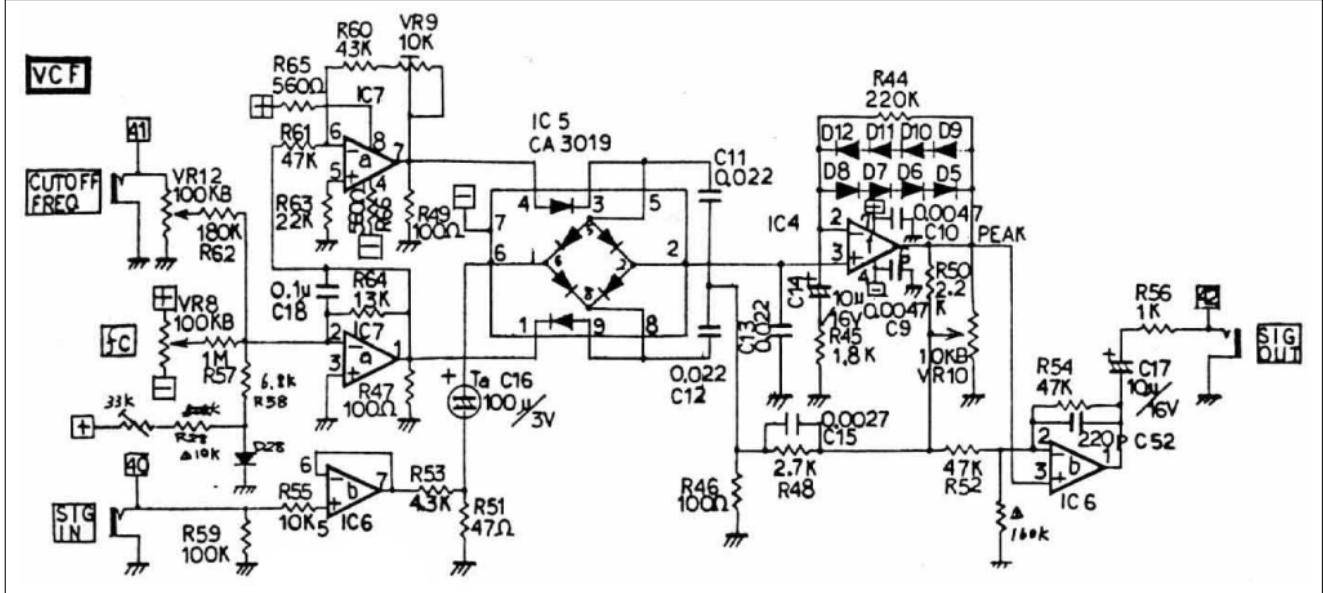


Figure 1: Original Korg MS-50 VCF circuit adopted from [15].

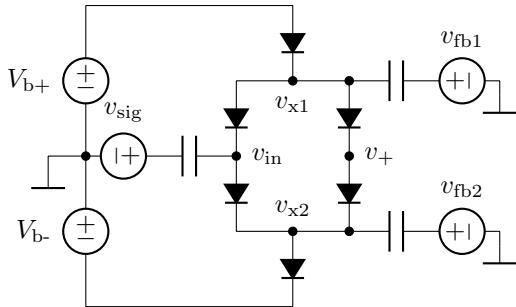


Figure 2: Voltage controllable diode bridge.

The basic idea of this topology is based around the small signal behavior of a single diode around its operating point and the resulting dynamic resistance r_d .

To derive an estimate of r_d for a single diode D , let us assume that the total voltage v_D across this diode can be written as a superposition of a constant biasing voltage V_b and a time varying small signal voltage v_s as

$$v_D = V_b + v_s, \quad (1)$$

with $v_s \ll V_b$. Substituting Eqn. (1) in the ideal diode equation $i_D = I_S \cdot e^{\frac{v_D}{n \cdot v_T}} - I_S$ and separating the constant term of the biasing current I_b yields

$$\begin{aligned} i_D &= I_S \cdot e^{\frac{V_b + v_s}{n \cdot v_T}} - I_S \\ &= I_S \cdot e^{\frac{V_b}{n \cdot v_T}} \cdot e^{\frac{v_s}{n \cdot v_T}} - I_S \\ &= I_b \cdot e^{\frac{v_s}{n \cdot v_T}} - I_S. \end{aligned} \quad (2)$$

Assuming also that $v_s \ll n \cdot v_T$ we can approximate Eqn. (2)

with a first order Taylor expansion around $v_s = 0$ V:

$$\begin{aligned} i_D &\approx I_b \cdot (1 + \frac{v_s}{n \cdot v_T}) - I_S = I_b + \frac{I_b \cdot v_s}{n \cdot v_T} - I_S \\ &= I_b + \frac{v_s}{r_d} - I_S \end{aligned} \quad (3)$$

In Eqn. (3), $r_d = \frac{n \cdot v_T}{I_b}$ is now defined as the dynamic resistance at the operating point set by V_b and thus the slope of the tangent at the operating point.

To apply these theoretical conclusions in practice, one must find a way to decouple the biasing voltage and the signal itself for superposition on the diode, as they almost always come from independent sources. One way to accomplish this is capacitive coupling. This is the method taken in the MS-50 filter. Figure 2 shows a simplified diode bridge as in the actual circuit but with ideal voltage sources for all connected voltages and the assumption that the capacitors are sufficiently large to have neglectable AC-impedance for the time-varying input and feedback signals.

Under these assumptions, the diode bridge network can be transformed into a linearized network of equivalent resistors r_{d1} - r_{d6} and further simplified into fewer resistors by calculating Thévenin equivalence resistances [24].

Figure 3a shows the linearized diode bridge with the bias sources V_{b+} and V_{b-} replaced by shorts. The dashed lines show the three different cases for calculating the equivalent resistances r_A , r_B , r_C and r_D seen by the sources v_{sig} , v_{fb1} and v_{fb2} towards nodes v_{x1} , v_{x2} and v_+ respectively as depicted in Fig. 3b. The equivalent resistances can be found as

$$r_A = ((r_{d4}||r_{d6}) + r_{d3} + r_{d5}) || r_{d1} || r_{d2} \quad (4)$$

$$r_B = ((r_{d1}||r_{d2}) + r_{d3} + r_{d5}) || r_{d4} || r_{d6} \quad (5)$$

$$r_C = (((r_{d2} + r_{d4}) || r_{d6}) + r_{d5}) || r_{d3} \quad (6)$$

$$r_D = (((r_{d2} + r_{d4}) || r_{d1}) + r_{d3}) || r_{d5} \quad (7)$$

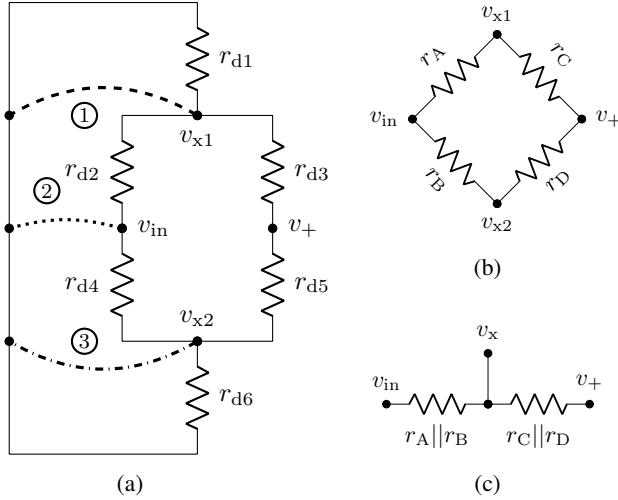


Figure 3: a) Linearized diode bridge to calculate Thévenin equivalence resistances for all three sources. (1) V_{fb1} replaced by a short to ground, (2) V_{sig} replaced by a short to ground, (3) V_{fb2} replaced by a short to ground; b) Thévenin equivalent resistances of linearized diode bridge; and c) simplified Thévenin equivalent resistors using symmetry.

If we further assume that the bridge is symmetrically biased and that the voltage sources v_{fb1} and v_{fb2} are equal as they are fed from the same branch in the original circuit ($v_{x1} = v_{x2}$), the pairs of r_A & r_B and r_C & r_D can be further summarized into two resistors with their respective parallel values (see Fig. 3c). Under the aforementioned assumptions we have thus transformed the diode bridge for small signals into two equivalent resistors and have further reduced the symmetric feedback branches into one common node. We will use these results in the next section to point out the Sallen-Key structure of the filter.

2.2. Sallen-Key Topology

Like the MS-20 filter from the same series of synths [19], the MS-50 VCF is also built around a Sallen-Key topology [16]. The Sallen-Key topology has only recently been the subject of study from a virtual analog perspective [25, 26, 27]. Figure 4 shows the relevant parts of the complete schematic with the linearized small signal diode bridge equivalent resistances substituted in. As the diode bridge is driven in the original circuit by two branches via C_{11} and C_{12} , they form a parallel capacitance in the linearized version.

From this circuit, the Sallen-Key topology with two resistive impedances in the forward path and two capacitive impedances on the op-amp input and feedback path is clearly visible, which is the characteristic configuration for a low-pass filter [16]. The variable resistor VR_{10} is additionally added to the standard configuration to control the amount of feedback. In the original schematic an additionally gain factor formed by R_{44} and R_{45} as well as clipping diodes around IC_4 are incorporated, which is omitted in Figure 4 for clarity.

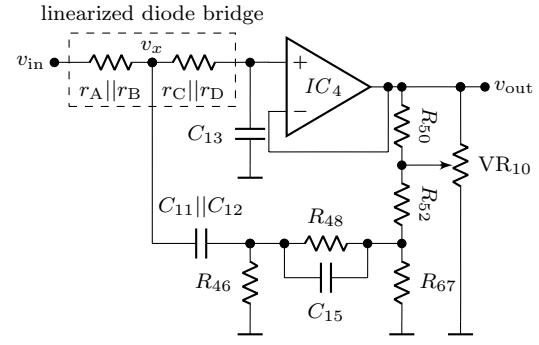


Figure 4: The filter's simplified, linearized Sallen-Key Structure.

3. WDF MODEL

Thanks to recent advances in the field of Wave Digital Filters [2, 3, 4, 6] it is possible to make a virtual analog model of such a diode bridge controlled filter including the non-linearities in the feedback path. Figure 5 shows the circuit which is used for modeling, the component values are given in Tab. 1.

Some parts of the original MS-50 schematic were simplified for this model, which are in the signal input, biasing and output stages. In the original circuit, the input signal is terminated by a simple load resistance of $R_{59} = 100\text{ k}\Omega$ followed by a unity gain amplifier. This stage is modelled as an ideal voltage source in the WDF. The second part which underwent simplifications is the biasing stage. In the original circuit, it consists of two op-amps in a summing configuration with weighted gains. These elements add external, manual and temperature compensation signals to a biasing voltage and its inverse. These voltages are symmetrically fed into the diode bridge. In the WDF model, the biasing op-amps are modelled as voltage sources with an output resistance of $R_{s+} = R_{s-} = 50\Omega$ and the biasing signal is treated as one composition of the individual original voltages. The next modification is performed around the main Sallen-Key op-amp IC_4 , which originally featured a couple of clipping diodes around its gain feedback path. These ones are omitted from this model to be able to independently study the behavior of the diode ring in the feedback path and its non-linear behavior. Finally, the output voltage is taken across an additional output load resistor R_L .

The WDF is created from this circuit using the MNA-based techniques described in [2, 3, 4, 6], especially for the complicated topology, the multiple non-linearities and the ideal op-amps. Figure 7 shows the resulting adaptor structure. All linear components are connected in branches to an \mathcal{R} -type adapter at the root. The non-linearities are arranged on the upper side of the adapter and the diodes are modelled using the Shockley model [28]. As there is no detailed datasheet for the CA 3019 diode array, parameter values of $I_s = 2.52\text{e}{-9}\text{ A}$ and $n = 1.752$ are used based on a 1N4148 fast small signal switching diode. The matrix of the \mathcal{R} -type describing the scattering behavior between all ports also incorporates the op-amps as Nullors, which results in ideal op-amp models [6]. The final adaptor structure is fully compatible with the current state of the RT-WDF Wave Digital Filter simulation library [17].

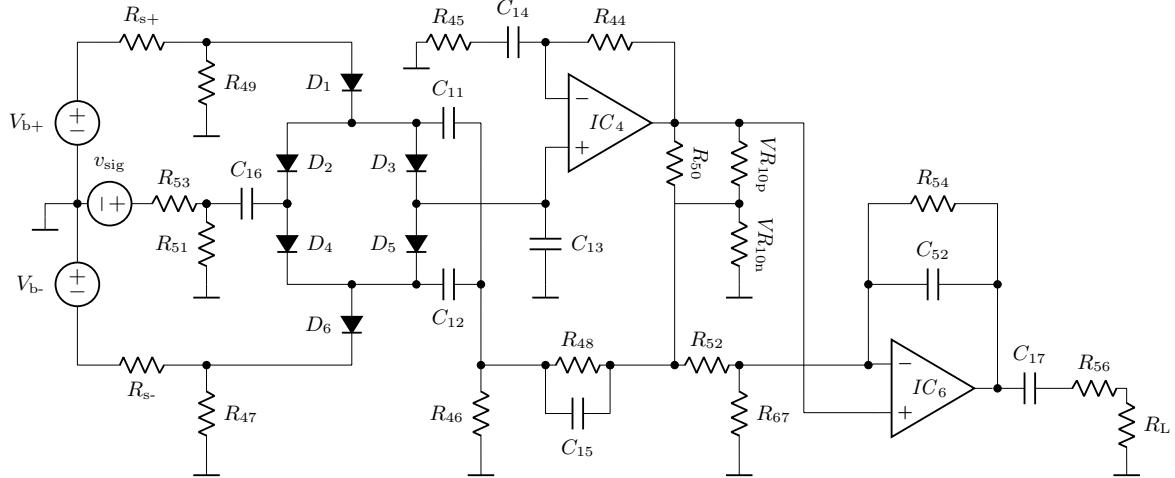


Figure 5: Simplified VCF circuit used for simulation.

Table 1: Component values used for simulation.

R_{s+}	50 Ω	R_{53}	4.3 kΩ	C_{16}	100 μF
R_{s-}	50 Ω	R_{54}	47 kΩ	C_{17}	10 μF
R_{44}	220 kΩ	R_{56}	1 kΩ	C_{52}	220 pF
R_{45}	1.8 kΩ	R_{67}	160 kΩ	D_1	1N4148
R_{46}	100 Ω	R_L	100 kΩ	D_2	1N4148
R_{47}	100 Ω	$VR_{10p} + VR_{10n}$	10 kΩ	D_3	1N4148
R_{48}	2.7 kΩ	C_{11}	22 nF	D_4	1N4148
R_{49}	100 Ω	C_{12}	22 nF	D_5	1N4148
R_{50}	2.2 kΩ	C_{13}	22 nF	D_6	1N4148
R_{51}	47 Ω	C_{14}	10 μF	IC_4	ideal
R_{52}	47 kΩ	C_{15}	2.7 nF	IC_6	ideal

4. RESULTS

The performance of the model is verified in three ways. Firstly, its linear behavior is measured by taking a small-signal impulse of 50 mV and processing it with both the WDF model and an equivalent SPICE model of the circuit from Fig. 5 in LTspice. This impulse produces a voltage on the right end of C_{16} of around 0.5 mV $\ll n \cdot v_T$, which keeps the diodes in an approximately linear region (see Sec. 2.1). The produced response thus characterizes the filter behavior in the linear region. Fig. 6 shows the magnitude responses of several such impulses for constant resonance setting and sweeping cutoff frequency. The match between the described WDF model and SPICE is good, with the exception of the presence of some small anomalies in the SPICE model, caused by the resampling algorithm used. The sampling rate of the WDF simulation was 176.4 kHz.

Secondly, the nonlinear behavior of the model is tested by examining its resonant behavior when driven by signals of varying amplitude. A sawtooth signal of 50 Hz is chosen for this purpose. Fig. 8 shows the output of the model and the equivalent SPICE model. Clearly visible with increasing input amplitude is the exaggeration of the initial transient of the sawtooth waveform while the following resonant behavior stays relatively constant. The output signals of the presented WDF model and SPICE are very close.

In order to further examine the variation in resonance frequency and amplitude with input level, a further test was per-

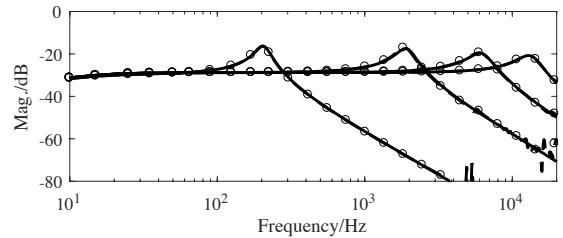


Figure 6: Mag. spectrum of calculated small-signal (50 mV) impulse responses of the circuit, at several cutoff frequencies. SPICE results are shown with a dotted line. Resonance is fixed at 0.7, bias voltage is set to [0.2, 0.3, 0.35, 0.38] V from left to right.

formed. The filter was given a resonance setting of 0.85, which produces self-oscillation. The filter input was then driven with a ramp waveform, peaking at 2 V, and the changing character of the self-oscillation observed. Fig. 9 shows the results of this process. The instantaneous frequency and amplitude of the output are estimated using a Hilbert transform of the output signal. The estimates are smoothed to remove higher frequency components that relate to the waveshape of the self-oscillation rather than the fundamental. The instantaneous frequency of the output is seen to decrease as the input ramp increases in voltage and thus re-biases the diode bridge, starting at around approximately 250 Hz and falling to approximately 200 Hz. The instantaneous amplitude of the output drops as the input voltage increases, with the self-oscillation almost completely damped as the input gets close to 2 V. Agreement between the SPICE and WDF models is close by these measures.

4.1. Performance

The maximum count of iterations for the six-dimensional nonlinear system in the Newton Solver with a stopping criteria of $\|\mathbf{F}\| \leq 1.5e-9$ did not exceed 2 at any time during calculation of the results presented here. No damping steps were needed. An increase of iterations was mostly noticed on sharp transients in the input signal and zero crossings of the output signal, both of which cause switching between the diodes.

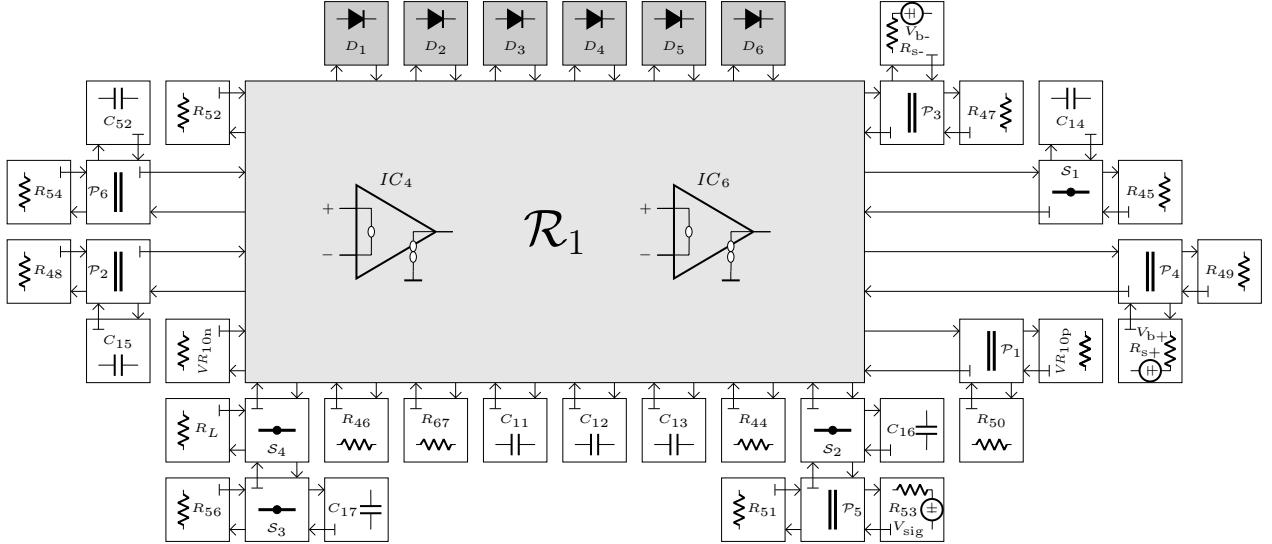


Figure 7: WDF Adaptor Structure.

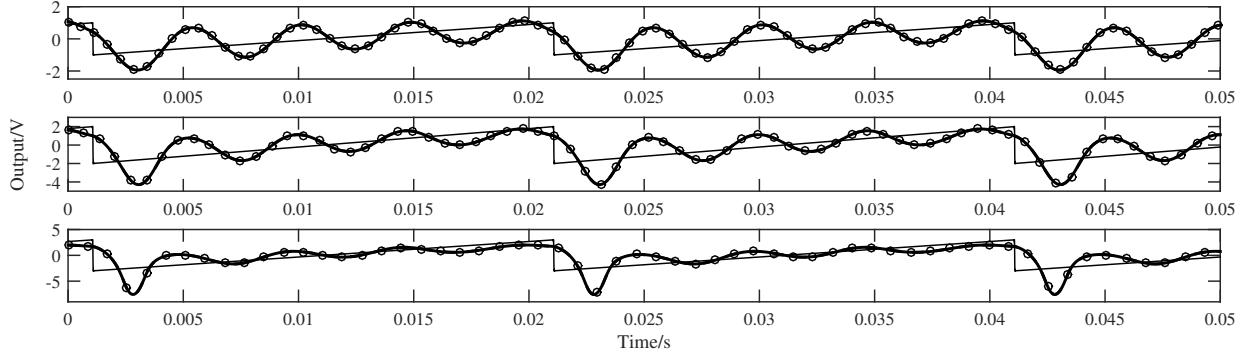


Figure 8: Output of model when driven by a 50 Hz sawtooth waveform of peak-to-peak amplitude of top: 1 V middle: 2 V, bottom: 3 V. SPICE results are shown with dots. Resonance is set to 0.8, and the bias voltage is 0.2 V. The input sawtooth is shown as a reference.

In terms of computational load, the implementation in RT-WDF [17] consumes approx. 80 % of one core at a sampling rate of 176.4 kHz on a laptop computer from 2013 with Intel i7 2.4 GHz CPU (4 cores) and 8 GB RAM running OS X 10.11.4. The *wdfRenderer* [17] was built with Apple LLVM 7.1 at optimization level -O3 and ran with a single rendering thread without any other considerable applications in the background or performance tweaking.

5. CONCLUSION

The circuit of the *Korg MS-50* voltage controlled filter was examined, and shown to fundamentally be a form of the Sallen-Key topology, controlled by a highly non-linear diode bridge. A Wave Digital Filter model of a simplified circuit was presented to highlight the influence of the diode bridge on the filter behavior and shown to match in most parts a reference implementation in SPICE. For small signal linear frequency analysis, small differences are mostly seen in terms of frequency warping effects in the WDF and aliasing artifacts in the SPICE results. Nonlinear frequency and amplitude variations exhibited by the filter are exam-

ined, with close agreement between the SPICE reference model and the WDF model shown. The model was implemented using the RT-WDF C++ framework, and is able to be run in real-time. Future research should analyze the original filter circuit behavior including the amplitude limiting clipping diodes and take advantage of the simplified diode bridge structure from Sec. 2 for a simplified model.

6. ACKNOWLEDGMENTS

Many thanks to Ross W. Dunkel for delightful initial discussions! Also, thanks to all reviewers for detailed comments.

7. REFERENCES

- [1] A. Fettweis, “Wave digital filters: Theory and practice,” *Proc. IEEE*, vol. 74, no. 2, Feb. 1986.
- [2] K. J. Werner, J. O. Smith III, and J. S. Abel, “Wave digital filter adaptors for arbitrary topologies and multiport linear

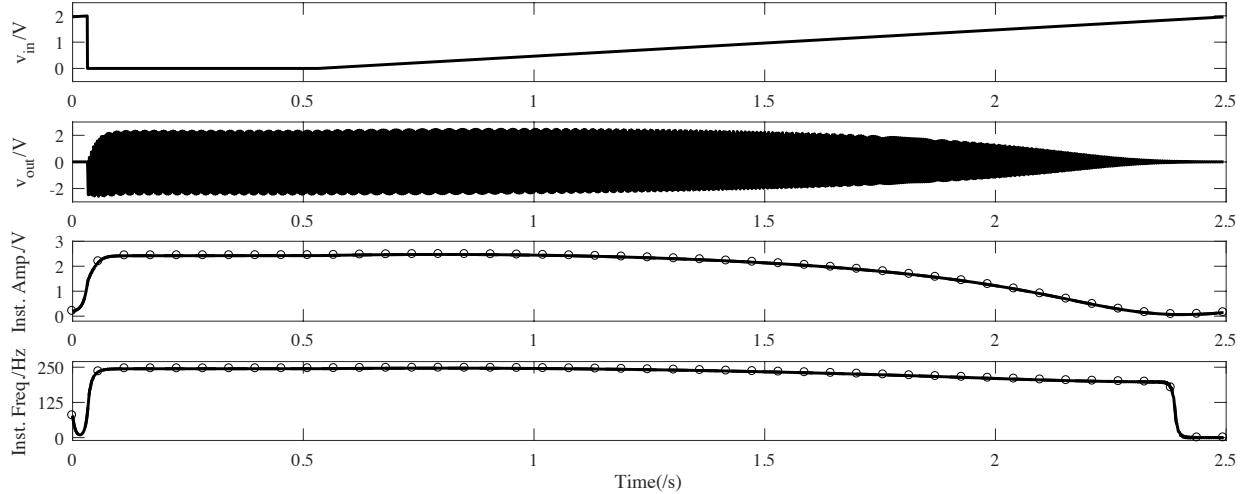


Figure 9: Output of models when set to self-oscillate with resonance of 0.85 and bias voltage of 0.2 V, driven by a slow 2 V ramp input. The input ramp v_{in} , model output v_{out} , and the estimated instantaneous frequency and amplitude of v_{out} are shown. SPICE results are shown with dots.

- elements,” in *Proc. Int. Conf. Digital Audio Effects (DAFx-15)*, Trondheim, NOR, Nov. 30 – Dec. 3 2015.
- [3] K. J. Werner, V. Nangia, J. O. Smith III, and J. S. Abel, “Resolving wave digital filters with multiple/multiport nonlinearities,” in *Proc. Int. Conf. Digital Audio Effects (DAFx-15)*, Trondheim, NOR, Nov. 30 – Dec. 3 2015.
 - [4] M. J. Olsen, K. J. Werner, and J. O. Smith III, “Resolving grouped nonlinearities in wave digital filters using iterative techniques,” in *Proc. Int. Conf. Digital Audio Effects (DAFx-16)*, Brno, CZ, Sept. 5–9 2016.
 - [5] A. Bernardini, K. J. Werner, A. Sarti, and J. O. Smith, “Modeling a class of multi-port nonlinearities in wave digital structures,” in *Proc. 23th Europ. Signal Process. Conf. (EUSIPCO)*, Nice, FR, Aug. 31 – Sept. 4 2015.
 - [6] K. J. Werner, W. R. Dunkel, M. Rest, M. J. Olsen, and J. O. Smith III, “Wave digital filter modeling of circuits with operational amplifiers,” in *Proc. 24th Europ. Signal Process. Conf. (EUSIPCO)*, Budapest, HU, Aug. 29 – Sept. 2 2016.
 - [7] 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, Oct. 2012.
 - [8] 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 (AES)*, New York, NY, USA, Oct. 29 – Nov. 1 2015.
 - [9] A. Bernardini and A. Sarti, “Biparametric wave digital filters,” *IEEE Trans. Circuits Syst. I: Reg. Papers*, 2017, To be published, DOI: 10.1109/TCSI.2017.2679007.
 - [10] M. Rest, W. R. Dunkel, K. J. Werner, and J. O. Smith, “RT-WDF—a modular wave digital filter library with support for arbitrary topologies and multiple nonlinearities,” in *Proc. Int. Conf. Digital Audio Effects (DAFx-16)*, Brno, CZ, Sept. 5–9 2016.
 - [11] 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. Int. Conf. Digital Audio Effects (DAFx-16)*, Brno, CZ, Sept. 5–9 2016.
 - [12] K. J. Werner, *Virtual Analog Modeling of Audio Circuitry Using Wave Digital Filters*, Ph.D. dissertation, Stanford University, Stanford, CA, USA, Dec. 2016.
 - [13] J. Parker, “A simple digital model of the diode-based ring-modulator,” in *Proc. Int. Conf. Digital Audio Effects (DAFx-11)*, Paris, FR, Sept. 19–23 2011.
 - [14] 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. II: Reg. Papers*, vol. 63, no. 8, Aug. 2016.
 - [15] KORG, *Model MS-50 Circuit Diagram*, Nov. 1978.
 - [16] R. P. Sallen and E. L. Key, “A practical method of designing RC active filters,” *IRE Trans. Circuit Theory*, vol. 2, no. 1, Mar. 1955.
 - [17] M. Rest, W. R. Dunkel, and K. J. Werner, “RT-WDF—a modular wave digital filter library,” 2016–2017, Available at <https://github.com/RT-WDF>, accessed Mar. 12 2017.
 - [18] A. Vladimirescu, *The SPICE Book*, John Wiley & Sons, New York, NY, USA, 1994.
 - [19] T. Stinchcombe, “A study of the Korg MS10 & MS20 filters,” Online, Aug. 30 2006, http://www.timstinchcombe.co.uk/synth/MS20_study.pdf, accessed Mar. 12 2017.
 - [20] Y. Nagahama, “Voltage-controlled filter,” Aug. 2 1977, US Patent 4,039,980.
 - [21] AMSynths on ModularGrid, “AM8319,” Online, Oct 30 2012, <https://www.modulargrid.net/e/amsynths-am8319>, accessed June 20 2017.
 - [22] RCA, *CA-3119 Ultra-Fast Low-Capacitance Matched Diodes*, Datasheet. RCA Corporation, Mar. 1970.

- [23] B. Brannon, *ICAN-5299 Application of the RCA-CA3019 Integrated-Circuit Diode Array*, RCA Linear Integrated Circuits and MOSFETS Applications. RCA Corporation, 1983.
- [24] A. S. Sedra and K. C. Smith, *Microelectronic Circuits*, Oxford University Press, New York, NY, USA, 5th edition, 2004.
- [25] J. Parker and S. D'Angelo, “A digital model of the Buchla lowpass-gate,” in *Proc. Int. Conf. Digital Audio Effects (DAFx-13)*, Maynooth, IE, Sept. 2–5 2013.
- [26] K. J. Werner, J. S. Abel, and J. O. Smith III, “The TR-808 cymbal: a physically-informed, circuit-bendable, digital model,” in *Proc. Int. Comput. Music Conf. (ICMC)*, Athens, EL, Sept. 14–20 2014.
- [27] M. Verasani, A. Bernardini, and A. Sarti, “Modeling Sallen-Key audio filters in the wave digital domain,” in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process. (ICASSP)*, New Orleans, LA, USA, Mar. 5–9 2017.
- [28] S. M. Sze, *Physics of semiconductor devices*, John Wiley & Sons, New York, NY, USA, 2nd edition, 1981.

COMPARISON OF GERMANIUM BIPOLAR JUNCTION TRANSISTOR MODELS FOR REAL-TIME CIRCUIT SIMULATION

Ben Holmes

Sonic Arts Research Centre,
School of Electronics, Electrical Engineering
and Computer Science
Belfast, U.K.
bholmes02@qub.ac.uk

Martin Holters

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

Maarten van Walstijn

Sonic Arts Research Centre,
School of Electronics, Electrical Engineering
and Computer Science
Belfast, U.K.
m.vanwalstijn@qub.ac.uk

ABSTRACT

The Ebers-Moll model has been widely used to represent Bipolar Junction Transistors (BJTs) in Virtual Analogue (VA) circuits. An investigation into the validity of this model is presented in which the Ebers-Moll model is compared to BJT models of higher complexity, introducing the Gummel-Poon model to the VA field. A comparison is performed using two complementary approaches: on fit to measurements taken directly from BJTs, and on application to physical circuit models. Targeted parameter extraction strategies are proposed for each model. There are two case studies, both famous vintage guitar effects featuring germanium BJTs. Results demonstrate the effects of incorporating additional complexity into the component model, weighing the trade-off between differences in the output and computational cost.

1. INTRODUCTION

Recent developments in processing power have allowed for real-time simulation of many audio circuits at a physical level, prompting the search for component models that achieve the highest accuracy whilst maintaining real-time compatibility. In the field of VA modelling, circuits featuring BJTs have been modelled successfully with a variety of different component models. Simplified models of the BJT are useful in applications featuring many BJTs. A notable case is exemplified in the modelling of the Moog Ladder filter in which current gain is presumed infinite, reducing a stage of two BJTs to one nonlinear equation [1]. In circuits featuring fewer BJTs, the large-signal Ebers-Moll model has been used [2]. This component model has been used in circuit models of complete guitar pedals, including wah-wah [3] and phaser effects [4].

Despite having been replaced by their silicon counterpart in most areas of electronics, vintage germanium BJTs (GBJTs) have remained consistently popular in guitar pedals, particularly fuzz effects. In previous work, two circuits featuring GBJTs have been studied: the Fuzz-Face [5] and Rangemaster [6] guitar pedals, both using the Ebers-Moll BJT model. However, each differs in how the parameters are derived: the Fuzz Face using parameters extracted from a datasheet, and the Rangemaster using parameters extracted through an optimisation procedure based on input/output data. Comparisons between the output of the circuits and the models demonstrate a good fit in certain regions of operation, though there are errors present in both models which remain unattributed. The component with the most complex behaviour in each circuit is the BJT which suggests it as a likely source of error.

A step further in complexity from the Ebers-Moll model is a possible solution to this issue. Significant improvements have been published, including: the Gummel-Poon model [7], the VBIC

model [8], and the MEXTRAM model [9]. These models have not yet featured in the field of VA, leaving the question of what difference they may make.

The aim of this paper is to provide an analysis of GBJTs within audio circuits for the purpose of VA modelling. This analysis consists of two primary sections: firstly the characterisation of models based on measurements, followed by the analysis of each model within the context of an audio circuit model. The analysed models include the Ebers-Moll model and models similar to Gummel-Poon, considering both additional DC and AC effects. Procedures for DC parameter extraction from measurements are discussed for all models. We revisit the case studies already presented: the Dallas Rangemaster Treble Booster and the Dallas Arbiter Fuzz-Face. In order to focus the analysis firmly on the differences between the BJT models, comparisons are made only between circuit models, as any separate circuit measurement would be subject to a range of further system variances and errors.

The rest of the paper is structured as follows: Section 2 describes the compared BJT models, Section 3 discusses extraction procedures for the DC parameters, Section 4 covers the case studies, the methodology, and results of the BJT model comparison, and Section 5 concludes with suggestions for modellers working on circuits featuring GBJTs.

2. BJT MODELS

This section describes the BJT models used in the analysis. Both GBJTs that are studied are PNP, which is reflected in the description of the models. The difference with an NPN model is only in notation, not behaviour.

In this work we define the term ‘external’ to refer to behaviour modelled by additional components i.e. resistors and capacitors. ‘Internal’ will refer to the remaining terms, modelled as voltage controlled current sources. This is illustrated by the differences between Figure 1 (a) and (b), in which the BJT in (b) is modelled by (a). External component values are modelled as being independent of the BJT bias point i.e. constant. Combination of the internal and external components will result in three models: the Ebers-Moll model, a DC Gummel-Poon model, and an AC Gummel-Poon (including capacitances).

Table 1 provides a reference for the name of each parameter in each model. The effect of changes in each parameter value will be discussed through the explanation of the extraction procedure in Section 3. Only necessary discussion is included about each BJT model; for a more comprehensive description see e.g. [10].

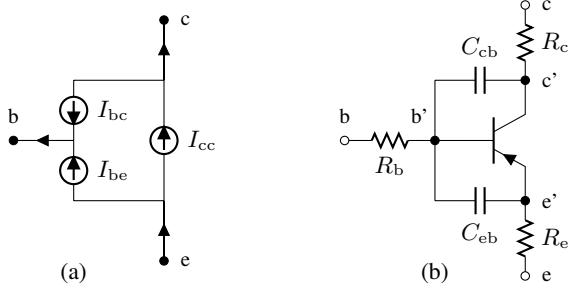


Figure 1: (a) Internal model schematic representation. (b) Complete model using additional components.

2.1. Ebers-Moll

The Ebers-Moll model can be understood as modelling the two *pn* junctions of the BJT as back to back diodes with coupling between junctions. This can be expressed as

$$I_f = I_s (e^{\frac{V_{eb}}{N_f V_t}} - 1), \quad I_r = I_s (e^{\frac{V_{cb}}{N_r V_t}} - 1) \quad (1)$$

$$I_{cc} = I_f - I_r, \quad I_{be} = \frac{1}{\beta_f} I_f, \quad I_{bc} = \frac{1}{\beta_r} I_r \quad (2)$$

$$I_c = I_{cc} - I_{bc}, \quad I_b = I_{be} + I_{bc} \quad (3)$$

where \$I_c\$ and \$I_b\$ are the currents through the collector and base, and \$V_{eb}\$ and \$V_{cb}\$ are the voltages across the emitter- and collector-base junctions. It is important to note that the remaining voltages and currents can be found by Kirchoff's circuit laws, i.e. \$V_{ec} = V_{eb} - V_{cb}\$ and \$I_e = I_c + I_b\$.

Intermediate terms are used in the description of the model to facilitate its extension and describe its behaviour in important regions of operation. \$I_f\$ and \$I_r\$ are forward and reverse currents which can be considered independent when the controlling voltage of the other current is zero. The terms \$I_{cc}\$, \$I_{be}\$ and \$I_{bc}\$ directly describe the schematic representation of the model, illustrated in Figure 1(a) as three current sources. Upon extension to the internal Gummel-Poon this representation remains the same, only requiring modification of the functions each current source represents.

2.2. Internal Gummel-Poon

To form the internal Gummel-Poon model several effects are added which change the behaviour in both low and high current regions, and also in response to changes in \$V_{ec}\$. Terms defined in (1) hold for the extended model whereas the intermediate current terms in (2) are replaced. The internal Gummel-Poon is then expressed as [11]

$$I_{be} = \frac{1}{\beta_f} I_f + I_{se} (e^{\frac{V_{eb}}{N_e V_t}} - 1) \quad (4)$$

$$I_{bc} = \frac{1}{\beta_r} I_r + I_{sc} (e^{\frac{V_{cb}}{N_c V_t}} - 1) \quad (5)$$

$$I_{cc} = \frac{2}{q_1(1 + \sqrt{1 + 4q_2})} (I_f - I_r) \quad (6)$$

where

$$q_1 = \frac{1}{1 - \frac{V_{eb}}{V_{ar}} - \frac{V_{cb}}{V_{af}}}, \quad q_2 = \frac{I_f}{I_{kf}} + \frac{I_r}{I_{kr}}. \quad (7)$$

The new terms in \$I_{be}\$ and \$I_{bc}\$ improve the modelling of the low current behaviour often referred to as leakage. A dependence of

\$I_c\$ on \$V_{ec}\$ is introduced through \$q_1\$ where two parameters control an increase to \$I_c\$ relative to the positive and negative values of \$V_{ec}\$ (consider the junction voltage that is dominant in each case). Finally, a current gain roll-off is introduced to the model through the inclusion of \$q_2\$ which reduces \$I_c\$ at high voltage values.

To reduce the internal Gummel-Poon to the Ebers-Moll model, parameters must be set to specific values: \$I_{se} = I_{sc} = 0\$ and \$V_{af} = V_{ar} = I_{kf} = I_{kr} = \infty\$.

2.3. External components

Five additional components are included in this work: a resistance at each terminal and a capacitance between both the base-emitter junction and the base-collector junction. Figure 1(b) illustrates the inclusion of these components to the internal BJT model.

More comprehensive models include two capacitances between each junction [10], both dependent upon the voltage across the junction; however, these are non-trivial to model using most VA techniques. Because of this, and also to reduce the difficulty in the measurement procedure, constant capacitance values were extracted from the datasheets of each GBJT.

3. PARAMETER EXTRACTION

DC parameter values for each model were extracted from measurements of GBJTs. The extraction strategy consists of a direct extraction followed by several targeted optimisations. This strategy is based on existing work [12, 13], but both of these tools are implemented in commercial systems which were unavailable. Therefore a straightforward approach was developed to operate on minimal measurements, with dedicated extraction procedures for the Ebers-Moll and Gummel-Poon models.

3.1. Measurement Strategy

Three sets of measurements were performed on each BJT: forward and reverse Gummel plots, which exposes the BJT behaviour in forward and reverse bias conditions, and the common-emitter output characteristic. Each measurement is designed to enable direct extraction of certain parameters by exposing specific behaviour. The circuits required for each measurement are illustrated in Figure 2. Table 2 contains the sourced currents and voltages for each measurement.

The common-emitter output characteristic is measured by specifying a base current, sweeping \$V_{ec}\$, and measuring \$I_c\$. This is repeated for different values of \$I_b\$ providing several snapshots of the relationship between \$V_{ec}\$, \$I_b\$ and \$I_c\$.

The forward Gummel plot fixes \$V_{ec}\$ at a positive bias while \$V_{eb}\$ is swept over a range and both \$I_c\$ and \$I_b\$ are measured. This is similar for the reverse Gummel plot, where \$V_{ec}\$ is biased negatively, \$V_{cb}\$ is swept, and \$I_e\$ and \$I_b\$ are measured. The applied methodology is described in [12], where \$V_{ec}\$ is biased between 2 V and half of the maximum voltage supplied in the common-emitter output characteristic. This creates a direct relationship between the common-emitter characteristic at that value of \$V_{ec}\$ and the Gummel plots, such that the voltages and currents should all be equal. This approach therefore biases the BJT in the active regions which provides confidence that the model will fit all measurements.

A Keithley 2602B Source Measure Unit was used to perform the measurements, enabling simultaneous measurement and sourcing of current and voltage. However, it should be noted that the Gummel plots can be measured using only voltages by placing a

Table 1: List of all parameters and extracted values from both the OC44 and AC128; constraints used in the intermediate optimisation stages; initial values used for parameters that were not found through direct extraction.

Parameter	Extracted Values				Optim. Constraints		Init. Values
	OC44		AC128		Lower Lim.	Upper Lim.	
	E.M.	G.P.	E.M.	G.P.			
I_s	Saturation current	2.029 μA	1.423 μA	23.75 μA	20.66 μA	-	-
β_f	Forward current gain	108.1	307.0	44.90	229.6	50	250
β_r	Reverse current gain	8.602	20.27	4.568	14.66	3	20
N_f	Forward ideality factor	1.062	1.022	1.168	1.133	-	-
N_r	Reverse ideality factor	1.105	1.025	1.171	1.140	-	-
(V_t)	Thermal voltage	25.5 mV	25.5 mV	25.5 mV	25.5 mV	-	-
V_{af}	Forward Early voltage	-	8.167 V	-	19.68 V	-	-
V_{ar}	Reverse Early voltage	-	14.84 V	-	88.28 V	-	-
I_{kf}	Forward knee current (gain roll-off)	-	43.82 mA	-	463.0 mA	10 μA	500 mA
I_{kr}	Reverse knee current (gain roll-off)	-	611.7 mA	-	241.5 mA	10 μA	500 mA
I_{se}	BE junction leakage current	-	30.54 nA	-	2.190 μA	0.1 fA	1 mA
I_{sc}	BC junction leakage current	-	213.5 nA	-	7.546 μA	0.1 fA	$I_s/2$
N_e	BE junction leakage emission coefficient	-	1.316	-	1.796	0.5	4
N_c	BC junction leakage emission coefficient	-	1.258	-	1.364	0.5	4
R_b	Base resistance	-	32.83 Ω	-	1.885 Ω	1 Ω	250 Ω
R_e	Emitter resistance	-	968.7 m Ω	-	306.4 m Ω	0.1 n Ω	2 Ω
R_c	Collector resistance	-	989.9 $\mu\Omega$	-	1.727 $\mu\Omega$	-	10 m Ω
C_{eb}	Emitter-base capacitance	-	410 pF	-	-	-	-
C_{cb}	Collector-base capacitance	-	10 pF	-	100 pF	-	-

Table 2: Ranges of the inputs to each measurement circuit. Specific values of I_b are provided on each measurement plot.

Meaurement	Input	OC44	AC128
Forward Gummel	V_{eb}	0 - 0.7 V	0 - 0.8 V
	V_{ec}	2 V	2 V
Reverse Gummel	V_{cb}	0 - 0.8 V	0 - 0.8 V
	V_{ec}	-2 V	-2 V
Common Emitter	I_b	3 - 50 μA	26 - 1000 μA
	V_{ec}	-5 - 5 V	-5 - 5 V

shunt resistor over which to measure the voltage drop. This is important as it reduces the equipment required for measurements, and as will be shown provides enough information to characterise the Ebers-Moll model.

3.2. Direct Extraction

Direct extraction of parameters is used to provide initial estimates upon which optimisation can then be performed. Estimates can be made using simplifications as the optimisation performs the majority of the extraction. However, it is essential to start the optimisation process in a position within the parameter space close to the optimum to avoid local minima which may halt the optimisation without providing the best model fit.

3.2.1. Ebers-Moll parameters

The Ebers-Moll parameters are extracted from the measured Gummel plots. Figure 3 illustrates the effects of the forward parameters and the saturation current I_s . This behaviour is equivalent in the reverse plot with the reverse parameters meaning that both regions can be described by analysing only one, in this work the forward region. To simplify the extraction procedure, the opposite current term, in this case, I_r is neglected, which is only valid if $V_{cb} = 0$. As the measurement strategy actually enforces $V_{cb} < 0$ there will

be an error introduced into the direct extraction, but the error is small, typically $I_r \leq I_s$. Further, this error is removed during the optimisation stages where there is no model simplification.

The thermal voltage V_t can be found directly through measuring the temperature of the room in which the measurement is taken. This relies on the assumption that the measurements are taken in such a way that avoids the BJT being heated by the current passing through it, and that the BJT is allowed to settle at room temperature prior to measurement. Using the temperature in kelvin T_K , $V_t = T_K \frac{k}{q}$ where k is Boltzmann's constant and q is the charge on an electron.

Following this, the ideality factor N_f can be found through finding the gradient of the log of I_c . While the model shown in Figure 3 is ideal, in measurements of BJTs the gradient of I_c will not be constant so it is important to find a suitable point at which to perform the extraction. One suitable method is to find the first minimum of the absolute value of the second derivative of I_c . Neglecting constant terms from I_f in (1) allows the formation of the expression

$$\frac{d\log(I_c)}{dV_{eb}} = \frac{1}{N_f V_t}. \quad (8)$$

Rearranging this equation provides a value for N_f . A value for I_s can then be found at the same value of V_{eb} , by solving the simplified expression of I_c for I_s , i.e.

$$I_c = I_s \exp\left(\frac{V_{eb}}{N_f V_t}\right), \quad I_s = \exp\left(\log(I_c) - \frac{V_{eb}}{N_f V_t}\right). \quad (9)$$

Examining this equation for when $V_{eb} = 0$ it is clear that I_s is the y-intercept of the Gummel-plot, as illustrated in Figure 3.

The extraction of β_f relies on the relationship between I_c and I_b , which from (1-3) can be expressed as $I_c = \beta_f I_b$. This relationship is illustrated in Figure 4. As V_{eb} approaches zero, I_b decreases such that β_f increases very rapidly. This does not provide practical values of β_f so values of β_f beneath the first turning point with respect to V_{eb} can be excluded. The maximum of the β_f is then used as the directly extracted parameter value.

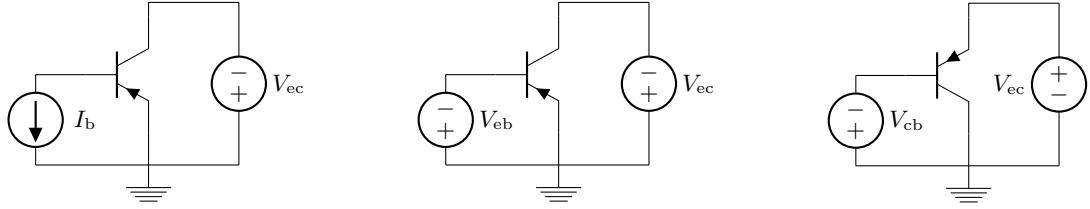


Figure 2: Measurement circuits for parameter extraction. Common-Emitter (left), Forward Gummel (middle), Reverse Gummel (right).

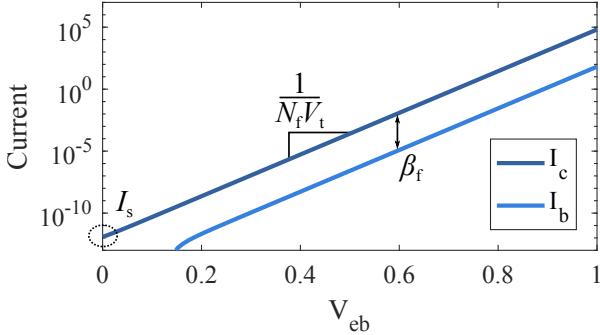


Figure 3: Example forward Gummel plot of the Ebers-Moll model illustrating the effects of the forward parameters N_f , β_f and also I_s . Currents I_c and I_b are plotted logarithmically in the current range against linear V_{eb} .

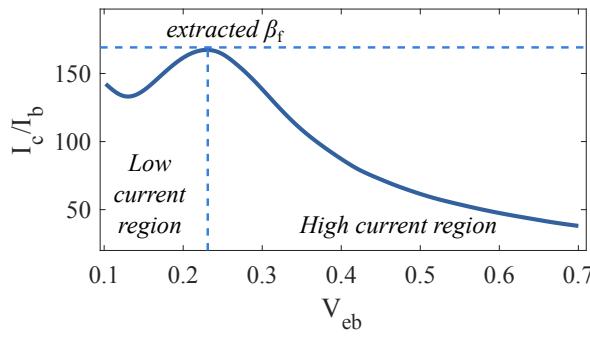


Figure 4: Example plot showing the relationship between I_c/I_b and V_{eb} . A nominal value of β_f is extracted from the maximum value, shown by the dashed line. High and low current effects cause reduction of β_f .

3.2.2. Gummel-Poon parameters

Values for I_{kf} and I_{kr} can be extracted from the current gain plots, the forward of which is illustrated in Figure 4. The extracted values for I_{kf} and I_{kr} are given by the value of I_c and I_e at which the current gain falls to half of its maximum value (respectively). If necessary the curve can be extrapolated to find a value.

It was not found necessary to implement a direct extraction technique for each parameter as initial values could be found through manually tuning the leakage parameters and terminal resistances. Due to the similarity of the GBJs being modelled, several parameters could be initialised at the same value for each GBJT and provide a good starting point for the optimisation stage of the extraction. The values for each parameter initialised using this method are shown in Table 1.

3.3. Optimisation Stages

After completing the direct extraction, each model's fit to measurements was optimised. Each optimisation was performed on a specific input range, selected to emphasise the relevant behaviour for each set of parameters being optimised. The ranges used can be found in Table 3. Different strategies were required for characterising the Ebers-Moll model and DC Gummel-Poon.

Two optimisation algorithms were used from MATLAB's optimisation toolbox, `fminsearch` which uses the Nelder-Mead simplex method [14], and `fmincon`, which uses the interior-point method [15]. The Nelder-Mead simplex method is useful in this scenario due to its ability to handle discontinuous surfaces. This enabled the use of objective functions that would return an infinite value if the parameters supplied were negative, preventing non-physical parameter sets. Experimentally it was found that this combination provided better convergence properties than using the interior-point method with a similar boundary. However, the interior-point method was useful in scenarios in which more complex boundaries were required to ensure the parameters retrieved would be a suitable starting point for the next simulation. The final stage of characterising each model was performed with the Nelder-Mead simplex method.

The same objective function was used for each optimisation with the only change being the data compared. The objective value is normalised with respect to both the number of data points and the values of the data points. An example of this can be expressed by

$$R(\theta, y) = \frac{1}{N} \sum_{n=1}^N \left(\frac{y[n] - \hat{y}(\theta)[n]}{y[n]} \right)^2 \quad (10)$$

where R is the objective value, y is the measured data, $\hat{y}(\theta)$ is the modelled data for a given parameter set θ , and N is the number of data points.

3.3.1. Ebers-Moll

Following the direct extraction, one optimisation stage was used in the extraction of the Ebers-Moll parameters. This was performed on the Gummel plots, using a low voltage range (see Table 3) to match the first 'ideal' region in which the gradient of the collector current is constant.

3.3.2. Gummel-Poon

The optimisation procedure for the DC Gummel-Poon model is illustrated in Figure 5. After the direct extraction stage, three stages of optimisations are used. The intermediate optimisation stages use constraints implemented using the interior-point method. Constraints for the intermediate optimisations can be seen in Table 1.

The first optimisation stage works on the current gain of the BJTs, given by e.g. I_c/I_b for the forward case. This significantly

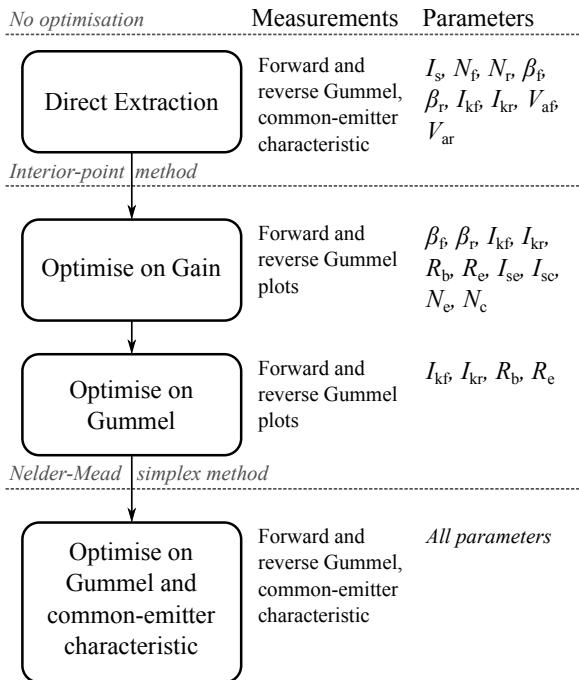


Figure 5: Optimisation strategy used to extract the parameters of the Gummel-Poon model. The arrow indicates the flow of stages of extraction. The resultant parameters of each stage are passed on to the next.

reduces the effects of I_s , V_{af} , V_{ar} , R_c , N_f , N_r , reducing the parameter set and thus the number of dimensions in the optimisation. A second stage was used to further tune a subset of these parameters to the Gummel plots. Finally, all parameters were optimised against all of the data sets. A weighting was applied to the objective function in the final optimisation, making the objective value of the common-emitter characteristic 10× higher than that of the Gummel plots.

3.4. Results

Multiple GBJT s were used in the comparison: 4 AC128 and 3 OC44 BJTs. Figures 6 and 7 show the results of the best fit of the DC Gummel-Poon to the measurements as determined by the objective function of the final optimisation. As these models had the best fit to the data, they were used in the comparison in the case studies. See Table 1 for the parameters of each model.

Thermal effects were noticed in the measurements despite considerations to reduce the effects. Post processing was used to reduce the observable effect of these; however there remains a possibility that some thermal error remains in the measurements which would affect the extracted parameters.

In the high-current region of the Gummel-plots the model deviates from the measurements. During the measurement stage, a current limit was enforced to prevent damage to components or the measurement system, set slightly above the maximum current specified by the GBJT datasheets. This limited the amount of high-current data that could be gathered. Further, the common-emitter plots were taken at low currents to ensure they were close to that of the ‘ideal’ region of operation, which reduces the amount of data about the high-current region. If the measurements were to

Table 5: Voltage ranges upon which each optimisation for both models were performed. Gummel plots were used in both the penultimate and ultimate stages for the Gummel-Poon model, and are labelled 1 and 2 to differentiate.

Model	Measurement	Input	Lower limit	Upper limit
Ebers-Moll	Gummel plots	V_{eb}, V_{cb}	10 mV	200 mV
Gummel-Poon	Current gain	V_{eb}, V_{cb}	110 mV	700 mV
	Gummel plots 1	V_{eb}, V_{cb}	100 mV	700 mV
	Gummel plots 2	V_{eb}, V_{cb}	50 mV	600 mV
	Common-emitter	V_{ec}	-5 V	5 V

be repeated it might be sensible to increase the base currents in the common-emitter characteristic, and consider increasing the current limit.

4. CASE STUDIES

The schematics of the Dallas Rangemaster and Dallas Arbiter Fuzz-Face can be found in Figures 8 and 9 respectively. These case studies were selected because each biases the BJT in different regions, exhibiting different behaviour. Three tests were performed on each case study: informal listening tests, waveform comparisons, and a comparison of the computational requirements. Each circuit was modelled using the Nodal DK-method (for reference see e.g. [3]) although it should be noted that the use of a different simulation technique would yield similar results for both audio and waveform comparisons. Computation time would however require evaluation for different simulation techniques. For each test all potentiometers of both case studies were set to maximum. Other potentiometer settings were tested but are omitted as they illustrate no substantial difference from those presented.

4.1. Informal listening tests

Several guitar signals were processed by both case studies at 8× oversampling as a means of comparing each model. Listeners agreed that differences could be heard between each model, with the Ebers-Moll model having the most high frequency content due to distortion and the AC Gummel-Poon having the least. Audio examples can be found on the first author’s website¹.

4.2. Waveform comparison

An objective comparison of each BJT model is achieved here using time-domain waveforms. Sinusoids at different frequencies and amplitudes were processed by both case studies and each model. To remove transient behaviour from the results, the final period of each of these signals are shown in Figure 10 and 11. Plots at 1200 Hz show the largest difference for the AC effects, illustrating the low-pass type behaviour of the capacitances. Differences due to the increased DC complexity are most prominent at lower amplitudes.

¹<http://bholmesqub.github.io/DAFx17/>

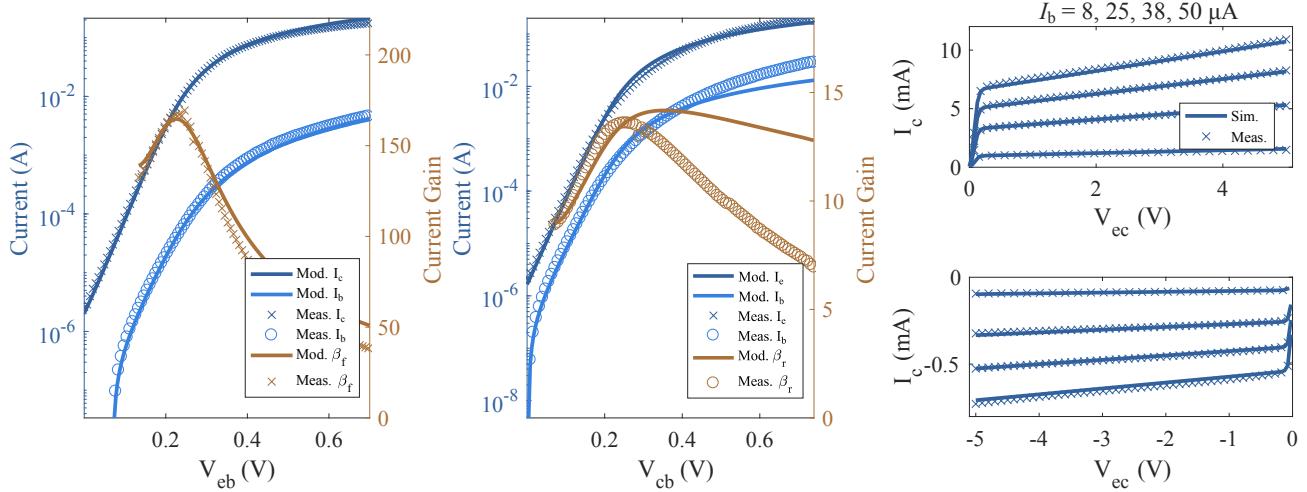


Figure 6: Forward and reverse Gummel plots and the common-emitter characteristic of the OC44 BJT.

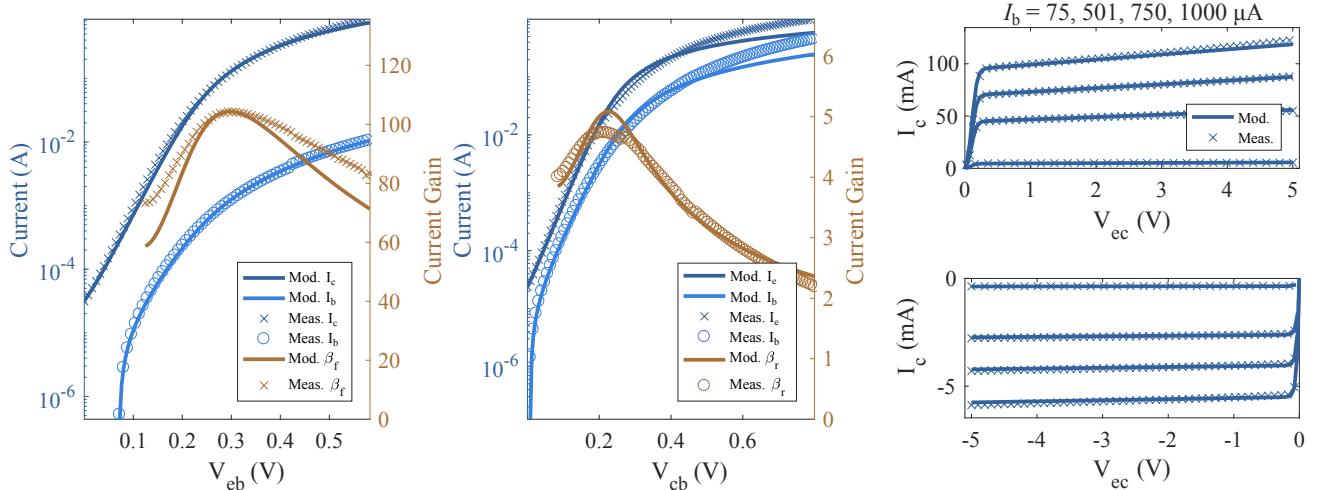


Figure 7: Forward and reverse Gummel plots and the common-emitter characteristic of the AC128 BJT.

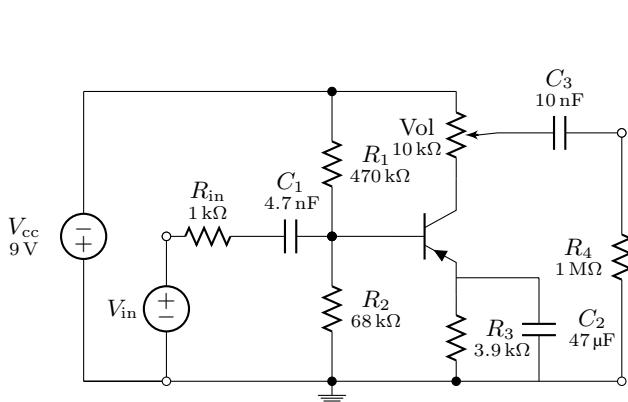


Figure 8: Schematic of the Rangemaster circuit.

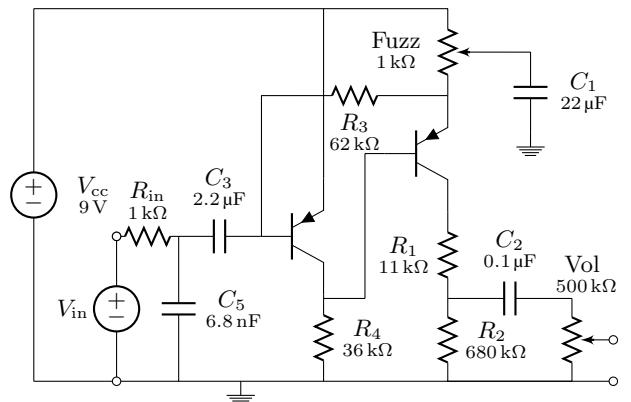


Figure 9: Schematic of the Fuzz Face circuit.

Table 4: Simulation time required to process one second of signal, average iterations per sample, and sub-iterations per sample of circuit models processing a guitar chord using different BJT models. The Rangemaster was tested over a peak voltage range of 0.1 – 2 V, the Fuzz-Face over a range of 10 – 100 mV.

Model	Rangemaster		Fuzz-Face	
	Sim. time/s	Mean Iter./Sub-iter. (ms)	Sim. time/s	Mean Iter./Sub-iter. (ms)
DC E.M.	95.9	3.53/0.20	341.6	3.62/0.04
AC E.M.	75.8	3.52/0.13	376.5	3.60/0.01
DC G.P.	371.8	3.47/0.07	819.1	3.04/0.03
AC G.P.	357.0	3.45/0.05	769.9	2.99/0.01

4.3. Computational efficiency

To understand the cost of increasing the complexity of the BJT model, computational requirements of each model were compared. The nonlinear equation of the circuit models was solved using damped Newton's method as described in [16], which uses an inner iterative loop to aid in convergence. This provides three metrics: time needed for one second of simulation, average iterations, and average sub-iterations.

A fourth model was included for this test: the Ebers-Moll model with C_{eb} and C_{cb} (AC Ebers-Moll) to provide an improved assessment of the cost of the capacitances. A guitar signal was processed by both case studies and each BJT model, with the peak amplitude of the signal set to 20 different levels. Computation time was then measured by MATLAB's `tic/toc` stopwatch functions. The results are shown in Table 4. It is clear from the results that increasing the DC complexity causes a significant increase in computation time, whereas including additional capacitances carries little cost. As iterations and sub-iterations decrease with increasing model complexity the increase in computation must be due to the increased complexity of evaluating the model equations. Decrease in computation cost when including the capacitances can be attributed to the reduction in high frequencies reducing the stress placed on the iterative solver, outweighing the increase in the complexity from including additional components.

5. CONCLUSION

A comparison of BJT models has been presented with a focus on GBJTs. Each model was characterised by extracting parameters from measured data using a multi-step optimisation strategy. The resultant models were compared through the use of two case study circuits covering both moderately and highly distorted circuit outputs. The circuit models were compared using three metrics: audible and waveform differences, and computational efficiency.

Results show that increase in model complexity does make a change to the behaviour of GBJTs in audio circuit models. This work has primarily focused on improving DC characterisation; however, the results show that AC effects are at least equally important. The improved DC characterisation has a significant increase in computational cost whereas the cost of the AC effects are minimal. These results indicate that any first extension to the Ebers-Moll model should be AC effects, and further extensions should then concern DC effects.

The core motivating factor for implementing and characteris-

ing more sophisticated BJT models was to reduce the error present in VA circuits featuring GBJTs. A fitting next step is to now use these models in conjunction with the design of models based on specific circuits. An implementation of the Gummel-Poon model has been included in ACME.jl² emulation tool for modellers interested in further investigation.

6. REFERENCES

- [1] A. Huovilainen, “Non-linear digital implementation of the Moog ladder filter,” in *Proc. of the International Conference on Digital Audio Effects (DAFx-04)*, Naples, Italy, 2004, pp. 61–64.
- [2] J. J. Ebers and J. L. Moll, “Large-signal behavior of junction transistors,” *Proceedings of the IRE*, vol. 42, no. 12, pp. 1761–1772, 1954.
- [3] M. Holters and U. Zölzer, “Physical Modelling of a Wah-Wah Pedal as a Case Study for Application of the Nodal DK Method to Circuits with Variable Parts,” in *Proc. of the 14th International Conference on Digital Audio Effects*, Paris, France, Sept. 2011, pp. 31–35.
- [4] F. Eichas, M. Fink, M. Holters, and U. Zölzer, “Physical Modeling of the MXR Phase 90 Guitar Effect Pedal,” in *Proc. of the 17th International Conference on Digital Audio Effects*, Erlangen, Germany, Sept. 2014, pp. 153–156.
- [5] K. Dempwolf and U. Zölzer, “Discrete State-Space Model of the Fuzz-Face,” in *Proceedings of Forum Acusticum*, Aalborg, Denmark, June 2011, European Acoustics Association.
- [6] B. Holmes and M. van Walstijn, “Physical model parameter optimisation for calibrated emulation of the Dallas Rangemaster Treble Booster guitar pedal,” in *Proc. of the 19th International Conference on Digital Audio Effects*, Brno, Czech Republic, Sept. 2016, pp. 47–54.
- [7] H. K. Gummel and H. C. Poon, “An integral charge control model of bipolar transistors,” *Bell System Technical Journal*, vol. 49, no. 5, pp. 827–852, 1970.
- [8] C. C. McAndrew, J. A. Seitchik, D. F. Bowers, M. Dunn, M. Foisy, I. Getreu, M. McSwain, S. Moinian, J. Parker, D. J. Roulston, and others, “VBIC95, the vertical bipolar inter-company model,” *IEEE Journal of Solid-State Circuits*, vol. 31, no. 10, pp. 1476–1483, 1996.
- [9] R. Van der Toorn, J. C. J. Paasschens, and W. J. Kloosterman, “The Mextram bipolar transistor model,” *Delft University of Technology, Technical report*, 2008.
- [10] I. Getreu, *Modeling the Bipolar Transistor*, Tektronix, 1976.
- [11] A. Vladimirescu, *The SPICE book*, J. Wiley, New York, 1994.
- [12] F. Sischka, “Gummel-Poon Bipolar Model: Model description, parameter extraction,” *Agilent Technologies*, 2001.
- [13] J. A. Seitchik, C. F. Machala, and P. Yang, “The determination of SPICE Gummel-Poon parameters by a merged optimization-extraction technique,” in *Proc. of the 1989 Bipolar Circuits and Technology Meeting*, 1989, pp. 275–278, IEEE.
- [14] J. C. Lagarias, J. A. Reeds, M. H. Wright, and P. E. Wright, “Convergence properties of the Nelder–Mead simplex method in low dimensions,” *SIAM Journal on optimization*, vol. 9, no. 1, pp. 112–147, 1998.
- [15] R. H. Byrd, J. C. Gilbert, and J. Nocedal, “A Trust Region Method Based on Interior Point Techniques for Nonlinear Programming,” *Mathematical Programming*, vol. 89, no. 1, pp. 149–185, 2000.
- [16] B. Holmes and M. van Walstijn, “Improving the robustness of the iterative solver in state-space modelling of guitar distortion circuitry,” in *Proc. of the 18th International Conference on Digital Audio Effects*, Trondheim, Norway, Dec. 2015, pp. 49–56.

²<https://github.com/HSU-ANT/ACME.jl>

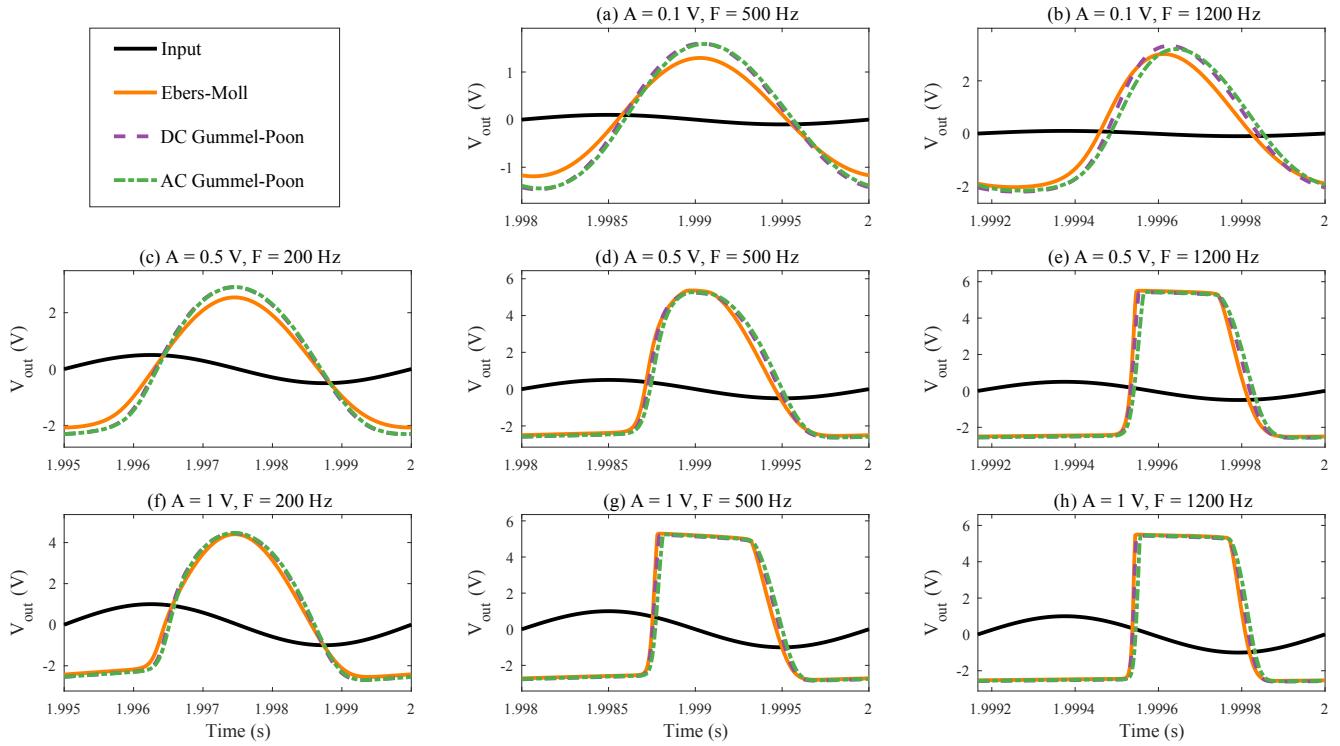


Figure 10: Single cycles of the Rangemaster output's response to a sinusoidal input signal.

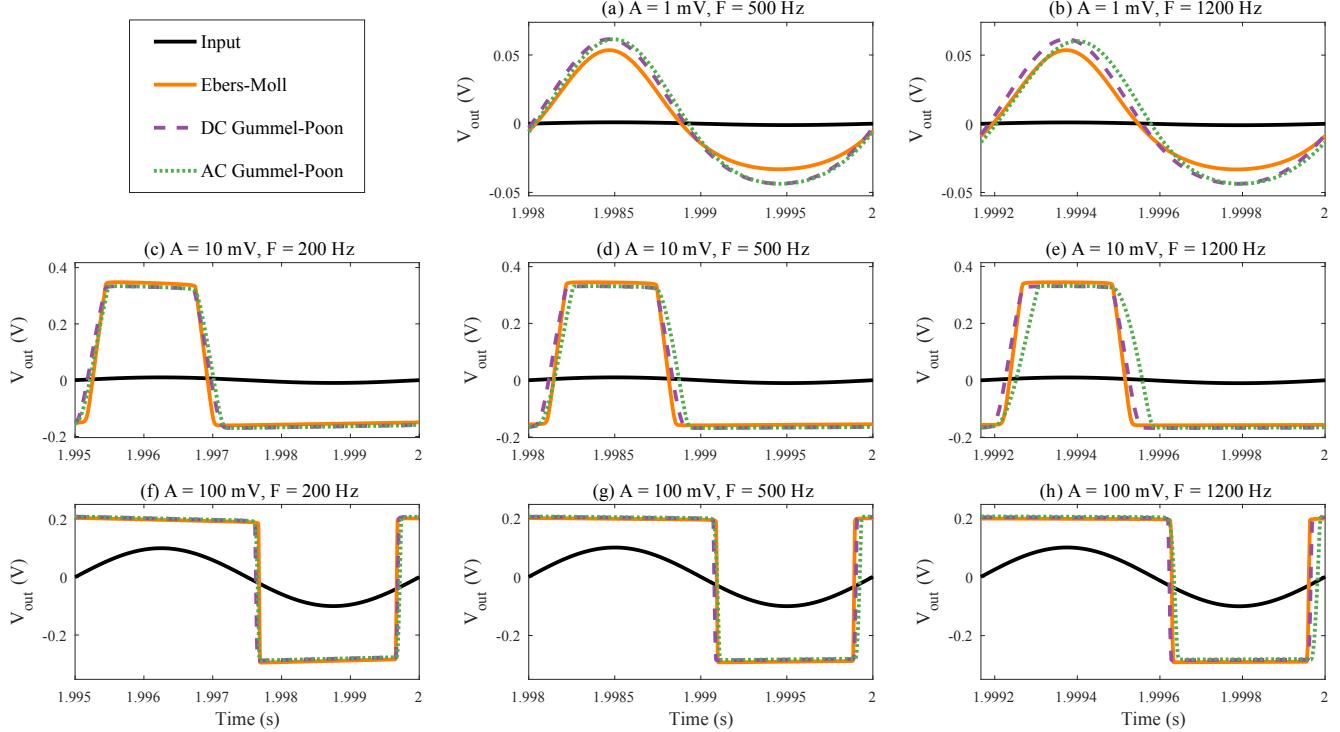


Figure 11: Single cycles of the Fuzz Face output's response to a sinusoidal input signal. The fuzz control of the circuit was set to maximum.

AUTOMATIC CONTROL OF THE DYNAMIC RANGE COMPRESSOR USING A REGRESSION MODEL AND A REFERENCE SOUND

Di Sheng

Centre for Digital Music (C4DM),
Queen Mary University of London
London, UK
d.sheng@qmul.ac.uk

György Fazekas

Centre for Digital Music (C4DM),
Queen Mary University of London
London, UK
g.fazekas@qmul.ac.uk

ABSTRACT

Practical experience with audio effects as well as knowledge of their parameters and how they change the sound is crucial when controlling digital audio effects. This often presents barriers for musicians and casual users in the application of effects. These users are more accustomed to describing the desired sound verbally or using examples, rather than understanding and configuring low-level signal processing parameters. This paper addresses this issue by providing a novel control method for audio effects. While a significant body of works focus on the use of semantic descriptors and visual interfaces, little attention has been given to an important modality, the use of sound examples to control effects. We use a set of acoustic features to capture important characteristics of sound examples and evaluate different regression models that map these features to effect control parameters. Focusing on dynamic range compression, results show that our approach provides a promising first step in this direction.

1. INTRODUCTION

The invention of recording in the late nineteenth century democratised music listening and gave rise to a new industry concerning the production, distribution and reproduction of music. Due to technical and aesthetic requirements, the industry developed an increasingly large number of tools for the manipulation of audio content to achieve desired sound qualities. Changing the dynamic range, timbre or frequency balance of recordings have first become widely possible with the introduction of analogue signal processing techniques, for instance, linear filters and non-linear effects like the compressor. Digital technologies such as software plugins or audio effects embedded in Digital Audio Workstations have significantly extended and, to some extent, replaced analogue effects. However, from the users' point of view, they rarely go beyond mimicking the operation of analogue counterparts.

Controlling effects requires significant experience and know-how, especially when used for aesthetic purposes during music production [1]. This often involves mapping a concept or idea concerning sound qualities to low-level signal processing parameters with limited meaning from a musical perspective. Knowledge of signal processing, which was requisite for engineers in early studios, as well as good understanding of their control parameters and function constitute the skills of sound engineers and producers. Acquiring these skills however present a high barrier to musicians and casual users in applying today's production tools. Consequently, the development of intelligent tools, as it has been done in other industries, may greatly benefit music production.

Substantial amount of works in this area are concerned with automating the mixing and mastering process (see e.g. [2] or [3]).

Our work is significantly different from previous studies in that it does not directly target multitrack mixing and mastering, or attempt to use high-level semantic descriptors to control effects. Our focus is on the novel task of estimating the parameters of audio effects given a sound example, such that the processed audio sounds similar in some relevant perceptual attributes (e.g. timbre or dynamics) to the reference sound. This has applications in various stages of music production. For instance, while creating an initial rough mix of a track, artists may describe how they would like an instrument to sound using an actual sound example [1]. An intelligent tool that provides audio effects settings based on a reference audio track is useful to meet this requirement. It may also help hobbyists and amateur to make their own music or create remixes, an activity encouraged by well-known bands such as Radiohead, by releasing stems and multitrack recordings.

It may be a considerable effort to develop an intelligent tool that estimates the parameters of different types of effects using complex audio material. To assess the feasibility of solving this problem, we first simplify the task and focus on a single effect: dynamic range compression, and simple audio material: mono-timbral notes and loops. The proposed solution consists of 1) an audio feature extractor that generates features corresponding to each parameter, 2) a regression model that maps audio features to audio effect parameters and 3) a music similarity measure to compare the processed and the reference audio.

The rest of the paper describes the proposed algorithm in detail. It is structured as follows: Section 2 outlines related work, Section 3 shows the workflow of our system, the evaluation and analysis is discussed in Section 4, followed by future work and conclusion in Section 5.

2. RELATED WORK

In this section, we provide a brief overview of intelligent audio production technologies with an emphasis on the dynamic range compressor (DRC), a non-linear time dependent audio effect. The area of intelligent audio production has become a burgeoning field over the last decade, with solutions ranging from automatic mixing systems [2] to intelligent audio editors [4][5]. These systems typically rely on audio feature extraction to analyse one or more channels and embed expert knowledge into procedural algorithms to automate certain aspects of the production workflow. The goal in many cases is the delivery of a technically correct mix by controlling the gain or loudness balance of sources in multitrack recordings. Finding the optimal dynamic range for each instrument[3] or reducing the number of user configurable parameters of an effect [6] have also been considered.

Other approaches aim at providing alternative control mecha-

nisms, such as the use of semantic descriptors or simplified graphical interfaces. Loviscach [7] for instance describes a control method for equalisation that allows drawing points or using free hand curves instead of setting parameters. Subsequent work [8] provides a creative method to map features to shapes. However, the shapes of this system do not link directly with the "meaning" of settings, rather they are classifications of the settings. Cartwright and Pardo [9] outline a control strategy using description terms such as warm or muddy, and demonstrate a method of applying high-level semantic controls to audio effects. Wilmering et al. [10] describe a semantic audio compressor that learns to associate control parameters with musical features such as the occurrence of chord patterns. This system provides intelligent recall functionality.

The idea of cross adaptive audio effects [2] was introduced in the context of automatic multitrack mixing. In this scenario, audio features extracted from multiple sources are utilised and automatic control is applied to optimise inter-channel relations and the coherence of the whole mix. These systems often follow expert knowledge obtained from the literature about music production practice, or interviews with skilled mixing engineers. In [3], the authors explored how different audio engineers use the compressor on the same material to train an intelligent compressor. An automatic multitrack compressor based on side-chain feature extraction is presented in [6] providing automatic settings for attack/release time, knee, and make up gain based on low-level features and heuristics. An alternative implementation of DRC using non-negative matrix factorisation (NMF) is proposed in [11]. Here, the authors consider raising the NMF activation matrix to $\frac{1}{R}$ to obtain a compressed signal with ratio R after re-synthesis. This is viewed as a compressor without a threshold parameter.

The goal of our work is substantially different from these solutions. At this initial stage, we focus on individual track compression, rather than trying to fit the signal into a mix, therefore we do not assume the presence of other channels. We do not aim to incorporate expert knowledge into the system or automate control parameters in a time varying manner, instead, we assume an audio example, and aim to configure the compressor to yield an output that sounds similar to the example. This requires the prediction of each parameter.

To estimate the settings of a compressor with hidden parameter values, Bitzer et. al. [12] proposes the use of purposefully designed input signals. Although this provides insight into predicting DRC parameters, in most cases, including ours, only the audio signal is available and the original compressor and its settings are not available for black-box testing. A reverse engineering process is proposed for the entire mix in [13]. The authors estimate a wide range of mixing parameters including those of audio effects. This work however focusses on the estimation of time-varying gain envelopes associated with dynamic non-linear effects rather than their individual parameters. A recent work proposes the use of deep neural networks (DNN) for the estimation of gain reduction [14]. The use of DNN in intelligent control of audio effects is quite novel, but this work targets only the mastering process and only considers the ratio factor.

Our work focuses on comparing linear and non-linear regression models to map low-level audio features discussed in Section 3 to common parameters of the dynamic range compressor. We propose using a reference audio example as target and evaluate the regression models in terms of how close the processed sounds get to the target in overall dynamic range and audio similarity. This is motivated by the need to analyse and compare changes in

the dynamic and spectral characteristic of the processed sounds, since both are affected by DRC. To this end, we measure peak-to-RMS ratio and also use a simple baseline model of audio similarity consisting of a Gaussian Mixture Model (GMM) trained on Mel Frequency Cepstrum Coefficients(MFCCs) [15]. The Kullback-Leibler (KL) divergence is a robust method to measure the similarity between single Gaussian distributions [16][17]. However, the divergence between multiple Gaussian models is not analytically tractable, therefore we use the approach proposed in [18] based on variational Bayes approximation. In the next section, we discuss regression model training and DRC parameter estimation.

3. METHODS

3.1. Training procedure

This study uses a single-channel, open-source dynamic range compressor developed in the SAFE project [19]. In the interest of brevity, we do not discuss the operation of the compressor and assume the reader is familiar with relevant principles[20, 6]. We consider the estimation of the most common parameters: threshold, ratio, attack and release time from reference audio and leave other parameters e.g. make up gain and knee width for future work. To form an efficient regression model, we need to choose the most relevant features first. Section 3.1.1 describes the features derived from a series of experiments. The system is then discussed in Section 3.1.2 outlining the data flow and system structure.

3.1.1. Feature extraction

Audio features are selected or designed for each specific effect parameter. Since DRC affects perceptual attributes in terms of loudness and timbre, six statistical features are selected for all four parameters. The RMS features reflect energy, which is related to loudness, while the spectral features reflect the spectral envelope, which is related to timbre. The statistical features are calculated frame-wise, with a frame size of 1024 samples and a 50% overlap. For spectral features, we use 40 frequency bins up to 11kHz. We assume this bandwidth is sufficient for the control of selected DRC parameters. We define the magnitude spectrogram $Y(n, k) = |X(n, k)|$ with $n \in [0 : N - 1]$ and $k \in [0 : K]$ where N is the number of frames and k is the frequency index of the STFT of the input audio signal with a window length of $M = 2(K + 1)$. We extract the spectral features described in Equations 1 - 4 as follows:

$$SC_{\text{mean}} = E\left[\frac{\sum_{k=0}^{K-1} k \cdot Y(n, k)}{\sum_{k=0}^{K-1} Y(n, k)}\right], \quad (1)$$

$$SC_{\text{var}} = Var\left[\frac{\sum_{k=0}^{K-1} k \cdot Y(n, k)}{\sum_{k=0}^{K-1} Y(n, k)}\right], \quad (2)$$

$$SV_{\text{mean}} = E[(E[Y(n, k)^2] - (E[Y(n, k)])^2)^{1/2}], \quad (3)$$

$$SV_{\text{var}} = Var[(E[Y(n, k)^2] - (E[Y(n, k)])^2)^{1/2}], \quad (4)$$

where SC stands for spectral centroid, and SV stands for spectral variance. The mean and variance in the equations are calculated across all M length frames.

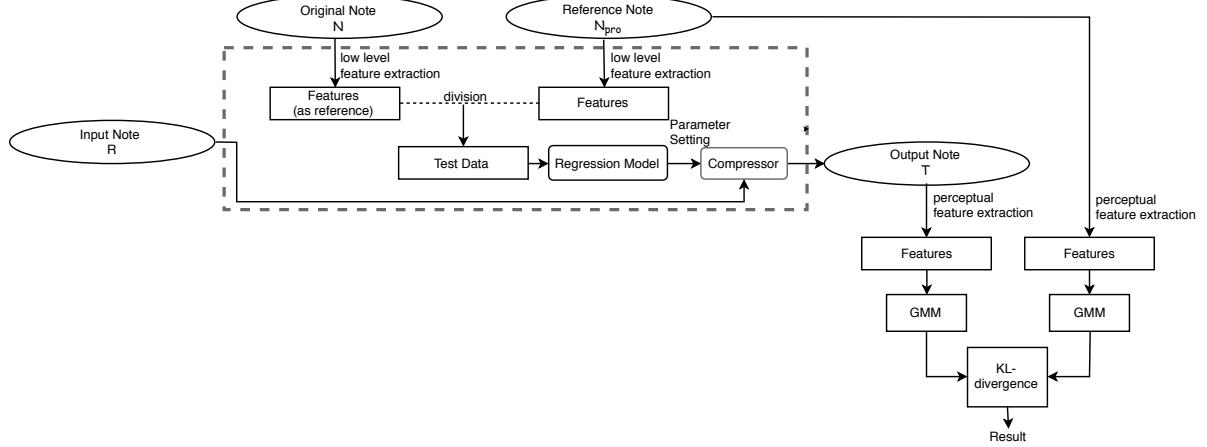


Figure 1: Workflow of initial system with access to the reference sound and its corresponding unprocessed version (see Section 3.2.2)

We also extract the following temporal features described in Equations 5 - 6:

$$\text{RMS}_{\text{mean}} = E[(\frac{1}{M} \sum_{m=0}^{M-1} x(m)^2)^{1/2}], \quad (5)$$

$$\text{RMS}_{\text{var}} = \text{Var}[(\frac{1}{N} \sum_{m=0}^{N-1} x(m)^2)^{1/2}], \quad (6)$$

where $x(m)$ represents the magnitude of audio sample m within each M length frame, and the mean and variance are calculated across all the N time frames as with the previous spectral features.

We designed four types of time domain features related to the attack and release of the notes as well as the speed of the compressor. The attack and release times $T_A = T_{\text{end}A} - T_{\text{start}A}$ and $T_R = T_{\text{end}R} - T_{\text{start}R}$ are calculated using the RMS envelope through a fixed threshold method (c.f. [21]) that determines start and end times of the attack and release parts of the sound. The end of the attack is considered to be the first peak that exceed 50% of the maximum RMS energy. The RMS curve is smoothed by a low-pass filter with a normalised cut-off frequency of 0.47 rad/s. We also extract the RMS amplitude at the end of the attack and the start of the release $\text{rms}(T_{\text{end}A})$ and $\text{rms}(T_{\text{start}R})$ respectively, as well as the mean amplitude during the attack and release parts of the sound.

$$A_{\text{att}} = \frac{1}{T_A} \sum_{n=T_{\text{start}A}}^{T_{\text{end}A}} \text{rms}(n), \quad (7)$$

$$A_{\text{rel}} = \frac{1}{T_R} \sum_{n=T_{\text{start}R}}^{T_{\text{end}R}} \text{rms}(n), \quad (8)$$

where T_{start} and T_{end} are indices of the start and end of the attack or release. Finally, we compute a feature related to how fast the compressor operates. We first calculate the ratio between the time-varying amplitudes of input or original sound and the reference sound $s(n) = \text{rms}_{\text{ref}}(n)/\text{rms}_{\text{orig}}(n)$. We then calculate the amount of time for $s(n)$ to reach a certain value using a fixed

threshold. This relates to the speed of compressor to reach the desired compression ratio, which is controlled primarily by its attack time. The same process is applied at the end of the note to extract how fast the ratio curve drops back to one. The design of these features was motivated by visual inspection of the signal. They are shown to improve the ability of the regression model to predict the parameters (see Section 4).

3.1.2. Regression model training

This section outlines the datasets and training procedure for the regression models that are used to map features to effect parameter settings. In the first stage of our research, we consider two types of instruments: snare drum and violin. The former is one of the most common instruments that requires at least a light compression to even out dynamics. The drum samples are typically short and, considering a typical energy envelope, exhibit only the attack and release (AR) part. The violin recordings typically consist of a long note with fairly clear attack, decay, sustain and release (ADSR) phase. All audio samples in our work are taken from the RWC isolated note database [22].

Table 1 describes the four violin note datasets denoted A, \dots, D that are used for training. In each dataset, one parameter of the effect is varied while the others are kept constant. The number of training samples in each dataset equals to the number of notes, i.e., 60 in case of the violin dataset, times the number of grid points (subdivisions) for each changing parameter. In this study, we use 50 settings for threshold and ratio, and 100 settings for attack and release time as it is shown in the first column of Table 1. The same process is applied to 12 snare drum samples to form the drum dataset. Each training set A, \dots, D is used for predicting a specific parameter.

Training sets (size)	Conditions			
	Thr(dB)	Ratio	Att(ms)	Rel(ms)
A (60*50)	0:1:49	2	5	200
B (60*50)	37.5	1:0.4:20	5	200
C (60*100)	37.5	2	1:1:100	200
D (60*100)	37.5	2	5	50:10:1000

Table 1: Training set generation

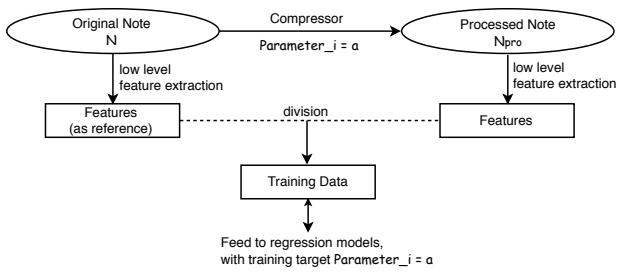


Figure 2: Flowchart of training data generation

Figure 2 describes the workflow. Taking training set A as an example, the original notes N are the recorded violin notes. The processed notes denoted N_{pro} on the right hand side of the figure are generated from N , which are processed by the compressor with different threshold values. There are 60 different notes N , and each N generates 50 processed notes N_{pro} . This yields $60 \times 50 = 3000 N_{pro}$ in the training dataset for threshold. Because the features from N_{pro} are highly correlated with the original note N , the ratio between them is used to focus on how the features change as a result of dynamic compression. Therefore, it is the difference we actually used to train the regression model. There are 6 features related to the threshold extracted from each note, therefore training data A comprises $3000 N_{pro} \times 6$ feature vectors. In the following, this training data is used to generate the regression model. The same principle applies to training sets B, C and D.

In our work, two regression models are compared and evaluated (see Section 4), simple linear regression, as well as random forest regression [23]. Random forest uses averaging over subsamples from the dataset to improve the predictive accuracy of the model as well as to mitigate over-fitting problems. The use of this latter model is motivated by the hypothesis that the relationship between the audio features and the compressor parameters may not be modelled accurately enough with simple linear regression due to the non-linearities in the process. In the evaluation, we use the implementations available in [24].

3.2. System design and testing procedure

3.2.1. Numerical test at the isolated note level

In this paper, we propose two system designs. The first aims only at verifying the basic idea behind the use of a reference sound (or note) to be approximated. This is not a realistic scenario, because it assumes we have access to both the processed and unprocessed (original) version of the sound used as reference. This is needed because the predictor variables, i.e., the input to the regression model is calculated as the ratio of the audio feature data extracted from these recordings. This requires access to all pairs of (N, N_{pro}) in the training data. This scenario is presented in Figure 1 consisting of two parts. The components within the dashed line box represent the actual control system for the compressor with three inputs: the input note R to be processed, the reference note N_{pro} to be approximated, and its corresponding original note N from the training set. The output note T outside of the dashed line box will be used in the evaluation, where we compare the similarity of the output and the reference. As it is mentioned in

Section 3.1.2, the regression model is trained on the ratio, therefore, we need to provide the same data for the model to predict the compressor parameters. The original note N is used only in the process of generating feature vectors.

Before we use the similarity model depicted on the right hand side of Figure 1, we first evaluate the regression model accuracy. This first study compares predicted parameter values with the actual ones. The workflow is the same as Figure 2, providing a standard testing step for regression models. In this study, we use repeated random sub-sampling validation (Monte Carlo variation). 10% of each feature vectors are used for testing, while the remaining 90% are used as training data. This experiment is repeated 100 times and the average results are reported in Section 4.

3.2.2. Similarity assessment at the isolated note level

Considering the motivations and use cases described in Section 1, the desired output of this algorithm is to make an unrelated note R sound similar to the reference note N_{pro} , where N_{pro} is generated through a compressor with e.g. its threshold set to x dB. However, even if the prediction is perfect with $x_p = x$, the same compressor for note R and note N can give different perceptual results. Therefore, a similarity model which takes this into account is used to evaluate the similarity between N_{pro} and the algorithm output T . The processing and evaluation workflow is represented in Figure 1 with the structure within the dashed line box used to control the system while the components on the right hand side are used in the similarity test.

In the similarity assessment, we use a simple and frequently used model of audio similarity [15] as well as a simple feature which is a good (although partial) indicator of the overall dynamics of the signal. First, we consider the crest factor and report the difference between the reference and the processed sound. Second, we follow the similarity model using a Gaussian Mixture Model trained on Mel Frequency Cepstrum Coefficients. Accordingly, the feature extraction in the workflow indicates the calculation of the divergence between two multiple Gaussian models, which provides the similarity information. We use the symmetrised Kullback-Leibler (KL) divergence, which is commonly used for Gaussian models, and since we use a GMM, an approximation of the KL divergence is calculated using the approach presented in [18]. The results of this test and analyses are provided in Section 4.

3.2.3. Note level similarity assessment in a realistic scenario

In a real world scenario, if the reference N_{pro} is a commercial audio track, its corresponding unprocessed original sound N is not likely to be available. In this case, we propose to use the system design outlined in Figure 3, where the input of the system are limited to the input note R to be processed and the reference note N_{pro} . In the feature computation workflow, the original note N is replaced by the input note R , because the features capture the difference between the reference note N and the input R can be seen more reasonable and closer to a real world scenario.

3.2.4. Loop level similarity assessment in a realistic scenario

This study extends the objective of the experiment from using mono-timbral notes to longer mono-timbral loops. The loops we used are approximately 5 seconds long, consisting of violin loops taken from the RWC Instrument Sound Database [22]. Under the

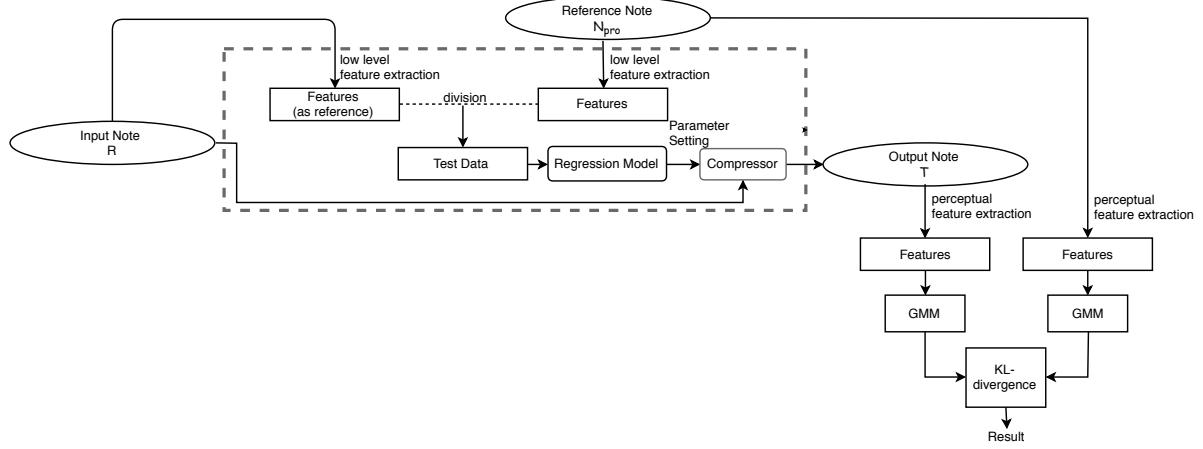


Figure 3: *Realistic workflow with only the input and reference sound available (see Section 3.2.3)*

constraint of mono-timbre, i.e., only a single instrument is sounding at a time, we assume that the statistical properties of the underlying features remain constant. Therefore, the rest of the algorithm remains the same. A possible method to apply our algorithm to loops is to consider them as notes, with the attack of the first note in the loop and release of the last. It is a reasonable simplification in practice, noting that the rest of the attack and release parts of notes are likely to be overlapping. As before, the results are reported and discussed in Section 4.3.

4. EVALUATION

4.1. Direct assessment of parameter estimation

Using the test procedure outlined in Section 3.2.1, here we report the accuracy of direct parameter estimation using random subsampling validation. Two regression models are compared and evaluated: simple Linear Regression (LR) and Random Forest regression (RF) [23] available in [24]. Table 2 & 3 show the absolute errors for both instruments and regression models. Since the observed feature values are relatively small, we linearly scale the feature values to [0, 1] and compare the errors. The highlighted values in the table show that the smallest error is always observed when using the scaled features and the random forest regression model. For completeness, the range for the four parameters are (0,50) dB for threshold, [1,20] for ratio, (0,100] ms for attack time, and (0,1000] ms for release time. Scaling is reasonable in this pilot experiment, because the test data (reference) is selected randomly from the database mentioned above, which means all N_{pro} has its original N available. However, in a real world scenario, the reference sound is not taken from the database prepared to train the regression models. It is more likely to be a produced sound or track without access to its unprocessed version. Therefore scaling will not be used in the subsequent studies. Please note that we do not overfit the models, because in the evaluation the reference note and the corresponding original is excluded from the training set of the regression model.

The results also illustrate that the prediction accuracy for drums is better than for violins in all cases. One reason is that drum samples are shorter and exhibit a simpler structure - short sustain, followed by release and there is no pitched content. It shows that the

system can predict the compressor parameters from drums better than for violins. In subsequent studies, we will therefore focus on the more complex case of similarity measurement for violin notes and loops.

Violin		LR	RF
Threshold(dB)	error	3.756	2.601
	scaled error	1.860	1.731
Ratio	error	2.065	1.583
	scaled error	0.110	0.091
Attack(ms)	error	15.503	0.719
	scaled error	1.012	0.686
Release(ms)	error	210.43	13.973
	scaled error	78.913	10.583

Table 2: Numerical test using linear and random forest regression model for violin notes

Snare Drum		LR	RF
Threshold(dB)	error	1.185	0.800
	scaled error	0.408	0.345
Ratio	error	1.571	0.999
	scaled error	0.669	0.305
Attack(ms)	error	6.867	0.860
	scaled error	2.260	0.017
Release(ms)	error	23.045	0.999
	scaled error	40.960	6.851

Table 3: Numerical test using linear and random forest regression model for snare drum samples

4.2. Results of similarity assessment between notes

In this section, we evaluate the changes in estimated similarity using the system outlined in the right hand side of Figure 1 and Figure 3. Firstly, we extract the crest factor, i.e., peak-to-RMS ratio as the similarity feature because it is correlated with the overall dynamic range of the signal. Based on our design, the crest factor of the reference note N_{pro} should be closer to the output note T than the input note R . An example of this test is given in Figure 4 with 25 test cases. The crest factor of the input signal is represented by

the constant at the top of the figure and the crest factor of a series of reference notes are depicted by the blue curve at the bottom. The crest factor of the output signal from the system is shown in the middle (green curve). It is consistently brought closer to the reference which fits our expectation here.

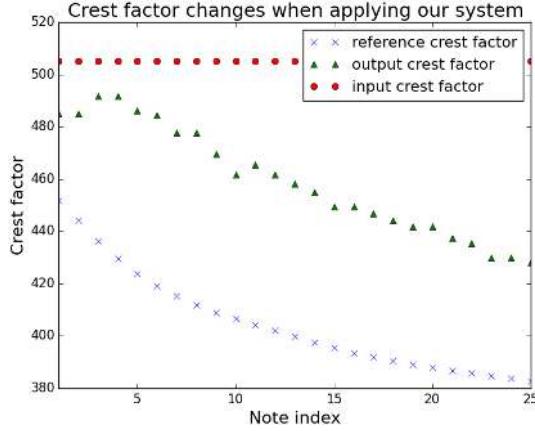


Figure 4: Example of changing crest factor with a fixed input and decaying reference sound

We test 50 reference notes and present the results in Table 4 for violin notes and Table 5 for snare drum, where $D_{Crest}(A, B) = \text{mean}(|\text{Crest}(A) - \text{Crest}(B)|)$. The results indicate that the system manages to bring the output closer to the reference using both regression models. In all parameters except threshold, the random forest model performs better than simple linear regression.

Violin	Threshold	Ratio	Attack	Release
$D_{Crest}(N_{pro}, R)$	60.31	94.13	104.93	85.31
$D_{Crest}(N_{pro}, T)_{LR}$	12.53	39.72	46.76	48.62
$D_{Crest}(N_{pro}, T)_{RF}$	15.27	38.24	45.23	47.19

Table 4: Average of Crest factor difference - Violin

Snare Drum	Threshold	Ratio	Attack	Release
$D_{Crest}(N_{pro}, R)$	49.14	70.04	50.47	70.68
$D_{Crest}(N_{pro}, T)_{LR}$	27.99	43.85	27.28	44.63
$D_{Crest}(N_{pro}, T)_{RF}$	29.33	43.45	27.20	43.44

Table 5: Average of Crest factor difference - Snare drum

Next, we discuss the results of similarity assessment as described in Section 3.2.2. At this stage, we use a simple audio similarity model to test the efficiency of the system. We use MFCC coefficients as features and fit a GMM on the MFCC vectors. An approximation of the symmetrised KL divergence is then calculated and used as a distance measure. Using the same procedure as in the previous part, the compressor settings provided by this algorithm should bring the output note T closer to the N_{pro} compared to the input note R . Thus it is reasonable to assume that $D(N_{pro}, R) > D(N_{pro}, T)$ holds and the performance of the regression models can be tested. In this experiment, we select 50 N_{pro} and change one parameter at a time. Table 6 & 7 indicate the efficiency of the algorithm. Since the similarity algorithm theoretically captures the timbre information as well, it will yield different

results on different instruments. In this test, the distance reduction achieved by the system is larger for violins, i.e., the violin notes exhibit better results than the snare drums. This is possibly due to the fact that the MFCC features used here do not model the drum sounds well enough. Finding a better feature representation for percussive instruments constitutes future work.

Violin	Threshold	Ratio	Attack	Release
$D(N_{pro}, R)$	38.122	53.187	44.018	55.206
$D(N_{pro}, T)_{LR}$	19.799	20.911	22.852	20.994
$D(N_{pro}, T)_{RF}$	19.742	20.856	22.213	20.807

Table 6: KL Divergences for the first workflow in 3.2.2 - Violin

Snare Drum	Threshold	Ratio	Attack	Release
$D(N_{pro}, R)$	77.497	112.368	73.559	91.487
$D(N_{pro}, T)_{LR}$	73.749	88.574	73.307	85.022
$D(N_{pro}, T)_{RF}$	73.696	88.487	73.238	86.081

Table 7: KL Divergences for the first workflow in 3.2.2 - Drum

Finally, we investigate how the proposed algorithm works in a more realistic scenario. When the original note N is not available, it is reasonable to use the input note R in place of N . Under this condition, we repeat the same test for both crest factor and the MFCC-based similarity model. The results for crest factor are provided in Table 8 for violin and in Table 9 for snare drum samples. The system is still able to bring the crest factor of the output T closer to the reference N_{pro} , but the efficiency is worse compared to the case when the original note is available. Random forest regression still yields better performance in almost all cases.

Violin	Threshold	Ratio	Attack	Release
$D_{Crest}(N_{pro}, R)$	60.31	94.13	104.93	85.31
$D_{Crest}(N_{pro}, T)_{LR}$	34.86	29.37	68.74	75.94
$D_{Crest}(N_{pro}, T)_{RF}$	30.11	25.36	67.82	54.02

Table 8: Average of Crest factor difference - Violin

Snare Drum	Threshold	Ratio	Attack	Release
$D_{Crest}(N_{pro}, R)$	49.14	70.04	50.47	70.68
$D_{Crest}(N_{pro}, T)_{LR}$	38.01	66.18	21.37	44.05
$D_{Crest}(N_{pro}, T)_{RF}$	42.90	45.24	27.37	42.75

Table 9: Average of Crest factor difference - Drum

The result using the MFCC-based similarity model is illustrated in Figure 5 & 6, with $D(N_{pro}, R)$ on the left, $D(N_{pro}, T)_{LR}$ in the middle and $D(N_{pro}, T)_{RF}$ on the right of each subplot. In Figure 5 for violin notes, the average divergence is not as promising, especially when comparing with the results in Table 6, but it is clear that even if the given reference sounds have a large diversity, the algorithm reduces this significantly, and shows a very stable improvement in the similarity result. On average, the random forest regression performs better than linear regression in all cases except when predicting threshold. This shows the benefit of modelling non-linearities. Therefore we will build our system using random forest regression. Figure 6 shows that the output of the system did not manage to achieve a dramatic reduction in the similarity distance in case of the snare drum. As explained before, we

need to further investigate the influence of timbre on this similarity algorithm. Furthermore, due to the size of the snare drum dataset, we have a limitation in the choice of test data.

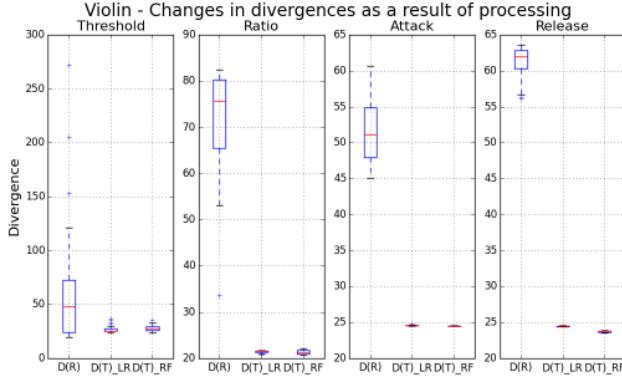


Figure 5: Similarity for four parameters in the second workflow, assuming the origin note N is not available. - Violin

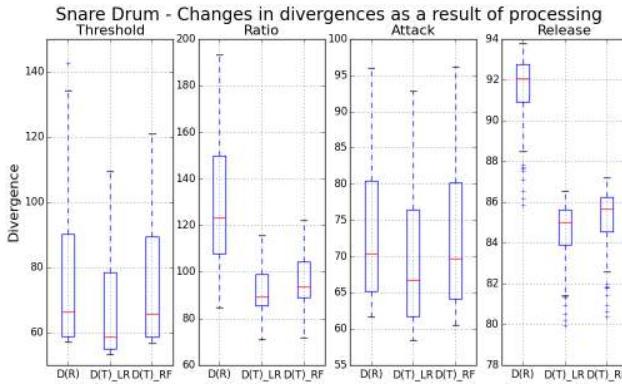


Figure 6: Similarity for four parameters in the second workflow, assuming the origin note N is not available. - Snare drum

4.3. Results of similarity between loops

We can extend the study by changing the audio material from mono-timbral notes to mono-timbral loops. The final experiment tests the efficiency of the workflow in Figure 3 without using the original notes, and replaces both R and N_{pro} with violin loops. The results displayed in Table 10 and Figure 7 correspond to 50 violin loops which have the duration of 3-5 seconds. A promising trend is observed using the average divergence. However, unlike in the previous studies, the divergence does not drop very significantly. These results indicate that the algorithm works better for attack/release time, but the performance is not yet satisfactory for threshold. A possible reason is that we selected six features for threshold/ratio but ten for attack/release while the features used may not be sufficient or accurate enough in this more generic case.

	Threshold	Ratio	Attack	Release
$D_{Crest}(N_{pro}, R)$	545.48	580.10	574.33	569.69
$D_{Crest}(N_{pro}, T)_{LR}$	301.59	237.24	326.68	314.11
$D_{Crest}(N_{pro}, T)_{RF}$	301.75	209.06	325.08	321.23

Table 10: Average of Crest factor difference - Loops

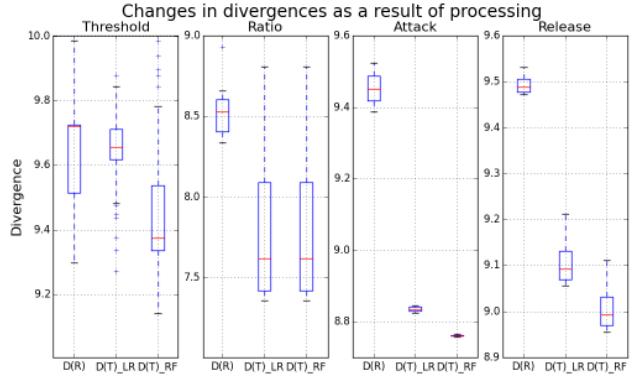


Figure 7: Similarity for four parameters in the second workflow, assuming the origin note N is not available. - Loops

5. CONCLUSION AND FUTURE WORK

In this paper, we proposed a method to estimate dynamic range compressor settings using a reference sound. We demonstrate the first steps towards a system to configure audio effects using sound examples, with the potential to democratise the music production workflow. We discussed progress from using a linear regression model to a random forest model and from a designated test case to a real world scenario. The evaluation shows a promising trend in most cases and provides an initial indication of the utility of our system. Our current research focuses on simple audio material, notes and mono-timbral loops. The study discussed in Section 3.2.4 considers loops as signal notes, which ignores a lot of information. In future work, we will assess the use of an onset event detection to identify notes from loops or more complex recordings and measure their attributes using the method applied to individual notes. A useful intelligent audio production tool will be devised for audio tracks that are more complex, while polyphonic tracks will also be considered in future research.

The algorithm itself may also need improvement. The features we chose to train the regression model can be extended. Thorough research on how to design audio features specific to compressor and other audio effects parameters would be beneficial. At the same time, the regression models employed in this work may be improved using optimisation techniques that take the similarity features into account. An improved similarity model may be used as an objective function, rather than being used only during evaluation. We note however that the currently employed technique in the assessment of similarity is only considered a starting point. More realistic and complex auditory models will be applied in future work. Additionally, we plan to evaluate the system using real human perceptual evaluation, i.e., using a listening test. In conclusion, this paper provides an innovative and feasible method for intelligent control of dynamic range compressor. However, this research is still in early phase and further considerations are needed to optimise feature design and the regression model.

6. ACKNOWLEDGEMENTS

This work has been part funded by the FAST IMPACt EPSRC Grant EP/L019981/1 and the European Commission H2020 research and innovation grant AudioCommons 688382.

7. REFERENCES

- [1] Sean McGrath, Alan Chamberlain, and Steve Benford, “Making music together: An exploration of amateur and pro-am grime music production,” in *Proceedings of the Audio Mostly 2016*. ACM, 2016, pp. 186–193.
- [2] Joshua D Reiss, “Intelligent systems for mixing multichannel audio,” in *17th International Conference on Digital Signal Processing (DSP)*. IEEE, 2011, pp. 1–6.
- [3] Zheng Ma, Brecht De Man, Pedro DL Pestana, Dawn AA Black, and Joshua D Reiss, “Intelligent multitrack dynamic range compression,” *Journal of the Audio Engineering Society*, vol. 63, no. 6, pp. 412–426, 2015.
- [4] Roger B Dannenberg, “An intelligent multi-track audio editor,” in *Proceedings of international computer music conference (ICMC)*, 2007, vol. 2, pp. 89–94.
- [5] Yuxiang Liu, Roger B Dannenberg, and Lianhong Cai, “The intelligent music editor: towards an automated platform for music analysis and editing,” in *Advanced Intelligent Computing Theories and Applications. With Aspects of Artificial Intelligence*, pp. 123–131. Springer, 2010.
- [6] Dimitrios Giannoulis, Michael Massberg, and Joshua D Reiss, “Parameter automation in a dynamic range compressor,” *Journal of the Audio Engineering Society*, vol. 61, no. 10, pp. 716–726, 2013.
- [7] Jörn Loviscach, “Graphical control of a parametric equalizer,” in *Audio Engineering Society Convention 124*. Audio Engineering Society, 2008.
- [8] Philipp Kolhoff, Jacqueline Preub, and Jörn Loviscach, “Music icons: procedural glyphs for audio files,” in *2006 19th Brazilian Symposium on Computer Graphics and Image Processing*. IEEE, 2006, pp. 289–296.
- [9] Mark Brozier Cartwright and Bryan Pardo, “Social-eq: Crowdsourcing an equalization descriptor map.,” in *Proceedings of the 14th International Conference on Music Information Retrieval (ISMIR)*, 2013.
- [10] Thomas Wilmering, György Fazekas, and Mark B Sandler, “High-level semantic metadata for the control of multitrack adaptive digital audio effects,” in *Audio Engineering Society Convention 133*. Audio Engineering Society, 2012.
- [11] Ryan Sarver and Anssi Klapuri, “Application of nonnegative matrix factorization to signal-adaptive audio effects,” in *Proc. DAFX*, 2011, pp. 249–252.
- [12] Joerg Bitzer, Denny Schmidt, and Uwe Simmer, “Parameter estimation of dynamic range compressors: models, procedures and test signals,” in *Audio Engineering Society Convention 120*. Audio Engineering Society, 2006.
- [13] Daniele Barchiesi and Joshua Reiss, “Reverse engineering of a mix,” *Journal of the Audio Engineering Society*, vol. 58, no. 7/8, pp. 563–576, 2010.
- [14] Drossos K. Virtanen T. Mimalakis, S.I. and G. Schuller, “Deep neural networks for dynamic range compression in mastering applications,” in *Audio Engineering Society Convention 140*. Audio Engineering Society, 2016.
- [15] Jean Julien Aucouturier and Francois Pachet, “Music similarity measures: What’s the use?,” in *Proceedings of the 3th International Conference on Music Information Retrieval (ISMIR)*, 2002, pp. 13–17.
- [16] Yossi Rubner, Carlo Tomasi, and Leonidas J Guibas, “The earth mover’s distance as a metric for image retrieval,” *International journal of computer vision*, vol. 40, no. 2, pp. 99–121, 2000.
- [17] Beth Logan and Ariel Salomon, “A music similarity function based on signal analysis.,” in *ICME*, 2001.
- [18] John R Hershey and Peder A Olsen, “Approximating the kullback leibler divergence between gaussian mixture models,” in *2007 IEEE International Conference on Acoustics, Speech and Signal Processing-ICASSP’07*. IEEE, 2007, vol. 4, pp. IV–317.
- [19] Ryan Stables, Sean Enderby, BD Man, György Fazekas, Joshua D Reiss, et al., “Safe: A system for the extraction and retrieval of semantic audio descriptors,” *15th International Society for Music Information Retrieval Conference (ISMIR 2014)*.
- [20] Udo Zolzer, *DAFx: Digital Audio Effects*, Wiley Publishing, 2nd edition, 2011.
- [21] Geoffroy Peeters, “A large set of audio features for sound description (similarity and classification) in the cuidado project,” 2004.
- [22] Masataka Goto, Hiroki Hashiguchi, Takuichi Nishimura, and Ryuichi Oka, “Rwc music database: Music genre database and musical instrument sound database,” *Proceedings of the 4th International Conference on Music Information Retrieval (ISMIR 2003)*, pp. 229–230, 2003.
- [23] Leo Breiman, “Random forests,” *Journal of Machine Learning Research*, vol. 45, no. 1, pp. 5–32, Oct. 2001.
- [24] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, “Scikit-learn: Machine learning in Python,” *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.
- [25] Danilo P Mandic and Vanessa Su Lee Goh, *Complex valued nonlinear adaptive filters: noncircularity, widely linear and neural models*, vol. 59, John Wiley & Sons, 2009.
- [26] Xavier Serra and Julius Smith, “Spectral modeling synthesis: A sound analysis/synthesis system based on a deterministic plus stochastic decomposition,” vol. 14, no. 4, pp. 12–24, 1990.
- [27] Sebastian Heise, Michael Hlatky, and Jörn Loviscach, “A computer-aided audio effect setup procedure for untrained users,” in *Audio Engineering Society Convention 128*. Audio Engineering Society, 2010.
- [28] Roger B Dannenberg, “Music representation issues, techniques, and systems,” in *Computer Music Journal*. 1993, vol. 17, pp. 20–30, JSTOR.
- [29] Olivier Lartillot, Petri Toivainen, and Tuomas Eerola, “A matlab toolbox for music information retrieval,” in *Data analysis, machine learning and applications*, pp. 261–268. Springer, 2008.
- [30] Petri Toivainen and Carol L Krumhansl, “Measuring and modeling real-time responses to music: The dynamics of tonality induction,” vol. 32, no. 6, pp. 741–766, 2003.

FIXED-RATE MODELING OF AUDIO LUMPED SYSTEMS: A COMPARISON BETWEEN TRAPEZOIDAL AND IMPLICIT MIDPOINT METHODS

François G. Germain

Center for Computer Research in Music and Acoustics (CCRMA),
Stanford University, Stanford, CA, USA
francois@ccrma.stanford.edu

ABSTRACT

This paper presents a comparison framework to study the relative benefits of the typical trapezoidal method with the lesser-used implicit midpoint method for the simulation of audio lumped systems at a fixed rate. We provide preliminary tools for understanding the behavior and error associated with each method in connection with typical analysis approaches. We also show implementation strategies for those methods, including how an implicit midpoint method solution can be generated from a trapezoidal method solution and vice versa. Finally, we present some empirical analysis of the behavior of each method for a simple diode clipper circuit and provide an approach to help interpret their relative performance and how to pick the more appropriate method depending on desirable properties. The presented tools are also intended as a general approach to interpret the performance of discretization approaches at large in the context of fixed-rate simulation.

1. INTRODUCTION

Computational modeling of audio lumped systems (i.e., virtual analog modeling) is a major topic of interest, including emulation of electronic and acoustic systems such as vintage audio effects or acoustic instruments. This main goal of this modeling process is the design of discrete-time systems emulating the behavior of a continuous-time system. When the set of differential equations driving the dynamics of that system are known, a common procedure is to discretize it using a discretization schemes [1–3].

Those methods have a variety of advantages and drawbacks. Discretization schemes are generally designed following concepts such as order of accuracy and stability. These properties guarantee the versatility of those methods for consistently generating discrete-time models with some level of accuracy. In the context of virtual analog modeling, a large part of the literature has been developed focusing on using the trapezoidal method [3–10]. This method provides a good compromise between simplicity (as a one-step method), accuracy (as a second-order method) and behavior (as an unconditionally stable method). A less-studied numerical method with those same properties is the implicit midpoint method [11]. While they share those properties, the two methods differ in behavior for nonlinear systems, and results from previous papers have hinted to the possibility that the implicit midpoint method could produce better-behaved simulations for some classes of systems [12, 13]. One particular point of interest to compare the two methods is the oscillatory behavior for the simulation of stiff systems [14].

Virtual analog modeling is also generally focused on fixed-rate simulation, meaning that that controlling the error through the modulation of the simulation rate is seldom considered. This is mostly due to the computational cost of such control, as real-time

simulation is a desirable property of virtual analog approaches [15]. In that context, typical error analysis methods have limited interpretation since they mostly describe the behavior of the methods as the simulation rate tends to infinity. Simulation rates in virtual analog systems tend to fall in regions where those asymptotic properties provide limited insight on the simulation behavior. We then want to draw alternative design methods for discretization methods that rather target the fixed-rate context [14, 16–18].

This paper presents a preliminary framework for the comparison and error analysis of the typical trapezoidal method and the lesser-used implicit midpoint method in the context of fixed-rate simulation of audio lumped systems, as well as practical information regarding their implementation in several existing frameworks. Sec. 2 shows the general state-space formalism that we use for our analysis. Sec. 3 shows the definition and the general properties of the two methods. Sec. 4 shows a discussion of strategies to implement the midpoint method using current virtual analog modeling approaches. Finally, Sec. 5 shows an empirical comparison of the two methods on a diode clipper system [15, 19, 20].

1.1. Notation

In this paper, we use bolded letters (e.g., \mathbf{x} , \mathbf{f}) to denote multi-dimensional variables and multi-output functions. Discrete-time sequences are denoted with an overline (e.g., \overline{x} , $\overline{\mathbf{x}}$). Superscripts are used to denote the number of a sample in a sequence (e.g., \overline{x}^n is the n th sample in the sequence \overline{x}). To avoid confusion, power are indicated outside parentheses (e.g., $(\overline{x}^3)^2$ is the 3rd sample of the sequence \overline{x} raised to the 2nd power). Subscripts are used to denote differentiation (e.g., $f_{xu}(x, u)$ is the 2nd-order derivative of the function $f(x, u)$ with respect to x and u).

2. STATE-SPACE SYSTEM REPRESENTATION

2.1. Continuous-time state-space representation

A common way of representing time-invariant lumped systems is in the so-called state-space representation, where the system is characterized by the equations [21]

$$\mathbf{x}_t(t) = \mathbf{f}(\mathbf{x}(t), \mathbf{u}(t)), \quad (1a)$$

$$\mathbf{y}(t) = \mathbf{g}(\mathbf{x}(t), \mathbf{u}(t)), \text{ and} \quad (1b)$$

$$\mathbf{x}(0) = \mathbf{x}_0, \quad (1c)$$

where \mathbf{f} , \mathbf{g} are nonlinear functions, \mathbf{x} is a vector of state variables, \mathbf{u} is a vector of input variables, \mathbf{y} is a vector of output variables, \mathbf{x}_0 is a vector of initial conditions. For conciseness, we omit repeating Eqs. (1b) and (1c) in the rest of the paper as they remain unchanged by the discretization process.

2.2. System discretization

The exact solution to Eqs. (1a) and (1c) is theoretically given by

$$\mathbf{x}(t) = \mathbf{x}_0 + \int_0^t \mathbf{f}(\mathbf{x}(\tau), \mathbf{u}(\tau)) d\tau. \quad (2)$$

but we can also express the solution at times $t^n = nT$ (with a fixed interval T) by solving iteratively the problem

$$\mathbf{x}(t_{n+1}) = \mathbf{x}(t_n) + \int_{t_n}^{t_{n+1}} \mathbf{f}(\mathbf{x}(\tau), \mathbf{u}(\tau)) d\tau. \quad (3)$$

In typical cases, solving Eq. (2) or (3) analytically is intractable. One can however compute a discrete-time series $\bar{\mathbf{x}}^n$ approximating $\mathbf{x}(t_n)$. To do so, we use numerical integration methods to approximate the integral in Eq. (3) using some N past input values and approximated state values. The update equation of $\bar{\mathbf{x}}^n$ is then

$$\bar{\mathbf{x}}^{n+1} = \bar{\mathbf{x}}^n + T \bar{\mathbf{f}}(\bar{\mathbf{x}}^{n+1}, \dots, \bar{\mathbf{x}}^{n-N}, \bar{\mathbf{u}}^{n+1}, \dots, \bar{\mathbf{u}}^{n-N}). \quad (4)$$

Coupled with Eq. (1b), we can then form a sequence of approximated output values for our system of interest. Note that when $\bar{\mathbf{f}}$ depends on $\bar{\mathbf{x}}^{n+1}$, Eq. (4) becomes implicit and cannot always be solved analytically.

3. NUMERICAL INTEGRATION METHODS

Numerical integration methods aims at approximating the value of the integral of a function using a finite number of function evaluations. As stated earlier, we can apply those methods to Eq. (3) to form a discretized update equation as in Eq. (4). In this paper, we discuss specifically two common methods, the trapezoidal method and the implicit midpoint method.

3.1. Trapezoidal method

The trapezoidal method approximates the value of f in the interval $[t, t+T]$ as the average of the function values at the start and end point of the integral to compute the integral as [22]

$$\int_t^{t+T} \mathbf{f}(\tau) d\tau \approx \frac{T}{2} (\mathbf{f}(t+T) + \mathbf{f}(t)). \quad (5)$$

Eq. (3) is then discretized to form the implicit update equation for the trapezoidal method as

$$\bar{\mathbf{x}}^{n+1} = \bar{\mathbf{x}}^n + \frac{T}{2} \bar{\mathbf{f}}(\bar{\mathbf{x}}^{n+1}, \bar{\mathbf{u}}^{n+1}) + \frac{T}{2} \bar{\mathbf{f}}(\bar{\mathbf{x}}^n, \bar{\mathbf{u}}^n). \quad (6)$$

3.2. Implicit midpoint method

The midpoint method approximates the value of f in the interval $[t, t+T]$ as its value at midpoint of the integral to compute the integral as [22]

$$\int_t^{t+T} \mathbf{f}(\tau) d\tau \approx T \mathbf{f}\left(t + \frac{T}{2}\right). \quad (7)$$

To use this approach in Eq. (3), several implementations are possible. It can be implemented as the *explicit midpoint* method, but that method requires a way to compute the state and input values at time $t_{n+\frac{1}{2}}$. Additionally, this method (also called *leapfrog*

method) has poor stability properties as its stability region is reduced to the imaginary axis in the s-plane [1]. We focus instead on the *implicit midpoint* method whose implicit update equation is

$$\bar{\mathbf{x}}^{n+1} = \bar{\mathbf{x}}^n + T \mathbf{f}\left(\frac{1}{2}(\bar{\mathbf{x}}^{n+1} + \bar{\mathbf{x}}^n), \frac{1}{2}(\bar{\mathbf{u}}^{n+1} + \bar{\mathbf{u}}^n)\right). \quad (8)$$

For conciseness, we will refer to the implicit midpoint method simply as “midpoint method” in the rest of the paper.

3.3. Stability analysis and pole mapping

In the following sections, we use the scalar version of Eq. (4) to simplify the notation, as the results extend readily to the multidimensional case using diagonalization and multivariable calculus.

A typical way of studying the stability of discretization methods is by studying the solution to linear time-invariant ordinary differential equations (ODEs) of the form

$$x_t(t) = \lambda x(t), \lambda \in \mathbb{C}, \quad (9)$$

whose update equation can typically be written in the form

$$\bar{x}(t_{n+1}) = \sum_{m=0}^N a_m(\lambda) \bar{x}(t_{n-m}), a_0(\lambda), \dots, a_N(\lambda) \in \mathbb{C}. \quad (10)$$

If we denote $\bar{\lambda}_m(\lambda)$ the $N+1$ roots of the polynomial

$$p(z) = z^{N+1} - \sum_{m=0}^N a_m(\lambda) z^{N-m}, \quad (11)$$

the stability region of a method is then defined as the set of λ such that $\forall m \in \{0, \dots, N\}$, we have $|\bar{\lambda}_m(\lambda)| < 1$ (i.e., $\bar{\lambda}_m(\lambda)$ is inside the unit sphere). For both the trapezoidal method and the midpoint method, we have a single root $\bar{\lambda}$ written as

$$\bar{\lambda}(\lambda) = \frac{1 + T\lambda/2}{1 - T\lambda/2}, \quad (12)$$

so that the stability region corresponds to left half of the complex plane $\{\lambda, \text{Re}(\lambda) < 0\}$. This means that the two methods have the desirable property of being A-stable [23]. This also means that, for linear systems, the two methods map the system poles and zeros exactly the same way. We then expect both methods to exhibit similar qualitative behavior, such as resonant peaks near the Nyquist frequency for stiff systems (i.e., systems with strongly damped poles) and frequency warping [14].

3.4. Discretization error and order of accuracy

The discretization error of a method is typically characterized using the equation $x_t(t) = f(x(t))$ by deriving the error between $x(t_{n+1})$ solution of

$$x(t_{n+1}) = x(t_n) + \int_{t_n}^{t_{n+1}} f(x(\tau)) d\tau \quad (13)$$

and \bar{x}^{n+1} solution of the discretized equation

$$\bar{x}^{n+1} = x(t_n) + T \bar{f}(\bar{x}^{n+1}, x(t_n), \dots, x(t_{n-N})). \quad (14)$$

This error is typically computed in terms of a polynomial in T using the Taylor expansion of $x(t_{n+1}) - \bar{x}^{n+1}$ [1] so that

$$x(t_{n+1}) - \bar{x}^{n+1} = \sum_{n=0}^{+\infty} \epsilon(n) T^n, \quad (15)$$

where $\epsilon(t_n)$ can be expressed in terms of function derivatives f_{x^k} . The order of accuracy of a method is then defined as the integer n for which $\epsilon_m = 0$ for all $m \leq n$. As symmetric methods, both the trapezoidal method and the midpoint method are second-order accurate ($\epsilon_0 = \epsilon_1 = \epsilon_2 = 0$). Beyond that, for the trapezoidal method, we have the 3rd and 4th order error terms as

$$\begin{aligned}\epsilon_3^{\text{tr}} &= -\frac{(f)^2 f_{xx} + f(f_x)^2}{12} \quad \text{and} \\ \epsilon_4^{\text{tr}} &= -\frac{(f)^3 f_{xxx} + 4(f^2) f_x f_{xx} + f(f_x)^3}{24},\end{aligned}\quad (16)$$

and, for the midpoint method, as

$$\begin{aligned}\epsilon_3^{\text{md}} &= \frac{(f)^2 f_{xx} - 2f(f_x)^2}{24} \quad \text{and} \\ \epsilon_4^{\text{md}} &= \frac{(f)^3 f_{xxx} - (f)^2 f_x f_{xx} - 4f(f_x)^3}{48}.\end{aligned}\quad (17)$$

We then see that depending on the nonlinear function characteristics, different patterns of constructive or destructive interference between the different terms will lead to different behaviors between the two methods.

3.5. First-order fixed-point behavior

Many systems of interest usually tend towards a steady-state solution after infinite time. Note however that not all systems behave this way (e.g., relaxation oscillators [24, 25]). Those steady-state (*equilibrium*) solutions x^e solve the implicit equation:

$$f(x^e) = 0. \quad (18)$$

A typical analysis of the equilibrium is to look at the value $f_x(x^e)$ to find if the equilibrium is *stable* ($f_x(x^e) < 0$) or *unstable* ($f_x(x^e) > 0$). Physical systems composed with passive and dissipative components typically present one or more stable equilibria due to the energy of the system being dissipated over time. For such equilibria, as the state variable of a system approaches x^e , the Hartman-Grobman theorem [26] guarantees they will behave similarly as the solutions of the linearized system around x^e which follow the exponentially decaying profile

$$x(t) \approx x(0) \exp(tf_x(x^e)) + x^e. \quad (19)$$

Sampled at times t_n , the update formula is expressed as

$$x(t_{n+1}) \approx (x(t_n) - x^e) \exp(Tf_x(x^e)) + x^e. \quad (20)$$

By construction, the equilibria of the discretized system using either the trapezoidal rule or the midpoint rule are identical to those of the original system as they also verify Eq. (18). In the vicinity of those equilibrium, both methods then behave as the linearized:

$$\bar{x}^{n+1} \approx \alpha(\bar{x}^n - x^e) + x^e, \quad (21)$$

with α related to $f_x(e)$ following Eq. (12), meaning

$$\alpha = \frac{1 + Tf_x(x^e)/2}{1 - Tf_x(x^e)/2}. \quad (22)$$

The stability properties of both methods guarantees that stable equilibria ($f_x(x^e) < 0$) are necessarily stable for the discretized sequences ($|\alpha| < 1$). However, we have no guarantee on the sign

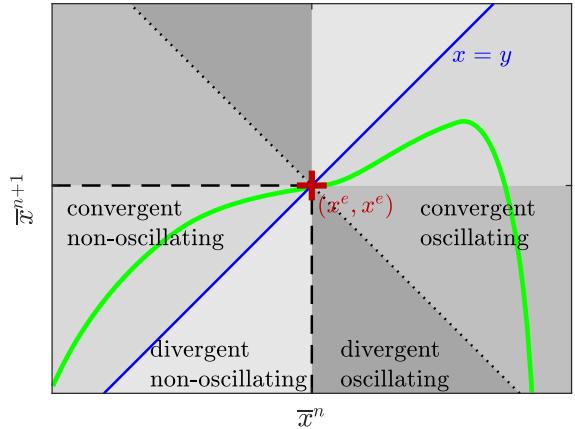


Figure 1: Solution sequence behavior regions for \bar{x}^{n+1} as a function of \bar{x}^n (in green) with reference to the point (x^e, x^e) .

of α , so that if $\alpha < 0$, the sign of $\bar{x}^n - x^e$ near the equilibrium alternates at each iteration, creating an oscillation that is not present in the original $x(t_n) - x^e$ which remains of the same sign according to Eq. (19). This oscillatory behavior matches the well-known oscillations exhibited by the solution of stiff systems discretized with the trapezoidal rule [14].

3.6. General fixed-point behavior

Further from an equilibrium, the first-order behavior from Eq. (21) may be insufficient to understand the behavior of the system in some regimes. Another tool we can use is to compute the transition equation \bar{x}^{n+1} as a function of \bar{x}^n . Once that transition function is known, we can deduce regions indicating whether the method is converging or diverging, i.e., (respectively):

$$|\bar{x}^{n+1} - x^e| < |\bar{x}^n - x^e| \quad \text{or} \quad |\bar{x}^{n+1} - x^e| > |\bar{x}^n - x^e| \quad (23)$$

and whether the method is oscillating or not, i.e., (respectively):

$$|\bar{x}^{n+1} - x^e| \cdot |\bar{x}^n - x^e| < 0 \quad \text{or} \quad |\bar{x}^{n+1} - x^e| \cdot |\bar{x}^n - x^e| > 0 \quad (24)$$

Those properties translate graphically as shown in Fig. 1. In that representation, a quasi-exponential decay such as the one described in Eq. (21) for $\alpha > 0$ corresponds to a linear section of curve in the convergent non-oscillating region. An oscillating exponential decay ($\alpha < 0$) corresponds to a linear section of curve in the convergent oscillating region.

3.7. Discretization error with variable input

When considering the influence of the input variables, the equation of interest becomes $x_t(t) = f(x(t), u(t))$. The error terms are then expressed as function of the partial derivatives $f_{x^k u^l}$ of f .

Expectedly, with the added input variables, both methods remain second-order accurate ($\epsilon_0 = \epsilon_1 = \epsilon_2 = 0$). The 3rd-order error term for the trapezoidal method becomes

$$\begin{aligned}\epsilon_3^{\text{tr}} &= -((f)^2 f_{xx} + 2u_{tf} f_{xu} + (u_t)^2 f_{uu})/12 \\ &\quad - ((f f_x + u_t f_u) f_x + u_{tt} f_u)/12,\end{aligned}\quad (25)$$

and one for the midpoint method becomes

$$\begin{aligned}\epsilon_3^{\text{md}} = & ((f)^2 f_{xx} + 2u_t f f_{xu} + (u_t)^2 f_{uu})/24 \\ & - ((ff_x + u_t f_u) f_x + u_{tt} f_u)/12.\end{aligned}\quad (26)$$

Here again, we see how the two methods will present different patterns of interference altering their behavior as a function of f as the influence of the input on the error terms differ as well.

4. IMPLEMENTATION CONSIDERATIONS

Many publications have presented various implementations of the trapezoidal method for the simulation of audio lumped systems so we refer the reader to those articles for more details [3–10]. We detail here approaches regarding the implementation of the midpoint method in typical audio circuit simulation frameworks. Generally, the implementation of the midpoint method can be derived in two ways: either through simple modifications of the system obtained using the more typical trapezoidal method or directly from the output of that trapezoidal-rule system with a simple transformation of the output as detailed below.

4.1. Direct implementation of the midpoint method

A direct implementation of the midpoint method computes the $\bar{x}_{\text{md}}^{n+1}$ by solving the implicit equation given in Eq. (8), similarly as what we do for the trapezoidal method when solving Eq. (6). Depending on the equation, similar strategies can be exploited, using analytical inverse functions when available or numerical root-finding methods otherwise. One potential downside of the midpoint method is that the expressions relative to the time step t_{n+1} and the time step t_n are grouped together inside the nonlinear function \mathbf{f} which may complicate the derivation and use of analytical inverse functions. On the other hand, notice that the update equation does not depend on two independent input samples $\bar{u}_{\text{md}}^{n+1}$ and \bar{u}_{md}^n but on the combined $\frac{1}{2}(\bar{u}_{\text{md}}^{n+1} + \bar{u}_{\text{md}}^n)$ so that the dimensionality of the update equation with respect to the input can be reduced compared to the trapezoidal method.

4.2. Midpoint method using the trapezoidal rule

The midpoint method and the trapezoidal method are conjugate methods [11]. More precisely, if the sequence \bar{x}_{md}^n verifies Eq. (8) for the input sequence \bar{u}_{md}^n , we can show that the sequence $\bar{x}_{\text{tr}}^n = \frac{1}{2}(\bar{x}_{\text{md}}^n + \bar{x}_{\text{md}}^{n-1})$ verifies Eq. (6) for the modified input sequence $\bar{u}_{\text{tr}}^n = \frac{1}{2}(\bar{u}_{\text{md}}^n + \bar{u}_{\text{md}}^{n-1})$ [27]. This means that if we want to implement the midpoint rule to simulate the state equation

$$\mathbf{x}_t(t) = \mathbf{f}(\mathbf{x}(t), \mathbf{u}(t)), \quad (27)$$

we can do so by simulating that same equation using the trapezoidal rule replacing the input $\mathbf{u}(t)$ by $\frac{1}{2}(\mathbf{u}(t) + \mathbf{u}(t-T))$. As a consequence, if a system has been designed to generate a simulated sequence \bar{x}_{tr}^n of equation (27) using the trapezoidal rule, we can obtain a simulated sequence \bar{x}_{md}^n for the input sequence \bar{u}_{md}^n iteratively as follows:

- Generate a new sample $\bar{x}_{\text{tr}}^{n+1}$ from the trapezoidal rule system using the input samples defined as

$$\begin{cases} \bar{u}_{\text{tr}}^{n+1} = \frac{1}{2}(\bar{u}_{\text{md}}^{n+1} + \bar{u}_{\text{md}}^n), \\ \bar{u}_{\text{tr}}^n = \frac{1}{2}(\bar{u}_{\text{md}}^n + \bar{u}_{\text{md}}^{n-1}). \end{cases} \quad (28)$$

- Generate a new sample $\bar{x}_{\text{md}}^{n+1}$ using either the (recurrent) equation:

$$\bar{x}_{\text{md}}^{n+1} = 2\bar{x}_{\text{tr}}^{n+1} - \bar{x}_{\text{tr}}^n \quad (29)$$

or the (direct) equation:

$$\bar{x}_{\text{md}}^{n+1} = \bar{x}_{\text{tr}}^{n+1} + \frac{T}{2}\mathbf{f}(\bar{x}_{\text{tr}}^{n+1}, \bar{u}_{\text{tr}}^{n+1}). \quad (30)$$

We also need to use the appropriate initial conditions for the trapezoidal-rule system based on the initial conditions of the desired midpoint sequence. In a typical case, those initial conditions correspond to $\bar{x}_{\text{md}}^0 = \mathbf{x}(0)$ for the state variables and $\bar{u}_{\text{md}}^0 = \mathbf{u}(t_0)$ for the input variables, typically based on a specified input function $\mathbf{u}(t)$ defined for $t \geq 0$. In addition to specifying the initial state value \bar{x}_{tr}^0 , the trapezoidal-rule system requires to also define \bar{u}_{tr}^0 which depends on the unspecified \bar{u}_{md}^{-1} . Note that \bar{x}_{tr}^0 and \bar{u}_{tr}^0 cannot be chosen independently. We can pick any initial condition pair $(\bar{x}_{\text{tr}}^0, \bar{u}_{\text{tr}}^0)$ as long as it verifies

$$\bar{x}_{\text{tr}}^0 + \frac{T}{2}\mathbf{f}(\bar{x}_{\text{tr}}^0, \bar{u}_{\text{tr}}^0) = \bar{x}_{\text{md}}^0. \quad (31)$$

A typical assumption is that the system was in a steady-state condition before the simulation started, so that the initial condition are given as a function of \bar{x}_{md}^0 by solving

$$\begin{cases} \bar{x}_{\text{tr}}^0 = \bar{x}_{\text{md}}^0, \\ \mathbf{0} = \mathbf{f}(\bar{x}_{\text{md}}^0, \bar{u}_{\text{tr}}^0). \end{cases} \quad (32)$$

4.3. Trapezoidal method using the midpoint rule

A similar principle can be used to generate a simulation based on the trapezoidal rule using a system designed to simulate that same system using the midpoint rule using the procedure:

- Generate a new sample $\bar{u}_{\text{md}}^{n+1}$ from the midpoint rule system using the input sample defined recursively as

$$\bar{u}_{\text{md}}^{n+1} = 2\bar{u}_{\text{tr}}^{n+1} - \bar{u}_{\text{md}}^n. \quad (33)$$

- Generate a new sample $\bar{x}_{\text{tr}}^{n+1}$ using either

$$\bar{x}_{\text{tr}}^{n+1} = \frac{1}{2}(\bar{x}_{\text{md}}^{n+1} + \bar{x}_{\text{md}}^n) \quad (34)$$

or solving the implicit equation

$$\bar{x}_{\text{tr}}^{n+1} + \frac{T}{2}\mathbf{f}(\bar{x}_{\text{tr}}^{n+1}, \bar{u}_{\text{tr}}^{n+1}) = \bar{x}_{\text{md}}^{n+1}. \quad (35)$$

Finding the initial conditions is less complex in that case. The initial source sample \bar{u}_{md}^0 can be chosen arbitrarily. If the input function $\mathbf{u}(t)$ is known for $t \geq 0$, a possible choice would be $\bar{u}_{\text{md}}^0 = \mathbf{u}(\frac{T}{2})$. An alternative option is $\bar{u}_{\text{md}}^0 = \frac{1}{2}(\bar{u}_{\text{tr}}^1 + \bar{u}_{\text{tr}}^0)$, which does not require explicit knowledge of $\mathbf{u}(t)$. The initial state \bar{x}_{md}^0 is obtained as

$$\bar{x}_{\text{md}}^0 = \bar{x}_{\text{tr}}^0 + \frac{T}{2}\mathbf{f}(\bar{x}_{\text{tr}}^0, \bar{u}_{\text{tr}}^0). \quad (36)$$

4.4. Considerations for typical audio systems

A large share of the audio systems presented in the literature are characterized by having the dynamical elements only be linear (e.g., capacitor, inductor), and having the nonlinear elements only be memoryless (e.g., diode, transistor, operational amplifier). Several simulation frameworks for audio systems (e.g., wave digital filters [9, 28], nodal DK method [29], generalized state space [8])

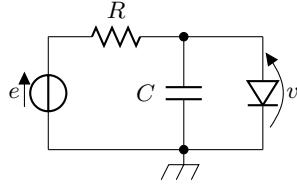


Figure 2: Diode clipper circuit.

segregate the system between sets of memoryless nonlinear equations and sets of linear equations between the system variables.

For the midpoint method, discretizing the linear equations is the same process as for the trapezoidal method, so the discrete update equations are identical. Memoryless nonlinear equations (e.g., voltage–current characteristics) are often of the form $\mathbf{b}(t) = \mathbf{h}(\mathbf{a}(t))$ (e.g., with \mathbf{a} voltages and \mathbf{b} currents for a voltage–current characteristic). On the contrary to the trapezoidal method where the update equations remain memoryless as $\bar{\mathbf{b}}^n = \mathbf{h}(\bar{\mathbf{a}}^n)$, the midpoint method equations become

$$\bar{\mathbf{b}}^{n+1} = -\bar{\mathbf{b}}^n + 2\mathbf{h}\left(\frac{1}{2}(\bar{\mathbf{a}}^{n+1} + \bar{\mathbf{a}}^n)\right). \quad (37)$$

This transformation would then be applied to the nonlinear elements at the root of a wave digital filter tree [9, 28], or to the nonlinear characteristic equations in the nodal DK approach [3] and in the generalized state-space [8].

Alternatively, we can use the process described in Sec. 4.2 to exploit simulations that use the trapezoidal method, applying the appropriate transformation to the input sequence (following Eq. (28)), and adding the conversion equations (following Eqs. (29) and (30)) to get the state variables for the midpoint simulation.

5. CASE STUDY

We study the diode clipper [14, 19, 20, 30] as shown in Fig. 2 to illustrate the concepts developed in the previous sections, as we study and compare the behavior of both methods in several scenarios and provide tools to understand and forecast such behavior.

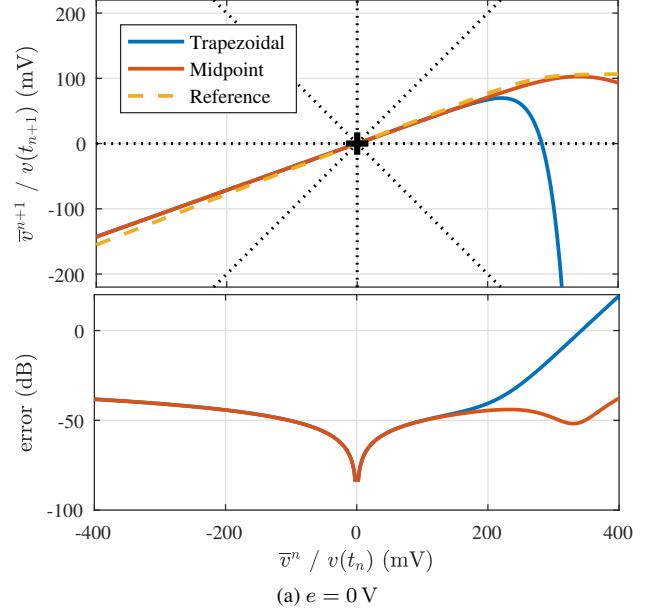
5.1. Circuit description

The circuit consists of a voltage input (i.e., a source), a resistor, a capacitor and a diode. The diode and capacitor are in parallel, their combination is in series with the resistor and the voltage source, and we wish to simulate the voltage v across the diode as a function of the source voltage e . The system behavior can then be summarized as the state-space system

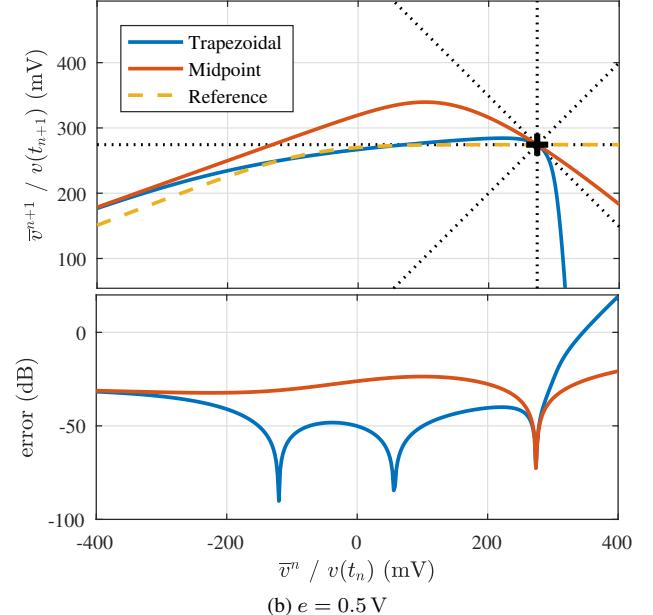
$$v_t(t) = \frac{e(t) - v(t)}{RC} - \frac{g(v(t))}{C}, \quad (38)$$

Table 1: Simulation parameters.

Name	Value	Description
f_s	48 kHz	sampling frequency
T	20.83 μ s	sampling period
R	2.2 k Ω	resistor
C	10 nF	capacitor
I_S	2.52 nA	N914 saturation current
V_T	25.85 mV	thermal voltage



(a) $e = 0$ V



(b) $e = 0.5$ V

Figure 3: $v(t_{n+1})/\bar{v}^{n+1}$ and error for as function of $v(t_n)/\bar{v}^n$ for two input conditions. The equilibria are indicated with a black +.

with g is the diode voltage–current characteristic. We then have:

$$\begin{aligned} v_t(t) &= f(v(t), e(t)), \text{ with} \\ f(x, u) &= (u - x)/RC - g(x)/C. \end{aligned} \quad (39)$$

We use the common Shockley diode model [31] for which

$$g(x) = I_S(e^{x/V_T} - 1), \quad (40)$$

with I_S the saturation current and V_T the thermal voltage.

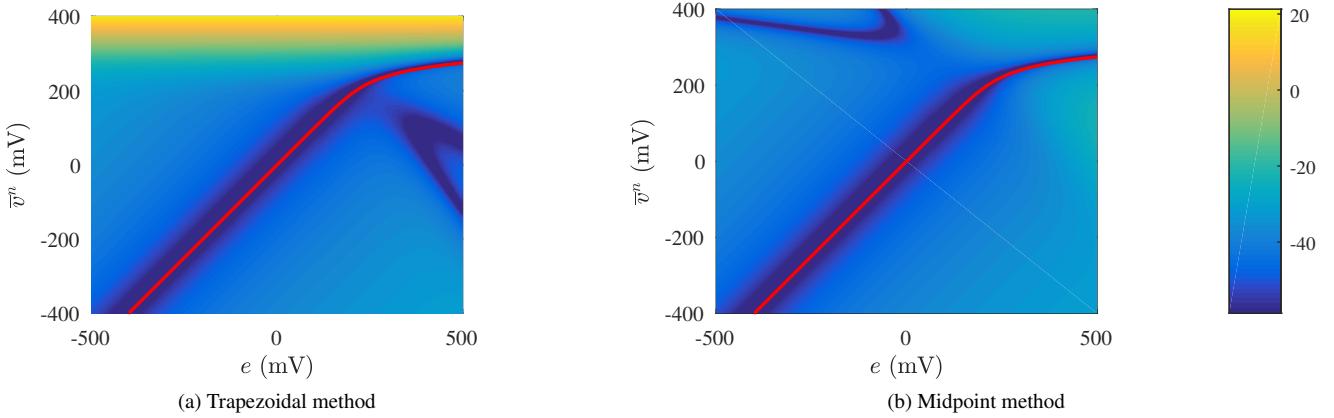


Figure 4: Error (in dB) for both methods. The equilibria are indicated in red.

5.2. Simulation parameters

We adopt typical parameter values for fixed-rate virtual analog simulations as shown in Tab. 1. The response of the reference system is approximated using the MATLAB adaptive solver `ode45` set to work with a target error at machine precision. The equilibria and the implicit equations for both numerical methods are computed using the MATLAB root finder `fzero` set to work with a target error at machine precision.

5.3. Constant-input study

Many typical inputs are characterized by a constant-input regime. For example, an impulse response will lead to a constant zero input after the impulse. Similarly, a step response will lead to a constant non-zero input after the transient. In such a case, the system can be interpreted as an ODE (i.e., as a system without input), with the input voltage becoming a constant in the equation such that

$$v_t(t) = h(v(t)) = f(v(t), e(t)). \quad (41)$$

First, we investigate the error made by each method on a single simulation step, i.e. when computing the new state \bar{v}^{n+1} (or $v(t_{n+1})$) from a current state \bar{v}^n (or $v(t_n)$). Fig. 3 show the new state \bar{v}^{n+1} and the error $|v(t_{n+1}) - \bar{v}^{n+1}|$ as a function of the current state \bar{v}^n (or $v(t_n)$) for two constant voltage source values (respectively $e = 0$ and $0.5V$). The plots show that while the two methods expectedly compare with one another and the reference for lower $v(t_n)$ values where the system behaves quasi-linearly, the error for the trapezoidal rule rises sharply for higher values (roughly 200 mV and higher). That region corresponds to the region where the system is highly non-linear and stiff. Those observations extend readily to other choices of input voltage and state values as shown in Fig. 4 where we display the error as a function of the current state $v(t_n)$ and the constant input value e .

5.4. Fixed-point analysis

Using the analysis described in Secs. 3.5 and 3.6 for Fig. 1, we further understand the behavior for the two methods and the reference. In Fig. 3a, we see that for both methods and the reference,

the first-order behavior around the system equilibrium is a non-oscillating decaying exponential. Furthermore, the transition functions are quasi-linear functions for a large region of state values around the equilibrium, where we have a quasi-exponential decay of the state sequence once a state value falls within that region. For high state voltages however, while the midpoint method stays close to the reference in the convergent non-oscillating region, the trapezoidal method drops sharply in the oscillating region. This means that a state sample in that region will iterate to a sample with a much lower voltage than the true solution (i.e., overshoot the true voltage) before entering the convergent non-oscillating regime.

In Fig. 3b, we see clearly that the first-order behavior near the equilibrium is substantially different for the reference on the one hand and the two methods on the other hand. The reference follows again a non-oscillating exponential decay in that region, while the two methods present an oscillating exponential decay as shown by the negative slope of the transition function at the equilibrium. Beyond that region, the behavior of the two methods are actually very different. The midpoint method shows a decaying oscillating behavior on a large section of state values both above and below the equilibrium and only match the non-oscillating exponential decay behavior of the reference solution for low state values ($\lesssim -100$ mV). On the other hand, the trapezoidal method presents a non-oscillating quasi-exponential decay close to the reference for almost all state values below the equilibrium, with small oscillation only close to the equilibrium. However, the behavior above the equilibrium is again characterized by a transition function dropping again sharply in the oscillating region.

Fig. 5 shows the behavior of both methods for a wide range of e and \bar{v}^n values. Knowing that the reference is always converging non-oscillating (i.e., light gray), its behavior is matched by the trapezoidal method only if the state variable does not exceed a value with a similar order of magnitude across the tested e values. The method becomes however oscillating above those values, becoming even diverging oscillating for very high \bar{v}^n . The midpoint method is convergent non-oscillating over a wide range of values for e and \bar{v}^n values. As hinted in Fig. 3b, we also see that high values of \bar{v}^n , for which the trapezoidal method is oscillating, keep a converging non-oscillating behavior if the input value e is low. However, the midpoint method is oscillating for an increasing range of high \bar{v}^n values as the input value e becomes high.

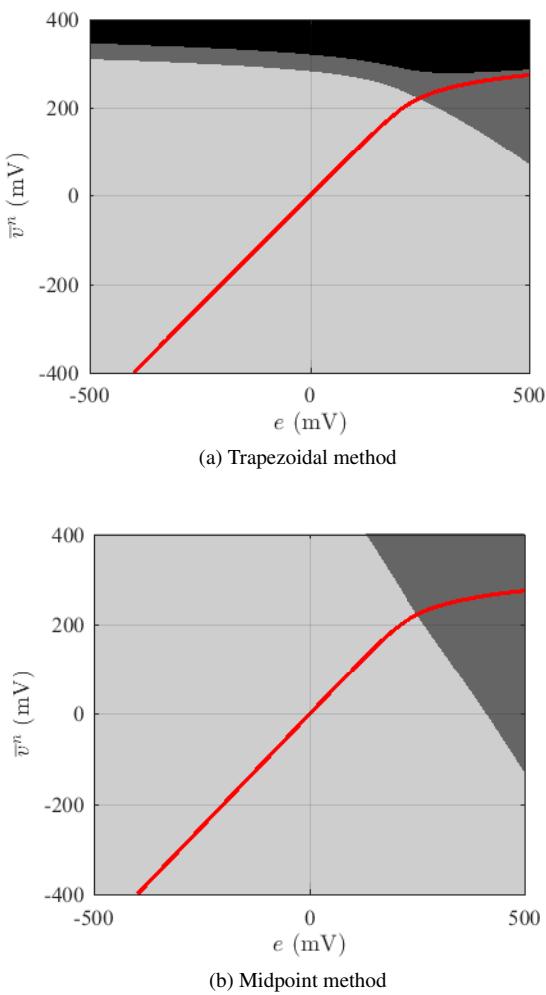


Figure 5: Behavior regions for both methods (Light gray: convergent non-oscillating, dark gray: convergent oscillating, black: divergent oscillating). Equilibria are indicated in red.

5.5. Step response

Finally, we look at the step response of the system for various initial states $v(0) = v^0$ and input values e in Fig. 6. The behavior matches the results from the previous section with all systems converging to the equilibrium v^e (with $f(v^e, e) = 0$). In particular, we can observe how, in the case of a low input voltage with a high initial state, the trapezoidal method systematically overshoots the true response, resulting in a significant simulation error. We also observe in Fig. 6d that for a high input voltage and a high initial state, both the trapezoidal method and the midpoint method oscillate significantly around the equilibrium. Finally, Fig. 6c shows how for a high input voltage and a low initial state, the oscillation amplitude for the midpoint is much greater. In general, we see that each method shows regions of that neither method is absolutely superior in all scenarios, so that the best method should rather be determined with a scenario-dependent approach. It also shows how some behaviors are shared among both methods and cannot be avoided, such as the oscillations in Fig. 6b and 6d.

6. CONCLUSION

In this paper, we presented a comparison of the implicit midpoint method with the more commonly used trapezoidal method as discretization methods for time-invariant audio lumped system simulation, with a specific focus on fixed-rate simulation. We presented some of the relevant theoretical similarities and differences between the two methods. We particularly focused on quantifying of the oscillatory behavior that both methods exhibit for stiff systems. We then discussed the practical implementation of the implicit midpoint method and how an implicit midpoint solution sequence could be computed from a trapezoidal-based implementation and vice versa. Finally, we compared the behavior of those two methods on a simple diode clipper system, drawing from our earlier theoretical analysis, in order to predict their behavior and identify cases of better performance for each method.

From a larger perspective, this paper is also meant as the preliminary of tools for the design and comparison of discretization methods for the specific purpose of fixed-rate simulation of audio lumped systems. Future work will focus on extending and improving those tools, as well as integrating into our analysis additional considerations relevant to the audio field, such as aliasing. Ultimately, we intend on applying those tools to a wide class of numerical methods to derive heuristics for the systematic design of accurate and efficient fixed-rate simulations.

Acknowledgment—The author wishes to thank Dr. Kurt J. Werner for his thoughtful comments and suggestions.

7. REFERENCES

- [1] P. Moin, *Fundamentals of engineering numerical analysis*, Cambridge Univ. Press, New York, NY, 2010.
- [2] S. Bilbao, *Numerical sound synthesis: Finite difference schemes and simulation in musical acoustics*, J. Wiley, Chichester, UK, 2009.
- [3] D. T. Yeh, J. S. Abel, A. Vladimirescu, and J. O. Smith, “Numerical methods for simulation of guitar distortion circuits,” *Comput. Music J.*, vol. 32, no. 2, pp. 23–42, 2008.
- [4] A. Fettweis, “Wave digital filters: Theory and practice,” *Proc. IEEE*, vol. 74, no. 2, pp. 270–327, 1986.
- [5] G. Borin, G. De Poli, and D. Rocchesso, “Elimination of delay-free loops in discrete-time models of nonlinear acoustic systems,” *IEEE Trans. Speech and Audio Process.*, vol. 8, no. 5, pp. 597–605, 2000.
- [6] S. Bilbao, *Wave and scattering methods for numerical simulation*, John Wiley & Sons, 2004.
- [7] 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. 715–27, 2010.
- [8] M. Holters and U. Zölzer, “A generalized method for the derivation of non-linear state-space models from circuit schematics,” in *Proc. 23rd European Signal Process. Conf.*, 2015.
- [9] K. J. Werner, V. Nangia, J. O. Smith, and J. S. Abel, “Resolving wave digital filters with multiple/multiport nonlinearities,” in *Proc. 18th Int. Conf. Digital Audio Effects*, 2015.
- [10] K. J. Werner, J. O. Smith, and J. S. Abel, “Wave digital filter adaptors for arbitrary topologies and multiport linear elements,” in *Proc. 18th Int. Conf. Digital Audio Effects*, 2015.
- [11] E. Hairer, C. Lubich, and G. Wanner, *Geometric numerical integration: structure-preserving algorithms for ordinary differential equations*, Springer, 2006.
- [12] A. Falaise, *Modélisation, simulation, génération de code et correction de systèmes multi-physiques audio: Approche par réseau de composants et formulation Hamiltonienne à Ports*, Ph.D. diss., Université Pierre et Marie Curie, Paris, France, 2016.

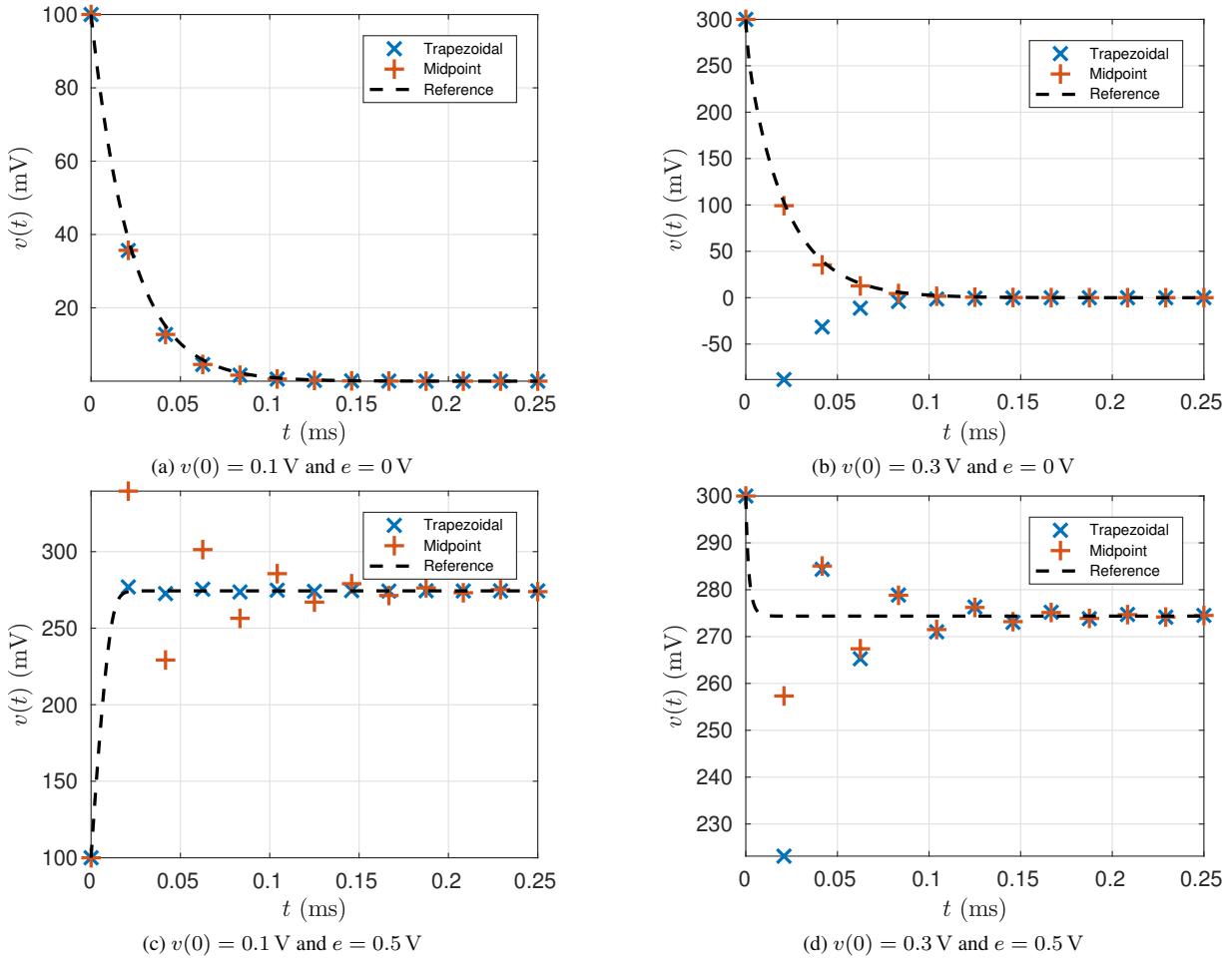


Figure 6: Step response simulation \bar{x}^n and reference $x(t)$ for different initial conditions and constant input values.

- [13] A. Falaize and T. Hélie, “Passive guaranteed simulation of analog audio circuits: A port-Hamiltonian approach,” *Appl. Sci.*, vol. 6, no. 10, 2016.
- [14] F. G. Germain and K. J. Werner, “Design principles for lumped model discretisation using Möbius transforms,” in *Proc. 18th Int. Conf. Digital Audio Effects*, 2015.
- [15] D. T. Yeh, J. S. Abel, and J. O. Smith, “Simulation of the diode limiter in guitar distortion circuits by numerical solution of ordinary differential equations,” in *Proc. 10th Int. Conf. Digital Audio Effects*, 2007.
- [16] T. Stilson, *Efficiently-variable non-oversampled algorithms in virtual-analog music synthesis*, Ph.D. diss., Stanford Univ., 2006.
- [17] F. G. Germain and K. J. Werner, “Joint parameter optimization of differentiated discretization schemes for audio circuits,” in *Proc. 142 Conv. Audio Eng. Soc.*, 2017.
- [18] F. G. Germain and K. J. Werner, “Optimizing differentiated discretization for audio circuits beyond driving point transfer functions,” in *Proc. IEEE Workshop Appl. Signal Process. Audio Acoust.*, 2017.
- [19] P. Daly, “A comparison of virtual analogue Moog VCF models,” M.S. thesis, Univ. of Edinburgh, Edinburgh, UK, 2012.
- [20] R. C. D. 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. 688–92, 2012.
- [21] K. M. Hangos, J. Bokor, and G. Szederkényi, *Analysis and control of nonlinear process systems*, Springer, 2006.
- [22] E. Süli and D. F. Mayers, *An introduction to numerical analysis*, Cambridge Univ. Press, 2003.
- [23] G. G. Dahlquist, “A special stability problem for linear multistep methods,” *BIT Numer. Math.*, vol. 3, no. 1, pp. 27–43, 1963.
- [24] P. Howoritz and W. Hill, *The Art of Electronics*, Cambridge Univ. Press, 2nd edition, 1989.
- [25] 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*, 2017.
- [26] E. A. Coayla-Teran, S.-E. A. Mohammed, and P. R. C. Ruffino, “Hartman-Grobman theorems along hyperbolic stationary trajectories,” *Discrete Contin. Dyn. Syst.*, vol. 17, no. 2, pp. 281–92, 2007.
- [27] G. G. Dahlquist and B. Lindberg, “On some implicit one-step methods for stiff differential equations,” Tech. Rep. TRITA-NA-73-02, The Royal Institute of Technology, Stockholm, Sweden, 1973.
- [28] 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. Digital Audio Effects*, 2016.
- [29] 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 Trans. Audio, Speech, Language Process.*, vol. 18, no. 4, pp. 728–37, 2010.
- [30] J. Macak and J. Schimmel, “Nonlinear circuit simulation using time-variant filter,” in *Proc. 12th Int. Conf. Digital Audio Effects*, 2009.
- [31] A. S. Sedra and K. C. Smith, *Microelectronic circuits*, New York: Oxford University Press, 1998.

GENERALIZING ROOT VARIABLE CHOICE IN WAVE DIGITAL FILTERS WITH GROUPED NONLINEARITIES

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

Michael Jørgen Olsen

Center for Computer Research
in Music and Acoustics (CCRMA)
Stanford University, CA, USA
mjolsen@ccrma.stanford.edu

Maximilian Rest

E-RM Erfindungsbüro
Berlin, Germany
m.rest@e-rm.de

Julian D. Parker

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

ABSTRACT

Previous grouped-nonlinearity formulations for Wave Digital Filter (WDF) modeling of nonlinear audio circuits assumed that nonlinear (NL) devices with memoryless voltage–current characteristics were modeled as voltage-controlled current sources (VCCSs). These formulations cannot accommodate nonlinear devices whose equations cannot be written as NL VCCSs, and they cannot accommodate circuits with cutsets composed entirely of current sources (including NL VCCSs). In this paper we generalize independent and dependent variable choice at the root of WDF trees to accommodate both these cases, and review two graph theorems for avoiding forbidden cutsets and loops in general.

1. INTRODUCTION

Along with State Space Modeling [1–4] and Port Hamiltonian Systems [5–8], the *Wave Digital Filter* (WDF) [9–15] formalism is a major approach to Virtual Analog modeling of audio circuits. In short, the Wave Digital Filter approach reframes an electrical circuit into a tree-like structure that separates electrical elements from their topological connections, represents each electrical element and topological connection mathematically with explicit input–output relationships in the (typically voltage or power) wave domain, discretizes reactive elements (most commonly using the trapezoidal rule), and resolves delay-free loops in the resulting structure by tuning the port resistance parameter of the wave variable definition at each port in the circuit. For circuits built entirely out of series and parallel connections, this approach is entirely modular. Good numerical behavior and incremental passivity can normally be inherited from passivity of the reference circuit and the use of wave variables [16]. The original WDF formalism could accommodate a single nonlinear electrical element at the root of the WDF tree [17]; since typical audio circuits rather contain multiple nonlinear devices, handling multiple nonlinearities is currently a main subject of WDF research. Three recent approaches to handling multiple nonlinearities involve the resolution of fictitious delays in non-tree-like structures using techniques from multidimensional signal processing with convergence guaranteed in some cases by the contractive properties of passive circuit elements [18–21], “dynamic adaptation” of one-port nonlinearities which are not at the root [22], or global iterative solution over a WDF structure [23].

In this paper, we'll deal with another recent thread in WDF research starting with [24] that is focused on solving the scattering matrices of complicated “ \mathcal{R} -type” adaptors [11] and the application of the topological insights of [25] (decomposition of a circuit graph into an SPQR tree) to generalized formulations for circuits involving multiple nonlinearities [12,13]. This approach groups all

nonadaptable elements (most importantly nonlinearities) together at the root of a WDF tree¹ and interfaces that multiport root element to standard WDF subtrees using an \mathcal{R} -type adaptor. Representing the mathematical relationships in this structure directly causes a number of delay-free loops in the signal flow graph, which may be resolved by a variety of methods.

Findings from this thread have enabled WDF simulation of circuits with complicated non-series/parallel topologies that may involve absorbed multiport linear elements [11] as well as multiple non-adaptable linear circuit elements [15] or nonlinear elements [12] grouped together at the root of a tree structure. These include audio circuits that were previously out of scope for WDF modeling: guitar tone stacks, active tone control circuits [11], the Hammond organ vibrato/chorus [15], circuits using operational amplifiers (modeled as ideal or using macromodels) [14] or operational transconductance amplifiers [27], Sallen–Key filters [28,29], guitar distortion stages [12], transistor [13] and triode [30] amplifiers, the Fender 5F6-A preamp stage [31], the Korg MS-50 Filter [32], relaxation oscillators [33], and the bass drum circuit from the Roland TR-808 Rhythm Composer [34].

However, two classes of circuits which may appear to be handled by those techniques on a topological level actually fail for reasons related to the choice of independent and dependent variables in the constituent equations of the circuits' nonlinear devices. The first class of circuits involves nonlinear devices whose constituent equations are inherently written in a way that is incompatible with the previous WDF formulation [12]. The second class of circuits involves forbidden topological combinations of nonlinear devices with a certain mathematical description and ideal sources. Specifically these are circuits involving:

1. cutsets on the circuit graph composed entirely of nonlinearities with current as the dependent variable and ideal independent or controlled current sources; or
2. loops on the circuit graph composed entirely of nonlinearities with voltage as the dependent variable and ideal independent or controlled voltage sources.

These restrictions also apply in the context of State Space and Port Hamiltonian System modeling. In the State Space context, it is acknowledged that any node with only nonlinearities attached creates an inconsistent system if they are treated as controlled current sources, motivating the development of the Generalized State Space approach [4]. This is a special case of the restriction that a circuit may not contain any cutsets composed entirely of devices with current as the dependent variable. In the Port Hamiltonian System context, “implicit” formulations which in practice corre-

¹Similar to [26], although allowing root topologies other than parallel.

spond to series combinations of voltage-controlled (controlled current) components or parallel combinations of current-controlled (controlled voltage) components require extensions to the standard approach [6]. These appear to be two special cases of the cutset and loop restrictions respectively. It is possible that the broader topological restrictions may not be widely recognized in audio circuit modeling since the most common problematic arrangement of diodes—series combinations of diodes and anti-parallel combinations of diodes in series—can be reasonably and intuitively modeled as a single one-port device in many circumstances [35].

In this paper, we extend a WDF formalism involving grouped nonlinearities at the root of an SPQR tree [11–13] to accommodate a wider variety of independent-dependent variable descriptions of nonlinear elements, enabling simulation of the two problematic classes of circuits mentioned above. For the first class of circuits, we exploit the proposed generalization to accommodate whatever pair of independent-dependent variables is required for each nonlinearity. For the second class of circuits, we explain the meaning of the forbidden topological connections and variable choices in terms of graph and network theory, give principles for choosing proper independent-dependent variable pairs, and show how to implement these principles using the proposed generalization. In this way, the class of problematic circuits which have been accommodated in the State Space and Port Hamiltonian context may be accommodated in the Wave Digital Filter context as well.

The rest of the paper is structured as follows. §2 outlines the proposed generalization to the WDF approach. §3 explains the first problematic class of circuits and how they can be accommodated using the proposed generalization. §4 explains the second problematic class of circuits and how they can be accommodated using the proposed generalization. To support a general search for problematic cutsets and loops, §5 reviews two graph theorems.

2. WDFS WITH GROUPED NONLINEARITIES

In this section we review the method of grouped nonlinearities in WDFs, simultaneously extending it to handle nonlinearities and nonadaptable linear circuit elements expressed with a wide variety of independent and dependent variables.

2.1. Overview

The method of grouped nonlinearities in WDFs is outlined diagrammatically in Fig. 1. In this formulation there are five conceptual relationships to consider:

- All nonlinearities of the circuit grouped together into a nonlinear multiport element at the root of a tree structure (labeled “nonlinearities”). The behavior of the group of nonlinearities is expressed mathematically by $\mathbf{y}_c = \mathbf{f}(\mathbf{x}_c)$, where \mathbf{x}_c is a vector of independent network port variables and \mathbf{y}_c is a vector of dependent network port variables.
- A change of variables \mathbf{x}_c and \mathbf{y}_c to the vectors of incident and reflected voltage waves \mathbf{a}_c and \mathbf{b}_c (labeled “change of variables”), the subscript c denoting “converter.”² This change of variables is expressed mathematically by the matrix \mathbf{C} with partitions \mathbf{C}_{11} , \mathbf{C}_{12} , \mathbf{C}_{21} , and \mathbf{C}_{22} .
- A compatibility relationship between \mathbf{a}_c and \mathbf{b}_c and “internal” port wave variables \mathbf{a}_i and \mathbf{b}_i , simply enforcing port

²This relates to earlier approaches to interfacing State Space and WDF systems [36].

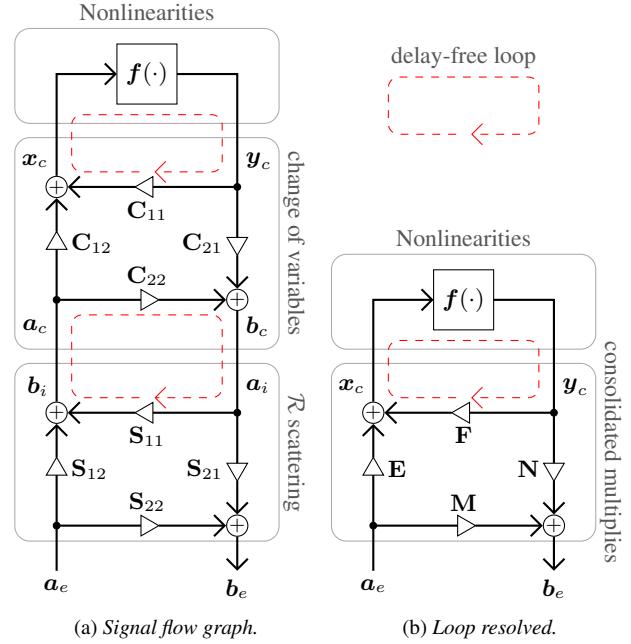


Figure 1: Proposed method signal flow graphs. \mathbf{a}_e is supplied by and \mathbf{b}_e delivered to classical WDF subtrees below.

connection criteria between the scattering matrix below and the multiport nonlinear element $\mathbf{y}_c = \mathbf{f}(\mathbf{x}_c)$. This is expressed mathematically by $\mathbf{a}_c = \mathbf{b}_i$ and $\mathbf{a}_i = \mathbf{b}_c$.

- Scattering between \mathbf{a}_i and \mathbf{b}_i and “external” port wave variables \mathbf{a}_e and \mathbf{b}_e . This is expressed mathematically by the scattering matrix \mathbf{S} with partitions \mathbf{S}_{11} , \mathbf{S}_{12} , \mathbf{S}_{21} , and \mathbf{S}_{22} . Because the circuit topology described by this scattering is usually a complicated \mathcal{R} -type topology [25], this is labeled “ \mathcal{R} scattering.”
- The rest of the WDF structure which is fed reflected waves \mathbf{b}_e by the root topology and provides incident waves \mathbf{a}_e . The rest of the structure has the form of a standard WDF “connection tree” [11, 37] and is not shown in the diagram.

These relationships are shown as a vector signal flow graph in Fig. 1a, where delay-free loops are represented by red dashed arrows. The root is described by the system of equations

$$\begin{array}{ll} \text{Nonlinearities} & \left\{ \begin{array}{l} \mathbf{y}_c = \mathbf{f}(\mathbf{x}_c) \\ \mathbf{x}_c = \mathbf{C}_{11}\mathbf{y}_c + \mathbf{C}_{12}\mathbf{a}_c \end{array} \right. & (1) \\ \text{change of variables} & \left\{ \begin{array}{l} \mathbf{b}_c = \mathbf{C}_{21}\mathbf{y}_c + \mathbf{C}_{22}\mathbf{a}_c \\ \mathbf{a}_c = \mathbf{b}_i \end{array} \right. & (2) \\ \text{compatibility} & \left\{ \begin{array}{l} \mathbf{a}_i = \mathbf{b}_c \\ \mathbf{a}_i = \mathbf{b}_c \end{array} \right. & (3) \\ \text{scattering} & \left\{ \begin{array}{l} \mathbf{b}_i = \mathbf{S}_{11}\mathbf{a}_i + \mathbf{S}_{12}\mathbf{a}_e \\ \mathbf{b}_e = \mathbf{S}_{21}\mathbf{a}_i + \mathbf{S}_{22}\mathbf{a}_e . \end{array} \right. & (4) \\ & & (5) \\ & & (6) \\ & & (7) \end{array}$$

Some of the delay-free loops can be resolved using matrix algebra [11], yielding a consolidated version of (1)–(7)

$$\mathbf{y}_c = \mathbf{f}(\mathbf{x}_c) \quad (8)$$

$$\mathbf{x}_c = \mathbf{E}\mathbf{a}_e + \mathbf{F}\mathbf{y}_c \quad (9)$$

$$\mathbf{b}_e = \mathbf{M}\mathbf{a}_e + \mathbf{N}\mathbf{y}_c \quad (10)$$

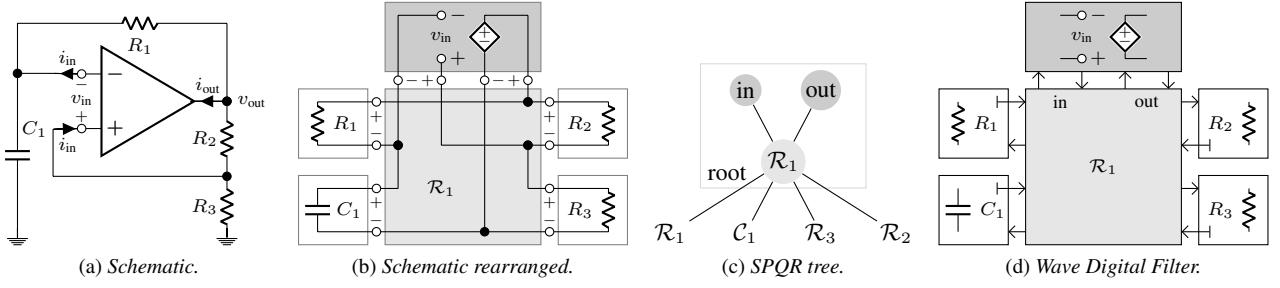


Figure 2: Relaxation oscillator schematic and derivation of Wave Digital Filter structure.

with

$$\begin{aligned} \mathbf{E} &= \mathbf{C}_{12}(\mathbf{I} + \mathbf{S}_{11}\mathbf{H}\mathbf{C}_{22})\mathbf{S}_{12}, & \mathbf{F} &= \mathbf{C}_{12}\mathbf{S}_{11}\mathbf{H}\mathbf{C}_{21} + \mathbf{C}_{11} \\ \mathbf{M} &= \mathbf{S}_{21}\mathbf{H}\mathbf{C}_{22}\mathbf{S}_{12} + \mathbf{S}_{22}, & \mathbf{N} &= \mathbf{S}_{21}\mathbf{H}\mathbf{C}_{21} \end{aligned}$$

where \mathbf{I} is the identity matrix and $\mathbf{H} = (\mathbf{I} - \mathbf{C}_{22}\mathbf{S}_{11})^{-1}$. This system of equations is shown as a vector signal flow graph in Fig. 1b; notice that one delay-free loop through \mathbf{F} and $f(\cdot)$ still remains. This delay-free loop can be resolved, e.g., by Newton-Raphson iteration [13], table lookup / the K method [12, 38], custom nonlinear solvers [33], or a final matrix inversion in the case that all nonadaptable elements are linear [15].

2.2. Populating \mathbf{S}

To use the method just outlined, matrices \mathbf{S} and \mathbf{C} must be populated. \mathbf{S} can be found using the techniques of [11], which applies Modified Nodal Analysis [39] to the \mathcal{R} -type topology where instantaneous Thévenin equivalents represent the incident wave at each port, combining with the WDF wave definition to solve for \mathbf{S} . If the \mathcal{R} -type adaptor contains no absorbed non-reciprocal elements, the method of [40] may be used to solve for \mathbf{S} .

2.3. New Considerations for \mathbf{C}

In previous work, the nonadaptable root elements were usually modeled in the Kirchhoff domain by $i = f(v)$.³ This aligns with standard models for common electrical elements like diodes (Shockley model), BJT transistors (Ebers-Moll model), and triodes. So, the derivation of the \mathbf{C} matrix assumed these Kirchhoff variables were to be converted to the wave variables a_c and b_c .

In this work, we generalize that approach so that each entry in the vector of independent (\mathbf{x}_c) and dependent (\mathbf{y}_c) root variables may be a voltage, current, or wave variable. These may be “mixed and matched” in two senses. First, it is not required that each nonlinear element have the same independent nor dependent variables. Second, it is possible to have, e.g., a Kirchhoff variable as an independent variable and a wave variable as a dependent variable (provided this still represents a single-valued function). It should be obvious that the independent and dependent variables at any port may not be linear combinations of one another.

In general, the relationship between the incident wave a_n and reflected wave b_n at a port n and two other variables which may

Table 1: $t_{11,n}$, $t_{12,n}$, $t_{21,n}$, $t_{22,n}$ for different dependent and independent variables at a port n .

x_n	independent variable		dependent variable		
	$t_{11,n}$	$t_{12,n}$	y_n	$t_{21,n}$	$t_{22,n}$
v_n	1/2	1/2	i_n	$1/(2R_n)$	$-1/(2R_n)$
i_n	$1/(2R_n)$	$-1/(2R_n)$	v_n	1/2	1/2
a_n	1	0	b_n	0	1

be linear combinations of a_n and b_n is expressed by

$$\begin{bmatrix} x_n \\ y_n \end{bmatrix} = \begin{bmatrix} t_{11,n} & t_{12,n} \\ t_{21,n} & t_{22,n} \end{bmatrix} \begin{bmatrix} a_n \\ b_n \end{bmatrix}. \quad (11)$$

Solving for x_n and b_n yields an equation in the form of \mathbf{C}

$$\begin{bmatrix} x_n \\ b_n \end{bmatrix} = \begin{bmatrix} c_{11,n} & c_{12,n} \\ c_{21,n} & c_{22,n} \end{bmatrix} \begin{bmatrix} y_n \\ a_n \end{bmatrix}, \quad (12)$$

where

$$\begin{bmatrix} c_{11,n} & c_{12,n} \\ c_{21,n} & c_{22,n} \end{bmatrix} = \begin{bmatrix} \frac{t_{12,n}}{t_{22,n}} & \frac{t_{11,n}t_{22,n} - t_{12,n}t_{21,n}}{t_{22,n}} \\ \frac{1}{t_{22,n}} & -\frac{t_{21,n}}{t_{22,n}} \end{bmatrix}. \quad (13)$$

The matrix partitions \mathbf{C}_{11} , \mathbf{C}_{12} , \mathbf{C}_{21} , \mathbf{C}_{22} of \mathbf{C} are typically diagonal, with “stamps” along the diagonals determined by the independent and dependent variables of each nonlinearity. For a root topology with N nonlinearities they take the form

$$\mathbf{C}_{11} = \text{diag}(c_{11,1}, c_{11,2}, \dots, c_{11,N}) \quad (14)$$

$$\mathbf{C}_{12} = \text{diag}(c_{12,1}, c_{12,2}, \dots, c_{12,N}) \quad (15)$$

$$\mathbf{C}_{21} = \text{diag}(c_{21,1}, c_{21,2}, \dots, c_{21,N}) \quad (16)$$

$$\mathbf{C}_{22} = \text{diag}(c_{22,1}, c_{22,2}, \dots, c_{22,N}), \quad (17)$$

with entries given in terms of x_n , y_n for each port n by (13).

Values for t_{11} , t_{12} , t_{21} , t_{22} entries that accommodate voltage, current, and incident and reflected waves are given in Tab. 1.

2.4. New Considerations for $f(\cdot)$

The \mathbf{E} , \mathbf{M} , and \mathbf{N} matrix multiplies in Fig. 1b can be computed with no special considerations. However, the \mathbf{F} matrix multiply and evaluation of the $f(\cdot)$ vector nonlinear function evaluation form a delay-free loop, or implicit relationship. These can be combined into a zero-finding function

$$\mathbf{x}_c = \mathbf{E}\mathbf{a}_e + \mathbf{F}f(\mathbf{x}_c) \rightarrow \mathbf{h}(\mathbf{x}_c) = \mathbf{E}\mathbf{a}_e + \mathbf{F}f(\mathbf{x}_c) - \mathbf{x}_c. \quad (18)$$

³In [15], they were modeled in the wave domain as $\mathbf{b} = f(\mathbf{a})$.

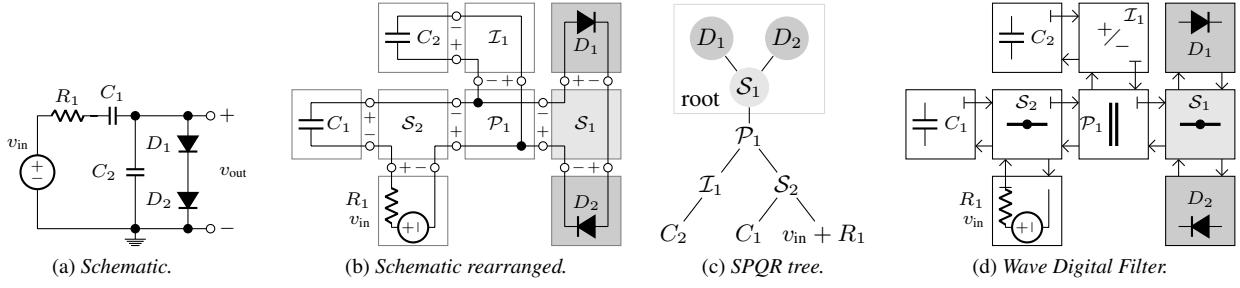


Figure 3: Series clipper schematic and derivation of Wave Digital Filter structure.

Table 2: Determinant of $(\mathbf{I} - \mathbf{C}_{22}\mathbf{S}_{11})$ for different variable choices for series clipper (Fig. 3). $\Gamma = R_1 + R_2 + R_3$.

	$v_2 \rightarrow i_2$	$v_2 \rightarrow b_2$	$i_2 \rightarrow v_2$	$i_2 \rightarrow b_2$	$a_2 \rightarrow v_2$	$a_2 \rightarrow i_2$	$a_2 \rightarrow b_2$
port 1 variables: $x_1 \rightarrow y_1$	0	$2R_1/\Gamma$	$4R_1/\Gamma$	$2R_1/\Gamma$	$4R_1/\Gamma$	0	$2R_1/\Gamma$
	$2R_2/\Gamma$	1	$2(R_1 + R_3)/\Gamma$	1	$2(R_1 + R_3)/\Gamma$	$2R_2/\Gamma$	1
	$4R_2/\Gamma$	$2(R_2 + R_3)/\Gamma$	$4R_3/\Gamma$	$2(R_2 + R_3)/\Gamma$	$4R_3/\Gamma$	$4R_2/\Gamma$	$2(R_2 + R_3)/\Gamma$
	$2R_2/\Gamma$	1	$2(R_1 + R_3)/\Gamma$	1	$2(R_1 + R_3)/\Gamma$	$2R_2/\Gamma$	1
	$4R_2/\Gamma$	$2(R_2 + R_3)/\Gamma$	$4R_3/\Gamma$	$2(R_2 + R_3)/\Gamma$	$4R_3/\Gamma$	$4R_2/\Gamma$	$2(R_2 + R_3)/\Gamma$
	0	$2R_1/\Gamma$	$4R_1/\Gamma$	$2R_1/\Gamma$	$4R_1/\Gamma$	0	$2R_1/\Gamma$
	$2R_2/\Gamma$	1	$2(R_1 + R_3)/\Gamma$	1	$2(R_1 + R_3)/\Gamma$	$2R_2/\Gamma$	1

$\mathbf{h}(\mathbf{x}_c) = \mathbf{0}$ is solved (i.e., finding a suitable \mathbf{x}_c) using Newton–Raphson iteration [13] by first providing an initial guess $\mathbf{x}_c^{(0)}$ and iterating according to

$$\mathbf{x}_c^{(k+1)} = \mathbf{x}_c^{(k)} - \mathbf{J}(\mathbf{x}_c^{(k)})^{-1} \mathbf{f}(\mathbf{x}_c^{(k)}) \quad (19)$$

where index k is the iteration count and $\mathbf{J}(\cdot)$ is the Jacobian operator. A typical initial guess at time step n is $\mathbf{x}_c^{(0)}(n) = \mathbf{x}_c(n-1)$; an alternative is given in [13].

Given a set of incident waves \mathbf{a}_e on the root, (18) can be solved using Newton–Raphson iteration [13] to yield \mathbf{x}_c . $\mathbf{y}_c = \mathbf{f}(\mathbf{x}_c)$ is evaluated to find \mathbf{y}_c , and \mathbf{y}_c and \mathbf{a}_e are finally used to find \mathbf{b}_e .

In the new generalized framework, elements of the vectors of independent root variables \mathbf{x}_c and dependent root variables \mathbf{y}_c may be voltages, currents, or any linear combination. Earlier we discussed the implications of that for forming \mathbf{C} . Of course, this also affects the definition of $\mathbf{f}(\cdot)$.

3. FIRST CLASS OF PROBLEMATIC CIRCUITS

The first class of circuits that require the generalization presented in this paper are those circuits whose constituent equations are written in a way that cannot be rewritten in the form $\mathbf{i}_c = \mathbf{f}(\mathbf{v}_c)$. These include, e.g., clipping operational amplifier models, clipping operational transconductance amplifier models, analog multipliers, logic gate models, and one-ports with non-functional $v-i$ curves, e.g. “s-type” nonlinear one-ports [41].

3.1. Example of First Class

As an example, we consider the family of clipping operational amplifiers (op amp) models. Op amps are two-port devices character-

ized by four network variables: v_{in} , i_{in} , v_{out} , and i_{out} . In some circuits, op amps may exhibit non-ideal voltage clipping behavior, which may be modeled with a $\tanh(\cdot)$ -based function [4]. In some circuits op amps are configured to act as voltage comparators, which may be modeled with a $\text{sgn}(\cdot)$ -based function. As nonlinear two-ports, op amps are modeled by

$$\begin{bmatrix} v_{out} \\ i_{in} \end{bmatrix} = \begin{bmatrix} f_1(i_{out}, v_{in}) \\ f_2(i_{out}, v_{in}) \end{bmatrix} = \mathbf{f} \left(\begin{bmatrix} i_{out} \\ v_{in} \end{bmatrix} \right) \quad (20)$$

Typically we have $f_2(i_{out}, v_{in}) = 0$, but $f_1(i_{out}, v_{in})$ can take a number of forms, e.g.

$$\text{clipper: } v_{out} = V_{\max} \tanh(A v_{in}) \quad (21)$$

$$\text{comparator: } v_{out} = V_{\max} \text{sgn}(v_{in}), \quad (22)$$

where A is the op amp’s gain in the clipper model (100000 is typical) and V_{\max} is the op amp saturation voltage in each model.

An example taken from [33] of an audio circuit using an op amp configured as a comparator is a relaxation oscillator, shown in Fig. 2a. This circuit is rearranged as shown in Fig. 2b, yielding the WDF structure in Fig. 2d. The two ports of the nonlinearity are labeled “in” and “out.” The input variables \mathbf{x}_c and output variables \mathbf{y}_c are defined as in (20). To accommodate this set of independent and dependent variables which are necessary for the op amp models just discussed we derive

$$\begin{aligned} t_{11,\text{out}} &= \frac{1}{2R_{\text{out}}}, & t_{12,\text{out}} &= \frac{-1}{2R_{\text{out}}}, & t_{21,\text{out}} &= \frac{1}{2}, & t_{22,\text{out}} &= \frac{1}{2} \\ t_{11,\text{in}} &= \frac{1}{2}, & t_{12,\text{in}} &= \frac{1}{2}, & t_{21,\text{in}} &= \frac{1}{2R_{\text{in}}}, & t_{22,\text{in}} &= \frac{-1}{2R_{\text{in}}} \end{aligned}$$

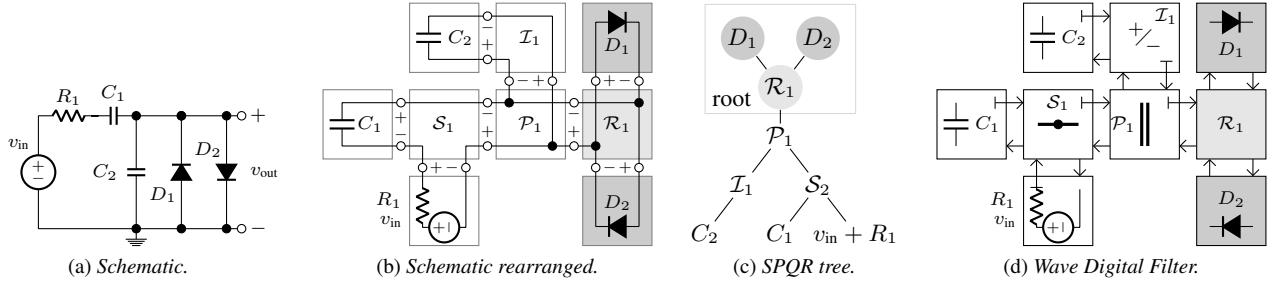


Figure 4: Parallel clipper schematic and Derivation of Wave Digital Filter structure.

Table 3: Determinant of $(\mathbf{I} - \mathbf{C}_{22}\mathbf{S}_{11})$ for different variable choices for parallel clipper (Fig. 4). $\Delta = G_1 + G_2 + G_3$ and $G_x = 1/R_x$.

	$v_2 \rightarrow i_2$	$v_2 \rightarrow b_2$	$i_2 \rightarrow v_2$	$i_2 \rightarrow b_2$	$a_2 \rightarrow v_2$	$a_2 \rightarrow i_2$	$a_2 \rightarrow b_2$
port 1 variables: $x_1 \uparrow y_1$	$v_1 \rightarrow i_1$	$4G_3/\Delta$	$2(G_2 + G_3)/\Delta$	$4G_2/\Delta$	$2(G_2 + G_3)/\Delta$	$4G_2/\Delta$	$4G_3/\Delta$
	$v_1 \rightarrow b_1$	$2(G_1 + G_3)/\Delta$	1	$2G_2/\Delta$	1	$2G_2/\Delta$	$2(G_1 + G_3)/\Delta$
	$i_1 \rightarrow v_1$	$4G_1/\Delta$	$2G_1/\Delta$	0	$2G_1/\Delta$	0	$4G_1/\Delta$
	$i_1 \rightarrow b_1$	$2(G_1 + G_3)/\Delta$	1	$2G_2/\Delta$	1	$2G_2/\Delta$	$2(G_1 + G_3)/\Delta$
	$a_1 \rightarrow v_1$	$4G_1/\Delta$	$2G_1/\Delta$	0	$2G_1/\Delta$	0	$4G_1/\Delta$
	$a_1 \rightarrow i_1$	$4G_3/\Delta$	$2(G_2 + G_3)/\Delta$	$4G_2/\Delta$	$2(G_2 + G_3)/\Delta$	$4G_2/\Delta$	$2(G_2 + G_3)/\Delta$
	$a_1 \rightarrow b_1$	$2(G_1 + G_3)/\Delta$	1	$2G_2/\Delta$	1	$2G_2/\Delta$	$2(G_1 + G_3)/\Delta$

using the stamps from Tab. 1 and procedure from §2.3. Using (13) this yields the appropriate \mathbf{C} matrix

$$\mathbf{C} = \begin{bmatrix} \mathbf{C}_{11} & \mathbf{C}_{12} \\ \mathbf{C}_{21} & \mathbf{C}_{22} \end{bmatrix} = \begin{bmatrix} -R_{in} & 0 & 1 & 0 \\ 0 & -1/R_{out} & 0 & 1/R_{out} \\ -2R_{in} & 0 & 1 & 0 \\ 0 & 2 & 0 & -1 \end{bmatrix}.$$

This choice of \mathbf{C} in the proposed general framework enables the simulation of the relaxation oscillator.⁴

4. SECOND CLASS OF PROBLEMATIC CIRCUITS

A second class of circuits that require the generalization presented in this paper are circuits that include cutsets composed entirely of nonlinearities (and current sources). In circuit theory, it is forbidden to have any cutset in a circuit graph composed entirely of current sources [41]. The presence of a cutset of current sources in a circuit creates the potential for a violation of Kirchhoff's current law: the sum of currents entering a node must equal the sum of currents leaving that node. A dual restriction is that no loop in a circuit graph may be composed entirely of voltage sources [41]. The presence of a loop of all voltage sources creates the potential for a violation of Kirchhoff's voltage law: the sum of voltages around any loop in the circuit must equal zero. This is a circuit theoretic argument but it is also expressed in the mathematics of the root topology. In the case of violating either the loop or the cutset criteria, it appears that in the WDF context the consequence is that the matrix $(\mathbf{I} - \mathbf{C}_{22}\mathbf{S}_{11})$ which needs to be inverted is made singular. Examples will be given in the next section.

⁴Anti-aliasing and multi-rate methods can be used to improve the simulation of the relaxation oscillator; these are described in Olsen *et al.* [33].

To fix this class of circuit, start with the standard choice of independent and dependent network variables for each nonlinearity. Identify each loop and cutset in the circuit by hand or using the graph theorems discussed in §5. If any problematic cutsets or loops exist, choose new dependent variables for some of the nonlinearities that make up that cutset or loop to avoid the issue.

4.1. Examples of Second Class

To discuss these issues, we consider three circuits: the “series diode clipper” shown in Fig. 3a, the “parallel diode clipper” shown in Fig. 4a, and the “series–parallel diode clipper” shown in Fig. 5.

Consider the series diode clipper and its WDF structure derivation shown in Fig. 3. Notice that inside the root topology \mathcal{S}_1 in Fig. 3b, we have a node with only diodes connected to it. If the diodes were written in the form $i = f(v)$, a current-source-only cutset is created. A remedy for this circuit would be to write either or both of the diodes in the form $v = f(i)$, or more broadly in any form that does not have current as the dependent variable. In Tab. 2, the determinant of matrix $(\mathbf{I} - \mathbf{C}_{22}\mathbf{S}_{11})$ is shown for the 49 different possible choices of x_n and y_n at port 1 (diode D_1) and port 2 (diode D_2). Notice that for the combinations that are forbidden according to circuit theory (both diodes with current as the dependent variable), the matrix which needs to be inverted becomes singular (its determinant is 0)—hence the circuit cannot be simulated using those variables.

It is not always a valid solution to choose voltage as the dependent variable. Consider the parallel diode clipper and its WDF structure derivation shown in Fig. 4. Notice that inside the root topology \mathcal{R}_1 in Fig. 4b there is a loop composed only of diodes. If the diodes were both written in the form $v = f(i)$, a voltage-source-only loop is created, $(\mathbf{I} - \mathbf{C}_{22}\mathbf{S}_{11})$ becomes singular, and

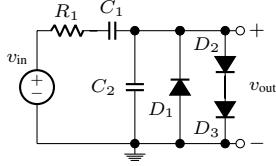


Figure 5: Series-parallel clipper schematic.

the circuit cannot be simulated. For this circuit, it suffices that one or both diodes is written in the form $i = f(v)$, or more broadly in any form that does not have voltage as the dependent variable. In Tab. 3, the determinant of matrix $(\mathbf{I} - \mathbf{C}_{22}\mathbf{S}_{11})$ is shown for the 49 different possible choices of \mathbf{x}_n and \mathbf{y}_n at port 1 (diode D_1) and port 2 (diode D_2). Notice that for the combinations that are forbidden according to circuit theory (both diodes with voltage as the dependent variable), the matrix which needs to be inverted becomes singular (its determinant is 0)—hence the circuit cannot be simulated using those variables.

Sometimes it is necessary to choose different variables for each nonlinearity. For example, consider the series-parallel diode clipper shown in Fig. 5. Writing all three diodes in the form $i = f(v)$ creates a current-source-only cutset but writing all three diodes in the form $v = f(i)$ creates a voltage-source-only loop: both forbidden cases causing a singular matrix. Here the solution is that at least one diode is written with current as the dependent variable to avoid the loop, and that at least one of D_2, D_3 is written with voltage as the dependent variable to avoid the cutset.

5. ENUMERATING LOOPS AND CUTSETS ON GRAPHS

In the previous section we discussed circuits whose cutsets and loops were easily identifiable at a glance. Unfortunately in general circuits may be very complex and involve too many loops and cutsets to enumerate at a glance. To handle problematic circuits in general it is necessary to examine all loops and cutsets in the circuit graph and choose the dependent variables of the nonlinearities to avoid current-source-only cutsets and voltage-source-only loops. Here we review dual graph theorems for enumerating loops and cutsets in a graph (an alternative to identifying them by visual inspection as in the previous section) and demonstrate their application to two circuits discussed in §4.

5.1. Find All Loops in a Circuit

Here we review a theorem for enumerating the set of all loops in an electrical circuit and give example applications of the theorem for the parallel diode clipper.

The theorem is stated as follows [42, p. 50]:

- Let the circuit be represented by a nonoriented, connected graph \mathcal{G} with v vertices and e edges, where each vertex represents a node in the circuit and each edge represents a one-port electrical element in the circuit.
- Choose a tree \mathcal{T} on \mathcal{G} . Form a set of fundamental loops with respect to \mathcal{T} by reinstating each edge of the co-tree \mathcal{T}' one at a time to create fundamental loops each involving one edge and a tree path formed by some or all of the branches of \mathcal{T} . The fundamental loops are represented mathematically by a matrix \mathbf{B}_f where each of the $e - v + 1$ fundamental loops is a row, and the e edges of \mathcal{G} are the columns.

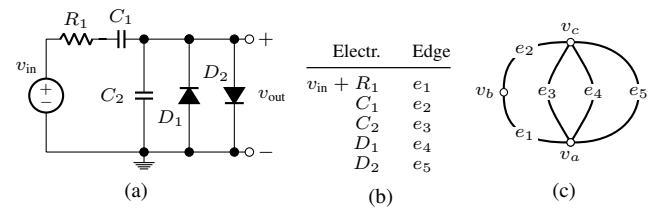


Figure 6: Parallel clipper (a) Schematic, (b) Mapping from circuit elements to graph edges, and (c) Graph.

- Form an intermediate matrix \mathbf{B}_1 by adding to \mathbf{B}_f all the possible ring sums among the rows of \mathbf{B}_f . The ring sum of two sets S_1 and S_2 is the set $S_1 \oplus S_2$ that has edges in S_1 or S_2 but not both [42, p. 14].
- Eliminate redundant rows in \mathbf{B}_1 , i.e. rows that appear more than once, to form \mathbf{B}_a , the *loop matrix* encoding all valid loops on \mathcal{G} . Redundant rows represent edge-disjoint unions of loops [42, p. 44].

Now, an example on the parallel diode clipper will illustrate the application of this theorem.

5.1.1. Parallel Diode Clipper Loops

Consider the parallel diode clipper whose schematic is shown in Fig. 6a. Using the mapping between electrical component and graph edges shown in Fig. 6b, a graph \mathcal{G} of the parallel clipper is formed in Fig. 6c. \mathcal{G} has 3 nodes $\{v_a, v_b, v_c\}$ and 5 edges $\{e_1, e_2, e_3, e_4, e_5\}$, i.e., $v = 3$ and $e = 5$.

To enumerate all loops on \mathcal{G} we choose a tree $\mathcal{T} = \{e_2, e_3\}$ and corresponding co-tree $\mathcal{T}' : \{e_1, e_4, e_5\}$. Combining each edge $c \in \mathcal{T}'$ with other edges chosen from \mathcal{T} yields $e - v + 1 = 3$ fundamental loops $\{c_{e_1}, c_{e_4}, c_{e_5}\}$ encoded in a matrix \mathbf{B}_f

$$\mathbf{B}_f = \begin{matrix} & e_1 & e_2 & e_3 & e_4 & e_5 \\ c_{e_1} & 1 & 1 & 1 & 0 & 0 \\ c_{e_4} & 0 & 0 & 1 & 1 & 0 \\ c_{e_5} & 0 & 0 & 1 & 0 & 1 \end{matrix} \begin{matrix} c_1 \\ c_2 \\ c_3 \end{matrix}$$

We take all possible ring sums among the rows of \mathbf{B}_f to find the intermediate matrix

$$\mathbf{B}_1 = \begin{matrix} & e_1 & e_2 & e_3 & e_4 & e_5 \\ c_{e_1} & 1 & 1 & 1 & 0 & 0 \\ c_{e_4} & 0 & 0 & 1 & 1 & 0 \\ c_{e_5} & 0 & 0 & 1 & 0 & 1 \\ c_{e_1} \oplus c_{e_4} & 1 & 1 & 0 & 1 & 0 \\ c_{e_1} \oplus c_{e_5} & 1 & 1 & 0 & 0 & 1 \\ c_{e_4} \oplus c_{e_5} & 0 & 0 & 0 & 1 & 1 \\ c_{e_1} \oplus c_{e_4} \oplus c_{e_5} & 1 & 1 & 1 & 1 & 1 \end{matrix} \begin{matrix} c_1 \\ c_2 \\ c_3 \\ c_4 \\ c_5 \\ c_6 \\ \text{redundant} \end{matrix}$$

The last ring sum represents an invalid loop, so it is discarded in forming \mathbf{B}_a . The final loop matrix \mathbf{B}_a and graphical representation of the six loops $\{c_1, c_2, c_3, c_4, c_5, c_6\}$ is given by

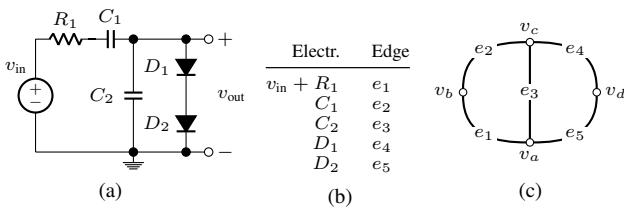
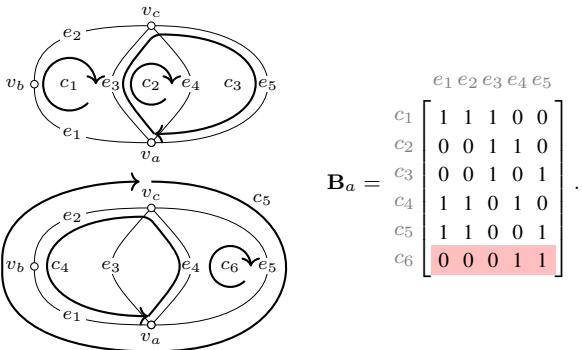


Figure 7: Series clipper (a) Schematic, (b) Mapping from circuit elements to graph edges, and (c) Graph.



The last loop $c_6 = \{e_4, e_5\}$ would be composed entirely of voltage sources if diodes D_1 and D_2 (edges e_4 and e_5) are both modeled in the form $v = f(i)$, as we saw earlier in §4.

5.2. Find All Cutsets in a Circuit

Here we review a theorem for enumerating the set of all cutsets in an electrical circuit and give an example application of the theorem to a circuit discussed in §4: the series diode clipper.

The theorem is stated as follows [42, p. 58]:

- Let the circuit be represented by a nonoriented, connected graph \mathcal{G} with v vertices and e edges.
- Choose a tree \mathcal{T} on \mathcal{G} . Form a set of fundamental cutsets with respect to \mathcal{T} by removing each edge of the cotree \mathcal{T}' one at a time to create fundamental cutsets each involving one tree edge and some or all of the edges of \mathcal{T}' . The fundamental cutsets are represented mathematically by a matrix \mathbf{Q}_f where each of the $v - 1$ fundamental cutsets is a row, and the e edges of \mathcal{G} are the columns.
- Form an intermediate matrix \mathbf{Q}_1 by adding to \mathbf{Q}_f all of the possible ring sums among the rows of \mathbf{Q}_f .
- Eliminate redundant rows in \mathbf{Q}_1 to form \mathbf{Q}_a , the *cutset matrix* encoding all valid cutsets on \mathcal{G} .

Now, an examples on the series diode clipper will illustrate the application of this theorem.

5.2.1. Series Diode Clipper Cutsets

Consider the series diode clipper whose schematic is shown in Fig. 7a. Using the mapping between electrical component and graph edges shown in Fig. 7b, a graph \mathcal{G} of the parallel clipper is formed in Fig. 7c. \mathcal{G} has 4 nodes $\{v_a, v_b, v_c, v_d\}$ and 5 edges $\{e_1, e_2, e_3, e_4, e_5\}$, i.e., $v = 4$ and $e = 5$.

To enumerate all cutsets on \mathcal{G} we choose a tree $\mathcal{T} = \{e_2, e_3, e_4\}$ and corresponding cotree $\mathcal{T}' = \{e_1, e_5\}$. Removing each edge $k \in \mathcal{T}$ with other edges chosen from \mathcal{T}' yields $v - 1 = 3$ fundamental cutsets $\{k_{e_2}, k_{e_3}, k_{e_4}\}$ encoded in a matrix \mathbf{Q}_f

$$\mathbf{Q}_f = \begin{matrix} & e_1 & e_2 & e_3 & e_4 & e_5 \\ \begin{matrix} k_{e_2} \\ k_{e_3} \\ k_{e_4} \end{matrix} & \left[\begin{array}{ccccc} 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 \end{array} \right] & \begin{matrix} k_1 \\ k_2 \\ k_3 \end{matrix} \end{matrix}$$

We take all possible ring sums among the rows of \mathbf{Q}_f to find the intermediate matrix \mathbf{Q}_1

$$\mathbf{Q}_1 = \begin{matrix} & e_1 & e_2 & e_3 & e_4 & e_5 \\ \begin{matrix} k_{e_2} \\ k_{e_3} \\ k_{e_4} \\ k_{e_2} \oplus k_{e_3} \\ k_{e_2} \oplus k_{e_4} \\ k_{e_3} \oplus k_{e_4} \\ k_{e_2} \oplus k_{e_3} \oplus k_{e_4} \end{matrix} & \left[\begin{array}{ccccc} 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 1 & 0 & 1 \\ 1 & 1 & 0 & 1 & 1 \\ 1 & 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 \end{array} \right] & \begin{matrix} k_1 \\ k_2 \\ k_3 \\ k_4 \\ k_5 \\ k_6 \\ k_7 \end{matrix} \end{matrix}$$

There are no redundancies in \mathbf{Q}_1 , so the final cutset matrix \mathbf{Q}_a and seven cutsets $\{k_1, k_2, k_3, k_4, k_5, k_6, k_7\}$ are given by

$$\mathbf{Q}_a = \begin{matrix} & e_1 & e_2 & e_3 & e_4 & e_5 \\ \begin{matrix} k_1 \\ k_2 \\ k_3 \\ k_4 \\ k_5 \\ k_6 \\ k_7 \end{matrix} & \left[\begin{array}{ccccc} 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 1 & 0 & 1 \\ 1 & 1 & 0 & 1 & 1 \\ 1 & 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 \end{array} \right] & \end{matrix}$$

The third cutset $k_3 = \{e_4, e_5\}$ would be composed entirely of current sources if diodes D_1, D_2 (edges e_4, e_5) are both modeled in the form $i = f(v)$, i.e., the standard Shockley ideal diode law.

6. CONCLUSION

This paper extended the class of circuits that can be modeled using Wave Digital Filters with grouped nonlinearities at the root of a tree [11–15] by generalizing independent/dependent variable choice for nonlinearities. This accommodates nonlinear devices (e.g. clipping op amps) which require flexibility in variable choice as well as circuits with loops composed entirely of nonlinearities (and ideal voltage sources) or cutsets composed entirely of nonlinearities (and ideal current sources).

When possible, choosing reflected waves as the dependent variable for nonlinearities will avoid problematic loops and cutsets. A circuit theory interpretation of this is that voltage waves “look like” *resistive* voltage sources (Thévenin source) or equivalently *resistive* current sources (Norton sources) [11]. A Thévenin source will never contribute to a voltage-source-only loop since any loop involving the source will also involve its resistance, and a Norton source will never contribute to a current-source-only cutset since any cutset involving it will also involve its resistance.

Future work should explore the potential for root variable choice to have computational benefits. If there are any computational or

other benefits of flexibility in dependent variable choice for nonlinearities, the insights of this paper could be used in the other WDF formulations [18–23] which currently use wave variables only but also currently should not produce problematic cutsets or loops.

7. REFERENCES

- [1] D. T. Yeh, J. S. Abel, and J. O. Smith III, “Automated physical modeling of nonlinear audio circuits for real-time audio effects—Part I: Theoretical development,” *IEEE Trans. Audio, Speech, Language Process.*, vol. 18, no. 4, pp. 728–737, 2010.
- [2] D. T. Yeh, “Automated physical modeling of nonlinear audio circuits for real-time audio effects—Part II: BJT and vacuum tube examples,” *IEEE Trans. Audio, Speech, Language Process.*, vol. 20, no. 4, pp. 1207–1216, 2012.
- [3] M. Holters and U. Zölzer, “Physical modelling of a wah-wah effect pedal as a case study for application of the nodal DK method to circuits with variable parts,” in *Proc. 14th Int. Conf. Digital Audio Effects*, Paris, France, 2011.
- [4] M. Holters and U. Zölzer, “A generalized method for the derivation of non-linear state-space models from circuit schematics,” in *Proc. 23rd European Signal Process. Conf.*, Nice, France, 2015.
- [5] A. Falaize-Skrzak and T. Hélie, “Simulation of an analog circuit of a wah pedal: a port-Hamiltonian approach,” in *Proc. 135th Conv. Audio Eng. Soc.*, New York, NY, 2013.
- [6] A. Falaize and T. Hélie, “Passive guaranteed simulation of analog audio circuits: A port-Hamiltonian approach,” *Appl. Sci.*, vol. 6, no. 10, 2016, Article #273.
- [7] A. Falaize, *Modélisation, simulation, génération de code et correction de systèmes multi-physiques audio: Approche par réseau de composants et formulation Hamiltonienne à ports*, Ph.D. thesis, Université Pierre et Marie Curie, Paris, France, July 2016.
- [8] A. Falaize and T. Hélie, “Passive simulation of the nonlinear port-Hamiltonian modeling of a Rhodes piano,” *J. Sound Vibration*, vol. 390, pp. 289–309, 2017.
- [9] A. Fettweis, “Wave digital filters: Theory and practice,” *Proc. IEEE*, vol. 74, no. 2, pp. 270–327, 1986.
- [10] 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.
- [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. Digital Audio Effects*, Trondheim, Norway, 2015.
- [12] K. J. Werner, V. Nangia, J. O. Smith III, and J. O. Abel, “Resolving wave digital filters with multiple/multiport nonlinearities,” in *Proc. 18th Int. Conf. Digital Audio Effects*, Trondheim, Norway, 2015.
- [13] 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. Digital Audio Effects*, Brno, Czech Republic, 2016.
- [14] K. J. Werner, W. R. Dunkel, M. Rest, M. J. Olsen, and J. O. Smith III, “Wave digital filter modeling of circuits with operational amplifiers,” in *Proc. 24th European Signal Process. Conf.*, Budapest, Hungary, 2016.
- [15] 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. Digital Audio Effects*, Brno, Czech Republic, 2016.
- [16] A. Fettweis, “Pseudo-passivity, sensitivity, and stability of wave digital filters,” *IEEE Trans. Circuit Theory*, vol. 19, no. 6, pp. 668–673, Nov. 1972.
- [17] K. Meerkötter and R. Scholz, “Digital simulation of nonlinear circuits by wave digital filter principles,” in *IEEE Int. Symp. Circuits Syst.*, Portland, OR, 1989.
- [18] T. Schwerdtfeger and A. Kummert, “A multidimensional signal processing approach to wave digital filters with topology-related delay-free loops,” in *IEEE Int. Conf. Acoust., Speech Signal Process.*, Florence, Italy, 2014.
- [19] T. Schwerdtfeger and A. Kummert, “A multidimensional approach to wave digital filters with multiple nonlinearities,” in *Proc. 22nd European Signal Process. Conf.*, Lisbon, Portugal, 2014.
- [20] T. Schwerdtfeger A. Kummert, “Newton’s method for modularity-preserving multidimensional wave digital filters,” in *Proc. IEEE Int. Work. Multidimensional Syst.*, Vila Real, Portugal, 2015.
- [21] T. Schwerdtfeger, *Modulare Wellendigitalstrukturen zur Simulation hochgradig nichtlinearer Netzwerk*, Ph.D. thesis, Bergischen Universität Wuppertal, Wuppertal, Germany, Dec. 2016.
- [22] A. Bernardini and A. Sarti, “Dynamic adaptation of instantaneous nonlinear bipoles in wave digital networks,” in *Proc. 24th European Signal Process. Conf.*, Budapest, Hungary, 2016.
- [23] A. Bernardini and A. Sarti, “Biparametric wave digital filters,” *IEEE Trans. Circuits Syst. I: Reg. Papers*, 2017, to be published, DOI: 10.1109/TCSI.2017.2679007.
- [24] 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 Work. Appl. Signal Process. Audio Acoust.*, New Paltz, NY, 2015.
- [25] 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.
- [26] S. Petrausch and R. Rabenstein, “Wave digital filters with multiple nonlinearities,” in *Proc. 12th European Signal Process. Conf.*, Vienna, Austria, 2004.
- [27] Ó. Bogason and K. J. Werner, “Modeling circuits with operational transconductance amplifiers using wave digital filters,” in *Proc. 20th Int. Conf. Digital Audio Effects*, Edinburgh, UK, 2017.
- [28] Ó. Bogason, “Digitizing analog circuits containing op amps using wave digital filters,” Blog Post, Mar. 20 2016, Online: <http://obogason.com/emulating-op-amp-circuits-using-wdf-theory/>.
- [29] M. Verasani, A. Bernardini, and A. Sarti, “Modeling Sallen-Key audio filters in the wave digital domain,” in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, New Orleans, LA, 2017.
- [30] M. Rest, W. R. Dunkel, K. J. Werner, and J. O. Smith, “RT-WDF—a modular wave digital filter library with support for arbitrary topologies and multiple nonlinearities,” in *Proc. 19th Int. Conf. Digital Audio Effects*, Brno, Czech Republic, 2016.
- [31] 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. Digital Audio Effects*, Brno, Czech Republic, 2016.
- [32] M. Rest, J. D. Parker, and K. J. Werner, “WDF modeling of a KORG MS-50 based non-linear diode bridge VCF,” in *Proc. 20th Int. Conf. Digital Audio Effects*, Edinburgh, UK, 2017.
- [33] 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*, Edinburgh, UK, 2017.
- [34] K. J. Werner, *Virtual Analog Modeling of Audio Circuitry Using Wave Digital Filters*, Ph.D. diss., Stanford Univ., Stanford, CA, USA, Dec. 2016.
- [35] 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, NY, 2015.
- [36] S. Petrausch and R. Rabenstein, “Interconnection of state space structures and wave digital filters,” *IEEE Trans. Circuits Syst. II: Expr. Briefs*, vol. 52, no. 2, pp. 90–93, February 2005.
- [37] 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.
- [38] 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, no. 5, pp. 597–605, Sept. 2000.
- [39] C.-W. Ho, A. E. Ruehli, and P. A. Prennan, “The modified nodal approach to network analysis,” *IEEE Trans. Circuits Syst.*, vol. 22, no. 6, pp. 504–509, June 1975.
- [40] G. O. Martens and K. Meerkötter, “On N-port adaptors for wave digital filters with application to a bridged-tee filter,” in *Proc. IEEE Int. Symp. Circuits Syst.*, Munich, Germany, 1976.
- [41] L. O. Chua, *Computer-Aided Analysis of Electronic Circuits*, Prentice-Hall, 1975.
- [42] S.-P. Chan, *Introductory Topological Analysis of Electrical Networks*, Holt, Rinehart and Winston, Inc., New York, NY, 1969.

BLOCK-ORIENTED GRAY BOX MODELING OF GUITAR AMPLIFIERS

Felix Eichas, Stephan Möller, Udo Zölzer

Department of Signal Processing and Communications,
Helmut Schmidt University
Hamburg, Germany
felix.eichas@hsu-hh.de

ABSTRACT

In this work, analog guitar amplifiers are modeled with an automated procedure using iterative optimization techniques. The digital model is divided into functional blocks, consisting of linear-time-invariant (LTI) filters and nonlinear blocks with nonlinear mapping functions and memory. The model is adapted in several steps. First the filters are measured and afterwards the parameters of the digital model are adapted for different input signals to minimize the error between itself and the analog reference system. This is done for a small number of analog reference devices. Afterwards the adapted model is evaluated with objective scores and a listening test is performed to rate the quality of the adapted models.

1. INTRODUCTION

Musical distortion circuits, especially guitar amplifiers, have been the subject of virtual analog modeling for years. There exist two main modeling approaches, like white-box modeling and black- or gray-box modeling. White-box modeling makes use of everything known about the reference system, for example its circuit and the characteristics of the circuit elements. In [1–3] the circuit of each reference is modeled by creating a (nonlinear) state-space model. To be able to do that, detailed knowledge about the circuit diagram as well as the nonlinear characteristics of the circuit elements is required. In [4] an alternative white-box modeling approach is described, where wave digital filters are used to model the circuit of a reference device. This approach has already been used in [5] to model a guitar pre-amplifier with four vacuum triodes and a complex circuit topology. Both white-box approaches give very good results which can reproduce all relevant characteristics of a reference device. If the sound of a specific analog device should be replicated with high accuracy, the white-box modeling approaches are preferable. The drawback of both approaches is the computational load of the model. Without simplifications and pre-calculations, the model of a circuit with complex topology and a lot of nonlinear elements will barely be real-time capable.

In [6] a distortion circuit for electric guitars is modeled with a gray-box modeling approach. The reference system is measured with an exponential sine sweep, which allows to construct a multi-branch Hammerstein model where each branch represents a harmonic oscillation of the fundamental frequency of the input signal. Each branch is then filtered with the corresponding filter obtained with the exponential sine sweep analysis. This method gives good results, but the polynomials used as nonlinear mapping curves in each branch of the model are amplitude dependent, which means that the model gives perfect results if the input signal has the same amplitude as the identification signal, but might not perform as well for other amplitudes.

This work describes a gray-box modeling approach, which is similar to [6], but with a different model structure and iterative optimization to adjust the parameters of the digital model. The only assumptions made about the reference system are its basic structure. The modeling procedure is completely automatic and uses solely input-output measurements and iterative optimization to adapt the digital model to the reference device. No knowledge about the circuit is required.

System identification or modeling approaches are already used in commercial products. In [7] the modeling procedure is not automated and it becomes obvious that it is a quite tedious process. The patent [8] details a gray box modeling approach for guitar amplifiers. A Wiener-Hammerstein model is used, consisting of an input filter in series with a memoryless nonlinearity and an output filter. In the patent the modeling process is detailed only vaguely, for example, the mathematical basis for the nonlinear mapping function is not explained. Nevertheless, the results of this method speak for themselves, since a lot of musicians already use the commercial product, because of its flexibility in sound design.

One major drawback of the gray-box modeling techniques is, that the user controls (e.g. knobs on the amplifier) can not be modeled without creating one model for every possible combination of user controls and then interpolate between the model's parameters, according to the current user control setting.

This work describes the structure of the proposed digital model in Section 2. Section 3 details the measurement setup which is used to measure all guitar amplifiers. Sections 4 and 5 explain the steps used to adapt the model with the iterative optimization routine and show objective and subjective results. In Section 6 conclusions are drawn.

2. DIGITAL MODEL

The overall structure of the digital model is straightforward and has been used in virtual analog modeling before. The model consists of linear-time-invariant (LTI) blocks and nonlinear blocks which introduce harmonic distortion. In [7] this structure has been described as ‘the fundamental principle of guitar tone’, since every guitar-specific audio system, regardless if it is an analog or a digital system, operates in this manner. As Fig. 1 depicts, the input signal is filtered by the first filter, afterwards it is distorted by the first nonlinear block, which corresponds to the nonlinear behavior of the pre-amplifier. The output of the pre-amplifier is then filtered by the next filter in the cascade, which corresponds to the tone-section of the guitar amplifier. Finally, the signal passes through the second nonlinear block, corresponding to the power stage of the guitar amplifier and is then filtered by the output filter. The first filter in an analog amplifier are mostly first order RC-highpass or RC-bandpass filters and the output filter is determined

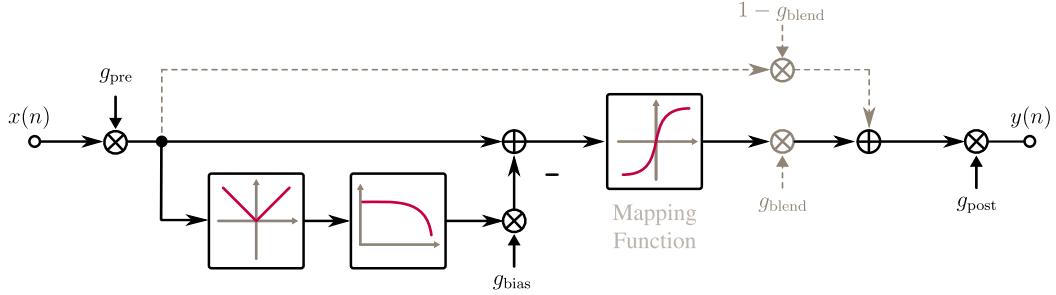


Figure 2: Signal flow graph of a nonlinear block. Blend stage is omitted in the second nonlinear block.

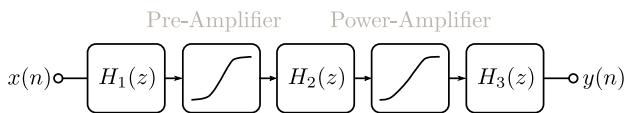


Figure 1: Block diagram of a guitar amplifier and structure of the digital model.

by the frequency behavior of the output transformer.

For analog guitar amplifiers, the distinction between the blocks is not always this clear. For example many amplifiers are designed in such a way, that turning the drive- or gain-knob down, the frequency response of the input filter is also changed.

Please note that the last filter $H_3(z)$ does not correspond to the impulse response of a loudspeaker. The amplifiers were measured without the influence of any speaker.

2.1. Pre-Amplifier Nonlinearity

All nonlinear blocks are structured as depicted in Fig. 2. This nonlinear block originated from [9] and has already been used in distortion effect modeling. The most important part of each nonlinear block is the mapping function because it defines the spectral shape of the harmonics.

The first nonlinear block consists of a polynomial mapping function which is complemented with pre- and post-gains, as well as a blend parameter, allowing dry/wet mixing of the output signal. The advantage of polynomial wave-shaping functions is the mathematical relationship between the coefficients of the polynomial and the shape of the harmonic overtones in the spectrum. Consider a polynomial function,

$$p(x) = a_0 + a_1x + a_2x^2 + \dots + a_Nx^N, \quad (1)$$

where x is the input variable (corresponding to the amplitude of the input signal) and a_n with $n \in [0, N]$ are the coefficients of the polynomial. Substituting x with $\tilde{x} = u \cdot \cos(\omega t)$, it is possible to separate the different harmonic oscillations of the fundamental frequency,

$$p(\omega t) = k_0 + k_1 \cos(\omega t) + k_2 \cos(2\omega t) + \dots + k_N \cos(N\omega t), \quad (2)$$

where the variables a_n and u have been combined into the harmonic variables k_n . Each k_n describes the amplitude of the n -th

harmonic to the fundamental frequency f_0 or ω_0 for a fixed input amplitude u . Fig. 3 depicts the harmonic variables k_n in frequency domain.

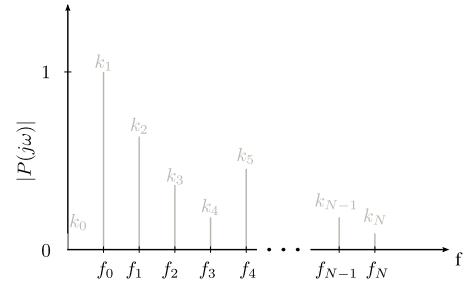


Figure 3: Overtones of a sinusoidal signal after the polynomial mapping function.

The relationship between harmonic variables k_n and polynomial coefficients a_n can be written in matrix form,

$$\begin{pmatrix} A_{11} & A_{12} & \dots & A_{1N} \\ A_{21} & A_{22} & \dots & A_{2N} \\ A_{31} & A_{32} & \dots & A_{3N} \\ \vdots & \vdots & \ddots & \vdots \\ A_{N1} & A_{N2} & \dots & A_{NN} \end{pmatrix} \begin{pmatrix} a_0 \\ a_1 \\ a_2 \\ \vdots \\ a_N \end{pmatrix} = \begin{pmatrix} k_0 \\ k_1 \\ k_2 \\ \vdots \\ k_N \end{pmatrix}$$

and solved for every a_n . With this technique it is possible to calculate the polynomial mapping function which creates the desired shape of overtones.

As an example the matrix equation is shown for 4 harmonics.

$$\begin{pmatrix} 1 & 0 & u^2/2 & 0 & 3u^4/8 \\ 0 & u & 0 & 3u^3/4 & 0 \\ 0 & 0 & 0 & u^3/4 & 0 \\ 0 & 0 & 0 & 0 & u^4/8 \end{pmatrix} \begin{pmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \\ a_4 \end{pmatrix} = \begin{pmatrix} k_0 \\ k_1 \\ k_2 \\ k_3 \\ k_4 \end{pmatrix}$$

Another extension, which has been made to the nonlinear blocks of the model, has been proposed in [10]. The envelope of the input signal is calculated and added to the signal, directly before the nonlinear mapping function. This behavior simulates the signal-dependent bias-point shift that is happening in tube amplifiers due to a varying cathode voltage which alters the plate current and thus moving the bias point of the tube.

2.2. Power-Amplifier Nonlinearity

The nonlinear block of the digital model, corresponding to the power stage of the guitar amplifier is slightly different than the first nonlinear block. Instead of using a polynomial mapping function, a concatenation of three hyperbolic tangents is used, which allows to shape positive and negative half-waves separately. It was already used in [9,11] to model distortion audio circuits. The blend stage was omitted in this nonlinear block.

3. MEASUREMENT SETUP

In this work, all measurements are done with a digital audio interface. First, the interface is calibrated with a digital oscilloscope. The output gain was altered until a sine wave with a digital amplitude of ± 1 corresponded to a voltage of ± 1 V at the output of the interface.

As Fig. 4 illustrates, Output 1 of the interface is connected to the input of the guitar amplifier under test and the output of the amplifier is connected to a power attenuator, which matches the impedance of the amplifier output and provides a line-out, which is connected to input 1 of the audio interface. The direct con-

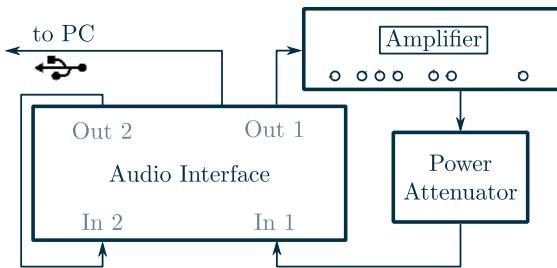


Figure 4: Measurement Setup.

nnection is used to design a compensation filter, described in [12], which reduces the influence of the audio interface when measuring frequency responses with sine sweeps. The recorded direct signal is also used as the input signal for the digital model. This is advantageous because the measured direct signal is automatically synchronized in time with the amplifier output, a crucial requirement for the following iterative optimization.

The used power attenuator was, unfortunately, purely resistive. A reactive power attenuator would be preferable because the amplifier might constitute a resonant behavior for certain frequencies, which does not occur with a purely resistive load.

4. SYSTEM IDENTIFICATION

This section describes the steps needed to adapt the digital model to a reference system. The process is subdivided into several steps to assure that the iterative optimization does not converge into a local minimum. At first the linear part of the reference system is measured and afterwards the parameters of the nonlinear blocks are optimized.

4.1. Filters

The method used in this work to measure the linear part of the reference system is the same as described in [6]. An exponential sine sweep is sent through the reference device and the recorded output is convolved with an inverse filter. The resulting impulse response contains the linear impulse response as well as different impulse responses for higher order harmonics. In this work only the impulse response corresponding to the linear part of the circuit is used.

To adapt the filters of the model several steps are used. First the small signal impulse response is measured with an exponential sine sweep from 10 Hz to 21 kHz and an amplitude of 0.01 V. This yields the filter $h_{\text{low}}(n)$. Afterwards the same measurement is repeated, but with an amplitude of 1 V, resulting in $h_{\text{high}}(n)$.

The low amplitude sweep is not exposed to the nonlinear behavior of the reference system and contains the influence of all its' filters. The high amplitude sine sweep gets distorted and the influence of some of the reference systems' filters is removed by the nonlinear parts of the reference system. This behavior is depicted in Fig. 5. The preceding filter $H(z)$ alters the amplitude of a sine wave which then passes through a nonlinear 'block' and is amplified back to the maximum amplitude, thus negating the influence of the preceding filter. The high amplitude sweep gets distorted and contains the influence of the last filter of the reference system. The obtained impulse responses are transformed into frequency-

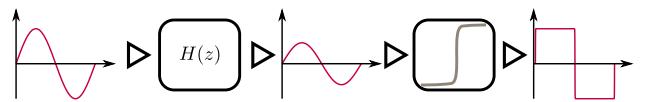


Figure 5: Influence of filters and nonlinear blocks.

domain with a discrete Fourier transform using 16384 samples to have a high frequency resolution. Afterwards, the small signal frequency response is divided by the large signal frequency response,

$$H_1(k) = \frac{H_{\text{low}}(k)}{H_{\text{high}}(k)}. \quad (3)$$

The resulting filter is transformed back into time-domain and used as the input filter of the digital model. The output is filtered with the measured impulse response of the high amplitude sweep $h_3(n) = h_{\text{high}}(n)$.

The results of this method are not perfect. The filters may be partially misidentified, because the gain of the last nonlinearity in the signal chain might not be high enough to negate the influence of the preceding filters. For this reason a 256 tap finite impulse response (FIR) filter $H_2(z)$ is adapted to match the small signal frequency response of the reference device. The FIR filter is located between the nonlinear blocks of the model.

The parameters for the linear part of the model are the FIR filters coefficients. The input signal for the adaptation is the above mentioned low amplitude sine sweep. The cost function calculates the difference of the magnitude spectrum of reference system and digital model output and the filter coefficients are adjusted to minimize the error between both spectra.

The length of the filter is a trade-off between computational complexity during optimization and frequency resolution and was

chosen empirically. The optimization algorithm approximates the derivative of the model output with respect to each parameter (in this case the 256 coefficients) using finite differences, leading to time-consuming calculations.

4.2. Nonlinear Blocks

After the small signal frequency response has been adapted, the filter coefficients of the FIR filter are not changed anymore, only the parameters of the nonlinear blocks can be altered.

Each nonlinear block features a multiplication with a variable gain (pre- and post-gain) of input signal and output signal. The intensity of the bias-point shift mentioned in Section 2.1 is adaptable for each nonlinear block, as well as the ‘blend’ stage, where dry and wet signal can be mixed by an adaptable parameter.

The first nonlinear block of the digital model also uses the parameters mentioned in Section 2.1. To limit the number of parameters, only the first 40 overtones $k_0 - k_{40}$ can be adapted. The optimization routine only alters the k_n parameters from which the polynomial coefficients a_n are computed. If the polynomial coefficients are used as parameters, too many unsuitable (or unstable) solutions would be possible and the optimization routine would not converge. The typical fundamental frequency region of an electric guitar in standard tuning ranges from 80 Hz to 1100 Hz, depending on the number of frets. 40 harmonics do not cover the whole frequency region for the tones with the lowest pitch, but usually the contribution to the overall spectrum of the 40th harmonic is negligible.

The nonlinear parameters are adapted for different input signals (of different complexity) and with different cost functions to assure convergence of the parameters into their global minimum. The used algorithm is always the Levenberg–Marquardt optimization routine, as described in [9] with different cost functions.

- At first, a grid search for the pre- and post-gain of the power-amplifier nonlinearity is performed because these parameters have the most influence on the shape of the output envelope of the digital model. The cost function calculates the difference of the envelopes of the output signals and the gain combination with the lowest error is chosen. The envelope is calculated by low-pass filtering the absolute value of the signal. The cut-off frequency of the used low-pass filter is $f_c = 10$ Hz.
- Afterwards all nonlinear parameters are adapted at the same time. The cost function, however, was designed differently in this optimization step. It calculates the sum of squares between digital model and reference system,

$$C(\mathbf{p}) = (y(n) - \hat{y}(n, \mathbf{p}))^2, \quad (4)$$

with $y(n)$ as the (digitized) output of the reference system and $\hat{y}(n, \mathbf{p})$ as the output of the digital model. \mathbf{p} is the parameter vector of the digital model. In this optimization step, all filters except the input filter H_1 are turned off during optimization. The chosen input signal is a 1000 Hz sine wave with amplitudes from 1 V down to 0.001 V. This first step helps to find a set of parameters which can be used as initial parameters in the next optimization step where all filters in the digital model are turned back on.

- The next optimization step is done with a multi-frequency sine wave. The phase shift between each frequency is chosen in such a way that the peak-factor of the sum of the

different frequencies is minimal and the signal has a flat power-spectrum [13]. The cost function in this case calculates the linear spectrogram of both signals and only compares the magnitude spectrogram, disregarding the phases. The spectrogram is initially calculated with the Fourier transform, but the frequency bins are merged into a semitone-spectrum, starting from $f_0 = 27.5$ Hz. Additionally the magnitude spectrogram is weighted with the inverted absolute threshold of hearing. Afterwards both spectrograms are subtracted from another and all values are squared and summed up to calculate the error value.

- For guitar amplifiers with nonlinear behavior it has been beneficial to add a last step, where the same cost function (spectrogram) is used but a recorded guitar track is used as input, which consists of a combination of guitar tones and chords to further refine the parameters of the digital model.

5. RESULTS

Evaluating the quality of the adapted model is not a trivial task. There are very few perceptually motivated objective scores and they are neither suited for virtual analog modeling evaluation nor are they available for free. There exist perceptually motivated scores like PEAQ or PEMO-Q [14, 15], but they were designed for a different purpose and therefore are not suited for quality assessment of virtual analog modeling.

For this reason the adapted digital model is rated with different methods, first it is evaluated with objective measures. If these objective metrics are close to zero, the result of the modeling process is always good. But in some cases the error is relatively high, but the quality of the adapted model is quite good from a perceptual point of view. This is why a listening test was conducted to assess the quality of the adapted models perceptually. The files which were used in the listening test were also used to calculate the objective scores.

The results are evaluated for different amplifier models and for different guitar signals. Some amplifiers are tested in multiple settings, creating distortion with the pre-amplifier, the power-amplifier or both at once. Other amplifiers are tested in an artist preferred setting, where the user controls of the amplifiers are not altered from the settings the artists used in the rehearsal room.

Figure 6 shows the amplifiers in the artist preferred setting. All amplifiers produced very little distortion in the output signal. The first amplifier (top), the Ampeg VT-22, did not feature separate controls for pre-amplifier and power-amplifier and the reverb was turned off. The Fender Bassman 100 (middle) and Fender Bassman 300 (bottom) were set up to introduce almost no distortion, as can be seen by the gain and volume controls.

Different input signals are used for each amplifier. Input signals from three guitars with different pick-ups are tested:

1. single coil pick-up (SC)
2. humbucker pick-up with medium output (HM1)
3. humbucker pick-up with high output (HM2)

Only the amplifiers which were modeled in the ‘artist preferred’ setting, were set up to have a clean sound, introducing very little distortion in the output signal. The amplifiers which introduced a lot of distortion in the output signal were modeled in multiple settings:

1. High gain and low volume (pre-amp distortion)

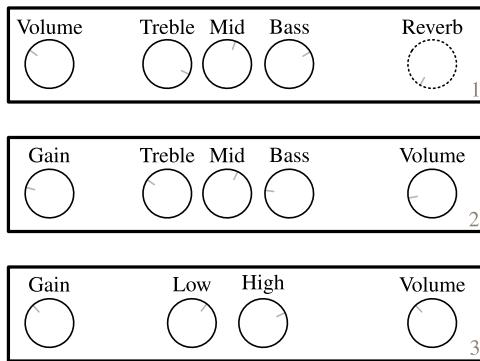


Figure 6: Settings for amplifiers in clean setting. 1.) Ampeg VT-22 (top) 2.) Fender Bassman 100 (middle) 3.) Fender Bassman 300 (bottom)

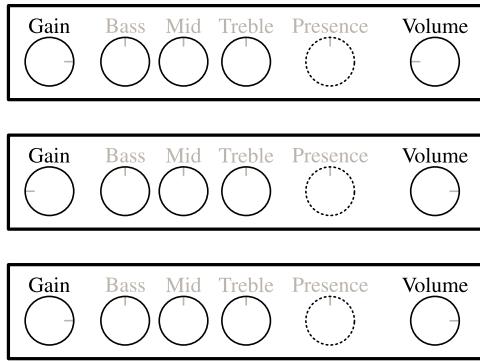


Figure 7: Settings for pre-amp distortion (top), power-amp distortion (middle) and heavy distortion (bottom).

- 2. Low gain and high volume (power-amp distortion)
- 3. high gain and high volume (heavy distortion)

These settings are illustrated in Fig. 7. The tone-section of the amplifiers were set to 12 o'clock and the presence knob is depicted with a dashed line, because only the Marshall - JCM 900 featured a presence control.

5.1. Objective Scores

Two scores are used to evaluate how well the model was adapted. The ‘error to signal ratio’ and the correlation coefficient. The error to signal ratio is defined as the energy of the time-domain error between reference device and digital model,

$$ESR = \frac{\sum_{n=-\infty}^{\infty} (y(n) - \hat{y}(n, \mathbf{p}))}{\sum_{n=-\infty}^{\infty} y(n)}. \quad (5)$$

The correlation coefficient describes the linear dependence of two random variables. In this case $y(n)$ and $\hat{y}(n, \mathbf{p})$ are consid-

Fender Bassman 100 (Blackface-Mod)	ESR	ρ
Single coil (SC)	0.0412	0.9795
Humbucker medium (HM1)	0.0779	0.9611
Humbucker high (HM2)	0.0518	0.9752

Table 1: Objective scores for the Bassman 100 with no distortion.

Ampeg VT-22	ESR	ρ
Single coil (SC)	0.0745	0.9631
Humbucker medium (HM1)	0.1170	0.9417
Humbucker high (HM2)	0.1742	0.9127

Table 2: Objective scores for the VT-22 with very little distortion.

ered as random variables and the correlation coefficient is calculated according to,

$$\rho(y(n), \hat{y}(n, \mathbf{p})) = \frac{\text{cov}(y(n), \hat{y}(n, \mathbf{p}))}{\sigma_{y(n)} \sigma_{\hat{y}(n, \mathbf{p})}}. \quad (6)$$

The results in Tabs. 1 and 2 show that the proposed method works very well with clean or almost clean amplifiers. For the Bassman 100 the ESR remains below 0.1 and the correlation coefficient never drops below 0.96.

The VT-22 also gives very good results, but when the input signal level is high, the error becomes higher too. This can be seen from the results in Tab. 2, where the ESR gets worse if the guitar input has a higher voltage. For the single coil guitar input the ESR is 0.0745 but if the input voltage is higher, which leads to more distortion, the ESR gets above 0.1. The correlation coefficient has the same tendency as the ESR.

Any reference device can add distortion either by increasing the gain, which leads to a clipping pre-amplification stage or by increasing the volume, which leads to a clipping power-amplification stage. The power-amplifier in the reference device is rarely turned up to high values, because it reaches very high sound pressure levels, when the amplifier is connected to a speaker [16], but while measuring only a dummy-load was connected to the reference device.

The results of the modeling process are shown in Tab. 3. Due to the nonlinear behavior of the reference device, the error does not increase proportionally with a rising input level and is already quite high.

Usually a guitarist will add distortion by increasing the gain knob on the amplifier. When comparing the same reference device with a clipping power-amplifier to a clipping pre-amplifier the objective error scores nearly double (see Tab. 4). But this impression is not reflected in the perceived difference between digital model and reference device.

Finally, the objective scores for the reference amplifier which introduced the most distortion in the output signal are shown in Tab. 5. In this case, the error energy is always higher than the actual signal energy, since the ESR is always greater than 1 for all test items. This is also the model which has the greatest deviation from the reference device from a perceptual point of view.

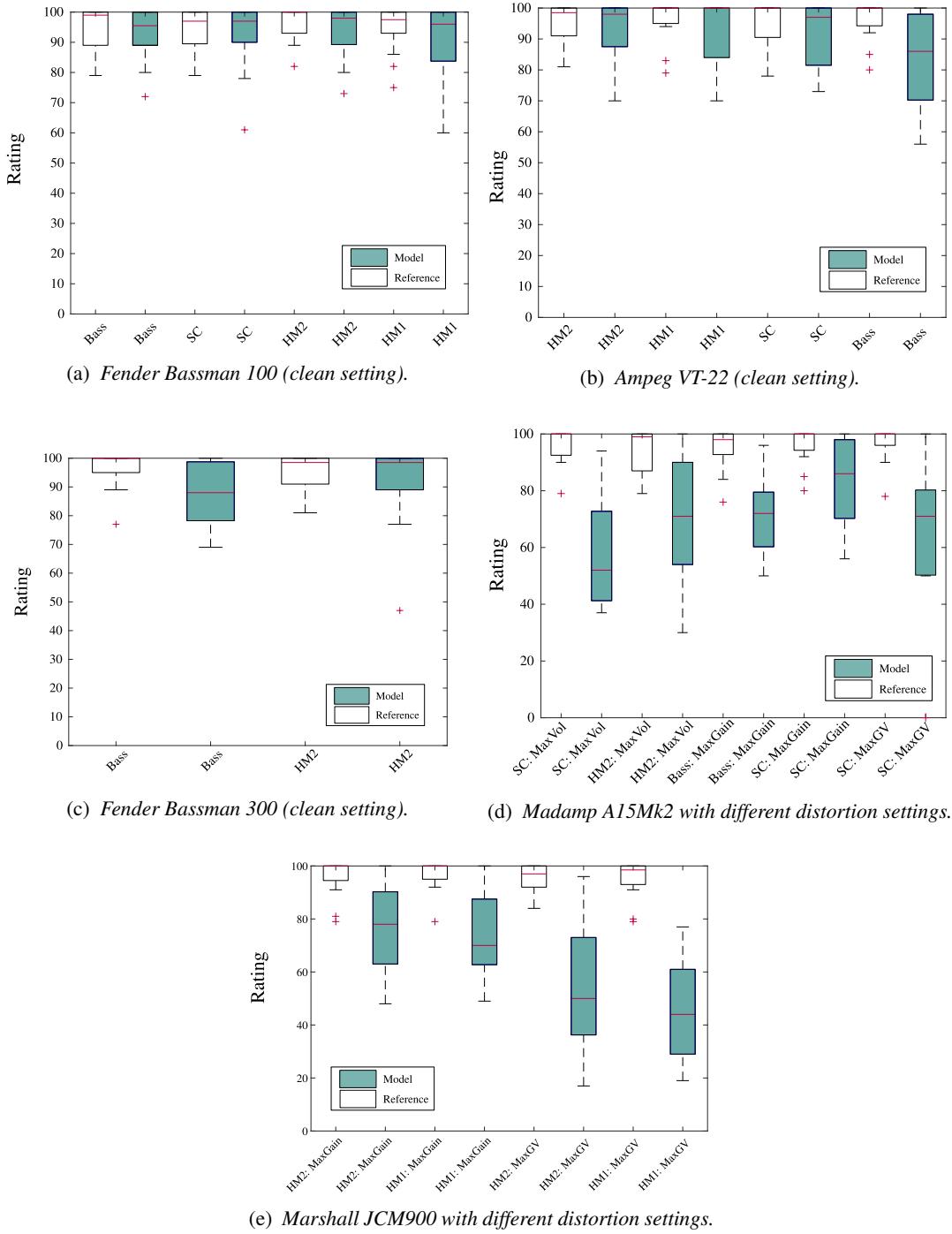


Figure 8: Results of the listening test for all tested amplifiers.

5.2. Listening Test

A listening test was conducted to see how well the adapted models perform for a human test subject. The listening test aimed at rating the adapted model in relation to the analog reference device. The test subjects were presented with a reference item and two test items. The items should be rated according to how **similar** they

sound to the reference, where 100 represents no detectable difference between the item and the reference and 0 represents a very annoying difference. One of the test items was a hidden reference, which was the same audio-file as the reference item.

The listening test featured 20 listening examples with hidden reference and digital model output and is currently still ongoing. So far 15 participants have taken the test from which 8 were ex-

Madamp A15Mk2	ESR	ρ
Single coil (SC)	0.4372	0.7762
Humbucker medium (HM1)	0.3709	0.8104
Humbucker high (HM2)	0.3145	0.8393

Table 3: Objective scores for the A15Mk2 with power-amp distortion (low gain, high volume).

Madamp A15Mk2	ESR	ρ
Single coil (SC)	0.8727	0.5383
Humbucker medium (HM1)	0.6596	0.6552
Humbucker high (HM2)	0.8236	0.5570

Table 4: Objective scores for the A15Mk2 with pre-amplifier distortion (high gain, low volume).

Marshall JCM900	ESR	ρ
Single coil (SC)	1.5277	0.4603
Humbucker medium (HM1)	1.5022	0.2486
Humbucker high (HM2)	1.5488	0.2237

Table 5: Objective scores for the JCM900 with maximum distortion (high gain, high volume).

perienced listeners, 4 were musicians and 2 were unexperienced listeners. At the end of the test, each participant had the option to comment on the test. The framework for the listening test was the ‘BeagleJS’ framework, described in [17]. It features example configurations for ABX and Mushra style listening tests. The Mushra configuration was adapted to fit the needs for model – reference comparison.

The test results have been cleaned by deleting the ratings where the hidden reference was rated with a score lower than 75, but only one test subject was removed from the evaluation completely, because for 13 of 20 items, the hidden reference was rated with scores much lower than 75. Figures 8a – 8e show the results of the listening test. The bar displays the 50% quantile (median) for each item. The lower and upper bounds of the box represent the 25% quantile or the 75% quantile respectively. Outliers are depicted as crosses.

Figures 8a and 8b show the results for the reference amplifiers in clean setting and the adapted models. The results show that the digital model is always rated in the same range as the analog reference device for the test items ‘Bass’, ‘Single Coil (SC)’, ‘Humbucker 1 (HM1)’ and ‘Humbucker 2 (HM2)’. These results confirm that the model is very well adapted as the objective scores, mentioned in Section 5.1, suggest.

The results for the Ampeg VT-22 are similar to the results of the Fender Bassman 100. In some cases there was an unwanted ‘crackling’ noise in the recording of the reference amplifier, which was not reproduced by the digital model. This made it possible to identify the difference between the hidden reference and the digital model output.

The last amplifier which is in an almost clean setting was the Fender Bassman 300 (Fig. 8c). Nevertheless, the HM2 (humbucker with high output voltage) test item had a nearly identical rating as the hidden reference. Only for the input signal from an

electric-bass, there were minor audible differences in the output signal. These results are in agreement with the comments from the participants. Several stated, that they could not perceive any difference when the amps were in a ‘clean’ or ‘almost clean’ setting.

The results of the optimization routine for distorted reference amplifiers are not as good as the results for the clean ones. The more nonlinear the amplifier becomes (more distortion), the higher is the perceivable difference between digital model and analog reference device. This assumption was already made, based on the objective scores from Section 5.1, but is confirmed by the results of the listening test.

The clipping power-amplifier of the Madamp A15Mk2 is rated worse than the clipping pre-amplifier, as shown by the single-coil (SC) items in Fig. 8d. This does not agree with the objective scores, since the error energy for the clipping power-amplifier is twice as low as the error energy for the clipping pre-amplifier. In the listening test, the clipping pre-amplifier was rated with ≈ 90 (median) and the clipping power-amplifier with ≈ 50 (median), in comparison with the hidden reference, which had a median of 100 in both cases. This suggests that these objective scores are not suitable for modeling amplifiers with a lot of distortion and a psycho-acoustically motivated cost-function would drastically improve the virtual analog modeling results for distorted amplifiers.

The listening test results for the last amplifier confirm the assumption, that a reference device with high nonlinear behavior is not identified as well as a system with little nonlinear behavior. The Marshall JCM 900 was rated worse if both pre- and power-amplifier were at high values, in comparison to the first 2 test items were only the pre-amplifier was set to a high value. A common comment from the participants was, that a difference in the noise floor between digital model and reference device made it possible to distinguish the reference from the model.

6. CONCLUSION

This work presents an approach for modeling guitar amplifiers with system identification methods. Input – output measurements are made on a reference device and a digital model, consisting of filters and nonlinear mapping functions, is adapted to recreate the characteristics of the reference device. The results showed that this method performs very good for reference amplifiers in a clean setting with almost no harmonics. If the amplifier introduces distortion, the modeling process does not perform as well.

It is possible to tune the digital model by hand, although it is not recommended. This is an indication that the model is able to recreate also highly nonlinear systems. Therefore a psycho-acoustically motivated cost-function for the iterative optimization routine needs to be developed to improve the results for highly nonlinear systems.

All signals were recorded while the amplifier was not connected to a cabinet. The influence of a cabinet could lead to reduced high frequency content in the output signal, which could lessen the perceived difference between reference device and digital model.

The amplitude of the sine sweep to measure the small signal frequency response of the reference device was set to 0.01 V, which might be too high for some amplifiers and lead to a distorted output. This was not the case for the tested amplifiers but to ensure a correct modeling result a total harmonic distortion measurement should be performed and the amplitude of the sweep should be adapted accordingly to avoid faulty measurements.

7. REFERENCES

- [1] David Te-Mao Yeh, *Digital implementation of musical distortion circuits by analysis and simulation*, Ph.D. thesis, Stanford University, 2009.
- [2] J. Macak, *Real-time Digital Simulation of Guitar Amplifiers as Audio Effects*, Ph.D. thesis, Brno University of Technology, 2011.
- [3] K. Dempwolf, *Modellierung analoger Gitarrenverstärker mit digitaler Signalverarbeitung*, Ph.D. thesis, Helmut-Schmidt-University, 2012.
- [4] K.J. Werner, J.O. Smith, and J.S. Abel, “Wave digital filter adaptors for arbitrary topologies and multiport linear elements,” in *Proc. Digital Audio Effects (DAFx-15)*, Trondheim, Norway, Nov. 30 - Dec. 3 2015.
- [5] 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. Digital Audio Effects.(DAFx-16)*, Brno, Czech Republic, Sept. 2016.
- [6] A Novak, L. Simon, P. Lotton, and J. Gilbert, “Chebyshev model and synchronized swept sine method in nonlinear audio effect modeling,” in *Proc. Digital Audio Effects (DAFx-10)*, Graz, Austria, Sept. 6-10, 2010.
- [7] Fractal Audio Systems, “Multipoint Iterative Matching and Impedance Correction Technology (MIMIC),” Tech. Rep., Fractal Audio Systems, April 2013.
- [8] C. Kemper, “Musical instrument with acoustic transducer,” June 12 2008, US Patent App. 11/881,818.
- [9] F. Eichas and U. Zölzer, “Black-box modeling of distortion circuits with block-oriented models,” in *Proc. Digital Audio Effects (DAFx-15)*, Trondheim, Norway, Nov 30 – Dec 3 2015.
- [10] J. Pakarinen and D.T. Yeh, “A review of digital techniques for modeling vacuum-tube guitar amplifiers,” *Computer Music Journal*, vol. 33, no. 2, pp. 85–100, 2009.
- [11] F. Eichas, S. Möller, and U. Zölzer, “Block-oriented modeling of distortion audio effects using iterative minimization,” in *Proc. Digital Audio Effects (DAFx-15)*, Trondheim, Norway, Nov 30 – Dec 3 2015.
- [12] A. Farina, “Advancements in impulse response measurements by sine sweeps,” in *Audio Engineering Society Convention 122*. Audio Engineering Society, 2007.
- [13] M. Schroeder, “Synthesis of low-peak-factor signals and binary sequences with low autocorrelation (corresp.),” *IEEE Transactions on Information Theory*, vol. 16, no. 1, pp. 85–89, 1970.
- [14] International Telecommunication Union, “Bs.1387: Method for objective measurements of perceived audio quality,” Available online at <http://www.itu.int/rec/R-REC-BS.1387> – accessed April 4th 2017.
- [15] Rainer Huber and Birger Kollmeier, “Pemo - q – a new method for objective audio quality assessment using a model of auditory perception,” *IEEE Transactions on audio, speech, and language processing*, vol. 14, no. 6, pp. 1902–1911, 2006.
- [16] M. Zollner, “Die dummy-load als lautsprecher ersatz (the dummy-load as speaker replacement),” in *GITEC Forum*, 2016.
- [17] S. Kraft and U. Zölzer, “Beaglejs: Html5 and javascript based framework for the subjective evaluation of audio quality,” in *Linux Audio Conference, Karlsruhe, DE*, 2014.

VIRTUAL ANALOG BUCHLA 259 WAVEFOLDER

Fabián Esqueda, Henri Pöntynen, Vesa Välimäki

Acoustics Lab, Dept. of Signal Processing and Acoustics
Aalto University
Espoo, Finland
`firstname.lastname@aalto.fi`

Julian D. Parker

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

ABSTRACT

An antialiased digital model of the wavefolding circuit inside the Buchla 259 Complex Waveform Generator is presented. Wavefolding is a type of nonlinear waveshaping used to generate complex harmonically-rich sounds from simple periodic waveforms. Unlike other analog wavefolder designs, Buchla's design features five op-amp-based folding stages arranged in parallel alongside a direct signal path. The nonlinear behavior of the system is accurately modeled in the digital domain using memoryless mappings of the input–output voltage relationships inside the circuit. We pay special attention to suppressing the aliasing introduced by the nonlinear frequency-expanding behavior of the wavefolder. For this, we propose using the bandlimited ramp (BLAMP) method with eight times oversampling. Results obtained are validated against SPICE simulations and a highly oversampled digital model. The proposed virtual analog wavefolder retains the salient features of the original circuit and is applicable to digital sound synthesis.

1. INTRODUCTION

To talk about Don Buchla is to talk about the history of the analog synthesizer. Motivated by his early experiments with *musique concrète*, California native Donald “Don” Buchla was drawn to the San Francisco Tape Music Center in 1963, where he began collaborating with composers Morton Subotnick and Ramon Sender [1]. Subotnick and Sender commissioned Buchla to design a voltage-controlled musical instrument that could manipulate the characteristics of sounds generated by function generators. This led to the development of Buchla’s first synthesizer, the Buchla 100 [1, 2], completed in 1964.

From the beginning, Buchla’s approach to sound synthesis was fundamentally different to that of his contemporaries, particularly Robert Moog. In Moog synthesizers, sounds are sculpted by filtering harmonically-rich waveforms with resonant filters. This method is known in the literature as “subtractive” synthesis and is commonly dubbed “East Coast” synthesis as a reference to Moog’s New York origins. In contrast, Buchla’s synthesis paradigm (known as “West Coast” synthesis) concentrates on timbre manipulation at oscillator level via nonlinear waveshaping, frequency modulation or phase locking. A trademark module in Buchla synthesizers is the lowpass gate, a filter/amplifier circuit capable of producing acoustic-like plucked sounds by using photoresistive opto-isolators, or “vactrols”, in its control path [3]. Buchla’s designs played a key role in the development of electronic music and can be heard across numerous recordings, such as in the works of the renowned composer Suzanne Ciani [4].

Recent years have seen a resurgence of interest in analog synthesizers, with music technology powerhouses such as Moog and Korg re-releasing modern versions of their now classic designs.

Similarly, contemporary manufacturers of modular synthesizers like Make Noise, Sputnik Modular and Verbos Electronics, to name a few, have reinterpreted Buchla’s designs, rekindling the interest in analog West Coast synthesis. This rise in popularity serves as the motivation to study classic analog devices and to develop virtual analog (VA) models which can be used within digital audio environments. VA instruments are generally more affordable than their analog counterparts, and are exempt from issues such as electrical faults and component aging [5].

In this work we present a novel VA model of the timbre circuit inside the seminal Buchla 259, a complex waveform generator released in 1970 as part of the Buchla 200 synthesizer. The 259 is a dual oscillator module with frequency modulation and waveform synchronization capabilities that provide a wide timbral palette. However, its most distinctive feature is its wavefolding circuit capable of producing the rich harmonic sweeps characteristic to West Coast synthesis. Wavefolding is a type of nonlinear waveshaping in which parts of the input signal that exceed a certain value are inverted or “folded back”. This process introduces high levels of harmonic distortion and thus alters the timbre of the signal.

The use of nonlinear distortion to generate complex sounds has been widely studied within the context of digital synthesis. Well-known methods include the use of nonlinear waveshaping functions, such as Chebyshev polynomials, to expand the spectrum of simple sinusoids [6–9], and frequency modulation (FM) synthesis [10]. Other methods include modified FM synthesis [11], bitwise logical modulation and vector phaseshaping synthesis [12, 13]. Previous research on VA modeling of nonlinear analog audio systems has covered a wide spectrum of topics, including Moog’s ladder filter [14–18], other nonlinear filters [3, 19–21], distortion circuits [22–26] and effects units [27–29].

One of the major challenges in VA modeling is to minimize the effects of aliasing distortion. Aliased components are known to be perceptually disturbing and unpleasant, but become negligible if attenuated sufficiently [30, 31]. The brute force method to reduce aliasing is oversampling, but, if the nonlinearity introduces high levels of distortion, the sample rate may have to be very high to obtain good audio quality. Aliasing suppression techniques have been thoroughly studied in the field of digital audio synthesis [32–35] and, more recently, in nonlinear audio processing [36–39]. In this work we propose the use of the previously introduced bandlimited ramp (BLAMP) method [36, 37] which can be used to bandlimit the corners, or edges, introduced by the wavefolding operation. The BLAMP method significantly reduces the oversampling requirements of the system.

This paper is organized follows. Section 2 details the analysis of the circuit. Section 3 deals with its implementation in the digital domain with emphasis on aliasing suppression. Finally, results and concluding remarks are presented in Sections 4 and 5, respectively.

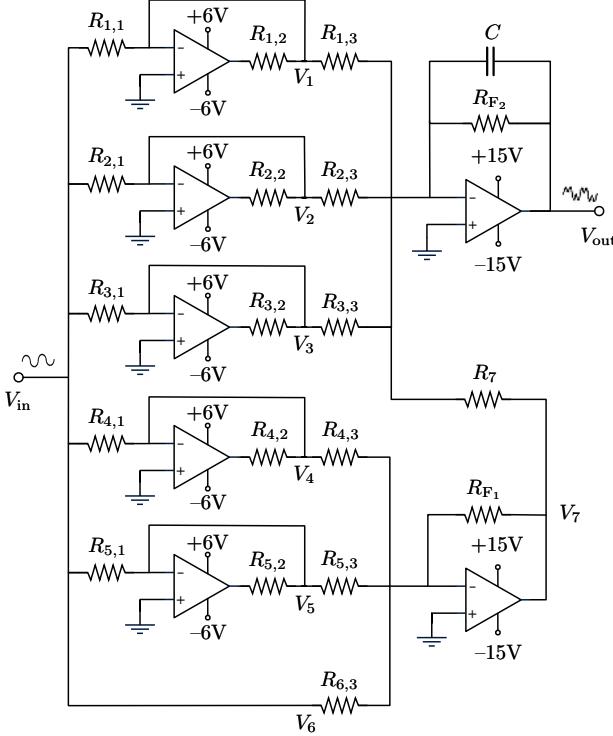


Figure 1: Simplified schematic of the Buchla 259 timbre circuit.

2. CIRCUIT ANALYSIS

Figure 1 shows a simplified schematic of the Buchla 259 timbre circuit. This figure has been adapted from Andre Theelen's DIY version of the circuit¹. The main difference between Fig. 1 and Buchla's original design² is the omission of the "Symmetry" and "Order" controls, which are not considered in this study. The following treatment of the circuit adheres, for the most part, to the analysis presented by Prof. Aaron Lanterman as part of his lecture series "Electronics for Music Synthesis" [40].

The wavefolder inside the Buchla 259 consists of five non-identical op-amp-based folding cells arranged in parallel alongside a direct signal path, as shown in Fig. 1. The two op-amps on the right-hand side of the schematic are set up as summing amplifiers and are used to combine the outputs of all six branches. Overall, this parallel topology differs from that of the more common transistor/diode-based wavefolders, where multiple folding stages are usually cascaded together, e.g. as in the middle section of the Serge Wave Multipliers³. The Intellijel μ Fold II⁴ and Toppobrillo Triple Wavefolder⁵ are examples of commercially-available designs built around a series topology.

To simplify the analysis of the circuit, we first derive the input-output voltage relationship of a single folding cell. Since the parallel paths share the same structure, this result can be applied to all

¹www.ecalpemos.nl/sdiy/buchlaesque-modular/mutant-259-timbre-modindex-section/

²rubidium.dyndns.org/~magnus/synths/companies/buchla/

³http://www.cgs.synth.net/modules/cgs52_folder.html

⁴www.intellijel.com/eurorack-modules/mu-fold-ii/

⁵www.toppobrillo.com/TWF/TWF.html

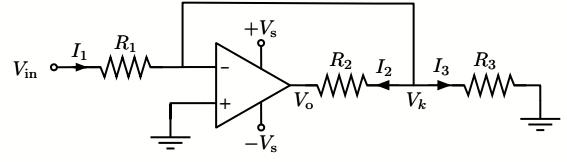


Figure 2: Circuit diagram for a single folding cell, cf. Fig. 1.

folding branches. Component values for the circuit are given in Table 1. Indices have been used to indicate branch number, e.g. $R_{5,2}$ denotes resistor R_2 in the fifth branch.

2.1. Single Folding Cell

Figure 2 shows the schematic for an op-amp circuit that is in the context of this work referred to as a folding cell. The variable V_{in} represents the voltage appearing at the input of all six branches. In the Buchla 259 the input of the timbre circuit is wired internally to the output of a sinusoidal oscillator. We denote the output voltage of each folding branch by V_k , where k is the branch number as counted from top to bottom. The value V_o denotes the voltage at the output terminal of the op-amp. Since R_3 is connected to the virtual ground node formed at the inverting input terminal of the succeeding summing amplifier (see Fig. 1), we assume loading effects between the branches to be minimal, and thus treat each folding cell individually.

First, we assume ideal op-amp behavior and apply Kirchhoff's voltage law (KVL). This results in the current–voltage relationships

$$V_{in} = R_1 I_1 + V_k \quad \text{and} \quad V_o = V_k - R_2 I_2, \quad (1)$$

where

$$V_k = R_3 I_3. \quad (2)$$

Rearranging these equations in terms of currents then gives us

$$I_1 = \frac{V_{in} - V_k}{R_1}, \quad I_2 = \frac{V_k - V_o}{R_2} \quad \text{and} \quad I_3 = \frac{V_k}{R_3}. \quad (3)$$

Next, we apply Kirchhoff's current law (KCL) at node V_k to establish the current relation

$$I_1 = I_2 + I_3. \quad (4)$$

Plugging (3) into (4) results in the expression

$$\frac{V_{in} - V_k}{R_1} = \frac{V_k - V_o}{R_2} + \frac{V_k}{R_3}, \quad (5)$$

Table 1: Component values for the Buchla 259 circuit in Fig. 1.

Name	Value	Name	Value	Name	Value
$R_{1,1}$	10 k Ω	$R_{1,2}$	100 k Ω	$R_{1,3}$	100 k Ω
$R_{2,1}$	49.9 k Ω	$R_{2,2}$	100 k Ω	$R_{2,3}$	43.2 k Ω
$R_{3,1}$	91 k Ω	$R_{3,2}$	100 k Ω	$R_{3,3}$	56 k Ω
$R_{4,1}$	30 k Ω	$R_{4,2}$	100 k Ω	$R_{4,3}$	68 k Ω
$R_{5,1}$	68 k Ω	$R_{5,2}$	100 k Ω	$R_{5,3}$	33 k Ω
—	—	C	100 pF	$R_{6,3}$	240 k Ω
R_7	24.9 k Ω	R_{F_1}	24.9 k Ω	R_{F_2}	1.2 M Ω

which we can solve for V_k as:

$$V_k = \frac{R_3 (R_2 V_{\text{in}} + R_1 V_o)}{R_1 R_3 + R_2 R_3 + R_1 R_2}. \quad (6)$$

Now, since the op-amp is in the inverting configuration, the value of V_o is defined as

$$V_o = -\frac{R_2}{R_1} V_{\text{in}}. \quad (7)$$

This definition implies that the op-amp can provide a fixed gain of $-\frac{R_2}{R_1}$ for all values of V_{in} . If we were to substitute (7) into (6) we would find that $V_k = 0$, as required by ideal op-amp behavior (i.e., the op-amp maintains the input terminals at the same potential) [41]. In practice, however, the value of V_o is limited by the supply voltages and the device is unable to maintain V_k at ground potential when the input voltage is high. Note that the op-amps in the folding branches are connected to lower supply voltages than the rest of the circuit.

Buchla's original design utilized CA3160 op-amps in its folding cells. This particular “rail-to-rail” op-amp features a CMOS output stage and is capable of swinging the output up to the supply voltages. As illustrated in its datasheet [42], the CA3160 exhibits a sharp saturating behavior similar to hard clipping. Therefore, we rewrite (7) as

$$V_o = \begin{cases} -\frac{R_2}{R_1} V_{\text{in}}, & \text{if } |V_{\text{in}}| \leq \frac{R_1}{R_2} V_s \\ -\text{sgn}(V_{\text{in}}) V_s, & \text{otherwise,} \end{cases} \quad (8)$$

where $V_s = 6$ V is the supply voltage of the op-amp and $\text{sgn}()$ is the *signum* function.

By combining (6) and (8), we can derive a piecewise expression for the output of each folding branch in the original circuit:

$$V_k = \begin{cases} \frac{R_{k,3} (R_{k,2} V_{\text{in}} - \text{sgn}(V_{\text{in}}) R_{k,1} V_s)}{R_{k,1} R_{k,3} + R_{k,2} R_{k,3} + R_{k,1} R_{k,2}}, & \text{if } |V_{\text{in}}| > \frac{R_{k,1}}{R_{k,2}} V_s \\ 0, & \text{otherwise.} \end{cases} \quad (9)$$

Figures 3(a)–(e) show the value of $V_{1–5}$ for values of V_{in} between -10 V and 10 V measured at 1 mV steps using SPICE. Since no publicly available SPICE model for the CA3160 seems to exist, LTC6088 was used in the simulations instead. This device is similar to the CA3160 in that it also features a “rail-to-rail”-capable CMOS output stage [43]. These plots show that the output of each folding cell has a “deadband” in the input voltage region where the op-amp displays ideal behavior and maintains V_k at ground potential. At larger input voltage values, the op-amp output saturates to the supply voltage and is unable to maintain the deadband.

2.2. Mixing Stages

Following the folding cells, the output voltages of the six parallel branches are combined with two inverting amplifiers. Voltage V_7 , the output of the lower amplifier (cf. Fig. 1), is formed as the weighted sum of the voltages from the three lower branches

$$V_7 = -R_{F1} \left(\frac{V_4}{R_{4,3}} + \frac{V_5}{R_{5,3}} + \frac{V_{\text{in}}}{R_{6,3}} \right). \quad (10)$$

This voltage is subsequently fed to the input of the upper amplifier along with voltages $V_{1–3}$. The upper amplifier is an active first-order integrator that lowpass filters the weighted combination of the input signals. Assuming that the op-amp is operating within

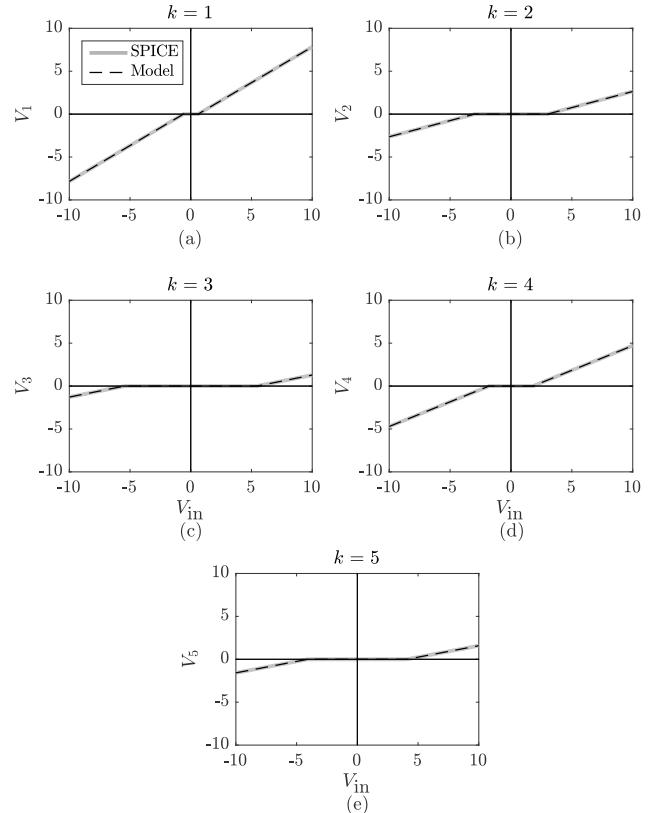


Figure 3: SPICE simulation of the input–output voltage relation of each folding branch against the proposed digital mappings.

its linear region, the summing and filtering operations commute. Therefore, we can simplify the analysis by representing this stage as an inverting amplifier cascaded with a first-order lowpass filter. By replacing capacitor C with an open circuit we can then derive an expression for V'_{out} , the output of the circuit before filtering:

$$V'_{\text{out}} = -R_{F2} \left(\frac{V_1}{R_{1,3}} + \frac{V_2}{R_{2,3}} + \frac{V_3}{R_{3,3}} + \frac{V_7}{R_7} \right). \quad (11)$$

Figure 4(a) shows a SPICE simulation of the input–output voltage relation of the entire circuit when the output filter is bypassed. It can be seen that the weighted sum of the individual branches (cf. Fig. 3) implements a piecewise linear waveshaping function. Figure 4(b) illustrates the outcome of driving the circuit with a sinusoidal signal. A fundamental frequency of 100 Hz and a peak voltage of 5 V were used in this simulation. The output of the circuit exhibits high levels of harmonic distortion which dramatically alters its timbral characteristics. In general, the output signal is perceived as harsher than the original input signal. Significant timbral variation can be achieved by simply modulating the amplitude of the input sinusoid. The filtering effect of the upper summing amplifier is discussed in Section 3.1.

3. DIGITAL IMPLEMENTATION

With the exception of the filtering stage at the output, the Buchla 259 timbre circuit can be categorized as a static system. This

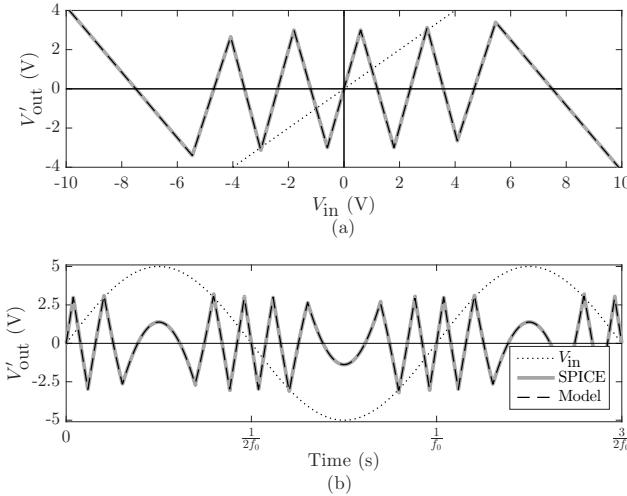


Figure 4: Comparison of the input–output relationship of a SPICE simulation of the Buchla 259 timbre circuit and the proposed digital model for (a) a DC voltage sweep and (b) 100-Hz sinusoidal input with 5-V peak gain.

means that we can derive a digital model using discrete memoryless mappings of the voltage relationships derived in the previous section. First, we define our discrete-time sinusoidal input as

$$V_{\text{in}}[n] = A \sin(2\pi f_0 n T), \quad (12)$$

where n is the sample index, A is peak amplitude, f_0 is the fundamental frequency and T is the sampling period, i.e. $T = 1/f_s$.

From (9) we can then define explicit discrete-time expressions for the output of each folding branch. To facilitate their implementation, terms containing resistor values have been evaluated and replaced for their corresponding approximate scalar values:

$$V_1[n] = \begin{cases} 0.8333V_{\text{in}}[n] - 0.5000s[n] & |V_{\text{in}}[n]| > 0.6000 \\ 0, & \text{otherwise,} \end{cases} \quad (13)$$

$$V_2[n] = \begin{cases} 0.3768V_{\text{in}}[n] - 1.1281s[n] & |V_{\text{in}}[n]| > 2.9940 \\ 0, & \text{otherwise,} \end{cases} \quad (14)$$

$$V_3[n] = \begin{cases} 0.2829V_{\text{in}}[n] - 1.5446s[n] & |V_{\text{in}}[n]| > 5.4600 \\ 0, & \text{otherwise,} \end{cases} \quad (15)$$

$$V_4[n] = \begin{cases} 0.5743V_{\text{in}}[n] - 1.0338s[n] & |V_{\text{in}}[n]| > 1.8000 \\ 0, & \text{otherwise,} \end{cases} \quad (16)$$

$$V_5[n] = \begin{cases} 0.2673V_{\text{in}}[n] - 1.0907s[n] & |V_{\text{in}}[n]| > 4.0800 \\ 0, & \text{otherwise,} \end{cases} \quad (17)$$

where $s[n] = \text{sgn}(V_{\text{in}}[n])$. From these branches we can then define a global summing stage:

$$\begin{aligned} V'_{\text{out}}[n] = & -12.000V_1[n] - 27.777V_2[n] - 21.428V_3[n] \\ & + 17.647V_4[n] + 36.363V_5[n] + 5.000V_{\text{in}}[n]. \end{aligned} \quad (18)$$

Figures 3 and 4 show the input–output relation of these mappings against the previously presented SPICE simulations. These results show a good match between the original and modeled behavior, with an absolute error in the range of 10^{-5} V.

3.1. Filtering Stage

The filter at the output of the system is a one-pole lowpass filter. In the Laplace domain, the transfer function of this filter is given by

$$H(s) = \frac{w_c}{s + w_c}, \quad (19)$$

where $w_c = 2\pi f_c$ and f_c represent the cutoff frequency in radians and Hz, respectively [44, 45]. From Fig. 1 the cutoff of the filter is derived as

$$f_c = \frac{1}{2\pi R_{F_2} C} \approx 1.33 \text{ kHz}. \quad (20)$$

This relatively low cutoff frequency indicates the purpose of the filter is simply to act as a fixed tone control, attenuating the perceived brightness of the output by introducing a gentle 6-dB/octave roll-off. Equation (19) can be discretized using the bilinear transform, which results in the z-domain transfer function

$$H(z) = \frac{b_0 + b_1 z^{-1}}{1 + a_1 z^{-1}}, \quad (21)$$

where

$$b_0 = b_1 = \frac{w_c T}{2 + w_c T} \quad \text{and} \quad a_1 = \frac{w_c T - 2}{w_c T + 2}.$$

Due to the low cutoff parameter, the warping effects of the bilinear transform can be neglected. This transfer function can be implemented digitally, e.g. using Direct Form II Transposed [44].

3.2. Antialiasing

Given the highly nonlinear nature of wavefolding, audio-rate implementations of the proposed model using (12)–(18) will suffer from excessive aliasing distortion. This problem can be attributed to the corners or edges introduced by the folding cells of the system (cf. Fig. 4). These corners indicate that the first derivative of the signal is discontinuous and, as such, has infinite frequency content. In the discrete-time domain, frequency components that exceed the Nyquist limit will be reflected into the audio band as aliases.

To ameliorate this condition we propose the use of the BLAMP method, which has previously been used in the context of ideal nonlinear operations such as signal clipping and rectification [36, 37]. This method consists of replacing the corners with bandlimited versions of themselves. It is an extension of the bandlimited step (BLEP) method used in subtractive synthesis [33–35], which is in turn based on the classic bandlimited impulse train (BLIT) synthesis method [32].

The BLAMP function is a closed-form expression that models a bandlimited discontinuity in the first derivative of a signal. It is derived from the second integral of the bandlimited impulse [35], or sinc, function and is defined as

$$R_{\text{BL}}(t) := t \left[\frac{1}{2} + \frac{1}{\pi} \text{Si}(\pi f_s t) \right] + \frac{\cos(\pi f_s t)}{\pi^2 f_s}, \quad (22)$$

where t is time and $\text{Si}(x)$ is the sine integral

$$\text{Si}(x) := \int_0^x \frac{\sin(t)}{t} dt. \quad (23)$$

Computing the difference between the BLAMP and the ideal ramp function

$$R(t) := \begin{cases} t, & \text{when } t \geq 0 \\ 0, & \text{when } t < 0 \end{cases} \quad (24)$$

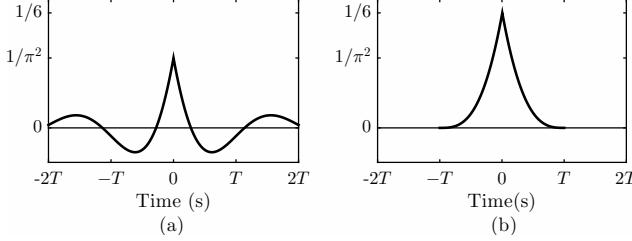


Figure 5: Time domain representation of (a) the central lobe of the BLAMP residual function and (b) its two-point polynomial approximation.

produces the BLAMP residual function shown in Fig. 5(a). In the discrete-time domain, this function is used to reduce aliasing by superimposing it on every corner within the waveform and sampling it at neighboring sample points. A crucial step in this process is centering the residual around the exact point in time where each discontinuity occurs, which is usually between samples.

Due to the high computational costs of evaluating (22), we will use its two-point polynomial approximation (polyBLAMP) instead [36]. Figure 5(b) illustrates the time-domain waveform of the two-point polyBLAMP residual function evaluated using the expressions given in Table 2. In this context $d \in [0, 1]$ is the fractional delay required to center the residual function between two samples.

In the case of the Buchla 259 timbre circuit, the BLAMP method is applied independently within each folding branch. To facilitate its implementation, we define an intermediate processing step in which the input–output relationships of the folding cells (13)–(17) are rewritten as inverse clippers. We then denote the output of the k th inverse clipper as V'_k , which can be written as

$$V'_k[n] = \begin{cases} V_{\text{in}}[n], & \text{if } |V_{\text{in}}[n]| > \frac{R_{k,1}}{R_{k,2}} V_s \\ \text{sgn}(V_{\text{in}}[n]) \frac{R_{k,1}}{R_{k,2}} V_s, & \text{otherwise.} \end{cases} \quad (25)$$

Figure 6 shows the input–output relation of this intermediate processing stage. The advantage of this seemingly unnecessary step is that now we can apply the BLAMP method following the same approach described in [36] and [37] for the case of the regular hard clipper. This process involves detecting the transition from non-clipping to clipping samples (i.e. detecting the corners), computing the exact fractional clipping point and adding the correction function to the samples immediately before and after each corner. Prior to addition, the polyBLAMP function must be scaled by the slope of the input signal at the clipping point. Since we know the input to the system is a sinusoidal waveform, we can compute the fractional clipping points and their respective slopes analytically, thus facilitating the implementation and improving the robustness of the method.

Table 2: Two-point polyBLAMP function and its residual [36].

Span	Two-point polyBLAMP $d \in [0, 1]$
$[-T, 0]$	$d^3/6$
$[0, T]$	$-d^3/6 + d^2/2 + d/2 + 1/6$
Span	Two-point polyBLAMP residual $d \in [0, 1]$
$[-T, 0]$	$d^3/6$
$[0, T]$	$-d^3/6 + d^2/2 - d/2 + 1/6$

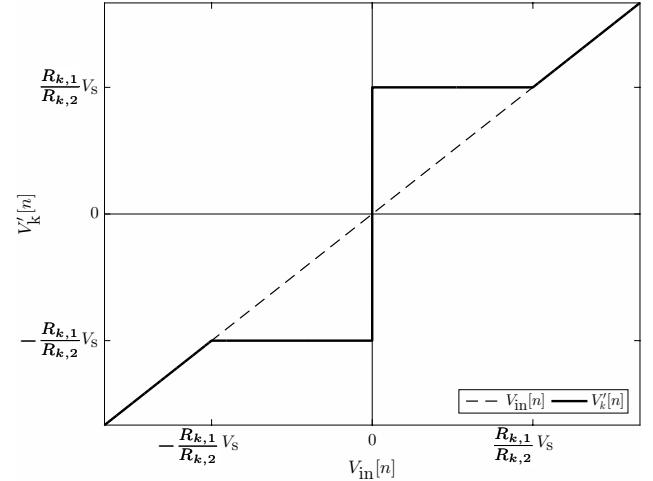


Figure 6: Input–output relationship of the proposed intermediate processing step, the inverse clipper (25).

For an arbitrary inverse clipper stage (25) driven by an f_0 -Hz sinewave starting at zero phase, the first clipping point (in seconds) is given by

$$t_1 = \frac{\sin^{-1}(V_s R_{k,1} / A R_{k,2})}{2\pi f_0}. \quad (26)$$

From this value, we can evaluate the three remaining clipping points within the first period of the signal:

$$t_2 = \frac{1}{2f_0} - t_1, \quad t_3 = \frac{1}{2f_0} + t_1 \quad \text{and} \quad t_4 = \frac{1}{f_0} - t_1. \quad (27)$$

Figure 7(a) shows the result of inverse-clipping the first period of a sinewave, all four clipping points are highlighted. Subsequent clipping points can then be computed by adding multiples of the fundamental period, i.e. $1/f_0$.

For a stationary sinewave, the magnitude of the slope is the same at all clipping points. Therefore, we can define a closed-form expression of the polyBLAMP scaling factor as

$$\mu = |2\pi f_0 A \cos(2\pi f_0 t_{1-4}) / f_s|. \quad (28)$$

Figure 7(b) illustrates the process of centering the polyBLAMP residual function at each clipping point, scaling it and sampling it at neighboring samples. The polarity must be adjusted according to the polarity of the signal at the clipping point. Although in this study we only consider the case of sinusoidal inputs, the same approach can be adapted when other periodic signals are used as input to the wavefolder, e.g. sawtooth and triangular waveforms.

Now, if we then define \tilde{V}_k' as the signal that results from applying the polyBLAMP method to V'_k , we can write an expression for \tilde{V}_k' , the antialiased output of each folding cell:

$$\begin{aligned} \tilde{V}_k'[n] = & \frac{R_{k,2} R_{k,3}}{R_{k,1} R_{k,3} + R_{k,2} R_{k,3} + R_{k,1} R_{k,2}} \left[\tilde{V}_k'[n] \right. \\ & \left. - \text{sgn}(\tilde{V}_k'[n]) \frac{V_s R_{k,1}}{R_{k,2}} \right]. \end{aligned} \quad (29)$$

This step basically undoes the intermediate processing step (25) while preserving the antialiased behavior.

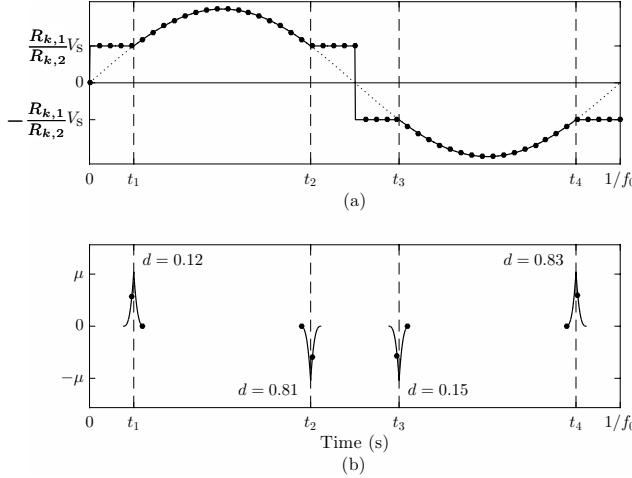


Figure 7: (a) Time-domain representation of a sinewave processed by the inverse clipping stage (25) and (b) the process of centering the polyBLAMP residual at each clipping point. The fractional delay is given for each corner.

A complete block diagram of the proposed wavefolder model, including the output filter, is given in Fig. 9. Boxes labeled w_{1-5} consist of the inverse clipper (25) followed by polyBLAMP correction and the mapping function (29). Once again, a tilde has been used to distinguish \tilde{V}_{out} , the output of the system with aliasing suppression, from V_{out} , its trivial counterpart.

4. RESULTS

Having compared the time-domain characteristics of the proposed model against SPICE simulations (cf. Figs. 3 and 4), in this section we move on to observe and evaluate its frequency-domain behavior. The spectrogram in Fig. 8 shows the effect of sweeping the input gain A from 0 to 10 for a sinewave with fundamental frequency $f_0 = 100$ Hz. Compared to typical saturating waveshapers (e.g. the tanh function or the hard clipper), where the level of introduced harmonics is directly proportional to input gain, wavefolding generates complex harmonic patterns reminiscent of FM synthesis. From a perceptual point of view, the folded waveform can be described as being brighter and more abrasive than the original input signal. It should be pointed out that due to the odd symmetry of the wavefolding operation [cf. Fig. 4(a)], the system introduces odd harmonics only.

Next, we analyze the effect of wavefolding on a static 890-Hz input sinewave with amplitude $A = 5$. Figures 10(a)–(b) show the waveform and magnitude spectrum, respectively, of the system’s output when implemented at audio rate (i.e. $f_s = 44.1$ kHz) and without polyBLAMP correction. The resulting signal is practically unusable, as it exhibits very high levels of audible aliasing distortion. In comparison, Figs. 10(c)–(d) show the outcome of operating at the same rate but employing the two-point polyBLAMP method. As expected, the overall level of aliasing has been considerably attenuated. Next, Figs. 10(e)–(f) show the output of the system for a sample rate $f_s = 2.82$ MHz, i.e. 64 times the previous rate. This example was generated by synthesizing the input sinewave at the target rate and plotting only those frequency components below 20 kHz. The output is virtually free from alias-

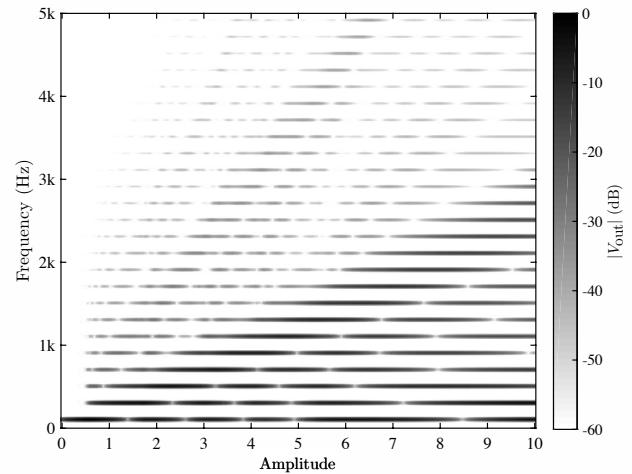


Figure 8: Magnitude response of the system for a 110-Hz sinusoidal input with amplitude ranging from 0 to 10.

ing, with only a handful of components laying above the -100 dB line. Lastly, Figs. 10(g)–(h) show the outcome of using eight times oversampling and the proposed polyBLAMP method. Results obtained are comparable to those in Fig. 10(f), which indicates that the proposed method reduces the oversampling requirements of the system.

The overall increase in signal quality provided by the two-point polyBLAMP method was measured for a larger set of input signals. Figure 11 shows the measured signal-to-noise ratio (SNR) at the output of the wavefolder for input sinewaves with fundamental frequency between 100 Hz and 5 kHz. In this context, we consider SNR to be the power ratio between the desired harmonics and aliasing components. This plot shows that the two-point polyBLAMP method provides an SNR increase of approx. 12 dB over a trivial audio-rate implementation. When combined with eight times oversampling the proposed method yields an average SNR increase of approx. 20 dB w.r.t. oversampling by factor 64.

In terms of computational costs, the two-point polyBLAMP method is highly efficient in that only samples around discontinuities are processed. Therefore, the complexity of the method increases as a function of fundamental frequency, not oversampling factor. For the case of a 5-kHz sinusoidal input (i.e. the worst-case scenario for the polyBLAMP method in terms of operation count), Matlab simulations indicated that the proposed method is approx. 6 times faster than oversampling by factor 64. This estimate does not include the costs of any resampling filters at the output of the system, which will also be more expensive for the case of oversampling by 64. An implementation of the proposed model and accompanying sound examples are available at <http://research.spa.aalto.fi/publications/papers/dafx17-wavefolder>.

5. CONCLUSIONS

In this work we have examined the underlying structure of the Buchla 259 wavefolder, also known as the timbre circuit. The analysis of the circuit provides a glimpse into the unconventional designs of Don Buchla and his approach to sound synthesis. A digi-

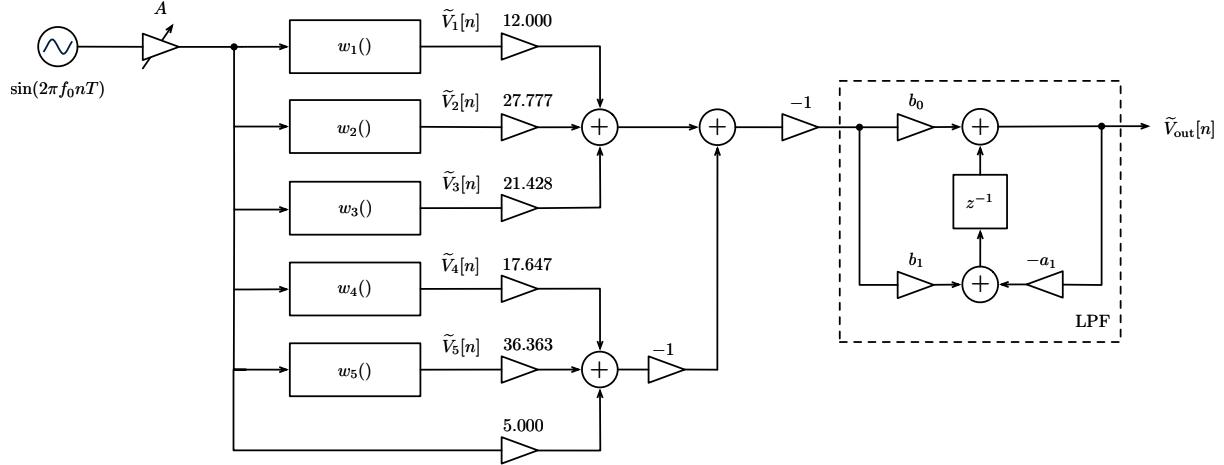


Figure 9: Block diagram of the proposed digital Buchla 259 wavefolder. The LPF block is the lowpass filter at the output of the system.

tal model of the wavefolder is derived using nonlinear memoryless mappings based on the input–output voltage relationships within the circuit. In an effort to minimize the high levels of aliasing distortion caused by the inherent frequency-expanding behavior of the system, the use of the BLAMP method has been proposed,

more specifically in its two-point polynomial form. This method reduces the oversampling requirements of the system, allowing us to accurately process sinusoidal waveforms with fundamental frequencies up to 5 kHz at a sample rate of 352.8 kHz, which is eight times the standard audio rate. The proposed model is free from perceivable aliasing and can be implemented as part of a real-time digital music synthesis environment.

6. ACKNOWLEDGMENTS

The work of Fabián Esqueda is supported by the Aalto ELEC Doctoral School. Part of this work was conducted during Julian Parker’s visit to Aalto University in March 2017.

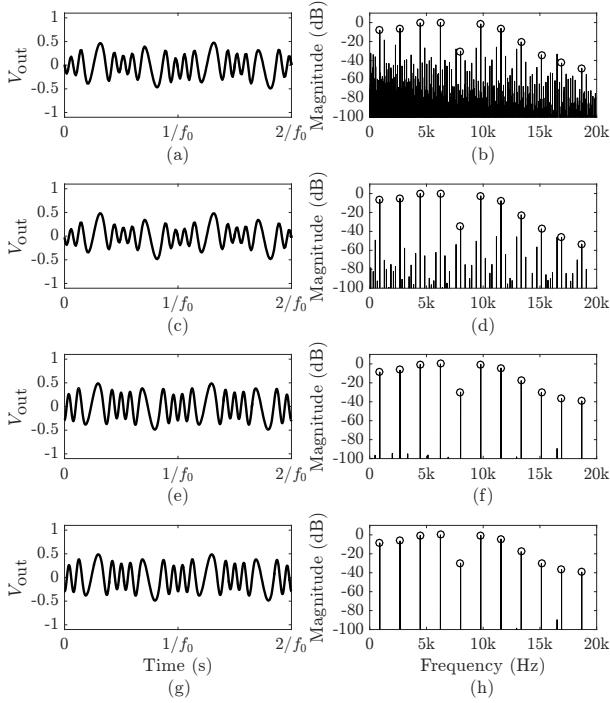


Figure 10: Waveform and magnitude spectrum of an 890-Hz sinewave ($A = 5$) processed using the proposed model (a)–(b) at audio rate ($f_s = 44.1$ kHz), (c)–(d) at audio rate with the two-point polyBLAMP method, (e)–(f) using 64 times oversampling ($f_s = 2.82$ MHz) and (g)–(h) with 8 times oversampling ($f_s = 352.8$ kHz) and the two-point polyBLAMP method.

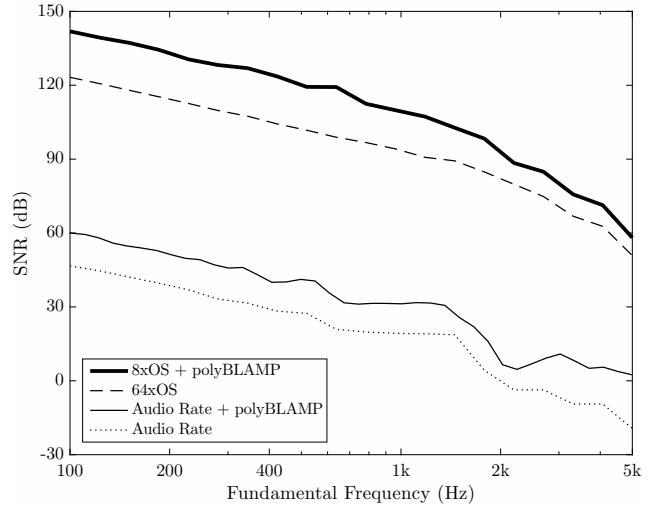


Figure 11: Measured SNRs for the proposed wavefolder model under sinusoidal input ($A = 5$) implemented trivially ($f_s = 44.1$ kHz), with the two-point polyBLAMP method, with oversampling by factor 64 ($f_s = 2.82$ MHz) and with oversampling by factor 8 and the two-point polyBLAMP method ($f_s = 352.8$ kHz).

7. REFERENCES

- [1] D. Bernstein, *The San Francisco Tape Music Center: 1960s Counterculture and the Avant-Garde*, University of California Press, Ltd., Berkeley, CA, 2008.
- [2] Buchla Electronic Musical Instruments, “The history of Buchla,” <https://buchla.com/history/>, Accessed June 19, 2017.
- [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, pp. 278–285.
- [4] S. Ciani, “Buchla Concerts 1975,” Finders Keepers Records, 2016.
- [5] V. Välimäki, F. Fontana, J. O. Smith, and U. Zölzer, “Introduction to the special issue on virtual analog audio effects and musical instruments,” *IEEE Trans. Audio, Speech, Language Process.*, vol. 18, no. 4, pp. 713–714, May 2010.
- [6] D. Arfib, “Digital synthesis of complex spectra by means of multiplication of non linear distorted sine waves,” in *Proc. 59th Conv. Audio Eng. Soc.*, Hamburg, Germany, Mar. 1978.
- [7] M. Le Brun, “Digital waveshaping synthesis,” *J. Audio Eng. Soc.*, vol. 27, no. 4, pp. 250–266, Apr. 1979.
- [8] C. Roads, “A tutorial on non-linear distortion or waveshaping synthesis,” *Comput. Music J.*, vol. 3, no. 2, pp. 29–34, 1979.
- [9] J. Lane, D. Hoory, E. Martinez, and P. Wang, “Modeling analog synthesis with DSPs,” *Comput. Music J.*, vol. 21, no. 4, pp. 23–41, 1997.
- [10] J. Chowning, “The synthesis of complex audio spectra by means of frequency modulation,” *J. Audio Eng. Soc.*, vol. 21, no. 7, pp. 526–534, Sept. 1973.
- [11] V. Lazzarini and J. Timoney, “New perspectives on distortion synthesis for virtual analog oscillators,” *Comput. Music J.*, vol. 34, no. 1, 2010.
- [12] J. Kleimola, “Audio synthesis by bitwise logical modulation,” in *Proc. 11th Int. Conf. Digital Audio Effects (DAFx-08)*, Espoo, Finland, Sept. 2008, pp. 60–70.
- [13] J. Kleimola, V. Lazzarini, J. Timoney, and V. Välimäki, “Vector phaseshaping synthesis,” in *Proc. 14th Int. Conf. Digital Audio Effects (DAFx-11)*, Paris, France, Sept. 2011, pp. 233–240.
- [14] T. Stilson and J. O. Smith, “Analyzing the Moog VCF with considerations for digital implementation,” in *Proc. Int. Comput. Music Conf.*, Hong Kong, Aug. 1996, pp. 398–401.
- [15] 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. 61–164.
- [16] T. Hélie, “Volterra series and state transformation for real-time simulations of audio circuits including saturations: Application to the Moog ladder filter,” *IEEE Trans. Audio, Speech, Language Process.*, vol. 18, no. 4, pp. 747–759, May 2010.
- [17] 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.
- [18] D. Medine, “Dynamical systems for audio synthesis: Embracing nonlinearities and delay-free loops,” *Appl. Sci.*, vol. 6, no. 5, 2016.
- [19] D. Rossum, “Making digital filters sound analog,” in *Proc. Int. Comput. Music Conf.*, San Jose, CA, USA, Oct. 1992, pp. 30–33.
- [20] M. Civolani and F. Fontana, “A nonlinear digital model of the EMS VCS3 voltage-controlled filter,” in *Proc. Int. Conf. Digital Audio Effects (DAFx-08)*, Espoo, Finland, Sept. 2008, pp. 35–42.
- [21] G. Moro and A. P. McPherson, “Approximating non-linear inductors using time-variant linear filters,” in *Proc. Int. Conf. Digital Audio Effects (DAFx-15)*, Trondheim, Norway, Nov. 2015, pp. 249–256.
- [22] D. T. Yeh, J. Abel, and J. O. Smith III, “Simulation of the diode limiter in guitar distortion circuits by numerical solution of ordinary differential equations,” in *Proc. Int. Conf. Digital Audio Effects (DAFx-07)*, Bordeaux, France, Sept. 2007, pp. 197–204.
- [23] J. Macak, J. Schimmel, and M. Holters, “Simulation of Fender type guitar preamp using approximation and state-space model,” in *Proc. 15th Int. Conf. Digital Audio Effects (DAFx-12)*, York, UK, Sept. 2012, pp. 209–216.
- [24] 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.
- [25] K. J. Werner, V. Nangia, A. Bernardini, J. O. Smith, and A. Sarti, “An improved and generalized diode clipper model for wave digital filters,” in *Proc. Audio Eng. Soc. Conv.*, New York, USA, Oct.–Nov. 2015.
- [26] A. Bernardini, K. J. Werner, A. Sarti, and J. O. Smith III, “Modeling nonlinear wave digital elements using the Lambert function,” *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 63, no. 8, pp. 1231–1242, Aug. 2016.
- [27] A. Huovilainen, “Enhanced digital models for analog modulation effects,” in *Proc. Int. Conf. Digital Audio Effects (DAFx-05)*, Madrid, Spain, Sept. 2005, pp. 155–160.
- [28] 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, Sept. 2010, pp. 50–56.
- [29] 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.
- [30] H.-M. Lehtonen, J. Pekonen, and V. Välimäki, “Audibility of aliasing distortion in sawtooth signals and its implications for oscillator algorithm design,” *J. Acoust. Soc. Am.*, vol. 132, no. 4, pp. 2721–2733, Oct. 2012.
- [31] J. Schimmel, “Audible aliasing distortion in digital audio synthesis,” *Radioengineering*, vol. 21, no. 1, pp. 56–62, Apr. 2012.
- [32] 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.
- [33] E. Brandt, “Hard sync without aliasing,” in *Proc. Int. Comput. Music Conf.*, Havana, Cuba, Sept. 2001, pp. 365–368.
- [34] V. Välimäki and A. Huovilainen, “Antialiasing oscillators in subtractive synthesis,” *IEEE Signal Process. Mag.*, vol. 24, no. 2, pp. 116–125, Mar. 2007.
- [35] V. Välimäki, J. Pekonen, and J. Nam, “Perceptually informed synthesis of bandlimited classical waveforms using integrated polynomial interpolation,” *J. Acoust. Soc. Am.*, vol. 131, no. 1, pp. 974–986, Jan. 2012.
- [36] F. Esqueda, S. Bilbao, and V. Välimäki, “Aliasing reduction in clipped signals,” *IEEE Trans. Signal Process.*, vol. 60, no. 20, pp. 5255–5267, Oct. 2016.
- [37] F. Esqueda, V. Välimäki, and S. Bilbao, “Rounding corners with BLAMP,” in *Proc. Int. Conf. Digital Audio Effects (DAFx-16)*, Brno, Czech Republic, Sept. 2016, pp. 121–128.
- [38] J. Parker, V. Zavalishin, and E. Le Bivic, “Reducing the aliasing of nonlinear waveshaping using continuous-time convolution,” in *Proc. Int. Conf. Digital Audio Effects (DAFx-16)*, Brno, Czech Republic, Sept. 2016, pp. 137–144.
- [39] S. Bilbao, F. Esqueda, J. D. Parker, and V. Välimäki, “Antiderivative antialiasing for memoryless nonlinearities,” *IEEE Signal Process. Lett.*, vol. 24, no. 7, pp. 1049–1053, July 2017.
- [40] Georgia Institute of Technology, “Aaron Lanterman’s Home Page,” <http://lanterman.ece.gatech.edu/>, [Online], Accessed April 9, 2017.
- [41] B. Razavi, *Fundamentals of Microelectronics*, Wiley, Hoboken, NJ, 1st edition, 2008.
- [42] Intersil, “CA3160 – 4MHz, BiMOS Operational Amplifier with MOSFET Input/CMOS Output,” 2004.
- [43] Linear Technology, “LTC6087/LTC6088 Dual/Quad 14MHz, Rail-to-Rail CMOS Amplifiers,” 2007.
- [44] J. O. Smith, *Introduction to Digital Filters with Audio Applications*, W3K Publishing, <http://www.w3k.org/books/>, 2007.
- [45] V. Zavalishin, *The Art of VA Filter Design*, Native Instruments, Berlin, Germany, 2012.

NETWORK VARIABLE PRESERVING STEP-SIZE CONTROL IN WAVE DIGITAL FILTERS

Michael Jørgen Olsen¹, Kurt James Werner² and François G. Germain¹

¹Center for Computer Research in Music and Acoustics (CCRMA)
Stanford University
660 Lomita Drive, Stanford, CA 94305, USA
[mjolsen|francois]@ccrma.stanford.edu

²The Sonic Arts Research Centre (SARC)
School of Arts, English and Languages
Queen's University Belfast, UK
k.werner@qub.ac.uk

ABSTRACT

In this paper a new technique is introduced that allows for the variable step-size simulation of wave digital filters. The technique is based on the preservation of the underlying network variables which prevents fluctuation in the stored energy in reactive network elements when the step-size is changed. This method allows for the step-size variation of wave digital filters discretized with any passive discretization technique and works with both linear and nonlinear reference circuits. The usefulness of the technique with regards to audio circuit simulation is demonstrated via the case study of a relaxation oscillator where it is shown how the variable step-size technique can be used to mitigate frequency error that would otherwise occur with a fixed step-size simulation. Additionally, an example of how aliasing suppression techniques can be combined with physical modeling is given with an example of the polyBLEP antialiasing technique being applied to the output voltage signal of the relaxation oscillator.

1. INTRODUCTION

The physical modeling of analog reference systems is typically done using three main paradigms: wave digital filters (WDFs) [1–5], state space filters [6–8] and port-Hamiltonian [9] modeling. The WDF technique was developed in the 1970s with the digitization of classical ladder/lattice filters in mind. A need to retain the passivity of the original reference circuits was required so that the simulations would be stable under finite numeric conditions.

Since the early days of WDFs, the theory has evolved considerably in part due to interest from the physical modeling community in modeling historic audio circuits. The interest from this community has led to the development of new techniques that allow for the simulation of circuits containing complex topologies [4, 10, 11] and multiple/multiport nonlinearities [3, 12, 13]. There are also many interesting case studies of the simulation of reference circuits containing nonlinear network elements including distortion circuits containing diodes [14–16], tube amplifiers containing triodes [17–20] and circuits containing operational amplifiers (op amps) [14, 21].

The digital synthesis of classic analog waveforms and the antialiasing techniques needed to reduce their aliasing in the digital domain is another important thread of research. While there are a large number of techniques for synthesizing these signals, one of main methods involves bandlimiting the analog versions of the waveforms (or their derivatives) and either directly sampling the bandlimited version or forming approximate correction functions from it using polynomial interpolation [22–24]. A variant of the latter technique has also been applied to antialiasing in nonlinear waveshaping [25, 26]. Recently, a new antialiasing technique has been introduced in [26] and expanded upon in [27] that is based

upon the differentiation of the antiderivatives of memoryless nonlinearities. The idea of combining physical modeling and antialiasing techniques has been suggested in [27].

To motivate the use of the variable step-size simulation method as well as provide an example of how physical modeling and antialiasing techniques can be combined, we study a square wave oscillator circuit commonly known as a relaxation oscillator. This circuit comes from a larger class of more complex oscillator circuits and was chosen to study due to its simplicity as well as two challenges that its simulation presents: significant frequency error which occurs when running the simulation with a fixed step-size and the need for an antialiasing method to suppress the aliasing present in the output waveform.

These two challenges are addressed as follows. First, the frequency error is reduced using the new variable step-size WDF simulation technique. Secondly, antialiasing of the output waveform is accomplished using the polyBLEP method [24] where domain knowledge of the circuit is used to estimate the exact location of the discontinuities in the output waveform.

In Section 2, a previous variable step-size WDF technique is reviewed and the new method is presented. The case study of the relaxation oscillator is given in Section 3. The circuit and its components are described in detail in Section 3.1. Section 3.2 explains how the WDF model of the circuit is derived and, in particular, how the op amp is implemented. The issues encountered when running the simulation with a fixed step-size are detailed in Section 3.3 and the way in which the new variable step-size technique is used to counter them is presented in Section 3.4. The technique in which polyBLEP is used to suppress output signal aliasing appears in Section 3.6 which is followed by the concluding thoughts in Section 4.

2. VARIABLE STEP-SIZE SIMULATION

Historically, WDFs were formulated to run at a fixed step-size using voltage waves [1] which are defined:

$$\begin{cases} a = \frac{1}{2}(v + i) \\ b = \frac{1}{2R}(v - i) \end{cases} \quad (1)$$

where a is called the “incident” wave, b is called the “reflected” wave and R is a free variable called the port resistance which is used to tune the system to eliminate delay free loops. A technique for simulating WDFs with a variable step-size was presented in [28] where it was shown that varying the step-size of an RLC circuit (Figure 1) discretized with the trapezoidal method led to an increase in energy in a zero-input simulation when it should have been passive. Their solution to the problem was to run the

Table 1: *Reactive Elements*

Element	Laplace Domain	Wave Domain	Port Resistance
capacitor	$v(s) = i(s)/(sC)$	$b[n] = a[n - 1]$	$R = T/(2C)$
inductor	$v(s) = sLi(s)$	$b[n] = -a[n - 1]$	$R = 2L/T$

variable step-size simulation on an equivalent circuit where the inductor was replaced with a network equivalent gyrator and capacitor pair. They stated that the equivalent circuit corresponded to discretization using the Gauss (midpoint) method and demonstrated that it exhibited the expected behavior of the reference circuit. Since their method depends on the replacement of inductors with network equivalent devices it is unclear whether the method will be guaranteed to work for all possible reference circuits.

In this section we introduce a new technique which allows WDFs discretized with any passive discretization technique, in particular the trapezoidal method, to be simulated with variable step-sizes. The development of the new technique was motivated by the fact that it was not obvious how the technique from [28] can be used with the relaxation oscillator circuit presented in the case study in Section 3. In Section 2.1, the general technique will be introduced and described. Later, in Section 3.4, it will be shown how varying the step-size fails to work with the example circuit and how the new technique can be used to mitigate the encountered issues.

2.1. Variable Step-Size WDF Simulation Technique

The variation of step-size in a WDF discretized with the trapezoidal rule (or other passive discretization technique) requires careful consideration due to the intricate relationship between the sampling rate and reactive network elements. Looking at the definition of wave variables (Equation (1)) and the port resistance values for reactive elements (Table 1) it is clear that the wave variables of these elements are directly coupled to the sampling rate.

The stored energy of a passive network is given by [28]:

$$\mathbf{w}_{k+1}^T \mathbf{G} \mathbf{w}_{k+1} - \mathbf{w}_k^T \mathbf{G} \mathbf{w}_k \leq \mathbf{x}_k^T \mathbf{G}_x \mathbf{x}_k, \quad (2)$$

where \mathbf{w}_k is the vector of state values at time k , \mathbf{G} is the diagonal matrix of the corresponding port conductances, \mathbf{x}_k is the vector of source values and \mathbf{G}_x is the diagonal matrix of source port conductances. In the case of zero input, Equation (2) implies that the stored energy will monotonically decrease towards zero. While Equation (2) is guaranteed to hold for passive networks under a fixed step-size, potential issues arise when one varies the step-size which changes the corresponding port resistances and the meaning of the stored energy.

Therefore, when changing the step-size it is not sufficient to just update the sampling rate, port resistances and matrices relating to the \mathcal{R} -type adaptor. It is also necessary (unless the technique from [28] can be used) to convert the wave variables to the representation that is consistent with the network variables and the new step-size before continuing the simulation.

The new step-size variation technique is based on maintaining the consistency of the network variables across step-size changes. Doing so ensures that the stored energy and the instantaneous power remain consistent as well. To do so, after changing the step-size and updating the port resistances, we then convert the stored energy by performing a wave variable transformation. Similar to the development of generalized C matrices in [29], the process is as

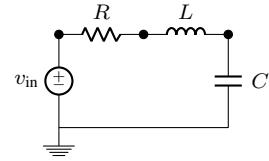


Figure 1: *RLC Circuit Schematic.*

follows to convert wave quantities a, b from sampling period T_k and port resistance R_k to new wave quantities \hat{a}, \hat{b} with sampling period T_{k+1} and port resistance R_{k+1} . From the definition of voltage waves (Equation (1)), voltage and current are given in terms of the original wave quantities and port resistance by:

$$\begin{cases} v = \frac{a + b}{2} \\ i = \frac{a - b}{2R_k} \end{cases}$$

We then define the new wave quantities in terms of voltage, current and the new port resistance by:

$$\begin{cases} \hat{a} = v + R_{k+1}i \\ \hat{b} = v - R_{k+1}i \end{cases}$$

Combining these two sets of equations in matrix form yields the following conversion matrix:

$$\begin{bmatrix} \hat{a} \\ \hat{b} \end{bmatrix} = \frac{1}{2R_k} \begin{bmatrix} R_k + R_{k+1} & R_k - R_{k+1} \\ R_k - R_{k+1} & R_k + R_{k+1} \end{bmatrix} \begin{bmatrix} a \\ b \end{bmatrix} \quad (3)$$

In practice, however, \hat{b} does not get used as the reflected waves of both inductors and capacitors only depend on \hat{a} . Through this transformation, both the power and the energy stored in the unit delays is preserved across step-sizes.

As a demonstration of the validity of this method, it was ran on the example RLC circuit used in [28] (Figure 1). The inductor incident waves from the simulation are shown in Figure 2 computed without correcting the wave quantities (as shown in the original paper), computed with our technique and also computed with a fixed step-size. As seen in the fixed step-size simulation trace, the traces should be exponentially decaying sinusoids. This correct behavior is seen in the trace of the simulation using the proposed technique. The trace is exponentially growing, however, when the wave quantities are not corrected.

To reiterate the process of correcting the wave variables, the following steps must be taken any time the sampling rate is altered:

1. Store the wave variables for all reactive elements and elements that appear above them in shared subtrees.
2. Readapt those network elements with the port resistance corresponding to the new sampling rate.
3. Convert the stored wave variables using the conversion matrix of Equation (3).
4. Update the relevant network elements with the converted wave variables.
5. If the structure contains an \mathcal{R} -type adaptor, update \mathbf{S} , \mathbf{H} , \mathbf{E} , \mathbf{F} , \mathbf{M} and \mathbf{N} using the updated port resistance values.
6. If wave quantities need to be output, convert them to a single unified representation (such as the one relating to audio rate).

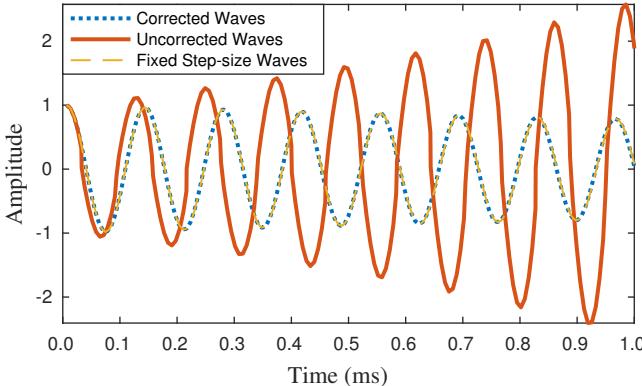


Figure 2: Inductor incident waves from RLC circuit [28].

3. RELAXATION OSCILLATOR CIRCUIT CASE STUDY

3.1. Circuit Description

The circuit to be modeled is called a relaxation oscillator [30] and is also known as an astable multivibrator circuit [31]. It contains the following components: a resistor R_1 , a capacitor C_1 , a voltage divider (R_2 and R_3) and an op amp. Throughout this paper it is assumed that $R_2 = R_3$ and that the op amp operates from rail-to-rail. The circuit schematic is shown in Figure 3.

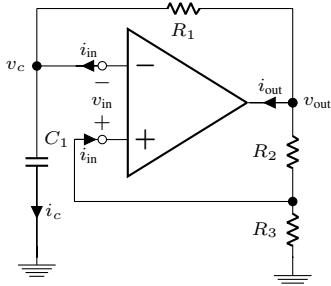


Figure 3: Relaxation Oscillator Schematic.

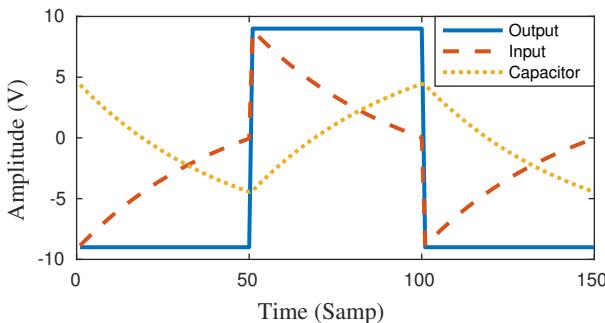


Figure 4: Traces of the op amp input voltage, op amp output voltage and the capacitor voltage.

The basic operation of the circuit is to behave as a comparator whose output swings high or low based on the whether the input voltage to the op amp is above or below a certain threshold value.

Table 2: Relaxation Oscillator Circuit Component Values

R_1	R_2	R_3	C_1	V_{\max}
10 kΩ	10 kΩ	10 kΩ	$1/(2 \log(3)R_1F_0)$ F	9 V

The comparator is configured as a Schmitt trigger by means of the positive feedback path between v_{out} and i_{in} . Assuming that the op amp is powered with a rail voltage of $\pm V_{\max}$ and that it begins with the op amp in positive saturation, the capacitor begins charging towards $+V_{\max}$ with the time constant defined by R_1C_1 . When the capacitor reaches $V_{\max}/2$, the op amp flips over to negative saturation and the capacitor begins discharging towards $-V_{\max}$. This behavior is illustrated in Figure 4 starting at sample number 50.

The period and frequency of oscillation are given by the following equations [31]:

$$\begin{cases} T_0 = 2 \log(3)R_1C_1 \\ F_0 = 1/T_0 \end{cases} \quad (4)$$

The component values used during all simulations are given in Table 2. The capacitor value was used to control the frequency of oscillation and thus set based on the desired simulation frequency F_0 per Equation (4).

3.1.1. Op Amp Modeling

Operational amplifiers are very important devices in audio circuit design. Since, on the component level, op amps are composed of many passive electrical devices and potentially dozens of transistors, they are often modeled during design and simulation using simplified models. Depending on the level of detail needed, various techniques can be used to model op amps. More complicated linear “macromodels” are built up using linear one-ports (resistors, capacitors and dc sources) and two-ports (controlled sources) and can account for a variety of observed op amp behaviors. In the virtual analog context these models may be too complex when only one particular aspect of the model is needed. In the context of the relaxation oscillator the op amp behaves as a comparator. Based on the difference in voltage between the positive and negative terminals of v_{in} the op amp output goes into either positive or negative saturation.

A comparator may be represented ideally as a two-port nonlinear voltage-controlled voltage source (VCVS). This two-port device enforces two network constraints. The first is the nonlinear function $v_{\text{out}} = V_{\max} \text{sgn}(v_{\text{in}})$ and the second is the no input current criteria $i_{\text{in}} = 0$. We call v_{out} , i_{in} the dependent variables and v_{in} , i_{out} the independent variables and let

$$\mathbf{x}_C = [v_{\text{in}} \ i_{\text{out}}]^T \quad \text{and} \quad \mathbf{y}_C = [i_{\text{in}} \ v_{\text{out}}]^T. \quad (5)$$

From this we define the nonlinear relationship as

$$\mathbf{y}_C = f(\mathbf{x}_C) = \begin{bmatrix} 0 \\ V_{\max} \text{sgn}(v_{\text{in}}) \end{bmatrix}. \quad (6)$$

This particular comparator model was chosen to enforce an instantaneous transition between $\pm V_{\max}$ which exactly matches the step discontinuity which the polyBLEP method is designed to work with (see Section 3.6).

3.2. WDF Model of the Circuit

We form the WDF model using the techniques from [3, 4]. Either by visual inspection or via graph theoretic techniques, the schematic is rearranged as in Figure 5a so that all series and parallel subtrees are separated from the two-port nonlinearity by the remaining complex connections which are collectively called the \mathcal{R} -type adapter.

The techniques of [3] are used to determine the scattering behavior of the \mathcal{R} -type adaptor which is represented by the matrix \mathbf{S} . The following system of equations fully describes the relationship between the \mathcal{R} -type adaptor and the two-port nonlinearity:

$$\begin{cases} \mathbf{y}_C = f(\mathbf{x}_C) & \mathbf{E} = \mathbf{C}_{12}(\mathbf{I} + \mathbf{S}_{11}\mathbf{H}\mathbf{C}_{22})\mathbf{S}_{12} \\ \mathbf{x}_C = \mathbf{E}\mathbf{a}_E + \mathbf{F}\mathbf{y}_C & \mathbf{F} = \mathbf{C}_{12}\mathbf{S}_{11}\mathbf{H}\mathbf{C}_{21} + \mathbf{C}_{11} \\ \mathbf{b}_E = \mathbf{M}\mathbf{a}_E + \mathbf{N}\mathbf{y}_C & \mathbf{M} = \mathbf{S}_{21}\mathbf{H}\mathbf{C}_{22}\mathbf{S}_{12} + \mathbf{S}_{22} \\ & \mathbf{N} = \mathbf{S}_{21}\mathbf{H}\mathbf{C}_{21} \end{cases} \quad (7)$$

where

$$\mathbf{S} = \begin{bmatrix} \mathbf{S}_{11} & \mathbf{S}_{12} \\ \mathbf{S}_{21} & \mathbf{S}_{22} \end{bmatrix}, \quad \mathbf{H} = (\mathbf{I} - \mathbf{C}_{22}\mathbf{S}_{11})^{-1}$$

and $\mathbf{a}_E = [a_3 \ a_4 \ a_5 \ a_6]$ is the vector of incident waves from the \mathcal{R} -type adaptor.

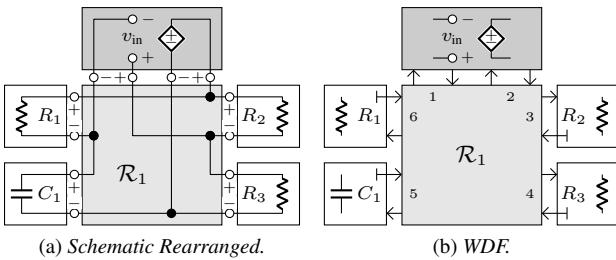


Figure 5: Relaxation Oscillator Rearranged Schematic and WDF.

As discussed in Section 3.1.1, the input variables, output variables and comparator model of the op amp are given by Equations (5) and (6). We then use the definition of voltage waves (Equation (1)) and the signal flow relationship of the nonlinear device quantities:

$$\begin{bmatrix} \mathbf{x}_C \\ \mathbf{b} \end{bmatrix} = \mathbf{C} \begin{bmatrix} \mathbf{y}_C \\ \mathbf{a} \end{bmatrix}$$

where

$$\mathbf{b} = \begin{bmatrix} b_1 \\ b_2 \end{bmatrix} \quad \text{and} \quad \mathbf{a} = \begin{bmatrix} a_1 \\ a_2 \end{bmatrix},$$

to determine the generalized conversion matrix \mathbf{C} [29]:

$$\mathbf{C} = \begin{bmatrix} \mathbf{C}_{11} & \mathbf{C}_{12} \\ \mathbf{C}_{21} & \mathbf{C}_{22} \end{bmatrix} = \left[\begin{array}{cc|cc} -R_1 & 0 & 1 & 0 \\ 0 & -G_2 & 0 & G_2 \\ \hline -2R_1 & 0 & 1 & 0 \\ 0 & 2 & 0 & -1 \end{array} \right]$$

where R_1 and G_2 refer to the port resistance and port conductance of ports 1 and 2, respectively.

3.2.1. Numerical Solution of the Nonlinearity

In [5] it was shown how, in general, nonlinearities can be solved using iterative techniques. In the case of this particular circuit,

however, the nonlinear relationship is modeled as a step discontinuity for which iterative techniques struggle to converge quickly if at all. Through inspection of the WDF equations representing the nonlinearity it was found that they are formed in such a way that a quasi-analytic solution can be determined.

From Equation (7), we get the following complete description of the nonlinear system of equations to be solved:

$$\mathbf{x}_C = \mathbf{E}\mathbf{a}_E + \mathbf{F}\mathbf{f}(\mathbf{x}_C) \quad (8)$$

where, for this particular circuit, \mathbf{E} is a 2×4 matrix and \mathbf{F} is a 2×2 matrix. We use domain knowledge of the circuit to simplify the nonlinear system of equations. Since a_3, a_4 and a_6 are the incident waves relating to the resistors, which are always zero, and since i_{in} is also zero, the following simplified expression is obtained:

$$\begin{bmatrix} v_{in} \\ i_{out} \end{bmatrix} = \begin{bmatrix} e_{13}a_5 + f_{12}V_{max} \operatorname{sgn}(v_{in}) \\ e_{23}a_5 + f_{22}V_{max} \operatorname{sgn}(v_{in}) \end{bmatrix}.$$

The values of v_{out} and i_{out} are dependent on the value and sign of v_{in} and the constant values from the \mathbf{E} and \mathbf{F} matrices. Therefore, the nonlinearity can be solved algorithmically using the following pseudocode:

Algorithm 1: Nonlinearity Solver

```

1  if  $n - 1 < 1$ 
2   initialize  $\operatorname{sgn}_{vin}$  to  $\pm 1$ 
3  else
4    $\operatorname{sgn}_{vin} \leftarrow \operatorname{sgn}(v_{in}[n - 1])$ 
5  end if
6
7   $v_{in}[n] \leftarrow e_{13}a_5[n] + f_{12}V_{max} \cdot \operatorname{sgn}_{vin}$ 
8   $\mathbf{x}_C \leftarrow \mathbf{E}\mathbf{a}_E[n] + \mathbf{F} \cdot [0 \ V_{max} \operatorname{sgn}(v_{in}[n])]^T$ 
9
10 if  $\mathbf{E}\mathbf{a}_E[n] + \mathbf{F} \cdot [0 \ V_{max} \operatorname{sgn}_{vin}]^T - \mathbf{x}_C \neq 0$ 
11   $\operatorname{sgn}_{vin} \leftarrow -\operatorname{sgn}_{vin}$ 
12   $v_{in}[n] \leftarrow e_{13}a_5[n] + f_{12}V_{max} \cdot \operatorname{sgn}_{vin}$ 
13   $\mathbf{x}_C \leftarrow \mathbf{E}\mathbf{a}_E[n] + \mathbf{F} \cdot [0 \ V_{max} \operatorname{sgn}(v_{in}[n])]^T$ 
14 end if
15
16  $\mathbf{y}_C \leftarrow [0 \ V_{max} \operatorname{sgn}(v_{in}[n])]^T$ 
17 return  $\mathbf{x}_C, \mathbf{y}_C$ 

```

The algorithm determines the correct values for v_{in} and v_{out} by guessing the sign of v_{in} (using the sign of the previous value of v_{in}) and checking whether or not the result is consistent with Equation (8). If it is consistent, then that is the correct value of v_{in} and it can be used to calculate \mathbf{x}_C and \mathbf{y}_C . Otherwise, the sign is flipped and then the only other possible solution is calculated.

3.3. Problems Resulting from the Fixed Step-size Simulation

3.3.1. Simulation Error

By running the simulation with a fixed step-size at a variety of frequencies and then analyzing the spectrum of the output voltage of the op amp it was determined that a noticeable amount of frequency error was occurring in the simulation. Upon more detailed analysis of the circuit output, it was determined that the majority of the error was occurring around the location of the discontinuities in the component voltages.

As shown in Figure 6, there are two related error scenarios. In the first, as seen in Figure 6a, it is possible that the error leads to the simulation running at a lower frequency than desired due to an overshoot at the first transition of the capacitor voltage. In the second scenario, Figure 6b, the opposite situation occurs and

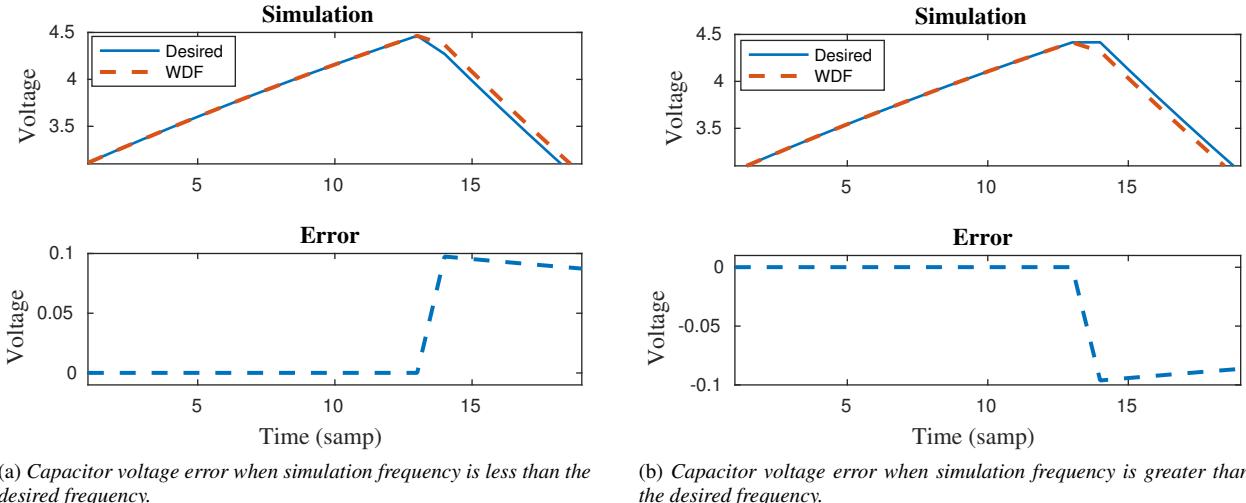


Figure 6: Capacitor voltage error scenarios.

the simulation runs at a higher frequency than desired. In both Figure 4 and Figure 6, the simulation was run at 44.1 kHz sampling rate with a desired period of 101 samples with the average period of the simulations being 102 and 100 for Figures 6a and 6b, respectively.

3.3.2. Simulation Frequency

It was determined experimentally that the overshoot/undershoot error identified in Section 3.3.1 causes the fundamental period T_0 of the simulation to become “phase-locked” to an even integer length period. In the lucky case that T_0 was an even integer, it was found that the fundamental period of simulation \hat{T}_0 was equal to T_0 and so the simulation did not contain any frequency error. For any other choice of T_0 , however, two possible values of \hat{T}_0 were observed:

$$\hat{T}_0^+ = \text{floor}(T_0) - (\text{floor}(T_0) \bmod 2) \quad (9a)$$

$$\hat{T}_0^- = \text{ceil}(T_0) + (\text{ceil}(T_0) \bmod 2) \quad (9b)$$

The phase-locking of the frequency that occurred from these errors occurred immediately such that \hat{T}_0 was the observed period of oscillation for every period of the simulation beginning with the first complete period.

Since it is not possible to derive an analytic expression for the actual simulation frequency error, we derive an expression for the worst case error in order to investigate the general behavior of the maximum possible frequency error. The worst case frequency error curve in cents can be derived using the following formula:

$$\Delta F_{\max} = 1200 \max \left(\left| \log_2 \left(\frac{\hat{F}_0^-}{F_0} \right) \right|, \left| \log_2 \left(\frac{\hat{F}_0^+}{F_0} \right) \right| \right) \quad (10)$$

where

$$\begin{cases} \hat{F}_0^+ = F_s / \hat{T}_0^+ \\ \hat{F}_0^- = F_s / \hat{T}_0^- \end{cases}$$

where F_s is the sampling rate. Equation (10) gives the worst case frequency error estimation as determined by the largest possible rounding error occurring in the simulation period.

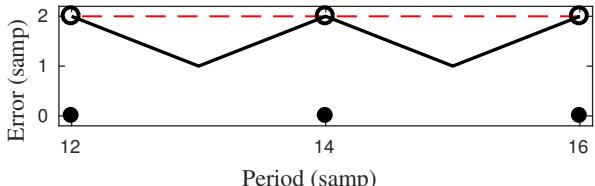
A portion of the resulting curve is shown in Figure 7a. The minimum points on the curve corresponding to an error of one sample occur at odd integer periods which are equal distance from the nearest even integer periods. From those points, the error curve increases in either direction approaching a theoretical limit of two samples (indicated by open circles). Finally, at the even integer period values, there is no error in the period which is denoted by the solid black dots. The dotted red line shows the maximum error limit which was used in the computation of the frequency error curves shown in Figure 7b.

Hypothetically assuming that a frequency error of six cents is imperceptible at all frequencies, the worst case error curves (Figure 7b) given an idea of the level of oversampling necessary for the fixed step-size simulation error to fall below that level. It is also evident that the maximum possible simulation error is frequency dependent and clearly increasing with frequency. Thus, the amount of oversampling necessary to mitigate the worst case error is coupled to the desired frequency of simulation.

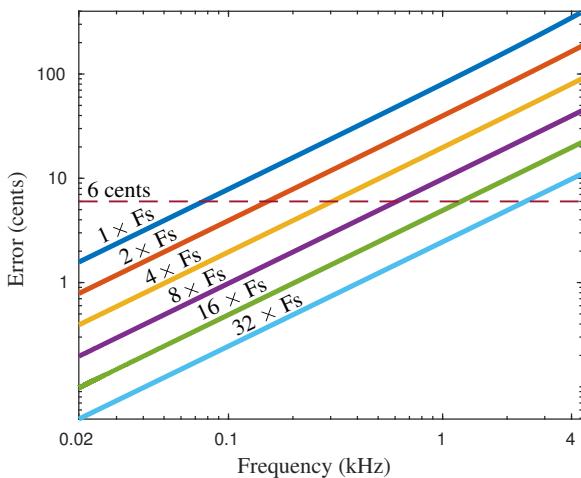
3.4. Variable Step-size Simulation of the Oscillator WDF

Variable step-size ODE solvers typically utilize an automatic step-size selection algorithm [28, 32]. The step-size is determined by estimating the amount of simulation error that would result from continuing to use the current step-size and then either increasing or decreasing the step-size based on the result of that error calculation.

Since the majority of the simulation error in the fixed step-size simulation of the relaxation oscillator occurs in the region of the discontinuities in the simulation voltages and the accuracy of the simulation frequency is coupled to the amount of oversampling performed, it would be highly desirable to limit oversampling to just those regions and to also have control over the rate of oversampling. This would have the benefit of reducing the computational cost as compared to just oversampling the simulation with a



(a) Local Maximum Possible Period Error in Samples.



(b) Maximum Frequency Error in Cents for Different Levels of Upsampling.

Figure 7: Worst Case Period and Frequency Error.

fixed step-size and would allow the user to control the amount of maximum possible frequency error in the simulation. To accomplish this, domain knowledge of the circuit is used to predict the occurrence of discontinuities and oversampling is performed only in that region.

3.4.1. Detection of Discontinuity Regions

As seen in Figure 4, the step discontinuities in the op amp output voltage occur whenever the op amp input voltage crosses zero which also corresponds to the point where the capacitor changes state. Thus, we use the input voltage and an extrapolation formula based on its theoretical behavior to predict the occurrences of zero-crossings and, therefore, of the step discontinuities.

This is done using the following extrapolation formula:

$$\hat{v}_{in}[n+1] = - \left(\frac{\text{sgn}(v_{in}[n]) V_{max}}{2} + v_{in}[n] \right) \left(1 - e^{\frac{-T}{R_1 C_1}} \right) + v_{in}[n], \quad (11)$$

where v_{in} is the input voltage of the op amp and sgn is the signum function.

The formula represents the theoretical charge and discharge exhibited in the input voltage of the op amp. Therefore, the detection of the discontinuities is expected to be highly accurate so that the simulation can be ran at the higher sampling rate for a limited number of samples.

3.4.2. Extrapolation and Variable Step-size Algorithm

Algorithm 2 gives pseudocode for the extrapolation and step-size modification routine used in the simulation. The algorithm detects if a zero-crossing will happen in the op amp input voltage during the next two future samples. If so, the system is updated to the higher sampling rate and runs the equivalent of three original rate samples at the higher rate. Only the higher rate samples corresponding to lower rate samples are stored (with wave quantities converted to the lower rate representation) so that the output of the simulation always corresponds to the base sampling rate. Finally, the system returns to running at the original sampling rate until the next discontinuity is detected. It is enforced that the higher sampling rate is an integer multiple of the base sampling rate so that interpolation can be avoided during decimation.

Algorithm 2: Extrapolation/Step-Size Algorithm

```

1  vin_nml <-  $v_{in}[n - 1]$ 
2  extrapolate  $v_{in\_n}$  from  $v_{in\_nml}$  using Equation (11)
3  extrapolate  $v_{in\_np1}$  from  $v_{in\_n}$  using Equation (11)
4  if  $v_{in\_n} \cdot v_{in\_np1} \leq 0$ :
5    nSamps = 3
6    K <- the oversampling factor
7    k <- nSamps · K
8    Update system to new sampling rate
9    for  $i = 1 : k$ 
10      iterate system 1 sample
11      if  $i \bmod K = 0$ 
12        save system values as  $(n + i/K - 1)$ -th values
13        convert saved wave quantities using Eq. (3)
14      end if
15    Update system back to original sampling rate
16  end if

```

It should be noted that anti-aliasing is not performed at the decimation stage. This is because the anti-alias filtering of v_{out} will change the dynamics of the system, through modification of the output voltage, affecting the energy stored in the capacitor therefore affecting the frequency of oscillation. In order for the simulation frequency to be as accurate as possible, which corresponds to the simulation error being as small as possible, we allow the aliasing to occur within the simulation. How to suppress the aliasing inherent in the output voltage is the subject of Section 3.6.

3.5. Variable Step-size Method with the Relaxation Oscillator Simulation

In Section 2, a new technique was introduced for the variable step-size simulation of a WDF. The need for the technique arose from the fact that the relaxation oscillator is a nonlinear circuit and also does not contain any inductors. Therefore, it was unclear how the technique from [28] could be implemented with this particular circuit. If the Gauss method is equivalent to the trapezoidal rule for any circuit that does not contain inductors, then variable step-size simulation without using the technique presented in Section 2 will lead to large fluctuations in the energy stored in the delay registers of the capacitor.

This can be clearly seen in Figure 8 which shows the incident waves and stored energy of the capacitor from three different simulations. The desired fundamental period of the simulation was 100.25 samples so that a fixed step-size simulation run at 8× oversampling would yield an output waveform with no frequency error. The corrected and uncorrected traces show the incident waves of the capacitor when trapezoidal discretization is used with and without the technique proposed in Section 2. The figure shows that large fluctuations in stored energy occur when the step-size

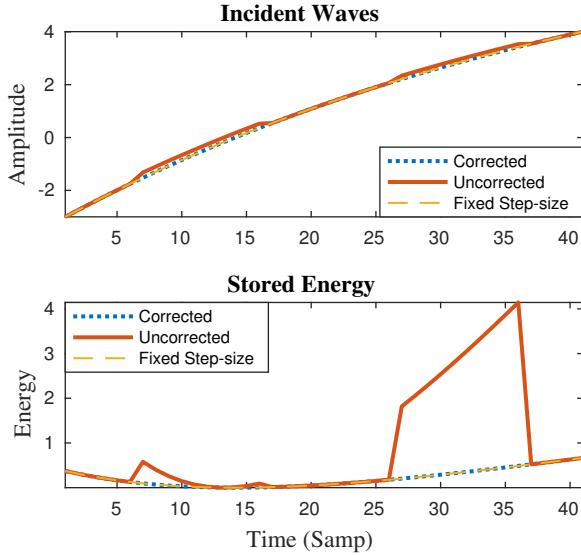


Figure 8: Incident waves and the stored energy in the capacitor for the relaxation oscillator circuit run with a fixed step-size and different variable step-size methods.

is changed without using the new technique. Conversely, when the new technique is used, the incident waves and energy perfectly match the corresponding traces from the fixed step-size simulation.

3.6. Aliasing Suppression in the Output Signal

Due to the use of the ideal comparator VCVS characteristic to model the op amp behavior, the output voltage of the op amp is a square wave that instantaneously switches between $\pm V_{\max}$. Thus, the presence of step discontinuities in the output waveform results in a similar amount of aliasing as arises from trivial synthesis of a square waveform. To minimize the aliasing, we use the polyBLEP [24] antialiasing technique.

In the original method, the square wave is generated using a phase accumulator which allows for an exact calculation of the location of the discontinuities. This fractional delay value is then used to determine the exact placement of samples of the polyBLEP residual function which are added to one or more samples of the output waveform before and after each discontinuity.

The WDF simulation of the example circuit in this paper does not lend itself to a simple determination of the phase of the op amp output voltage. We use a method based on the extrapolation technique developed in Section 3.4.1 for determining the fractional delay of the occurrence of each discontinuity. During the higher rate processing, after the polarity of the input voltage has flipped, a version of Equation (11) is used to determine the fractional delay in terms of the higher sampling rate:

$$dx_K = \frac{1}{K} + R_1 C_1 K F_s \log \left(\frac{\operatorname{sgn}(v_{in}[i-1]) V_{\max}}{\operatorname{sgn}(v_{in}[i-1]) V_{\max} + 2v_{in}[i-1]} \right).$$

From this, the fractional delay is expressed in terms of the original sampling rate by adding the distance to the next lower rate sample:

$$d = \frac{(K-i) \bmod K}{K}$$

to the appropriately scaled fractional delay value:

$$dx = \frac{dx_K}{K} + d$$

where i and K are as defined in Algorithm 2.

Depending on the chosen polyBLEP method, a number of v_{out} samples before and after the step discontinuity have to be altered to include the polyBLEP residual amount for the chosen scheme. This is done exactly the same way as in the original paper.

Due to the effect that the polyBLEP residual has on the dynamics of the simulations, the polyBLEP residual is applied to a separate output signal path so that it does not affect the internal dynamics of the simulation. Thus, the modified output voltage is not fed back through the feedback path of the circuit.

Third-order Lagrange polynomial polyBLEP residual functions were used to reduce the aliasing for a simulation run with a fundamental frequency of 100π at a sampling rate of 44.1 kHz with a higher rate of $16 \times F_s$ for a frequency error of $9.7e-4$ cents. The result of applying the polyBLEP residual function to the output voltage is shown in Figure 9 where the upper image shows the spectrum of the original output voltage and the lower image shows the spectrum of the output voltage with the polyBLEP residual applied. Clearly, the aliasing in the spectrum of the output voltage with polyBLEP residuals applied is falling off at a much steeper rate than the original waveform's spectrum. It is also evident that the amplitude of the higher harmonics in the spectrum of the alias suppressed waveform are falling off more quickly than the theoretical amplitudes. This can be fixed using a high shelf equalization filter. Example filter coefficients are given in [24].

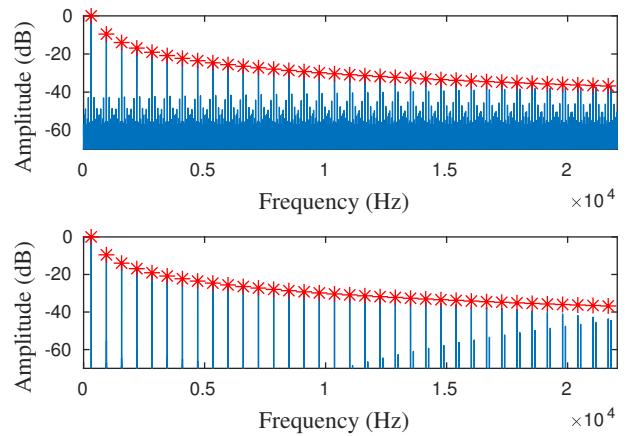


Figure 9: Spectrums of the output voltages of a simulation run without polyBLEP (top) and with (bottom). The theoretical amplitude of the harmonics are indicated by asterisks.

4. CONCLUSION

In this paper a new technique was introduced for the variable step-size simulation of wave digital filters. A conversion of the wave variable quantities of reactive network elements is performed any time that the sampling rate is changed. Our method is novel in that it ensures that the network variables are preserved which enforces that the energy stored in the reactive elements is preserved as well.

In [28], the variable step-size simulation of WDFs discretized with the trapezoidal rule was identified as an open problem. By identifying the relationship between wave variables and network variables it has been shown how the correct behavior can be achieved in a WDF model which directly discretizes the reactive elements and does not use network component equivalencies. The validity of the technique holds regardless of the discretization technique being used to discretize the reactive network elements. With WDFs, however, it is well known that passive discretization techniques must be used.

Through the case study of a relaxation oscillator circuit, the need for the new variable step-size technique was demonstrated. The variable step-size WDF simulation technique was used to mitigate frequency error related to the discontinuities in the device voltages. An extrapolation formula was used to estimate the occurrence of future zero crossings in the input voltage of the op amp. This allowed the system to be switched to the higher simulation rate for a short duration of three base rate samples encompassing the occurrence of the step discontinuity in the op amp output voltage. It was also demonstrated how physical modeling and antialiasing techniques can be combined to yield an accurate and alias-reduced simulation of the relaxation oscillator circuit. The polyBLEP antialiasing technique was used and the oversampling from the variable step-size technique had the additional benefit of improving the estimate of the fractional delay thus further improving the accuracy of the polyBLEP method.

5. REFERENCES

- [1] A. Fettweis, “Wave digital filters: Theory and practice,” *Proc. IEEE*, vol. 74, no. 2, pp. 270–327, Feb. 1986.
- [2] 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.
- [3] 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. Digit. Audio Effects*, Trondheim, Norway, Nov. 30 – Dec. 3 2015, pp. 387–394.
- [4] 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. Digit. Audio Effects*, Trondheim, Norway, November 30 – December 3 2015, pp. 379–386.
- [5] 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. Digit. Audio Effects*, Brno, Czech Republic, Sept. 5–9 2016, pp. 279–286.
- [6] 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 Trans. Audio, Speech, Language Process.*, vol. 18, no. 4, pp. 728–737, 2010.
- [7] M. Holters and U. Zölzer, “A generalized method for the derivation of non-linear state-space models from circuit schematics,” in *Proc. 23rd European Signal Process. Conf.*, Nice, France, Aug. 31 – Sept. 4 2015, pp. 1073–1077.
- [8] M. Holters and U. Zölzer, “A k-d tree based solution cache for the non-linear equation of circuit simulations,” in *Proc. 24th European Signal Process. Conf.*, Budapest, Hungary, Aug. 29 – Sept. 2 2016, pp. 1028–1032.
- [9] V. Duindam, A. Macchelli, S. Stramigioli, and H. Bruyninckx, *Modeling and Control of complex Physical Systems: The Port-Hamiltonian Approach*, Springer Berlin Heidelberg, 2009.
- [10] 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.
- [11] T. Schwerdtfeger and A. Kummert, “A multidimensional signal processing approach to wave digital filters with topology-related delay-free loops,” in *IEEE Int. Conf. Acoust., Speech, Signal Process.*, Florence, Italy, May 4–9 2014, pp. 389–393.
- [12] S. Petrausch and R. Rabenstein, “Wave digital filters with multiple nonlinearities,” in *Proc. 12th European Signal Process. Conf.*, Vienna, Austria, Sept. 6–10 2004, vol. 12, pp. 77–80.
- [13] T. Schwerdtfeger and A. Kummert, “Newton’s method for modularity-preserving multidimensional wave digital filters,” in *IEEE 9th Int. Workshop Multidimensional (nD) Syst. (nDS)*, Vila Real, Portugal, Sept. 7–9 2015.
- [14] R. C. D. 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.
- [15] D. T. Yeh and J. O. Smith, “Simulating guitar distortion circuits using wave digital and nonlinear state-space formulations,” in *Proc. 11th Int. Conf. Digit. Audio Effects*, Espoo, Finland, Sept. 1–4 2008, pp. 19–26.
- [16] 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. 139 Int. Audio Eng. Soc.*, New York, NY, Oct. 29 – Nov. 1 2015.
- [17] 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. Digit. Audio Effects*, Brno, Czech Republic, Sept. 5–9 2016, pp. 263–270.
- [18] 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.
- [19] 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 14–19 2006, vol. 5.
- [20] S. D’Angelo, J. Pakarinen, and V. Välimäki, “New family of wave-digital triode models,” *IEEE Trans. Audio, Speech, Language Process.*, vol. 21, no. 2, pp. 313–321, Feb. 2013.
- [21] K. J. Werner, W. R. Dunkel, M. Rest, M. J. Olsen, and J. O. Smith III, “Wave digital filter modeling of circuits with operational amplifiers,” in *Proc. 24th European Signal Process. Conf.*, Budapest, Hungary, Aug. 29 – Sept. 2 2016, pp. 1033–1037.
- [22] T. Stilson and J. Smith, “Alias-free digital synthesis of classic analog waveforms,” in *Proc. Int. Comput. Music Conf.*, Hong Kong, China, Aug. 19–24 1996, pp. 332–335.
- [23] Eli Brandt, “Hard sync without aliasing,” in *Proc. Int. Comput. Music Conf.*, Havana, Cuba, Sept. 17–23, 2001, pp. 365–368.
- [24] Vesa Välimäki, Jussi Pekonen, and Juhan Nam, “Perceptually informed synthesis of bandlimited classical waveforms using integrated polynomial interpolation,” *J. Acoust. Soc. America*, vol. 131, no. 1, pp. 974–986, 2012.
- [25] F. Esqueda and V. Välimäki, “Rounding corners with BLAMP,” in *Proc. 19th Int. Conf. Digit. Audio Effects*, Brno, Czech Republic, Sept. 5–9 2016, pp. 121–128.
- [26] J. D. Parker, V. Zavalishin, and E. Le Bivic, “Reducing the aliasing of nonlinear waveshaping using continuous-time convolution,” in *Proc. 19th Int. Conf. Digit. Audio Effects*, Brno, Czech Republic, Sept. 5–9 2016, pp. 137–144.
- [27] S. Bilbao, F. Esqueda, J. D. Parker, and V. Välimäki, “Antiderivative antialiasing for memoryless nonlinearities,” *IEEE Signal Process. Lett.*, 2017, to be published, DOI 10.1109/LSP.2017.2675541.
- [28] Dietrich Fränken and Karlheinz Ochs, “Automatic step-size control in wave digital simulation using passive numerical integration methods,” *AEU – Int. J. Electron. Commun.*, vol. 58, no. 6, pp. 391–401, 2004.
- [29] K. J. Werner, M. J. Olsen, M. Rest, and J. D. Parker, “Generalizing root variable choice in wave digital filters with grouped nonlinearities,” in *Proc. 20th Int. Conf. Digit. Audio Effects*, Edinburgh, U.K., Sept. 5–9 2017.
- [30] Paul Horowitz and Winfield Hill, *The Art of Electronics*, Cambridge University Press, Cambridge, UK, second edition, 1980.
- [31] Adel S. Sedra and Kenneth C. Smith, *Microelectronic Circuits*, Oxford University Press, Inc., New York, NY, USA, 6th edition, 2009.
- [32] R. L. Burden and D. J. Faires, *Numerical Analysis*, Brooks/Cole, Cengage Learning, Boston, MA, 9th edition, 2010.

ON THE DESIGN AND USE OF ONCE-DIFFERENTIABLE HIGH DYNAMIC RESOLUTION ATOMS FOR THE DISTRIBUTION DERIVATIVE METHOD

Nicholas Esterer

SPCL*

McGill University

Montreal, Quebec, Canada

nicholas.estrer@mail.mcgill.ca

Philippe Depalle

SPCL

McGill University

Montreal, Quebec, Canada

philippe.depalle@mcgill.ca

ABSTRACT

The accuracy of the Distribution Derivative Method (DDM) [1] is evaluated on mixtures of chirp signals. It is shown that accurate estimation can be obtained when the sets of atoms for which the inner product is large are disjoint. This amounts to designing atoms with windows whose Fourier transform exhibits low sidelobes but which are once-differentiable in the time-domain. A technique for designing once-differentiable approximations to windows is presented and the accuracy of these windows in estimating the parameters of sinusoidal chirps in mixture is evaluated.

1. INTRODUCTION

Additive synthesis using a sum of sinusoids plus noise is a powerful model for representing audio [2], allowing for the easy implementation of many manipulations such as time-stretching [3] and timbre-morphing [4]. In these papers, [2–4] the phase evolution of the sinusoid is assumed linear over the analysis frame, only the phase and frequency of the sinusoids at these analysis points are used to fit a plausible phase function after some the analysis points are connected to form a partial [5]. Recently, there has been interest in using higher-order phase functions [6] as the estimation of their parameters has been made possible by a new set of techniques of only moderate computational complexity using signal derivatives [7]. The use of higher-order phase models allows for accurate description of highly modulated signals, for example in the analysis of birdsong [8]. The frequency modulation information has also been used in the regularization of mathematical programs for audio source separation [9].

The sinusoidal model approximating signal s typically considered is

$$\tilde{s}(t) = \exp(a_0 + \sum_{q=1}^Q a_q t^q) + \eta(t) \quad (1)$$

where \tilde{s} is the approximating signal, t the variable of time, the $a_q \in \mathbb{C}$ coefficients of the argument's polynomial, and $\eta(t)$ white Gaussian noise. Although this technique can be extended to describe a single sinusoid of arbitrary complexity simply by increasing Q , it remains essential to consider signals featuring a sum of P such components, whether they represent the harmonic structure of a musical sound or the union of partials resulting from a mixture of multiple signal sources (e.g., recordings of multiple speakers or performers), i.e.,

$$x(t) = \sum_{p=1}^P x_p(t) + \eta(t) \quad (2)$$

with

$$x_p(t) = \exp(a_{p,0} + \sum_{q=1}^Q a_{p,q} t^q) \quad (3)$$

As regards the design and evaluation of signal-derivatives analysis techniques, previous work has generally assumed signals containing a single component, i.e., $P = 1$ or assumed the influence of other components to be negligible. Later we will refine when this assumption can be made. In [10] the authors provide a comprehensive evaluation of various signal-derivatives analysis methods applied to a single-component signal. In [11] the extent to which two components in mixture can corrupt estimations of the frequency slope ($\Im\{a_{0,2}\}$ and $\Im\{a_{1,2}\}$) is investigated in the context of the reassignment method, one of the signal-derivatives techniques, but the corruption of the other parameters is not considered.

In this paper, we revisit the quality of signal-derivatives estimation of *all* the a_q when analyzing a *mixture* of components. We focus on the DDM [1] analysis method for its convenience as it can simply be considered as an atomic decomposition (see Sec. 2), and does not require computing derivatives of the signal to be analysed.

The DDM does, however, require a once-differentiable analysis window. As we are interested in windows with lower sidelobes in order to better estimate parameters of sinusoidal chirp signals in mixture, we seek windows that combine these two properties. For this, a technique to design once-differentiable approximations to arbitrary symmetrical windows is proposed and presented along with a design example for a high-performance window. Finally we evaluate the performance of various once-differentiable windows in estimating the parameters a_q .

2. ESTIMATING THE PARAMETERS a_q

We will now show briefly how the DDM can be used to estimate the a_q . Based on the theory of distributions [12], the DDM makes use of “test functions” or atoms ψ . These atoms must be once differentiable with respect to time variable t and be non-zero only on a finite interval $[-\frac{L_t}{2}, \frac{L_t}{2}]$. First, we define the inner product

$$\langle x, \psi \rangle = \int_{-\infty}^{\infty} x(t) \bar{\psi}(t) dt \quad (4)$$

and the operator

$$\mathcal{T}^\alpha : (\mathcal{T}^\alpha x)(t) = t^\alpha x(t) \quad (5)$$

Consider the weighted signal

$$f(t) = x(t) \bar{\psi}(t) \quad (6)$$

* Sound Processing and Control Laboratory

differentiating with respect to t we obtain

$$\begin{aligned} \frac{df}{dt}(t) &= \frac{dx}{dt}(t)\bar{\psi}(t) + x(t)\frac{d\bar{\psi}}{dt}(t) = \\ &\quad \left(\sum_{q=1}^Q qa_q t^{q-1} \right) x(t)\bar{\psi}(t) + x(t)\frac{d\bar{\psi}}{dt}(t) \end{aligned} \quad (7)$$

Because ψ is zero outside of the interval $[-\frac{L_t}{2}, \frac{L_t}{2}]$, integrating $\frac{df}{dt}(t)$ we obtain

$$\begin{aligned} \int_{-\infty}^{\infty} \frac{df}{dt}(t) dt &= \\ \sum_{q=1}^Q qa_q \int_{-\frac{L_t}{2}}^{\frac{L_t}{2}} t^{q-1} x(t)\bar{\psi}(t) dt + \left\langle x, \frac{d\bar{\psi}}{dt} \right\rangle &= 0 \end{aligned} \quad (8)$$

or, using the operator \mathcal{T}^α ,

$$\sum_{q=1}^Q qa_q \langle \mathcal{T}^{q-1} x, \bar{\psi} \rangle = - \left\langle x, \frac{d\bar{\psi}}{dt} \right\rangle \quad (9)$$

Estimating coefficients a_q , $1 < q \leq Q$, simply requires R atoms ψ_r with $R \geq Q$ to solve the linear system of equations

$$\sum_{q=1}^Q qa_q \langle \mathcal{T}^{q-1} x, \bar{\psi}_r \rangle = - \left\langle x, \frac{d\bar{\psi}_r}{dt} \right\rangle \quad (10)$$

for $1 \leq r \leq R$.

To estimate a_0 we rewrite the signal we are analysing as

$$x(t) = \exp(a_0)\gamma(t) + \epsilon(t) \quad (11)$$

where $\epsilon(t)$ is the error signal, the part of the signal that is not explained by our model, and $\gamma(t)$ is the part of the signal whose coefficients have already been estimated, i.e.,

$$\gamma(t) = \exp \left(\sum_{q=1}^Q a_q t^q \right) \quad (12)$$

Computing the inner product $\langle x, \gamma \rangle$, we have

$$\langle x, \gamma \rangle = \langle \exp(a_0)\gamma, \gamma \rangle + \langle \epsilon, \gamma \rangle \quad (13)$$

The inner product between ϵ and γ is 0, by the orthogonality principle [13, ch. 12]. Furthermore, because $\exp(a_0)$ does not depend on t , we have

$$\langle x, \gamma \rangle = \exp(a_0) \langle \gamma, \gamma \rangle \quad (14)$$

so we can estimate a_0 as

$$a_0 = \log(\langle x, \gamma \rangle) - \log(\langle \gamma, \gamma \rangle) \quad (15)$$

As will be seen in subsequent sections, the DDM typically involves taking the discrete Fourier transform (DFT) of the signal windowed by both an everywhere once-differentiable function of finite support (e.g., the Hann window) and this function's derivative. A small subset of atoms corresponding to the peak bins in the DFT are used in Eq. 10 to solve for the parameters a_q .

3. ESTIMATING THE $a_{p,q}$ OF P COMPONENTS

We examine how the mixture model influences the estimation of the $a_{p,q}$ in Eq. 3. Consider a mixture of P components. If we define the weighted signal sum

$$g(t) = \sum_{p=1}^P x_p(t)\bar{\psi}(t) = \sum_{p=1}^P f_p(t) \quad (16)$$

and substitute g for f in Eq. 7 we obtain

$$\begin{aligned} \sum_{p=1}^P \int_{-\frac{L_t}{2}}^{\frac{L_t}{2}} \frac{df_p}{dt}(t) dt &= 0 = \\ \sum_{p=1}^P \left(\sum_{q=1}^Q qa_{p,q} \langle \mathcal{T}^{q-1} x_p, \bar{\psi} \rangle + \left\langle x_p, \frac{d\bar{\psi}}{dt} \right\rangle \right) \end{aligned} \quad (17)$$

From this we see if $\langle \mathcal{T}^{q-1} x_p, \bar{\psi}_r \rangle$ and $\left\langle x_p, \frac{d\bar{\psi}_r}{dt} \right\rangle$ are small for all but $p = p^*$ and a subset of R atoms¹, we can simply estimate the parameters $a_{p^*,q}$ using

$$\sum_{q=1}^Q qa_{p^*,q} \langle \mathcal{T}^{q-1} x_{p^*}, \bar{\psi}_r \rangle = - \left\langle x_{p^*}, \frac{d\bar{\psi}_r}{dt} \right\rangle \quad (18)$$

for $1 \leq r \leq R$. To compute $a_{p^*,0}$ we simply use

$$\gamma_{p^*}(t) = \exp \left(\sum_{q=1}^Q a_{p^*,q} t^q \right) \quad (19)$$

in place of γ in Eq. 15.

4. DESIGNING THE ψ_R

In practice, an approximation of Eq. 4 is evaluated using the DFT on a signal x that is properly sampled and so can be evaluated at a finite number of times nT with $n \in [0, N-1]$ and T the sample period in seconds. In this way, the chosen atoms $\psi_\omega(t)$ are the products of the elements of the Fourier basis and an appropriately chosen window w that is once differentiable and finite, i.e.,

$$\psi_\omega(t) = w(t) \exp(-j\omega t) \quad (20)$$

Defining $N = \frac{L_t}{T}$ and angular frequency at bin r as $\omega_r = 2\pi \frac{r}{N}$, the approximate inner product is then

$$\langle x, \psi_\omega \rangle \approx \sum_{n=0}^{N-1} x(Tn)w(Tn) \exp(-2\pi j r \frac{n}{N}) \quad (21)$$

i.e., the definition of the DFT of a windowed signal². The DFT is readily interpreted as a bank of bandpass filters centred at normalized frequencies $\frac{r}{N}$ and with frequency response described by

¹The notation x^* will mean the value of the argument x maximizing or minimizing some function.

²Notice however that this is an approximation of the inner product and should not be interpreted as yielding the Fourier series coefficients of a properly sampled signal x periodic in L_t . This means that other evaluations of the inner product that yield more accurate results are possible. For example, the analytic solution is possible if x is assumed zero outside of $[-\frac{L_t}{2}, \frac{L_t}{2}]$ (the ψ are in general analytic). In this case the samples of x are convolved with the appropriate interpolating sinc functions and the integral of this function's product with ψ is evaluated.

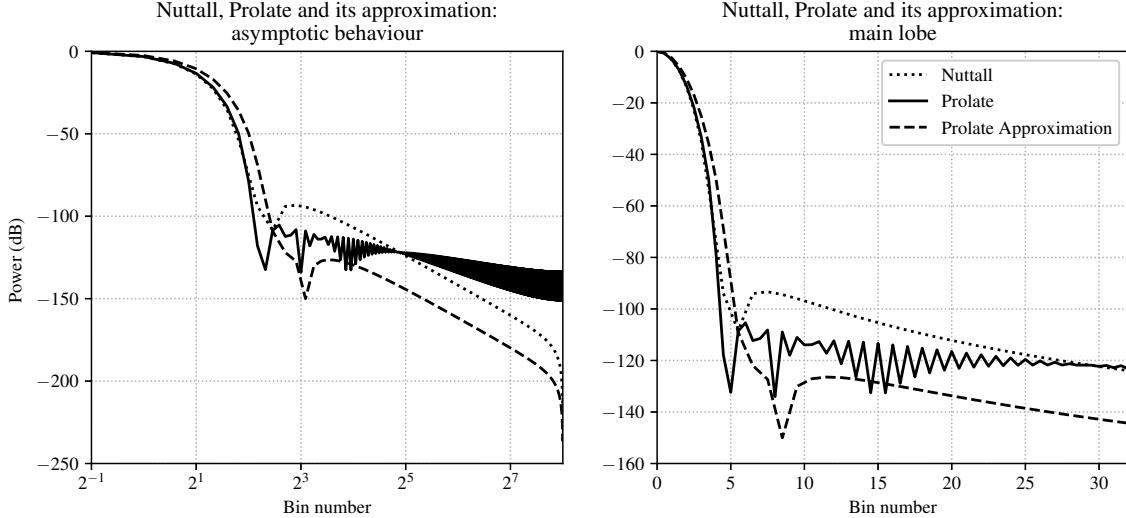


Figure 1: Comparing the main-lobe and asymptotic power spectrum characteristics of the continuous 4-term Nuttall window, the digital prolate window with $W = 0.008$, and the continuous approximation to the digital prolate window.

the Fourier transform of modulated w [14]. Therefore choosing ψ amounts to a filter design problem under the constraints that the impulse response of the filter be differentiable in t and finite. To minimize the influence of all but one component, granted the components's energy concentrations are sufficiently separated in frequency, we desire impulse responses whose magnitude response gives maximum out-of-band rejection or equivalently, windows whose Fourier transform exhibits the lowest sidelobes.

In all the publications reviewed on the DDM for this paper, the window used was the Hann window which is once-differentiable everywhere in the time-domain. In [11], a publication on the re-assignment method, other windows than the Hann are considered but these windows must be twice-differentiable. Nuttall [15] has designed windows with lower sidelobes than the canonical Hann window which are everywhere at least once-differentiable. It is also possible to design approximations to arbitrary symmetrical window functions using harmonically related cosines, as is discussed in the following section.

5. DIFFERENTIABLE APPROXIMATIONS TO WINDOWS

A differentiable approximation to a symmetrical window can be designed in a straightforward way. In [16] and [17] it is shown how to design optimal windows of length N samples using a linear combination of M harmonically related cosines

$$\tilde{w}(n) = \sum_{m=0}^{M-1} b_m \cos\left(2\pi m \frac{n}{N}\right) \mathcal{R}\left(\frac{n}{N}\right) \quad (22)$$

where \mathcal{R} is the *rectangle function*. This function is discontinuous at $n = \frac{\pm N}{2}$, and therefore not differentiable there, unless

$$\sum_{m=0}^{M-1} b_m \cos(\pm\pi m) = 0 \quad (23)$$

Rather than design based on an optimality criterion, such as the height of the highest sidelobe [17], a once-differentiable approximation to an existing window w is desired. To do this, we choose the b_m so that the window \tilde{w} 's squared approximation error to w is minimized while having $\tilde{w}\left(\frac{\pm N}{2}\right) = 0$, i.e. we find the solution $\{b_m^*\}$ to the mathematical program

$$\text{minimize} \sum_{n=0}^{N-1} (w(n) - \sum_{m=0}^{M-1} b_m \cos(2\pi m \frac{n}{N}))^2 \quad (24)$$

$$\text{subject to} \sum_{m=0}^{M-1} b_m \cos(\pi m) = 0 \quad (25)$$

which can be solved using constrained least-squares; a standard numerical linear algebra routine [18, p. 585].

6. A CONTINUOUS WINDOW DESIGN EXAMPLE

As a design example we show how to create a continuous approximation of a digital prolate spheroidal window.

Digital prolate spheroidal windows are a parametric approximation to functions whose Fourier transform's energy is maximized in a given bandwidth [19]. These can be tuned to have extremely low sidelobes, at the expense of main-lobe width. Differentiation of these window functions may be possible but is not as straightforward as differentiation of the sum-of-cosine windows above. Furthermore, the windows do not generally have end-points equal to 0. In the following we will demonstrate how to approximate a digital prolate spheroidal window with one that is everywhere at least once-differentiable.

In [20] it was shown how to construct digital prolate spheroidal windows under parameters N , the window length in samples, and a parameter W choosing the (normalized) frequency range in which the proportion of the main lobe's energy is to be maximized. We chose $N = 512$ based on the window length chosen in [1] for ease of comparison. Its W parameter's value was chosen by synthesizing windows with W ranging between 0.005 and 0.010 at a

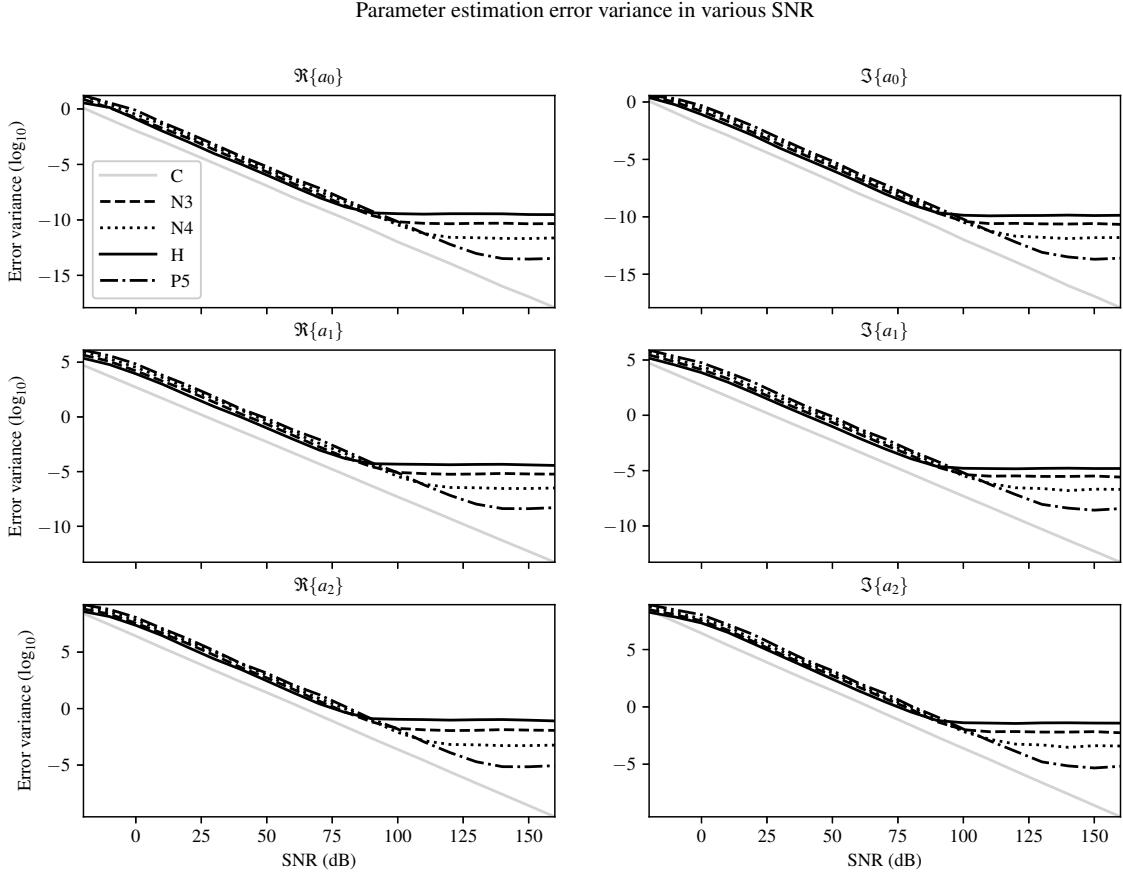


Figure 2: The estimation variance of random polynomial phase sinusoids averaged over $K_1 = 1000$ trials using atoms generated from various windows. C is the Cramér-Rao lower bound, $N3$ and $N4$ are the 3- and 4-cosine-term continuous Nuttall windows, H is the Hann window, and $P5$ is the continuous 5-cosine-term approximation to a digital prolate window as described in Sec. 6.

Table 1: The coefficients of the once-differentiable approximation to a digital prolate window designed in Sec. 6.

$$\begin{aligned} b_0 &= 3.128 \times 10^{-1} \\ b_1 &= 4.655 \times 10^{-1} \\ b_2 &= 1.851 \times 10^{-1} \\ b_3 &= 3.446 \times 10^{-2} \\ b_4 &= 2.071 \times 10^{-3} \end{aligned}$$

resolution of 0.001. The window with the closest 3 dB bandwidth to the 4-term Nuttall window was obtained with $W = 0.008$. Its magnitude response is shown in Fig. 1. We see that this window's asymptotic falloff is 6 dB per octave and therefore has a discontinuity somewhere in its domain [15].

We designed an approximate window using Eq. 24 for M varying between 2 and $N/8$ to find the best approximation to the digital prolate window's main lobe using a small number of cosines. The M giving the best approximation was 5. The magnitude re-

sponse of the approximation is shown in Fig. 1 and its coefficients are listed in Tab. 1; the temporal shape is very close to a digital prolate spheroidal window with $W = 0.008$ and is therefore omitted for brevity.

It is seen that a lower highest sidelobe level than the Nuttall and Prolate windows is obtained by slightly sacrificing the narrowness of the main lobe. More importantly, in Fig. 1 we observe that the falloff of the window is 18 dB per octave because it is once-differentiable at all points in its domain.

7. THE PERFORMANCE OF IMPROVED WINDOWS

7.1. Signals with single component

To compare the average estimation error variance with the theoretical minimum given by the Cramér-Rao bound we synthesized K_1 random chirps using Eq. 1 with $Q = 2$ and parameters chosen from uniform distributions justified in [1]. The original Hann window, the windows proposed by Nuttall and the new digital prolate based window were used to synthesize the atoms as described in Sec. 4 and their estimation error variance was compared (see Fig. 2). After performing the DFT to obtain inner products with the

atoms, the three atoms whose inner products were greatest were used in the estimations, i.e., $R = 3$ in Eq. 10. The windows with the lowest sidelobes only give the lowest error variance at very favourable SNRs, at real-world SNRs the original Hann window still performs best at estimating the parameters of a single component signal.

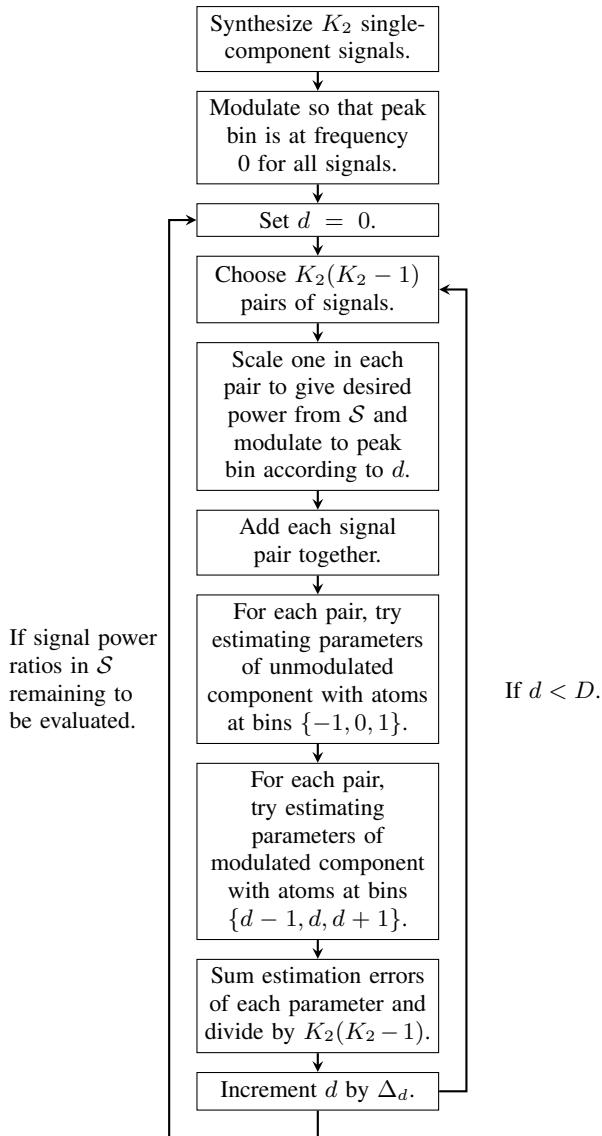


Figure 4: The evaluation procedure for 2-component signals.

7.2. Signals with 2 components

To evaluate the performance of the various windows when estimating the parameters of components in mixture we synthesized signals using Eq. 3 with $P = 2$ and $Q = 2$ and parameters chosen from the uniform distributions specified in [1]. We desired to see how the accuracy of estimation is influenced by the difference (in bins) between the locally maximized atoms and the difference in signal power between the two components. To obtain a set of components from which test signals exhibiting the desired differences

could be constructed, we synthesized a set \mathcal{C} of K_2 components for which the energy is maximized in bin 0. Test signals were obtained by choosing a pair of unique components from this set and modulating one to give the desired frequency and amplitude difference. This was carried out as follows: the atom r^* for which the inner product was maximized was determined for each unmixed chirp and the chirp was modulated by $\exp(-2\pi \frac{r^* n}{N} j)$ for $0 \leq n < N$ in order to move this maximum to $r = 0$. Then for each desired difference d , with $0 \leq d < D$ (for the evaluation $D = 40$), two unique chirps were selected from \mathcal{C} and one chirp was modulated by $\exp(2\pi \frac{nd}{N} j)$ for $0 \leq n < N$ in order to give the desired difference between maxima. This component was also scaled by a constant to give a desired signal power ratio from set S with the other component (the power ratios S tested were 0 dB and -30 dB). As we assume perfect peak-atom selection for this evaluation no inner-product maximizing r^* is chosen, rather atoms with angular frequencies $\omega = 2\pi \frac{\hat{d}}{N}$ for $\hat{d} \in \{d - 1, d, d + 1\}$ in Eq. 20 (again, $R = 3$) were chosen to carry out the estimation. d was incremented by $\Delta_d = 0.25$ and so \hat{d} was not generally integral valued in this case. The parameters of the unmodulated component were estimated using angular frequencies $\omega = 2\pi \frac{\hat{d}}{N}$ for $\hat{d} \in \{-1, 0, 1\}$ in Eq. 20. The squared estimation error for each parameter was summed and divided by $K_2(K_2 - 1)$ (the number of choices of two unique components) to give the averaged squared estimation error for each parameter at each difference d . The procedure is summarized in Fig. 4.

The behaviour of the windows when used to analyse mixtures of non-stationary signals is similar to the behaviour of windows used for harmonic analysis in the stationary case [16]; here we obtain further insight into how the estimation of each coefficient of the polynomial in Eq. 1 is influenced by main-lobe width and sidelobe height and slope. In Fig. 3 we see that there is generally less estimation error for components having similar signal power. This is to be expected as there will be less masking of the weaker signal in these scenarios. The estimation error is large when the atoms containing the most signal energy for each component are not greatly separated in frequency. This is due to the convolution of the Fourier transform of the window with the signal, and agrees with what was predicted by Eq. 17: indeed windows with a larger main lobe exhibit a larger “radius” (bandwidth) in which the error of the parameter estimation will be high. However, for signals where local inner-product maxima are from atoms sufficiently separated in frequency, windows with lower sidelobes are better at attenuating the other component and for these the estimation error is lowest.

8. CONCLUSIONS

Motivated by the need to analyse mixtures of frequency- and amplitude-modulated sinusoids (Eq. 3), we have shown that the DDM can be employed under a single-component assumption when components have roughly disjoint sets of atoms for which their inner products take on large values. This indicates the need for windows whose Fourier transform exhibits low sidelobes. We developed windows whose sidelobes are minimized while remaining everywhere once-differentiable: a requirement to generate valid atoms for the DDM. These windows were shown to only improve parameter estimation of $P = 1$ component with argument-polynomial of order $Q = 2$ in low amounts of noise. However, for $P = 2$ components of the same order in mixture without noise, granted the

components exhibited reasonable separation in frequency between the atoms for which the inner product was maximized, these new windows substantially improved the estimation of all but the first argument-polynomial coefficient.

Further work should evaluate these windows on sinusoids of different orders, i.e., $Q \gg 1$. Optimal main-lobe widths for windows should be determined depending on the separation of local maxima in the power spectrum. It should also be determined if these windows improve the modeling of real-world acoustic signals.

9. ACKNOWLEDGMENTS

This work was partially supported by grant from the Natural Sciences and Engineering Research Council of Canada awarded to Philippe Depalle (RGPIN-262808-2012).

10. REFERENCES

- [1] Michaël Betser, “Sinusoidal polynomial parameter estimation using the distribution derivative,” *IEEE Transactions on Signal Processing*, vol. 57, no. 12, pp. 4633–4645, 2009.
- [2] Xavier Serra, *A system for sound analysis/transformation/synthesis based on a deterministic plus stochastic decomposition*, Ph.D. thesis, Stanford University, 1989.
- [3] Sylvain Marchand and Martin Raspaud, “Enhanced time-stretching using order-2 sinusoidal modeling,” in *Proceedings of the 7th International Conference on Digital Audio Effects (DAFx-04)*, 2004, pp. 76–82.
- [4] Lippold Haken, Kelly Fitz, and Paul Christensen, “Beyond traditional sampling synthesis: Real-time timbre morphing using additive synthesis,” in *Analysis, Synthesis, and Perception of Musical Sounds*, pp. 122–144. Springer, 2007.
- [5] Robert J McAulay and Thomas F Quatieri, “Speech analysis/synthesis based on a sinusoidal representation,” *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol. 34, no. 4, pp. 744–754, 1986.
- [6] Wen Xue, “Piecewise derivative estimation of time-varying sinusoids as spline exponential functions,” in *Proceedings of the 19th International Conference on Digital Audio Effects (DAFx-16)*, 2016, pp. 255–262.
- [7] Brian Hamilton and Philippe Depalle, “A unified view of non-stationary sinusoidal parameter estimation methods using signal derivatives,” in *the proceedings of the 2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2012, pp. 369–372.
- [8] Dan Stowell, Sašo Mušević, Jordi Bonada, and Mark D Plumley, “Improved multiple birdsong tracking with distribution derivative method and Markov renewal process clustering,” in *the proceedings of the 2013 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2013, pp. 468–472.
- [9] Elliot Creager, “Musical source separation by coherent frequency modulation cues,” M.A. thesis, McGill University, 2016.
- [10] Brian Hamilton and Philippe Depalle, “Comparisons of parameter estimation methods for an exponential polynomial sound signal model,” in *Audio Engineering Society Conference: 45th International Conference on Applications of Time-Frequency Processing in Audio*. Audio Engineering Society, 2012.
- [11] Axel Röbel, “Estimating partial frequency and frequency slope using reassignment operators,” in *International Computer Music Conference*, 2002, pp. 122–125.
- [12] Laurent Schwartz and Institut de mathématique (Strasbourg), *Théorie des distributions*, vol. 2, Hermann Paris, 1959.
- [13] Steven M Kay, *Fundamentals of statistical signal processing, volume I: Estimation theory*, Prentice Hall, 1993.
- [14] Jont B Allen and Lawrence R Rabiner, “A unified approach to short-time Fourier analysis and synthesis,” *Proceedings of the IEEE*, vol. 65, no. 11, pp. 1558–1564, 1977.
- [15] Albert Nuttall, “Some windows with very good sidelobe behavior,” *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 29, no. 1, pp. 84–91, 1981.
- [16] Fredric J Harris, “On the use of windows for harmonic analysis with the discrete Fourier transform,” *Proceedings of the IEEE*, vol. 66, no. 1, pp. 51–83, 1978.
- [17] L Rabiner, Bernard Gold, and C McGonegal, “An approach to the approximation problem for nonrecursive digital filters,” *IEEE Transactions on Audio and Electroacoustics*, vol. 18, no. 2, pp. 83–106, 1970.
- [18] Gene H Golub and Charles F Van Loan, *Matrix computations*, The John Hopkins University Press, 3rd edition, 1996.
- [19] David Slepian, “Prolate spheroidal wave functions, Fourier analysis, and uncertainty V: The discrete case,” *Bell Labs Technical Journal*, vol. 57, no. 5, pp. 1371–1430, 1978.
- [20] Tony Verma, Stefan Bilbao, and Teresa HY Meng, “The digital prolate spheroidal window,” in *the proceedings of the 1996 IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*. IEEE, 1996, vol. 3, pp. 1351–1354.

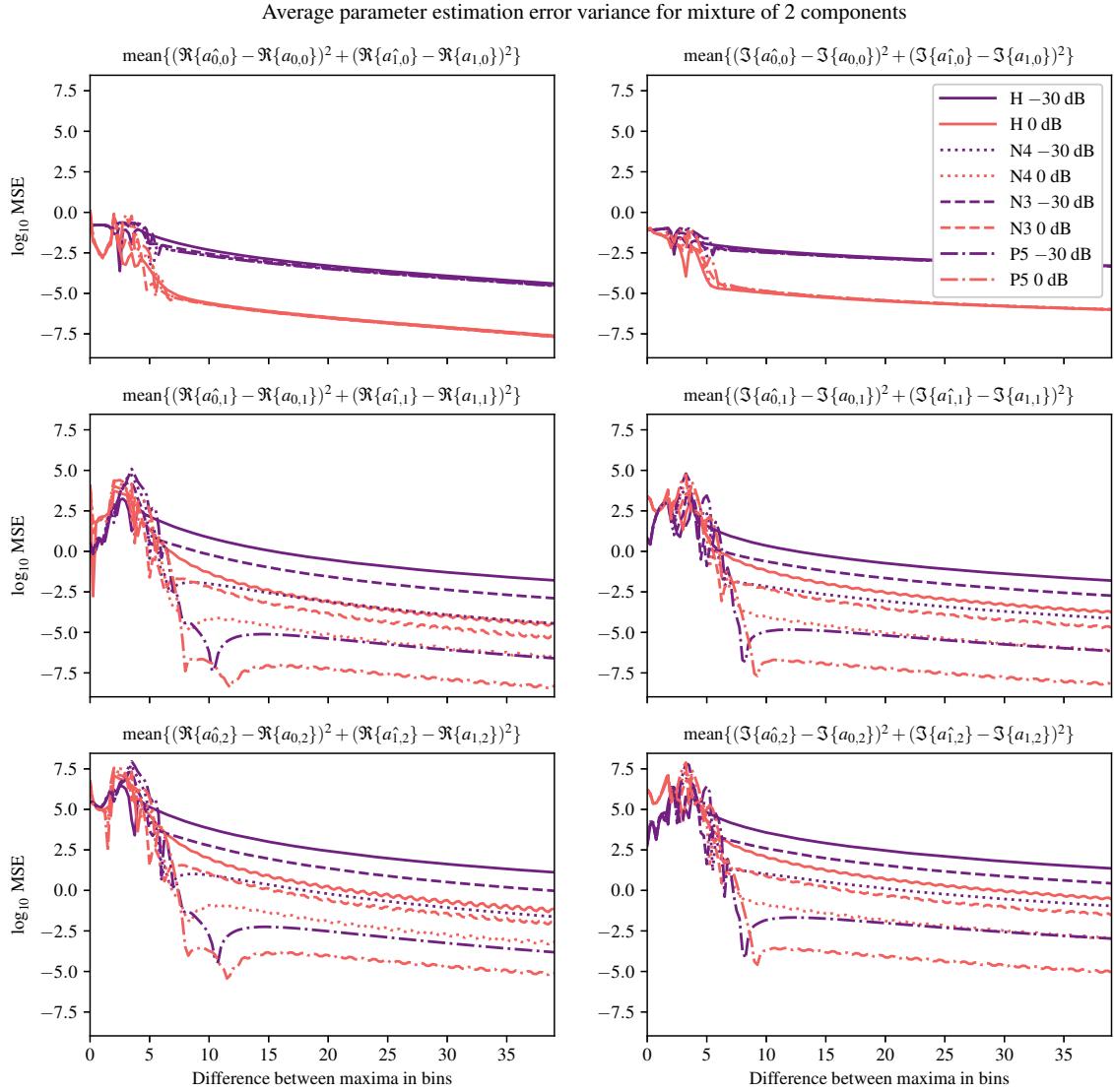


Figure 3: The mean squared estimation error for each parameter in an analysis of two components in mixture. A set of $K_2 = 10$ chirps was synthesized and each unique pair used for maximum bin differences $0 \leq d < 40$, with d varied in 0.25 bin increments. The signal power ratio between components is indicated with colours and the corresponding ratio in decibels is indicated in the plot legend. The names indicate the windows used to generate the atoms for estimation: $N3$ and $N4$ are the 3- and 4-cosine-term continuous Nuttall windows, H is the Hann window, and $P5$ is the continuous 5-cosine-term approximation to a digital prolate window as described in Sec. 6.

NICHT- NEGATIVEMATRIXFAKTORISIERUNG NUTZEN DES KLANGSYNTHESYSTEM (NIMFKS): EXTENSIONS OF NMF-BASED CONCATENATIVE SOUND SYNTHESIS

Michael Buch

Centre for Digital Music
Queen Mary University of London
London, UK
m.buch@se13.qmul.ac.uk

Elio Quinton

Centre for Digital Music
Queen Mary University of London
London, UK
e.quinton@qmul.ac.uk

Bob L. Sturm

Centre for Digital Music
Queen Mary University of London
London, UK
b.sturm@qmul.ac.uk

ABSTRACT

Concatenative sound synthesis (CSS) entails synthesising a “target” sound with other sounds collected in a “corpus.” Recent work explores CSS using non-negative matrix factorisation (NMF) to approximate a target sonogram by the product of a corpus sonogram and an activation matrix. In this paper, we propose a number of extensions of NMF-based CSS and present an open MATLAB implementation in a GUI-based application we name NiMFKS. In particular we consider the following extensions: 1) we extend the NMF framework by implementing update rules based on the generalised β -divergence; 2) We add an optional monotonic algorithm for sparse-NMF; 3) we tackle the computational challenges of scaling to big corpora by implementing a corpus pruning preprocessing step; 4) we generalise constraints that may be applied to the activation matrix shape; and 5) we implement new modes of interacting with the procedure by enabling sketching and modifying of the activation matrix. Our application, NiMFKS and source code can be downloaded from here: <https://code.soundsoftware.ac.uk/projects/nimfks>.

1. INTRODUCTION

Given a dataset of digitally recorded sound material called a “corpus”, the goal of concatenative sound synthesis (CSS) is to concatenate, or mix together, this sound material to approximate a “target” sound according to some criteria [1–7]. This criteria is typically in terms of distance within a descriptor space, e.g., features like spectral centroid and MFCCs, but can also involve considerations of context, e.g., concatenation cost and/or transformation cost [4, 7]. The motivations of CSS can be creative [2, 5, 8], and also to achieve high-quality sound synthesis [9, 10].

Recent work of Driedger et al. [3] explores the use of non-negative matrix factorization (NMF) for CSS. NMF [11] is an iterative procedure that attempts to express a non-negative matrix $\mathbf{V} \in \mathbb{R}_+^{N \times M}$ as a product of two other non-negative matrices, $\mathbf{V} \approx \mathbf{WH}$. The idea is to represent each column of \mathbf{V} as a linear combination of the columns of $\mathbf{W} \in \mathbb{R}_+^{N \times K}$, by the weights in $\mathbf{H} \in \mathbb{R}_+^{K \times M}$. As such, the columns of \mathbf{W} are referred to as *templates* and the rows of \mathbf{H} as *activations*. In audio applications, it is common for \mathbf{V} to be the magnitude or energy spectrogram (sonogram) [12], which is non-negative. NMF has found application in several areas of audio processing, e.g., polyphonic music transcription [12] and source separation [13–15].

In their CSS research, Driedger et al. [3] proposes to adapt NMF with additional constraints to enforce particular characteristics in the activation matrix \mathbf{H} . In particular, these modifications

aim to reduce repetition, suppress simultaneity, and preserve context of sound material in the corpus. As a follow-up, Su et al. [16] extends on this work to create a certain “8-bit music” and contrast different methods of NMF when used for CSS. They develop a simple time-domain synthesis method to avoid sonic artifacts in the inversion of complex STFT, a technique we also develop here as an option in our application (see Sec. 3.1).

The work we present here makes several contributions to NMF for CSS. First, we propose a generalisation of existing procedures and introduce some extensions. Namely, we generalise the NMF framework and the post-processing constraints, by using the generalised β -divergence as an optimisation objective and a general convolution kernel respectively. As an additional constraint on the activation matrix, we implement a monotonic algorithm to enforce sparsity constraints in the NMF optimisation [17]. Second, we address computation issues with NMF that makes it unsuitable for very large corpora (i.e. dictionary matrix \mathbf{W}). In order to overcome this limitation, we introduce a pruning strategy as a pre-processing step in which only a subset of the corpus frames are selected to be used in the NMF procedure (i.e. reducing the dimensionality of \mathbf{W}). Third, we present an open MATLAB Graphical User Interface (GUI), thereby making these techniques accessible. In addition, we propose and implemented other modes of interacting with the procedure, e.g., editing or “sketching” the activation matrix \mathbf{H} . To encourage further research and applications, we design our application NiMFKS to facilitate the integration of new modules through separation of concerns between the interface and implementation details and modular design with individual components responsible for different parts of the synthesis. Essentially one can simply plug-in desired NMF algorithms and modify existing ones without breaking the application.

2. PRIOR WORK

NMF aims to find non-negative matrices \mathbf{W} and \mathbf{H} such that $\mathbf{V} \approx \mathbf{WH}$. This takes the form of an optimisation problem where the factorisation attempts to minimise a reconstruction error function, which we denote $D(\mathbf{V} || \mathbf{WH})$. Common choices when applying NMF to audio data are Euclidian distance and Kullback-Leibler (KL) divergence [12–15].

Driedger et al. [3] use the KL divergence. At each iteration, after the NMF update, Driedger et al. [3] introduces subsequent updates to the activations to avoid particular behaviours, e.g., encouraging diagonal structures over horizontal and vertical ones. To suppress horizontal structures (e.g., corpus unit repetitions), they

modify \mathbf{H} after the NMF update at iteration l of L according to

$$[\mathbf{H}]_{km} \leftarrow \begin{cases} [\mathbf{H}]_{km}, & [\mathbf{H}]_{km} = \max\{[\mathbf{H}^T \mathbf{e}_k]_{m-r:m+r}\} \\ [\mathbf{H}]_{km}(1 - \frac{l+1}{L}), & \text{else} \end{cases} \quad (1)$$

where \mathbf{e}_k is the k^{th} standard vector, and $r \in \{0, 1, \dots, m\}$ describes the number of columns to the right and left of the m^{th} activation column over which one wishes to reduce repeated non-zero values.

To suppress non-zero values in each column of \mathbf{H} , i.e., discourage too many templates from being active simultaneously, they modify \mathbf{H} according to

$$[\mathbf{H}]_{km} \leftarrow \begin{cases} [\mathbf{H}]_{km}, & [\mathbf{H}]_{km} \in \max_p\{\mathbf{H}\mathbf{e}_m\} \\ [\mathbf{H}]_{km}(1 - \frac{l+1}{L}), & \text{else} \end{cases} \quad (2)$$

where $\max_p\{\cdot\}$ returns the subset of $p \in \{0, 1, \dots, m\}$ largest values of its vector argument.

To promote diagonal structures, i.e., continuity of the corpus units, Driedger et al. [3] process the activation matrix after its full update in iteration l by filtering. This can be expressed as a two-dimensional convolution between \mathbf{H} and a diagonal kernel

$$\mathbf{H} \leftarrow \mathbf{H} * \mathbf{G} \quad (3)$$

where $*$ is convolution, and $\mathbf{G} = \mathbf{I}_c$ (identity matrix of size c) in Driedger et al. [3].

The application of these constraints removes the NMF algorithm's convergence guarantee. Thus the sequence of updates (1) to (3) is repeated until a user-specified stopping criteria, such as number of iterations. In summary, this NMF approach to CSS requires the specification of the following parameters: the total number of iterations L , the horizontal neighbourhood r , the vertical magnitude neighbourhood p , the filter kernel size c , and finally the parameters involved in computing the STFT and its inverse.

3. EXTENSIONS

We now present our three extensions to the application of NMF to CSS: 1) time-domain synthesis; 2) sparse NMF; and 3) pruning large corpora.

3.1. Time-domain synthesis

Given a magnitude spectrogram, or the product \mathbf{WH} in the present case, the Griffin-Lim algorithm [18] can be used to project back to the time-domain. If we have the corpus as a time-domain waveform, then we can simply window, scale, and add the corresponding waveforms to create the synthesis. Since the m^{th} row of \mathbf{H} specifies the activation of the m^{th} column of \mathbf{W} , the time domain synthesis is achieved by concatenating the corresponding portions of the corpus waveform scaled by their activation (i.e. elements of \mathbf{H}). This approach to synthesis circumvents the restrictions on window shape and overlap for invertibility inherent to the Griffin-Lim algorithm [18], and also makes possible the use of other kinds of features, e.g., chroma. It can be parallelised to make it faster.

3.2. Sparse NMF

The generalised β -divergence offers a family of divergences that are parametrised by β [19, 20]. It is defined as follows:

$$D_\beta(x||y) = \begin{cases} \frac{x^\beta}{\beta(\beta-1)} + \frac{y^\beta}{\beta} - \frac{xy^{\beta-1}}{\beta-1} & \beta \in \mathbb{R} \setminus \{0, 1\} \\ x \log\left(\frac{x}{y}\right) - x + y & \beta = 1 \\ \frac{x}{y} - \log\left(\frac{x}{y}\right) - 1 & \beta = 0 \end{cases} \quad (4)$$

When $\beta = 2$, this becomes the Euclidean distance; the Kullback-Leibler (KL) divergence for $\beta = 1$, and the Itakuro-Saito (IS) divergence for $\beta = 0$. In our case, we denote the β -divergence of \mathbf{V} and \mathbf{WH} as $D_\beta(\mathbf{V}||\mathbf{WH})$. In this case, we apply (4) to each element of the two matrices, and then sum over all elements. Fevotte and Idier [21] show that one of the possible multiplicative update rules using β -divergence is given by

$$\mathbf{H} \leftarrow \mathbf{H} \odot \frac{\mathbf{W}^T [\mathbf{V} \odot (\mathbf{WH})^{(\beta-2)}]}{\mathbf{W}^T (\mathbf{WH})^{(\beta-1)}} \quad (5)$$

where \odot denotes the element-wise multiplication, and exponentiation as well as division are element-wise.

As an extension of Driedger et al. [3], we implement sparsity constraints in the NMF update, which effectively restrict both polyphony and the corpus diversity within the NMF framework. There exist a number of algorithms for applying sparsity constraints on the activation matrix because it has proven useful for a variety of audio and music tasks, see for example [22, 23]. In our implementation, we use the penalty introduced in [17], notated Υ here, so that the cost function we minimise with respect to \mathbf{H}, \mathbf{W} is:

$$D_\beta(\mathbf{V}||\mathbf{WH}) + \lambda \Upsilon \quad (6)$$

where λ is a parameter to control the weight of the penalty. In order for the regulariser Υ to enforce a sparsity constraint, it is common to set $0 < \beta \leq 1$. Typically, the closer β is to 0, the stronger the sparsity constraint will be. We set $\beta = 0.5$ in our application by default, as it enforces strong sparsity constraints, so that the templates that only account for little variance in the target matrix and their corresponding activation tend to zero during the NMF updates. By virtue of the multiplicative updates, components of zero magnitude remain zero for the remaining updates and are therefore effectively pruned out of the model [17]. The corresponding update rule for \mathbf{H} is [17]:

$$\mathbf{H} \leftarrow \mathbf{H} \odot \left[\frac{\mathbf{W}^T (\mathbf{V} \odot (\mathbf{WH})^{[-\frac{3}{2}]})}{\mathbf{W}^T (\mathbf{WH})^{[-\frac{1}{2}]} + \lambda \Psi_{\mathbf{H}}} \right]^\varphi \quad (7)$$

where $\Psi_{\mathbf{H}} = \mathbf{H}^{-1/2}$ is a term resulting from the application of the penalty Υ with $\beta = 0.5$ and $\varphi = \frac{2}{3}$ is a term ensuring the descent of the cost function at each iteration.

3.3. Working with Large Corpora

Optimisation approaches applied to CSS [3, 4, 7], carry far higher computational costs than “greedy” approaches, e.g., Catepillar [2] and MATConcat [5]. With the templates \mathbf{W} fixed, NMF need only iteratively update the activations \mathbf{H} . Using the Euclidean distance, the computational complexity of each iteration is then $\mathcal{O}(K^2 M)$. The KL divergence carries a higher complexity. These increase

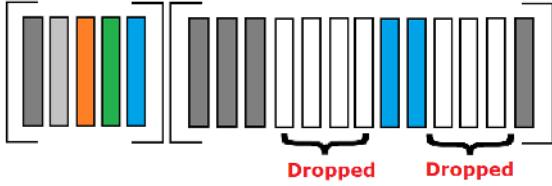


Figure 1: The corpus pruning algorithm removes those corpus frames that are dissimilar from any of the frames in the target. Algorithm 1 describes the procedure.

even more when we include the constraints detailed in section 2. Hence, the complexity is linear in the duration of the target, but quadratic in the duration of the corpus. This imposes limits on the size of the corpus one can work with in NMF for CSS, which can affect the quality of the results.

Algorithm 1: Corpus template pruning: Given two sequences of vectors in the same space \mathbb{R}^N — \mathcal{W} from the corpus and \mathcal{V} from the target — and three user-specified parameters — γ , ρ_{\min} , θ — produce a subset of the index into \mathcal{W} according to the dissimilarity function $d : \mathbb{R}^N \times \mathbb{R}^N \rightarrow \mathbb{R}_+$.

```

1 function Prune ( $\mathcal{W}, \mathcal{V}, \gamma, \rho_{\min}, \theta$ );
Input :
  1.  $\mathcal{W} = (\mathbf{w}_k \in \mathbb{R}^N)_{k \in \mathcal{K}}, \mathcal{K} = \{1, \dots, K\}$  (sequence of
     corpus frames)
  2.  $\mathcal{V} = (\mathbf{v}_m \in \mathbb{R}^N)_{m \in \mathcal{M}}, \mathcal{M} = \{1, \dots, M\}$  (sequence of
     target frames)
  3.  $\gamma \geq 0$  (similarity pruning parameter)
  4.  $\rho_{\min} \in [0, 1)$  (relative energy pruning parameter)
  5.  $\theta > 0$  (comparisons parameter)

Output:  $\mathcal{K}_{\text{pruned}} \subseteq \mathcal{K}$  (pruned index into  $\mathcal{W}$ )
2  $\mathcal{K}_{\text{remain}} \leftarrow \{k \in \mathcal{K} : \|\mathbf{w}_k\|_2 > \max_{l \in \mathcal{K}} \|\mathbf{w}_l\|_2 2\rho_{\min}\};$ 
3  $\mathcal{M}_{\text{remain}} \leftarrow \{m \in \mathcal{M} : \|\mathbf{v}_m\|_2 > \max_{l \in \mathcal{M}} \|\mathbf{v}_l\|_2 2\rho_{\min}\};$ 
4  $\mathcal{K}_{\text{pruned}} \leftarrow \emptyset;$ 
5 while  $\mathcal{M}_{\text{remain}}$  is not empty do
  6    $m \leftarrow \mathcal{M}_{\text{remain}}(1)$  (first element of set);
  7    $\mathcal{D} \leftarrow (d(\mathbf{v}_m, \mathbf{w}_k) : k \in \mathcal{K}_{\text{remain}})$  (dissimilarities of the
      target frame to remaining corpus frames);
  8    $\mathcal{J} \leftarrow \{k \in \mathcal{K}_{\text{remain}} : d(\mathbf{v}_m, \mathbf{w}_k) < (1 + \gamma) \min \mathcal{D}\}$ 
      (indices of corpus frames acceptably similar to the
      target frame);
  9    $\mathcal{K}_{\text{pruned}} \leftarrow \mathcal{K}_{\text{pruned}} \cup \mathcal{J}$  (store indices of corpus frames);
10    $\mathcal{K}_{\text{remain}} \leftarrow \mathcal{K}_{\text{remain}} \setminus \mathcal{J}$  (remove indices of corpus frames
      from further consideration);
11    $\mathcal{M}_{\text{remain}} \leftarrow \{m' \in \mathcal{M}_{\text{remain}} : d(\mathbf{v}_m, \mathbf{w}_{m'}) > \theta\}$ 
      (indices of target frames that acceptably dissimilar
      from current target frame);
12 end
13 return  $\mathcal{K}_{\text{pruned}}$ ;

```

To address this complexity issue, we propose preprocessing a corpus by pruning. Figure 1 depicts the basic concept. Frames in the target (left) and corpus (right) are colour coded by their content. Pruning keeps only the corpus frames that are measured similar enough to the target frames. We then execute NMF with an \mathbf{W}

built by concatenating the remaining templates. This assumes that the dissimilar corpus frames would not have been used in approximating \mathbf{V} anyway.

The template pruning algorithm is shown in Algorithm 1. The user must specify three parameters, and a dissimilarity function. The dissimilarity function we use is the cosine distance

$$d(\mathbf{v}, \mathbf{w}) := 1 - \frac{\mathbf{v}^T \mathbf{w}}{\|\mathbf{v}\|_2 \|\mathbf{w}\|_2}. \quad (8)$$

The parameter $\gamma \geq 0$ controls the severity of the pruning procedure, with smaller values producing a higher degree of pruning. For $\gamma = 0$, only one corpus frame will be kept for each target frame considered. The parameter $\theta > 0$ controls the number of comparisons between target frames and corpus frames. As $\theta \rightarrow 0$ every target frame is considered, but this can be unnecessary when there is high similarity between target frames. As θ grows, we potentially consider fewer and fewer target frames in the pruning. Finally, the parameter $\rho_{\min} \in [0, 1)$ controls a preprocessing pruning procedure, removing the indices of the corpus and target frames that have norms that are too small. The effects of this are bypassed if $\rho_{\min} = 0$; and if $\rho_{\min} \rightarrow 1$, only the frame with the largest norm is kept.

After pruning, we use the columns of \mathbf{W} indexed by $\mathcal{K}_{\text{pruned}}$ to create a \mathbf{W}' , and then use NMF as described above to find a \mathbf{H}' such that $\mathbf{V} \approx \mathbf{W}' \mathbf{H}'$. We then upsample the activation matrix \mathbf{H}' to form \mathbf{H} having a size commensurate with the original problem. This involves distributing the rows of \mathbf{H}' according to the indices $\mathcal{K}_{\text{pruned}}$, and leaving everything else zero. We can then apply modifications to \mathbf{H} , such as continuity enhancement and polyphony restriction, but at the conclusion of the NMF procedure.

4. THE MATLAB APPLICATION NiMFKS

Driedger et al. [3] do not supply a reproducible work package, but their procedure is described such that it can be reproduced. We have done so by implementing a GUI application in MATLAB, named NiMFKS. We organise the interface into four panes. Figure 2 shows an example screenshot of the interface.

The “Sound Files” pane, top left, lets the user load the audio files to be used as corpus and target. In this example figure, the user has selected multiple audio files, and a target instrumental recording of “Mad World” by “Gary Jules”. If several audio files are specified by the user, they are concatenated to form the corpus. If the sampling rates of the target and corpus are different, NiMFKS resamples the target signal to have the same sampling rate as the corpus.

The pane labeled “Analysis” lets the user set the parameters for computing features of both the corpus and target audio. In the example of Fig. 2, the user has specified the STFT feature to be computed using a Hann window of duration 400ms with 50% overlap.

In the “Synthesis” pane, the user may select the method used for synthesising the output audio waveform and set the related parameters. Building on this observation and work reported in [16] where Su et al. demonstrate the benefits of synthesis approaches alternative to the ISTFT, we make both Time-Domain (cf. section 3.1) and ISTFT synthesis available in NiMFKS. The “Synthesis Method” drop-down menu lets the user choose between these.

The NMF sub-pane lets the user set all the parameters that influence the factorisation. The “Prune” parameter, γ (cf. section

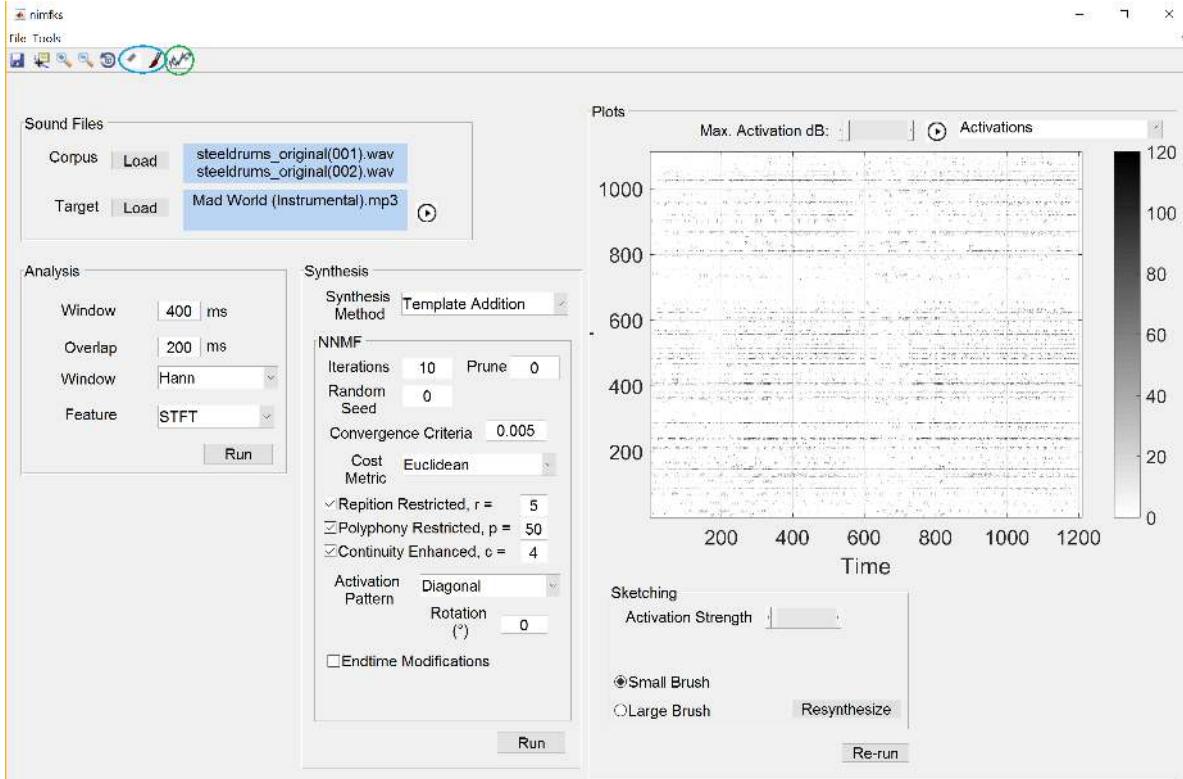


Figure 2: This screen shot shows the main window of NiMFKS. We see the target is “Mad World”, and the corpus is multiple samples of steel drums. The user presses the run button in the “Analysis” pane to compute STFT features using a Hann window of duration 400ms with overlap of 50%. When the user presses the “Run” button in the “Synthesis” pane, 10 iterations of NMF with the Euclidean distance will be performed with the additional constraints (see Sec. 2). No pruning is performed in this case. For synthesis “Template Addition” is specified. On the right we see a visualisation of the activation matrix. Next to the slider is the play button to hear the resulting synthesis. Circled in blue at top are the “Activation Sketching” and “Activation Eraser” features (see Sec. 5). Clicking one of these buttons will reveal the “Sketching” pane in which you can choose the “Paint Brush” to perform the actual sketching. If new activations are added or removed by hand, the buttons “Resynthesise” and “Re-run” repeat the synthesis and NMF procedure respectively using the new activation matrix.

3.3), allows to tune how aggressively the corpus loaded by the user should be pruned before it is fed to the NMF optimisation. Currently, NiMFKS defines ρ_{\min} to be -60 dB below the maximum observed in the corpus and target, and $\theta = 0.1$.

The NMF updates stop after a given number of iterations, or when the cost function reaches a convergence criteria. In the example of Fig. 2, NMF will perform no more than 10 iterations, but could exit before if the relative decrease in cost between subsequent iterations is less than 0.005.

NiMFKS relies on β -NMF update rules (cf. section 2) and provides three presets accessible from the “Cost Metric” dropdown menu: Euclidean ($\beta = 2$), Kullback-Leibler divergence ($\beta = 1$), and Sparse NMF with $\beta = 1/2$ as described in (4).

Finally, the remaining parameters control the modifications to be applied to the activation matrix \mathbf{H} either at each iteration of the NMF optimisation or only after convergence if the “Endtime Modifications” parameter is toggled. The user can set the parameters of the filtering kernel, as described in section 2.

The pane labeled “Plots” displays various results from the procedure, accessible from the dropdown menu at top-right. These include the sound waveforms, their sonograms, and the activation matrix. In the example of Fig. 2, the user has selected to view

the resulting activation matrix, and can adjust the contrast of the image by the slider labeled “Max. Activation dB”.

Finally, in order to enable the “Activation Sketching” feature (cf. section 5.2), the user must select the “Activation Sketching” item from the “Tools” menu, which enabled the buttons located at the very top left of the application window and circled on Figure 2. The sub-pane labeled “Sketching” then lets the user specify the settings to be used when sketching activations.

5. ARTISTIC EXTENSIONS

We now describe a few artistic extensions that we have implemented in NiMFKS.

5.1. Activation Sketching

NiMFKS allows one to treat the activation matrix as an image on which the user can directly erase, draw, inflate or deflate activations. This feature is labeled as “Activation Sketching” in the GUI. These can be resynthesized on the fly to see any effects. This proved to be a useful experimentation tool to understand NMF and the synthesis methods in action. Figure 3 shows an example of an

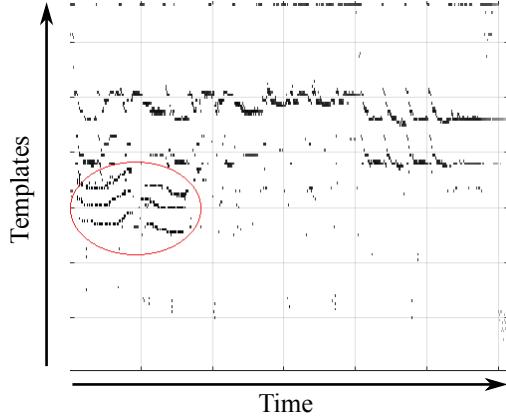


Figure 3: Image of an activation matrix produced by NMF using a specific corpus and target but on which we have sketched additional activations (circled).

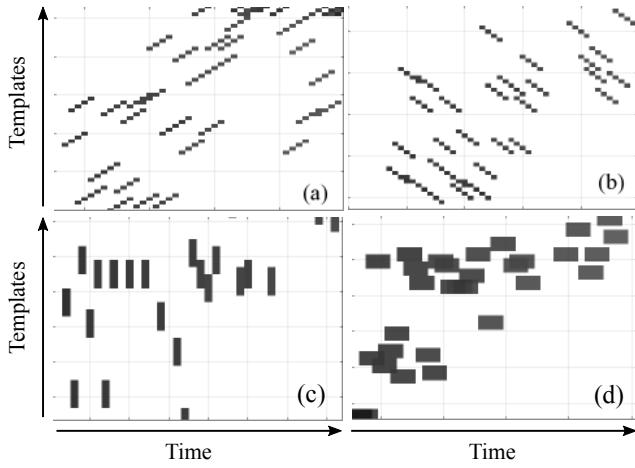


Figure 4: Clockwise from top-left, an activation matrix \mathbf{H} has been filtered (3) with a kernel \mathbf{G} that is diagonal, anti-diagonal, vertical bar and horizontal bar. More exotic options are possible.

activation matrix to which we have sketched additional activations. One can also remove activations with the eraser, and resynthesise the result.

5.2. Modifications of the activation matrix

Driedger et. al. [3] constrained the activation matrix to exhibit diagonal structures in order to preserve the original context of the corpus units. This was created by using a diagonal filter, i.e. setting $\mathbf{G} = \mathbf{I}_c$ in (3). We can define different kinds of filtering kernels, e.g., anti-diagonal, or block average in order to favour other types of structure. NiMFKS enables the user to choose from a selection of such kernels. For instance, the anti-diagonal filter promotes similar structures to the diagonal filter, but with a time-reversed context (the corpus units are not time-reversed, however). The vertical bar filter results in activating several templates simultaneously, while the horizontal bar filter produces repetitions of corpus units. Figure 4 shows several examples of activation matrices obtained using such structures. Additionally, the diagonal

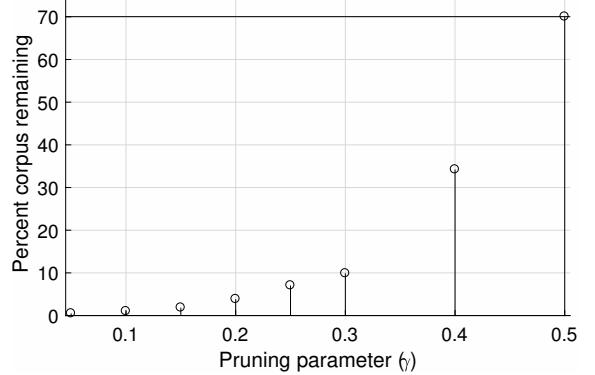


Figure 5: For several values of the pruning parameter, the percentage of the corpus remaining given a target. About 70% of the corpus remains after pruning by energy (keeping frames within 60 dB of the most energetic frame).

kernel may be rotated by an angle specified by the user.

6. DEMONSTRATION

We now demonstrate the pruning strategy. We build a 40m58s corpus from a collection of 44.1 kHz sampled vocalisations from a variety of monkeys and apes. Our analysis of this corpus uses a Hann window of 100ms duration, and 50ms hop. We compute the discrete Fourier transform of each frame, sampled at 1024 points from DC to the Nyquist frequency. In total, there are 49,164 frames, which means \mathbf{W} is a matrix of size 1024×49164 , or 50,343,936 elements. Our target is a recording of “President” Trump saying, “Sadly. The American. Dream. Is *dead!*” Figure 5 shows the percentage of the corpus remaining after the pruning procedure for several pruning parameters. At the most strict value tested, $\gamma = 0.05$, only 247 frames remain. At the least strict value, $\gamma = 0.5$, the number of frames remaining is 34,433. The pruning by energy removes 14,731 frames.

Figure 6 shows how the pruning parameter affects the cost of the factorisation over iteration. We see for this example that after 10 iterations the cost from using a larger corpus becomes smaller than when using the more pruned corpus. Figure 7 shows how the spectrogram of the original target matches with the results from NMF and three pruned corpora. The bottom spectrogram shows the results with a pruning $\gamma = 0.1$ and constraints with values $r = p = c = 3$. The convolution matrix used for continuity enhancement in (3) is diagonal with exponentially decreasing positive values along the diagonal.

7. DISCUSSION & CONCLUSION

The work of Driedger et al. [3] provides an excellent starting point for exploring NMF for CSS but their code is not available. We have implemented their approach and make the application freely available. Our application contributes a variety of extensions as well. We have implemented a time-domain synthesis method, which does not suffer from the limitations of STFT inversion. We make the modifications proposed in [3] applicable during the NMF update procedure, or at the end as a post-processing of the activation matrix. We also implement a pruning strategy for working with

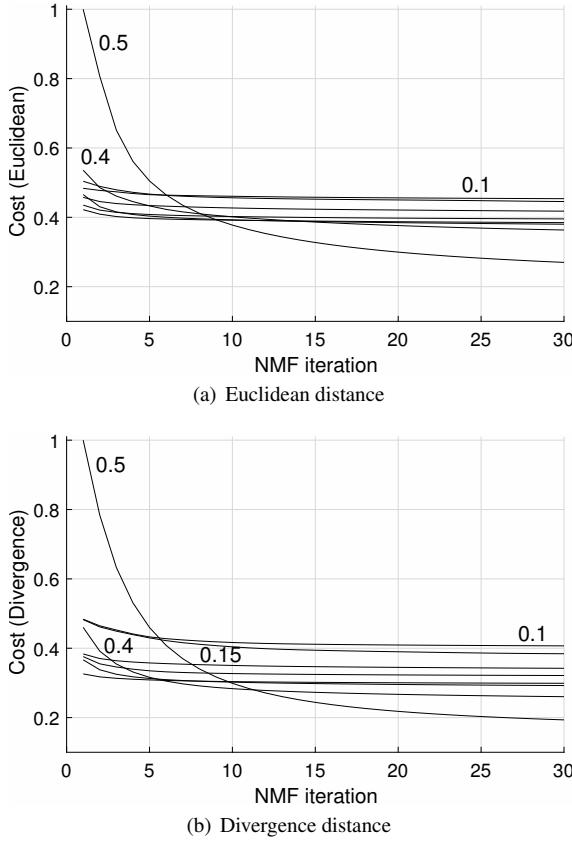


Figure 6: Change in cost over NMF iteration with pruned corpora (relative to highest cost for $\gamma = 0.5$).

large corpora. Although we find this hurts the cost in the factorising procedure, it can become useful when trying to get an idea of how a target and corpus could work together. In a more creative direction, we have implemented an interactive “sketching” procedure by which one may modify a resulting activation matrix, and more general activation filtering procedure.

Our future work will address the uniform segmentation of this NMF approach to CSS. We seek a way to make it compatible with a unit-based segmentation of a target and corpus. A simple way this could be solved is by performing parallel NMF with different time resolutions, and then a fusion of the results at the end with binary masking. We will also redesign the GUI to separate analysis, pruning, factorising, post-processing and synthesising.

8. ACKNOWLEDGMENTS

This work is supported by AHRC Grant No. AH/N504531/1.

9. REFERENCES

- [1] A. Zils and F. Pachet, “Musical mosaicing,” in *Proc. COST G-6 Conf. Digital Audio Effects*, Limerick, Ireland, 2001.
- [2] D. Schwarz, “Concatenative sound synthesis: The early years,” *J. New Music Research*, vol. 35, no. 1, pp. 3–22, 2006.
- [3] J. Driedger, T. Prätzlich, and M. Müller, “Let It Bee – towards NMF-inspired audio mosaicing,” in *Proc. Int. Conf. Music Information Retrieval*, Malaga, Spain, 2015, pp. 350–356.
- [4] J.-J. Aucouturier and F. Pachet, “Jamming with plunderphonics: Interactive concatenative synthesis of music,” *J. New Music Research*, vol. 35, no. 1, 2006.
- [5] B. L. Sturm, “Adaptive concatenative sound synthesis and its application to micromontage composition,” *Computer Music J.*, vol. 30, no. 4, pp. 46–66, Dec. 2006.
- [6] G. Bernardes, *Composing Music by Selection: Content-Based Algorithmic-Assisted Audio Composition*, Ph.D. thesis, Faculty of Engineering, University of Porto, 2014.
- [7] G. Coleman, *Descriptor Control of Sound Transformations and Mosaicing Synthesis*, Ph.D. thesis, Universitat Pompeu Fabra, 2016.
- [8] Juan José Burred, “Factorsynth: A max tool for sound analysis and resynthesis based on matrix factorization,” in *Proceedings of the Sound and Music Computing Conference 2016, SMC 2016, Hamburg, Germany*, 2016.
- [9] E. Lindemann, “Music synthesis with reconstructive phrase modeling,” *IEEE Signal Processing Magazine*, vol. 24, no. 2, pp. 80–91, March 2007.
- [10] M. Umbert, J. Bonada, M. Goto, T. Nakano, and J. Sundberg, “Expression control in singing voice synthesis: Features, approaches, evaluation, and challenges,” *IEEE Signal Processing Magazine*, vol. 32, no. 6, pp. 55–73, Nov 2015.
- [11] D. D. Lee and H. S. Seung, “Algorithms for non-negative matrix factorization,” in *Advances in Neural Information Processing Systems 13*, pp. 556–562. MIT Press, 2001.
- [12] P. Smaragdis and J. C. Brown, “Non-negative matrix factorization for polyphonic music transcription,” in *Proc. IEEE Workshop on App. Signal Proc. Audio and Acoustics*, 2003, pp. 177–180.
- [13] A. Ozerov and C. Févotte, “Multichannel nonnegative matrix factorization in convolutive mixtures for audio source separation,” *IEEE Trans. Audio, Speech, and Lang. Proc.*, vol. 18, no. 3, pp. 550–563, 2010.
- [14] T. Virtanen, “Monaural sound source separation by nonnegative matrix factorization with temporal continuity and sparseness criteria,” *IEEE Trans. Audio, Speech, and Lang. Proc.*, vol. 15, no. 3, pp. 1066–1074, 2007.
- [15] S. Ewert and M. Müller, “Using score-informed constraints for NMF-based source separation,” in *Proc IEEE Int. Conf. Acoustics, Speech and Signal Proc., 2012*. 2012, pp. 129–132, IEEE.
- [16] S. Su C. Chiu L. Su and Y. Yang, “Automatic conversion of pop music into chiptunes for 8-bit pixel art,” in *Proc. IEEE Int. Conf. Acoustics, Speech and Signal Proc.*, March 2017, pp. 411–415.
- [17] E. Quinton, K. O’Hanlon, S. Dixon, and M. B. Sandler, “Tracking Metrical Structure Changes with Sparse-NMF,” in *Proc. IEEE Int. Conf. Acoustics, Speech and Signal Proc.*, 2017.
- [18] D. W. Griffin, J., and S. Lim, “Signal estimation from modified short-time fourier transform,” *IEEE Trans. Acoustics, Speech and Signal Proc.*, pp. 236–243, 1984.

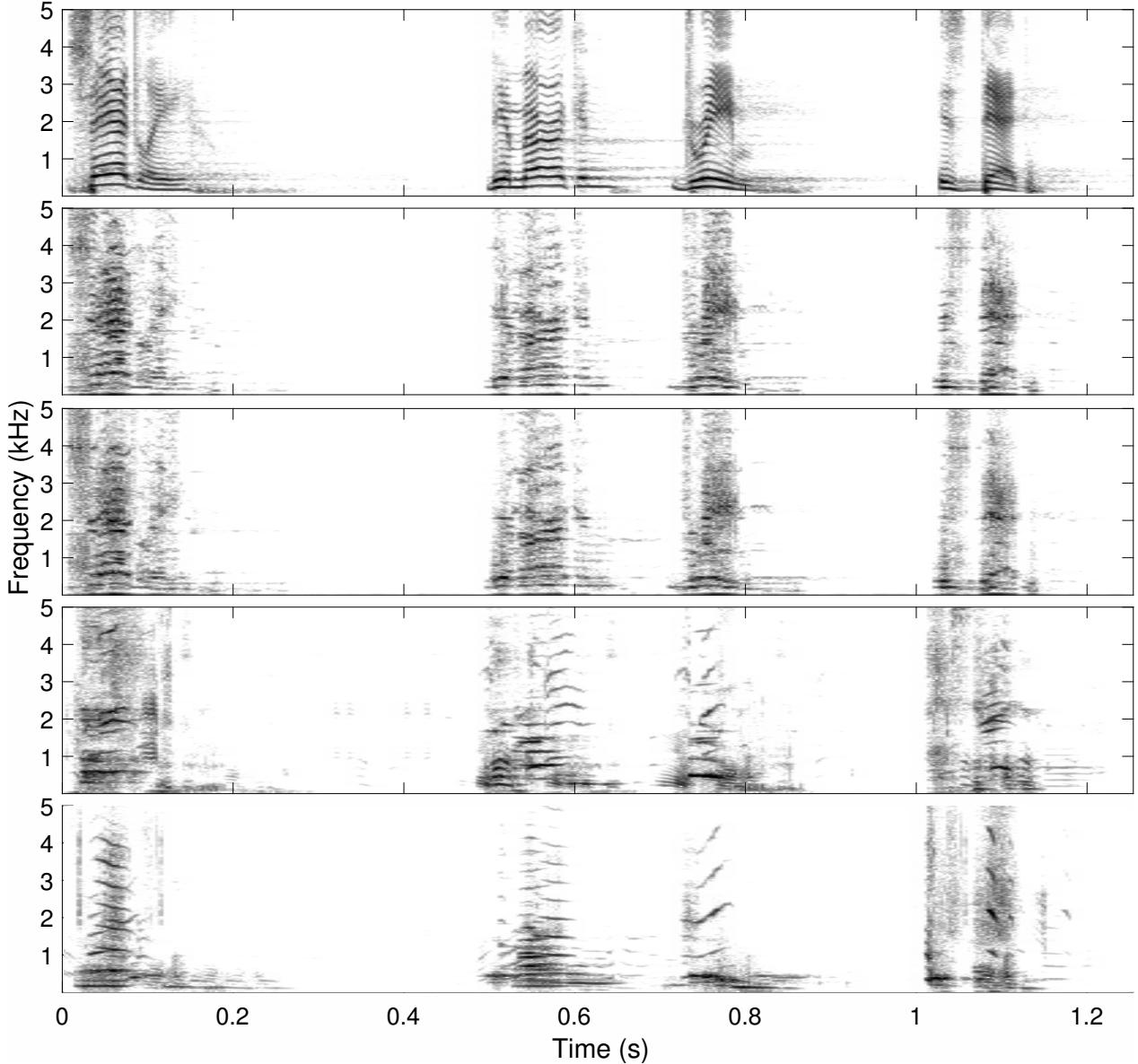


Figure 7: We use NMF-based CSS to resynthesise a recording of “President” Trump saying, “Sadly. The American. Dream. Is *dead!*” using a corpus of 40m58s of sampled vocalisations of monkeys, apes and other animals. The features are DFT magnitude frames. The spectrograms from top to bottom are: original signal; synthesis by NMF with Euclidean distance and pruning with $\gamma = 0.5$; synthesis by NMF with Euclidean distance and pruning with $\gamma = 0.3$; synthesis by NMF with Euclidean distance and pruning with $\gamma = 0.1$; synthesis by NMF with Euclidean distance and pruning with $\gamma = 0.1$, and constraints $r = p = c = 3$ with G in (3) diagonal that is exponentially decreasing.

- [19] R. Kompass, “A generalized divergence measure for nonnegative matrix factorization,” *Neural computation*, vol. 19, no. 3, pp. 780–791, 2007.
- [20] A. Cichocki, R. Zdunek, and S. Amari, “Csiszar’s divergences for non-negative matrix factorization: Family of new algorithms,” in *Proc. Int. Conf. on Independent Component Analysis and Signal Separation*, 2006, pp. 32–39.
- [21] C. Févotte and J. Idier, “Algorithms for nonnegative matrix factorization with the β -divergence,” *Neural computation*, vol. 23, no. 9, pp. 2421–2456, 2011.
- [22] P. O. Hoyer, “Non-negative matrix factorization with sparseness constraints,” *J. Machine Learning Research*, vol. 5, no. Nov, pp. 1457–1469, 2004.
- [23] J. Eggert and E. Korner, “Sparse coding and NMF,” in *Proc. of the Int. Joint Conference on Neural Networks*, 2004, vol. 4, pp. 2529–2533.

VALIDATED EXPONENTIAL ANALYSIS FOR HARMONIC SOUNDS

Matteo Briani*

Computational Mathematics,
Universiteit Antwerpen
Antwerp, BE
matteo.briani@uantwerpen.be

Annie Cuyt

Computational Mathematics,
Universiteit Antwerpen
Antwerp, BE
annie.cuyt@uantwerpen.be

Wen-shin Lee

Computational Mathematics,
Universiteit Antwerpen
Antwerp, BE
wenshin.lee@uantwerpen.be

ABSTRACT

In audio spectral analysis, the Fourier method is popular because of its stability and its low computational complexity. It suffers however from a time-frequency resolution trade off and is not particularly suited for aperiodic signals such as exponentially decaying ones. To overcome their resolution limitation, additional techniques such as quadratic peak interpolation or peak picking, and instantaneous frequency computation from phase unwrapping are used.

Parametric methods on the other hand, overcome the time-frequency trade off but are more susceptible to noise and have a higher computational complexity. We propose a method to overcome these drawbacks: we set up regularized smaller sized independent problems and perform a cluster analysis on their combined output. The new approach validates the true physical terms in the exponential model, is robust in the presence of outliers in the data and is able to filter out any non-physical noise terms in the model.

The method is illustrated in the removal of electrical humming in harmonic sounds.

1. INTRODUCTION

Multi-exponential models arise naturally in music and audio. They are used in audio effects, audio coding, source separation and music transcription. Multi-exponential models can be computed via a non-parametric or a parametric approach.

The non-parametric Fourier based techniques are widely used to decompose a signal into a sum of complex exponentials with equispaced frequencies on the unit circle. The frequency resolution is directly linked to the length of the time window used for the analysis. Longer time windows result in a higher frequency resolution, thus bounding these methods to quasi-stationary recordings. To overcome this limitation and improve the frequency resolution of short time windows, one can exploit the amplitude or phase information of consecutive windows. An overview of such techniques is found in [1].

On the other hand, parametric methods can overcome the frequency limitation of the Fourier based methods, but at the expense of being more susceptible to the noise present in the signal. Another advantage of parametric methods is that they can model complex exponentials whose amplitudes are modulated by exponential decays or polynomials [2]. These methods are successfully applied in different fields of audio signal processing [3, 4, 5]. To improve the robustness against noise, other approaches have also been developed [6, 7].

* This research is supported by the Instituut voor Wetenschap en Technologie - IWT

Recently multi-exponential analysis was generalized [8] to sub-Nyquist sampling rates. The ability to use a coarser time grid may also improve the numerical stability of the parametric method. Moreover, the use of a uniform non-consecutive sampling scheme permits to avoid outliers in the data. At the same time, connections of the method with sparse interpolation from computer algebra and Padé approximation from numerical approximation theory were studied in depth [9]. In the Sections 2, 3 and 4 all these recent results are recapped.

Making use of these recent developments, we present a new approach in Section 5, which can be combined with the use of existing parametric methods. In particular, it can be used on top of any parametric technique derived from Prony’s method. The new parametric analysis is performed on several decimated signals, each being sub-sampled. The use of a cluster detection algorithm allows to automatically validate the output of the method. In the same way it is possible to remove outliers from the data since the latter will not be validated as true frequencies.

When applying the proposed method in Section 6 to harmonic sounds, we see that it easily allows to detect and numerically refine the lowest partial of an harmonic sound, since the validated frequencies retrieved by the cluster analysis are stable frequencies. From the lowest partial, a denoised version of the source can subsequently be constructed.

2. THE MULTI-EXPONENTIAL MODEL IN SIGNAL PROCESSING

In order to proceed we introduce some notations. Let the real numbers $\psi_i, \omega_i, \beta_i$ and γ_i respectively denote the damping, frequency, amplitude and phase in each component of the signal

$$\phi(t) = \sum_{i=1}^n \alpha_i \exp(\phi_i t), \quad t \in \mathbb{Z}$$

$$i^2 = -1 \quad \alpha_i = \beta_i \exp(i\gamma_i), \quad \phi_i = \psi_i + i2\pi\omega_i. \quad (1)$$

For the moment, we assume that the frequency content of $\phi(t)$ is limited by

$$|\Im(\phi_i)/(2\pi)| = |\omega_i| < \Omega/2, \quad i = 1, \dots, n,$$

and we sample $\phi(t)$ at the equidistant points $t_j = j\Delta$ for $j = 0, 1, \dots, 2n-1, \dots, N$ with $\Delta \leq 1/\Omega$. In the sequel we denote

$$f_j := \phi(t_j), \quad j = 0, 1, \dots, 2n-1, \dots, N.$$

The aim is to find the model order n , and the parameters ϕ_1, \dots, ϕ_n and $\alpha_1, \dots, \alpha_n$ from the measurements $f_0, \dots, f_{2n}, \dots$ We further denote

$$\lambda_i := \exp(\phi_i \Delta), \quad i = 1, \dots, n.$$

With

$$H_n^{(r)} := \begin{pmatrix} f_r & \dots & f_{r+n-1} \\ \vdots & \ddots & \vdots \\ f_{r+n-1} & \dots & f_{r+2n-2} \end{pmatrix}, \quad r \geq 0, n \geq 1,$$

the λ_i are retrieved [10] as the generalized eigenvalues of the problem

$$H_n^{(1)} v_i = \lambda_i H_n^{(0)} v_i, \quad i = 1, \dots, n, \quad (2)$$

where v_i are the generalized right eigenvectors. From the values λ_i , the complex numbers ϕ_i can be retrieved uniquely because of the restriction $|\Im(\phi_i \Delta)| < \pi$. In a noisy context the Hankel matrices can be extended to be rectangular of size $m \times n$ with $m > n$ and equation (2) can be considered in a least square sense [11].

In the absence of noise, the exact value for n can be deduced from [12, p. 603] (for a detailed discussion see [13])

$$\begin{aligned} \det H_n^{(r)} &\neq 0, \\ \det H_\nu^{(r)} &= 0, \quad \nu > n. \end{aligned} \quad (3)$$

In the presence of noise and/or clusters of eigenvalues, the use of (3) is not very reliable though. We indicate in Section 5 how n is then to be detected numerically.

Finally, the α_i are computed from the interpolation conditions

$$\sum_{i=1}^n \alpha_i \exp(\phi_i t_j) = f_j, \quad j = 0, \dots, 2n-1, \quad (4)$$

either by solving the system in the least squares sense, in the presence of noise, or by solving a subset of n (consecutive) interpolation conditions in case of a noisefree $\phi(t)$. Note that

$$\exp(\phi_i t_j) = \lambda_i^j$$

and that the coefficient matrix of (4) is therefore a Vandermonde matrix. It is well-known that the conditioning of structured matrices is something that needs to be monitored [14, 15].

3. EXPONENTIAL ANALYSIS VIEWED AS PADÉ APPROXIMATION

With $f_j = \phi(t_j)$ we now define the noisefree

$$f(z) = \sum_{j=0}^{\infty} f_j z^j. \quad (5)$$

The Padé approximant $r_{m,n}(z)$ of degree m in the numerator and n in the denominator is defined as the irreducible form of the rational function $p(z)/q(z)$ satisfying

$$\frac{d^j(fq-p)(z)}{dt^j}(0) = 0, \quad j = 0, \dots, m+n.$$

This condition guarantees a high degree of contact between $f(z)$ and $p(z)/q(z)$. Since

$$f_j = \sum_{i=1}^n \alpha_i \exp(\phi_i j \Delta) = \sum_{i=1}^n \alpha_i \lambda_i^j,$$

we can rewrite

$$f(z) = \sum_{i=1}^n \frac{\alpha_i}{1 - z \lambda_i}. \quad (6)$$

So we see that $f(z)$ is itself a rational function of degree $n-1$ in the numerator and n in the denominator, with poles $1/\lambda_i$. Hence, from Padé approximation theory we know that $r_{n-1,n}(z)$ computed for (5) reconstructs $f(z)$, in other words

$$r_{n-1,n}(z) = f(z).$$

The partial fraction decomposition (6) is related to the Laplace transform of the exponential model (1), which explains why this approach is known as the Padé-Laplace method [16].

Now we add a white circular Gaussian noise term ϵ_j to each sample f_j . In the sequel we denote the noisy series by

$$f(z) + \epsilon(z) = \sum_{j=0}^{\infty} (f_j + \epsilon_j) z^j.$$

A number of strong approximation and convergence results exist for sequences of Padé approximants to $f(z) + \epsilon(z)$. They express what one would expect intuitively from such approximants: they are especially useful if the approximated function is meromorphic (i.e. has poles) in some substantial region of the complex plane [17], as is the case for $f(z)$ given by (6). The theorem of Nuttall, later generalized by Pommerenke, states that if $f(z) + \epsilon(z)$ is analytic throughout the complex plane except for a countable number of poles [18] and essential singularities [19], then the paradiagonal sequence $\{r_{\nu-1,\nu}(z)\}_{\nu \in \mathbb{N}}$ converges to $f(z) + \epsilon(z)$ in measure on compact sets. So no assertion is made about pointwise or uniform convergence. Instead, the result states that for sufficiently large ν , the measure of the set where the convergence is disrupted, so where $|f(z) + \epsilon(z) - r_{\nu-1,\nu}(z)| \geq \tau$ for some given threshold τ , tends to zero as ν tends to infinity. The pointwise convergence is actually disrupted by $\nu - n$ unwanted pole-zero combinations of the Padé approximants $r_{\nu-1,\nu}(z)$: near each spurious pole introduced by increasing the denominator degree beyond the true n , one finds an associated zero, the pole and zero effectively cancelling each other locally. These pole-zero doublets are referred to as Froissart doublets [20, 21, 22]. Because of the Padé convergence theorem, the n true physical poles can be identified as stable poles of successive $r_{\nu-1,\nu}(z)$, while the $\nu - n$ spurious nonphysical poles are distinguished by their instability [23, 24]. So these Froissart doublets offer a way to filter the noise $\epsilon(z)$ from the underlying signal $f(z)$ [25]. Because of their ability to model the noise, Froissart doublets should not be avoided in the computation, as in [26] and [27], but should be filtered out at a later stage in the computation, an approach we apply in this paper.

So n is revealed as the number of stable poles in successive Padé approximants $r_{\nu-1,\nu}(z)$ for (6). Before moving to an implementation of this idea, we need another tool.

4. REGULARIZATION OF THE PROBLEM BY DOWNSAMPLING

When replacing Δ by a multiple

$$\Delta(k) := k\Delta, \quad \Delta(k) \leq k/\Omega$$

and thus sampling at $t_{jk} = j\Delta(k) = jk\Delta$, the eigenvalues we retrieve from (2) are not λ_i , but

$$\lambda_i(k) = \lambda_i^k, \quad i = 1, \dots, n.$$

So we fill the Hankel matrices $H_n^{(r)}$ with the samples f_{ik} instead of the samples $f_i, i = 0, \dots, 2n - 1$. To avoid confusion we denote the latter ones by

$$H_n^{(r)}(k) := \begin{pmatrix} f_{rk} & \dots & f_{(r+n-1)k} \\ \vdots & \ddots & \vdots \\ f_{(r+n-1)k} & \dots & f_{(r+2n-2)k} \end{pmatrix}.$$

Again these Hankel matrices can be extended to be rectangular of size $m \times n$ with $m > n$. From $\lambda_i^k = \exp(k\phi_i\Delta)$ the imaginary part of ϕ_i cannot be retrieved uniquely anymore, because now

$$|\Im(k\phi_i\Delta)| < k\pi.$$

So aliasing may have occurred: because of the periodicity of the function $\exp(i2\pi k\omega_i\Delta)$ a total of k values in the $2k\pi$ wide interval can be identified as plausible values for $2\pi\omega_i$. Note that when the original λ_i are clustered, the powered λ_i^k may be distributed quite differently and unclustered. Such a relocation of the generalized eigenvalues may significantly improve the conditioning of the Hankel matrices involved.

What remains is to investigate how to solve the aliasing problem in the imaginary parts $2\pi\omega_i$ of the ϕ_i . So far, from λ_i^k , we only have aliased values for ω_i . But this aliasing can be fixed at the expense of a small number of additional samples. In what follows n can everywhere be replaced by $\nu > n$ when using $\nu - n$ additional terms to model the noise, and all square $\nu \times \nu$ matrices can be replaced by rectangular $\mu \times \nu$ counterparts with $\mu > \nu$. To fix the aliasing, we add n samples to the set $\{f_0, f_k, \dots, f_{(2n-1)k}\}$, namely at the shifted points

$$\begin{aligned} t_{kj+\kappa} &= jk\Delta + \kappa\Delta = jk\Delta + \kappa\Delta, \\ j &= r, \dots, r+n-1, \quad 0 \leq r \leq n. \end{aligned}$$

An easy choice for κ is a small number relatively prime with k (for the most general choice allowed, we refer to [28]). With the additional samples we proceed as follows.

From the samples $\{f_0, f_k, \dots, f_{(2n-1)k}, \dots\}$ we compute the generalized eigenvalues $\lambda_i^k = \exp(\phi_i k\Delta)$ and the coefficients α_i going with λ_i^k in the model

$$\begin{aligned} \phi(jk\Delta) &= \sum_{i=1}^n \alpha_i \exp(\phi_i jk\Delta) \\ &= \sum_{i=1}^n \alpha_i \lambda_i^{jk}, \quad j = 0, \dots, 2n-1. \end{aligned} \quad (7)$$

So we know which coefficient α_i goes with which generalized eigenvalue λ_i^k , but we just cannot identify the correct $\Im(\phi_i)$ from λ_i^k . The samples at the additional points $t_{jk+\kappa}$ satisfy

$$\begin{aligned} \phi(jk\Delta + \kappa\Delta) &= \sum_{i=1}^n \alpha_i \exp(\phi_i(jk + \kappa)\Delta) \\ &= \sum_{i=1}^n (\alpha_i \lambda_i^\kappa) \lambda_i^{jk}, \quad j = r, \dots, r+n-1, \end{aligned} \quad (8)$$

which can be interpreted as a linear system with the same coefficients matrix as (7), but now with a new left hand side and unknowns $\alpha_1 \lambda_1^\kappa, \dots, \alpha_n \lambda_n^\kappa$ instead of $\alpha_1, \dots, \alpha_n$. And again we

can associate each computed $\alpha_i \lambda_i^\kappa$ with the proper generalized eigenvalue λ_i^k . Then by dividing the $\alpha_i \lambda_i^\kappa$ computed from (8) by the α_i computed from (7), for $i = 1, \dots, n$, we obtain from λ_i^κ a second set of κ plausible values for ω_i . Because of the fact that we choose κ and k relatively prime, the two sets of plausible values for ω_i have only one value in their intersection [29]. Thus the aliasing problem is solved.

5. ADDING VALIDATION TO THE ANALYSIS METHOD

The Padé view from Section 3 can now nicely be combined with the regularization technique from Section 4. The downsampling option uses only $1/k$ of the overall sequence of samples $f_0, f_1, \dots, f_{2\nu-1}, \dots$ taken at equidistant points $t_j = j\Delta, j = 0, 1, 2, \dots$, plus an additional set of samples at shifted locations to get rid of some possible aliasing effect. So for a fixed $k > 0$ the full sequence of samples points can be divided into k downsampled subsequences

$$\begin{aligned} t_0, t_k, t_{2k}, \dots \\ t_1, t_{k+1}, t_{2k+1}, \dots \\ \vdots \\ t_{k-1}, t_{2k-1}, t_{3k-1}, \dots \end{aligned}$$

For each downsampled subsequence $t_{\ell+kj}, \ell = 0, \dots, k-1$ a sequence of shifted sample points $t_{\ell+kj+\kappa}$ can also be extracted from the original full sequence t_j , as long as $\gcd(k, \kappa) = 1$. Actually, the computation of λ_i^κ can be improved numerically by considering the following shift strategy instead of a single shift. After the first shift by κ of the sample points to $t_{\ell+kj+\kappa}$, multiples of the shift κ can be considered. By sampling at $t_{\ell+kj+h\kappa}, h = 1, \dots, H = \lfloor (N - k - 1 - 2nk)/\kappa \rfloor$ the values $\alpha_i \lambda_i^\kappa, \alpha_i \lambda_i^{2\kappa}, \dots, \alpha_i \lambda_i^{H\kappa}$ are obtained, which can be considered as the samples $\psi(1), \psi(2), \dots, \psi(H)$ of a mono-exponential analysis problem

$$\psi(h) = \alpha_i \lambda_i^{\kappa h}. \quad (9)$$

From these we can compute λ_i^κ using (2) for a single exponential term. For this the Hankel matrices are again best enlarged to rectangular matrices. This sparse interpolation replaces the division $\alpha_i \lambda_i^\kappa / \alpha_i$ by a more accurate procedure.

By repeating the computation of λ_i^k from (7) and λ_i^κ from (9) for each $\ell = 0, \dots, k-1$, we obtain k independent problems of the form (8) instead of just one. In each of these – assuming that we overshoot the true number of components n in (7) and (8) by considering $\nu - n$ – the true parameters ϕ_i from the model (1) appear as n stable poles in the Padé-Laplace method and the $\nu - n$ spurious noisy poles behave in an unstable way. In fact, each downsampled sequence can be seen as a different noise realization while the underlying function $\phi(t)$ remains the same. So the generalized eigenvalues related to the signal $\phi(t)$ cluster near the true λ_i^k , while the other generalized eigenvalues belong to independent noise realizations and do not form clusters anywhere [23, 24].

The gain of considering k downsampled multi-exponential analysis problems rather than one large problem is twofold:

- A cluster analysis algorithm can detect the number of clusters in the complex plane, and hence deliver the number n of components in the model of the form (1). So there is no need to estimate the number n separately, by means of an SVD of $H_\nu^{(0)}$ with $\nu > n$ for instance.

- Because the physically meaningful generalized eigenvalues form clusters of (about) k elements, their value can be estimated more accurately by computing the center of gravity of each cluster, where the cluster radius is a function of the ill-disposedness of that specific λ_i [15].

The cluster analysis method used in the examples below is DBSCAN [30]. Since the cluster radii may vary, we typically perform two runs of DBSCAN with different parameter settings. In a first run we retrieve the clusters with higher density, while a second run allows to detect the less dense clusters of generalized eigenvalues.

After obtaining a center of gravity as approximation for the λ_i^k and a center of gravity as approximation for the λ_i^κ associated to the λ_i^k , the intersection of the solution sets for ω_i can be taken. We simply build a distance matrix and look for the closest match between

$$\omega_i : \exp(k\phi_i\Delta) = \exp(k(\psi_i + i2\pi\omega_i)\Delta) = \lambda_i^k$$

and

$$\omega_i : \exp(\kappa\phi_i\Delta) = \exp(\kappa(\psi_i + i2\pi\omega_i)\Delta) = \lambda_i^\kappa.$$

We point out and emphasize that the above technique can be combined with any implementation to solve problem (1), more precisely (7) and (8), popular methods being [31, 10, 11].

To illustrate the validation aspect and how it is robust in the presence of outliers, we consider 400 audio samples of the sustained part of an A4 note played by a trumpet, corrupted by an outlier as shown in Figure 1. We put $k = 4$ and $\kappa = 3$ and compare the validation to a standard implementation of ESPRIT. As can be seen in the ESPRIT reconstruction in Figure 1 (top), it suffers from the presence of the outlier, as any parametric method would. The new method, illustrated in Figure 1 (bottom), deals with k decimated signals instead of the full signal and is more robust. In both approaches we choose $n = 20$. While the ESPRIT algorithm deals with a Hankel matrix of size 260×141 , the cluster analysis only needs Hankel matrices of size 70×30 . When the recording is corrupted by an outlier, here only one of the k decimated signals is affected. The decimated sample set that contains the outlier does not contribute to the formed clusters. But the cluster algorithm still detects clusters composed of at least $k - 1$ eigenvalues at the correct locations λ_i^k . Since one easily identifies the decimated signal that did not contribute to all clusters, the equations coming from that set of samples and contributing to (4) for the computation of the α_i , are best removed from the Vandermonde system.

6. ILLUSTRATION ON A HARMONIC SOUND

We consider a recorded sound of a guitar playing a D3 note corrupted by electrical humming [32], downloaded from the website freesound.org. The samples are collected at a rate of 48 kHz, for a duration of about 9 seconds in total (454 071 sample points t_j). We apply the method described above to audio windows of 1024 samples, with an overlap of 75% between the windows. The goal is to extract the sinusoidal tracks [33] that form the guitar partials. We choose the downsampling factor $k = 5$ and take $\kappa = 7$ (other combinations work as well, of course). So in each window the downsampled set contains 204 samples, which we use to extract $\nu = 61$ generalized eigenvalues, leaving us to Hankel matrices of size (at most) 143×61 . For the solution of (7) we use

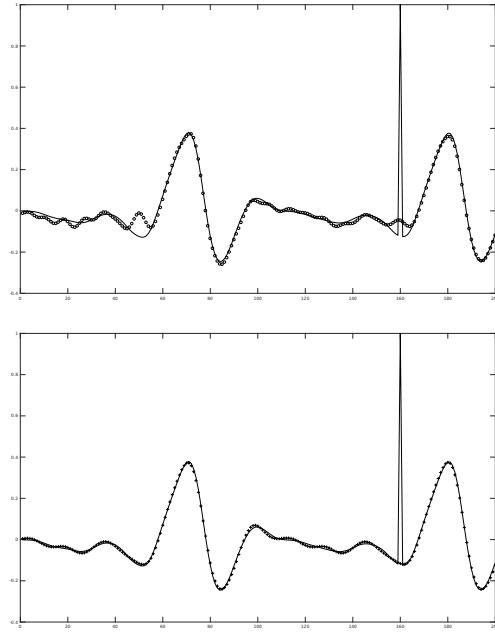


Figure 1: Corrupted trumpet recording reconstructed by ESPRIT (top, circles) and by the new method (bottom, crosses).

the ESPRIT algorithm [31]. After superimposing the $k = 5$ analyses of the downsampled audio windows, a cluster analysis using DBSCAN is performed for each window, thus retrieving the generalized eigenvalues λ_i^k most accurately.

To illustrate the regularization effect on the rectangular 143×61 analogon of (2) from choosing $k > 1$, we show in Figure 2 the distribution of the generalized eigenvalues $\lambda_i, i = 1, \dots, n$ of the full not downsampled 60-th windows starting at t_{15104} opposed to that of the $\lambda_i^5, i = 1, \dots, n$ of the downsampled set of samples from the same window.

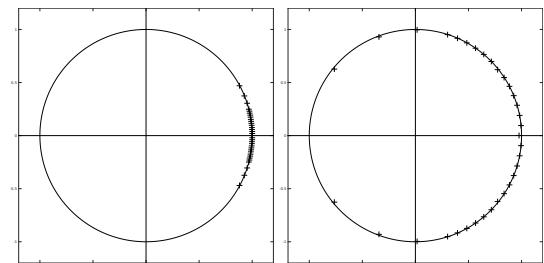


Figure 2: Generalized eigenvalues λ_i (left) versus λ_i^5 (right) from the 60-th window.

In the wake of the shift strategy discussed in Section 5 (we merely took $h = 1, \dots, 24 < H$), the value λ_i^κ can further be improved. After performing the cluster analysis on the superimposed results for λ_i^k , we can look at the λ_i^κ associated with each of these and compute their center of gravity (disregarding those that fall out of scope). Note that for the λ_i^κ no separate cluster analysis needs to be performed. The latter is illustrated in Figure 3 for $i = 16$, corresponding to the 16-th harmonic partial.

Since the technique is being applied to a harmonic sound, an

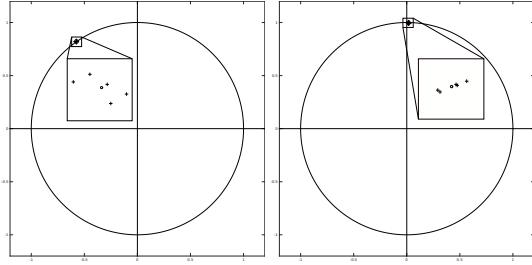


Figure 3: Cluster of λ_{15}^5 values (left) and λ_{15}^7 values (right).

additional step can be performed to estimate the base frequency (per window) more accurately. Once the stable frequencies $\phi_i, i = 1, \dots, n$ are retrieved, we look for a harmonic relation between them. We divide every detected harmonic partial ϕ_i by the integers $j = 1, \dots, 40$ (which is the largest number of partials expected) and we add these quotients to the discovered ϕ_i , in this way creating a new larger cluster at the base frequency, which we call ϕ_1 . The center of gravity of this larger cluster estimates the lowest partial of the harmonics. Using this estimate of the base frequency ϕ_1 , all higher harmonic partials $j\phi_1, j = -60, \dots, 60$ are reconstructed and substituted in one large rectangular 512×121 Vandermonde system (4), which serves as the coefficient matrix for the computation of the $\alpha_i, i = 1, \dots, 121$.

While moving from one window to the next over the course of the 9 seconds, the higher harmonic partials that are detected become weaker and fewer. So n decreases with time. We refer to Figure 4, where we again show the generalized eigenvalues λ_i and λ_i^5 , before and after regularization, now for one of the middle audio windows. Fortunately, the number of partials remains large enough during the whole audio fragment to rebuild the harmonics as described. Since the final reconstructed guitar sound only makes use of the ϕ_i from the stable generalized eigenvalues, the reconstruction, which can be downloaded from <https://www.uantwerpen.be/br-cu-ea-val-17/>, does not suffer from the electrical hum anymore.

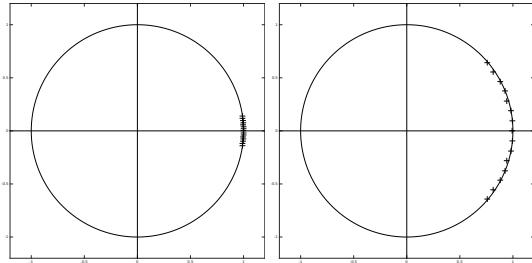


Figure 4: Generalized eigenvalues λ_i (left) versus λ_i^5 (right) from a middle window.

7. CONCLUSION

We present an approach that can be embedded in several parametric methods. We exploit the aliasing phenomenon caused by sub-Nyquist sampling and the connections with sparse interpolation and Padé approximation. The result is a parametric method that is able to discern between the stable and unstable components of

a multi-exponential model. It can thus remove outliers and/or the noisy part of the signal. We illustrate our approach on a harmonic sound where the validated output is used to refine the estimation of the lowest partial and reconstruct the signal thus eliminating the electrical humming present in the recording.

In our case study, we use a quasi-stationary sound and thus have not fully exploited the potential of the presented method yet. In the future we plan to apply the techniques described in Section 5 to decaying signals and signals with modulated amplitudes. Our illustrations pave the way for a wider field of unexplored applications and connections. Actually, every audio algorithm that makes use of a parametric method, may benefit from the ideas presented here.

8. REFERENCES

- [1] S. Marchand, “Fourier-based methods for the spectral analysis of musical sounds,” in *21st European Signal Processing Conference (EUSIPCO 2013)*, Sept 2013, pp. 1–5.
- [2] R. Badeau, G. Richard, and B. David, “Performance of ESPRIT for estimating mixtures of complex exponentials modulated by polynomials,” *IEEE Transactions on Signal Processing*, vol. 56, no. 2, pp. 492–504, Feb 2008.
- [3] J. Nieuwenhuijse, R. Heusens, and E. F. Deprettere, “Robust exponential modeling of audio signals,” in *Acoustics, Speech and Signal Processing, 1998. Proceedings of the 1998 IEEE International Conference on*, May 1998, vol. 6, pp. 3581–3584 vol.6.
- [4] J. Jensen, R. Heusdens, and S. H. Jensen, “A perceptual subspace approach for modeling of speech and audio signals with damped sinusoids,” *IEEE Transactions on Speech and Audio Processing*, vol. 12, no. 2, pp. 121–132, March 2004.
- [5] Kris Hermus, Werner Verhelst, Philippe Lemmerling, Patrick Wambacq, and Sabine Van Huffel, “Perceptual audio modeling with exponentially damped sinusoids,” *Signal Processing*, vol. 85, no. 1, pp. 163 – 176, 2005.
- [6] Romain Couillet, “Robust spiked random matrices and a robust G-MUSIC estimator,” *Journal of Multivariate Analysis*, vol. 140, pp. 139–161, 2015.
- [7] R Roy and T Kailath, “Total least squares ESPRIT,” in *Proc. of 21st Asilomar Conference on Signals, Systems, and Computers*, 1987, pp. 297–301.
- [8] Annie Cuyt and Wen-shin Lee, “Smart data sampling and data reconstruction,” Patent PCT/EP2012/066204.
- [9] Annie Cuyt and Wen-shin Lee, “Sparse interpolation and rational approximation,” 2016, vol. 661 of *Contemporary Mathematics*, pp. 229–242, American Mathematical Society.
- [10] Yingbo Hua and Tapan K. Sarkar, “Matrix pencil method for estimating parameters of exponentially damped/undamped sinusoids in noise,” *IEEE Trans. Acoust. Speech Signal Process.*, vol. 38, pp. 814–824, 1990.
- [11] Delin Chu and Gene H. Golub, “On a generalized eigenvalue problem for nonsquare pencils,” *SIAM Journal on Matrix Analysis and Applications*, vol. 28, no. 3, pp. 770–787, 2006.
- [12] P. Henrici, *Applied and computational complex analysis I*, John Wiley & Sons, New York, 1974.

- [13] Erich Kaltofen and Wen-shin Lee, “Early termination in sparse interpolation algorithms,” *J. Symbolic Comput.*, vol. 36, no. 3-4, pp. 365–400, 2003, International Symposium on Symbolic and Algebraic Computation (ISSAC’2002) (Lille).
- [14] Walter Gautschi, “Norm estimates for inverses of Vandermonde matrices,” *Numer. Math.*, vol. 23, pp. 337–347, 1975.
- [15] B. Beckermann, G.H. Golub, and G. Labahn, “On the numerical condition of a generalized Hankel eigenvalue problem,” *Numer. Math.*, vol. 106, no. 1, pp. 41–68, 2007.
- [16] Zeljko Bajzer, Andrew C. Myers, Salah S. Sedarous, and Franklyn G. Prendergast, “Padé-Laplace method for analysis of fluorescence intensity decay,” *Biophys J.*, vol. 56, no. 1, pp. 79–93, 1989.
- [17] G.A. Baker, Jr. and P. Graves-Morris, *Padé approximants* (2nd Ed.), vol. 59 of *Encyclopedia of Mathematics and its Applications*, Cambridge University Press, 1996.
- [18] J. Nuttall, “The convergence of Padé approximants of meromorphic functions,” *J. Math. Anal. Appl.*, vol. 31, pp. 147–153, 1970.
- [19] Ch. Pommerenke, “Padé approximants and convergence in capacity,” *J. Math. Anal. Appl.*, vol. 41, pp. 775–780, 1973.
- [20] J.L. Gammel, “Effect of random errors (noise) in the terms of a power series on the convergence of the Padé approximants,” in *Padé approximants*, P.R. Graves-Morris, Ed., 1972, pp. 132–133.
- [21] J. Gilewicz and M. Pindor, “Padé approximants and noise: a case of geometric series,” *J. Comput. Appl. Math.*, vol. 87, pp. 199–214, 1997.
- [22] J. Gilewicz and M. Pindor, “Padé approximants and noise: rational functions,” *J. Comput. Appl. Math.*, vol. 105, pp. 285–297, 1999.
- [23] P. Barone, “On the distribution of poles of Padé approximants to the Z-transform of complex Gaussian white noise,” *Journal of Approximation Theory*, vol. 132, no. 2, pp. 224 – 240, 2005.
- [24] L. Perotti, T. Regimbau, D. Vrinceanu, and D. Bessis, “Identification of gravitational-wave bursts in high noise using padé filtering,” *Phys. Rev. D*, vol. 90, pp. 124047, Dec 2014.
- [25] D. Bessis, “Padé approximations in noise filtering,” *J. Comput. Appl. Math.*, vol. 66, pp. 85–88, 1996.
- [26] Pedro Gonnet, Stefan Güttel, and Lloyd N. Trefethen, “Robust Padé approximation via SVD,” *SIAM Rev.*, vol. 55, pp. 101–117, 2013.
- [27] O.L. Ibryaeva and V.M. Adukov, “An algorithm for computing a Padé approximant with minimal degree denominator,” *J. Comput. Appl. Math.*, vol. 237, no. 1, pp. 529–541, 2013.
- [28] Annie Cuyt and Wen-shin Lee, “An analog Chinese Remainder Theorem,” Tech. Rep., Universiteit Antwerpen, 2017.
- [29] Annie Cuyt and Wen-shin Lee, “How to get high resolution results from sparse and coarsely sampled data,” Tech. Rep., Universiteit Antwerpen, 2017.
- [30] Martin Ester, Hans-Peter Kriegel, Jörg Sander, and Xiaowei Xu, “A density-based algorithm for discovering clusters in large spatial databases with noise,” in *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining*. 1996, KDD’96, pp. 226–231, AAAI Press.
- [31] R. Roy and T. Kailath, “ESPRIT-estimation of signal parameters via rotational invariance techniques,” *IEEE Trans. Acoust. Speech Signal Process.*, vol. 37, no. 7, pp. 984–995, July 1989.
- [32] Frederic Font, Gerard Roma, and Xavier Serra, “Freesound technical demo,” in *Proceedings of the 21st ACM International Conference on Multimedia*, New York, NY, USA, 2013, MM ’13, pp. 411–412, ACM.
- [33] Xavier Serra and Julius 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, 1990.
- [34] R. Schmidt, “Multiple emitter location and signal parameter estimation,” *IEEE Transactions on Antennas and Propagation*, vol. 34, no. 3, pp. 276–280, 1986.

A SYSTEM BASED ON SINUSOIDAL ANALYSIS FOR THE ESTIMATION AND COMPENSATION OF PITCH VARIATIONS IN MUSICAL RECORDINGS

Luís F. V. de Carvalho *

Electrical Eng. Program - COPPE/SMT,
Federal University of Rio de Janeiro
Rio de Janeiro, Brazil
luis.carvalho@smt.ufrj.br

Hugo T. de Carvalho

Institute of Mathematics
Federal University of Rio de Janeiro
Rio de Janeiro, Brazil
hugo.carvalho@smt.ufrj.br

ABSTRACT

This paper presents a computationally efficient and easily interactive system for the estimation and compensation of speed variations in musical recordings. This class of degradation can be encountered in all types of analog recordings and is characterized by undesired pitch variations during the playback of the recording. We propose to estimate such variations in the digital counterpart of the analog recording by means of sinusoidal analysis, and these variations are corrected via non-uniform resampling. The system is evaluated for both artificially degraded and real audio recordings.

1. INTRODUCTION

The problem of speed variations in old recordings is quite ubiquitous: for example, the puncture of vinyl and gramophone disks could be not well centered, and this kind of media, when subject to high temperature, could be bent; also, poorly stored magnetic tapes can be stretched. In both cases, when the degraded media is reproduced the playback speed will not be constant, causing an effect that is perceived as a pitch variation along the signal. Because of this audible effect, this defect is also known as “wow” in the literature. When considering historical collections, where it is very common to have only one copy of the recording available, it is then important to develop methods to identify and remove this effect of the degraded recording.

The study of quantification of “wow” dates back to the 40’s [1, 2]. Depending on the cause of the degradation, mechanical methods can be used to restore such recordings: for example, correctly centering the puncture on a disk is a quite efficient way of undoing the degradation, but it only works in this particular case. For more general causes of this degradation, more sophisticated methods are required, that enable the use of digital signal processing, since the basic idea behind all the proposed restoration methods is to re-sample the degraded signal in a non uniform way such that the speed variation is compensated [3]. Therefore, it is then necessary to firstly estimate the so-called *pitch variation curve* (PVC) from the degraded signal. Then, a time-varying resampling algorithm is applied on the PVC. One of the first proposed methods following this guideline is [4, 5], where the curve is estimated via a statistical procedure from the spectrogram of the degraded signal and then used to resample it. A drawback of this method is that it is quite computationally intensive. This same idea was also explored in [6, 7], where an improved method for estimating the

peaks in the spectrogram is proposed, as well as a different modeling for the pitch variation curve is employed. Since this modeling is parametric and sinusoidally-based, it can fail to describe more general curves. Other methods for determining the distortion are proposed in [8], although they are difficult to implement. Also in [8] an extensive discussion and comparison of several estimation methods is presented. Lastly, commercial tools are also available, for instance Capstan¹.

In this paper we propose a computationally efficient and non-parametric method which requires low amount of user interaction for determining the PVC based on a sinusoidal analysis of the degraded signals, as well as a time-varying resampling scheme that uses the estimated curve to restore the degraded signal.

The paper is organized as follows: in Section 2 an outline of the proposed solution is presented, and the next sections describe each step in more details; Section 3 presents the peak detection method employed in the sinusoidal analysis, followed by the peak tracking algorithm in Section 4; in Section 5 it is shown how the PVC is obtained from the estimated tracks, and Section 6 describes how the PVC is used in the time-varying resampling algorithm; results are presented and conclusions are drawn in Sections 7 and 8, respectively.

2. OUTLINE OF THE PROPOSED SOLUTION

The proposed solution has essentially three steps, as shown in Figure 1: the degraded signal is given as input to a sinusoidal analysis algorithm, whose estimated tracks are used to obtain the PVC, subsequently used in the time-varying resampling algorithm in order to restore the degraded signal. In this Section we briefly recall some aspects of the sinusoidal analysis and outline how these aforementioned steps are interconnected.

Sinusoidal analysis is a well-known multi-purpose technique in audio processing where small excerpts of a digital audio signal $x[n]$ are described as a sum of sinusoidal components [9]:

$$x[n] = \sum_j A_j[n] \cos(\Psi_j[n]), \quad (1)$$

where $A_j[n]$ and $\Psi_j[n]$ represent the time-varying envelope and the phase modulations of each component j , respectively.

The main goal is to estimate the parameters $A_j[n]$ and $\Psi_j[n]$ from the respective audio excerpt. With this set of parameters in hand, several tasks could be performed, for example, feature extraction or some modification and posterior re-synthesis of the signal [9]. In our case, this framework is used to estimate the PVC, based on the idea that all the frequencies present in an audio signal

* The authors thank CAPES and CNPq Brazilian agencies for funding this work.

¹<http://www.celemony.com/en/capstan>

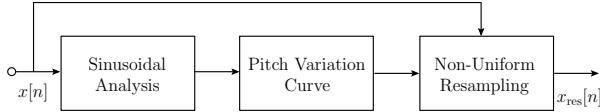


Figure 1: Main steps of the proposed method.

degraded by speed variation must contain some degree of deviation, more discussed in Section 5. Therefore, in this work it is more important to estimate the frequencies and amplitudes present within short excerpts of the audio signal than other quantities, and this estimation is performed as follows (see [10] for more details):

1. The whole signal $x[n]$ is segmented, with each segment being multiplied by a window function $w[n]$ (here the Hann window was employed) of length N_W , and contiguous segments have an overlap of N_H samples. Denote the n -th sample of the b -th block as $x_b[n]$, for $n = 1, \dots, N_W$;
2. The N_{FFT} -point DFT of each segment is computed, its result being denoted by $X(k, b)$, representing the coefficient of the k -th frequency bin in the b -th block.
3. The most prominent peaks of each block that are more likely to be related to tonal components of the underlying signal are estimated via a procedure described in Section 3, based on [11];
4. Finally, in order to form tracks with the peaks estimated in the previous step, a simple algorithm is applied, where essentially a peak in some particular block is associated with the closest peak in the following block. This is a modification of the MQ algorithm [12], and is explained in more details in Section 4. The frequency of the i -th track in the b -th block of signal is denoted by $f_i[b]$.

Now, using an weighted average of the previously obtained tracks, the PVC is estimated via a simple procedure described in detail in Section 5. Finally, the PVC is given, together with the degraded signal, as input to a non-uniform resampling algorithm, detailed in Section 6. In the next Sections, the aforementioned steps are thoroughly discussed.

3. PEAK DETECTION

Since the presented method for estimating the PVC is based on sinusoidal analysis, it is important to employ a peak detection algorithm that rejects those noise-induced. Several methodologies have been proposed to detect tonal peaks in audio signals, including threshold-based methods [9, 13] and statistical analysis [14].

To separate genuine peaks from spurious ones, we adopt the *Tonalness Spectrum*, introduced in [11], since it is an easily extensible and flexible framework for sinusoidal analysis. This is a non-binary representation which indicates the likelihood of a spectral bin to be a tonal or non-tonal component. In this metric, a set of spectral features $\mathbb{V} = \{v_1, v_2, \dots, v_V\}$ is computed from the signal spectrum and combined to produce the overall tonalness spectrum:

$$\mathcal{T}(k, b) = \left(\prod_{i=1}^V t_i(k, b) \right)^{1/\eta}, \quad (2)$$

where $t_i(k, b) \in [0, \dots, 1]$, which is referred to as the *specific tonal score*, is calculated for each extracted feature v_i according to:

$$t_i(k, b) = \exp \left\{ - [\epsilon_i \cdot v_i(k, b)]^2 \right\}. \quad (3)$$

This measure can be explained as the probability of the given feature present a tonal component in the bin k of block b . The factor ϵ_i is a normalization constant which ensures that all specific features will equally contribute to the tonalness spectrum when combining them, and it is obtained by setting in Eq. 3 the specific tonal score of the median of the feature in each block to 0.5. This yields the following expression for the normalization constant:

$$\epsilon_i = \frac{\sqrt{\log(2)}}{m_{v_i}}, \quad (4)$$

where $m_{v_i}(b)$ is the median value of feature i in the block b and $\overline{m_{v_i}}$ is the mean over all blocks of all files (in the case that a dataset is being analyzed).

The feature set comprises simple and established features, including a few that are purely based on information from the current magnitude spectrum of a block, such as frequency deviation, peakiness and amplitude threshold; some that are based on spectral changes over time, such as amplitude and frequency continuity; and one feature that is based on the phase and amplitude of the signal, which is the time window center of gravity.

Although the proposed method for estimating speed variations is based on a time-frequency representation, the peak detection stage is characterized by analyzing each frame individually. Therefore we take into account only those features which extract information from a single block.

The evaluation of results in [11] showed that, although the combination of features intuitively and empirically performs better than individual scores, combinations of more than three features did not achieve better representations. Moreover, it was also reported that combination with a simple product, that is, setting η to 1 in Eq. 2, yielded better results than with the distorted geometric mean.

Taking these information into account, our tonalness spectrum is formed by the combination of the *amplitude threshold* and *peakiness* features, since in [15] the combination of these features achieved good results on the detection of fundamental frequencies and their harmonics in music signals. Therefore, in our case, the expression in Eq. 2 can be simplified to:

$$\mathcal{T}(k, b) = t_{PK}(k, b) \cdot t_{AT}(k, b), \quad (5)$$

with $t_{PK}(k, b)$ and $t_{AT}(k, b)$ being the specific tonal scores of the peakiness and amplitude threshold, respectively.

The peakiness feature measures the height of a spectral sample in relation to its neighboring bins, and it is defined as:

$$v_{PK}(k, b) = \frac{|X(k+p, b)| + |X(k-p, b)|}{|X(k, b)|}, \quad (6)$$

where the distance p to the central sample should approximately correspond to the spectral main lobe width of the adopted window function when segmenting the signal, so side lobe peaks can be avoided.

The amplitude threshold feature measures the relation of the magnitude spectrum by an adaptive magnitude threshold, and it is defined as:

$$v_{AT}(k, b) = \frac{r_{TH}(k, b)}{|X(k, b)|}, \quad (7)$$

where $r_{\text{TH}}(k, b)$ is a recursively smoothed version of the magnitude spectrum:

$$r_{\text{TH}}(k, b) = \beta \cdot r_{\text{TH}}(k - 1, b) + (1 - \beta) \cdot |X(k, b)|, \quad (8)$$

with this filter being applied in both forward and backward direction in order to adjust the group delay, and $\beta \in [0, 1]$ being a factor that is empirically tweaked.

3.1. Peak Selection

After computing the tonalness spectrum, all peaks k_i are selected in each block, and those which do not fulfill the following criterion are discarded:

$$\mathcal{T}(k_i, b) \geq \mathcal{T}_{\text{TH}}, \quad (9)$$

where $\mathcal{T}_{\text{TH}} \in [0, 1]$ is an empirically adjusted likelihood threshold.

Moreover, since our tonalness spectrum measurement evaluates peaky components and their surroundings, independently of their absolute magnitude amplitudes, some small and insignificant peaks could present a high tonalness likelihood and thus be selected. Hence the following criterion is employed in each block to discard such irrelevant peaks:

$$|X(k_i, b)| \geq \gamma \cdot \max|X(k, b)|, \quad (10)$$

where γ is a percentage factor, and satisfactory peak selections were obtained by setting this factor to around 1%.

Figure 2 illustrates the peak selection stage in a block of an audio signal containing a single note being played by a flute. The criteria in Eq. 9 and 10 were set to 0.8 and 1%, respectively. It can be seen that the tonalness spectrum is a powerful representation of tonal components, when comparing it with the original magnitude spectrum.

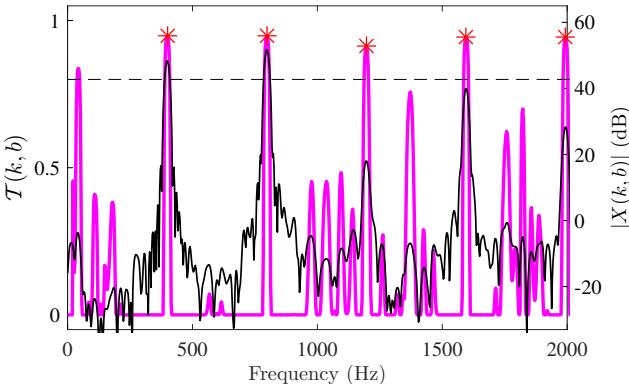


Figure 2: Illustration of the peak detection stage. The thicker and brighter line represents the tonalness spectrum of a block from an audio signal, whose magnitude spectrum is indicated on the thinner and darker line. The likelihood threshold is represented by the horizontal dashed line and the selected peaks are marked with asterisks.

4. PARTIAL TRACKING

To conclude the sinusoidal analysis stage, the spectral peaks selected in each frame are grouped into time-changing trajectories

in which both frequency and amplitude can vary. This process is referred to as *partial tracking*.

Several methods have been proposed to track spectral peaks. Relevant works include the classical McAuley & Quatieri (MQ) algorithm [16], its extended version using linear prediction [17], and a solution via recursive least-squares (RLS) estimation [18].

In this work, a modified version of the MQ algorithm is employed, which is described in [12], since this is a computationally efficient technique that is easily implementable and achieves good representation of sinusoidal tracks.

In this algorithm, a track can be marked with three different labels: it *emerges* when a peak is not associated with any existing track, *remains* active while it is associated with peaks, and *vanishes* when it finds no compatible peak to incorporate. Defining $f_{i,b}$ and $A_{i,b}$ the frequency and magnitude amplitude of the i th detected peak in frame b , the algorithm can be explained as follows:

1. For each peak $f_{j,b+1}$ a search is performed to find a peak $f_{i,b}$ from a track which had remained active until the frame b , satisfying the condition $|f_{i,b} - f_{j,b+1}| < \Delta f_{i,b}$. The parameter $\Delta f_{i,b}$ controls the maximum frequency variation, and is set to a quarter tone around $f_{i,b}$.
2. If the peak $f_{j,b+1}$ finds a corresponding track in the previous frame satisfying the condition described in step 1, it associates with this track, which remains active. If two or more peaks satisfy the condition, the peak that minimizes

$$J = (1 - \kappa) \frac{|f_{i,b} - f_{j,b+1}|}{f_{i,b}} + \kappa \frac{|A_{i,b} - A_{j,b+1}|}{A_{i,b}} \quad (11)$$

is selected, where $\kappa \in [0, 1]$ is a weighting parameter that controls the influence of the relative frequency and amplitude differences in the cost function in Eq. 11.

3. When the peak of a track in b is not associated with any peak in $b + 1$ satisfying the condition, it is marked as *vanishing* and a virtual peak with its same frequency and amplitude is created in $b + 1$. When the track reaches D consecutive virtual peaks, it is then terminated.

Except for the first frame, where all the peaks invariably start new tracks, these steps are performed in all frames, until all peaks are labeled. Lastly, short tracks whose length is less than a number E of frames are removed.

5. GLOBAL PITCH VARIATION CURVE

A consequence of speed variations in musical signals is the deviation of all their frequencies by the same percentage factor, that is, a pitch modification. Therefore, a curve which combines the variations of the main frequency components of the signal is a suitable metric for estimating the defect.

In [4, 5] a Bayesian procedure is proposed to estimate the pitch variation curve, but it is quite computationally expensive. An alternative is to determine the distortion using only the most prominent spectral component of the signal, a technique proposed in [8]. However, this method is not practical, and such spectral component may contain frequency modulations not corresponding to speed variations, interfering then with the correct curve estimation.

Our proposed approach consists of calculating a weighted average curve from the computed tracks, thus exhibiting an overall

behavior of how the tracks vary with time. If a global frequency variation in a part of the signal is detected, this might be a promising evidence of a speed variation in that part.

The weights in this average are the magnitude amplitudes of the detected peaks. The motivation for weighting the curve is that the most important tracks should contribute more to the PVC than the less prominent ones. This metric can be interpreted as an extended version of the method proposed in [8]. Since this metric takes into account more frequency components and it is refined by their magnitude amplitudes, it is expected that this overall pitch variation curve can be more reliable than taking only one component.

For aesthetic reasons, it is quite common the presence of notes played with ornaments in music recordings, such as vibrato. This is often seen in bowed string or woodwind instruments, and in singing voice as well. Thus, when tracking an instrument recording with such effect, parts where vibrato occurred could be erroneously detected as speed variations. However, if the instrument is in a recording with other musical instruments, which would not necessarily be synchronized with its vibrato, that effect would be attenuated by computing an average of the tracks. Nevertheless, the method presented in this paper can be implemented in a way that the user may select the specific parts of the audio signal to be restored.

5.1. Extraction of the weighted global average of the tracks

In the first step of the global pitch variation curve extraction, the mean of all tracks are shifted to zero, and then each of them are normalized by its respective frequency average in time, this way obtaining the percentage average of each sinusoidal trajectory. For a track i , this stage is mathematically described as:

$$\tilde{f}_i[b] = \frac{f_i[b] - \bar{f}_i}{\bar{f}_i}, \quad (12)$$

with $f_i[b]$ being the frequency of the i th track in the frame b , and \bar{f}_i being the arithmetic mean of all frequencies in the i th track.

Following that, the weighted average of each frame b is calculated:

$$\tilde{f}'[b] = \frac{\sum_i A_i[b]^{\alpha} \tilde{f}_i[b]}{\sum A_i[b]^{\alpha}}, \quad (13)$$

where $A_i[b]$ represents the magnitude amplitude of the i th track in the frame b and $\alpha \in [0, 1]$ is a parameter that controls the influence of large amplitudes over smaller ones, which we will from now on refer to as the *weighting factor*. The motivation for this parameter is that the magnitude amplitude of the harmonic components of tonal peaks drop sharply as the harmonic order increases, and one may desire to increase the participation of such medium and small amplitude harmonics in the computation of the pitch variation curve.

It can be noticed that when $\alpha = 0$, the Eq. 13 happens to be the arithmetic mean of the tracks, indicating that all tracks will equally contribute to the PVC; analogously, when $\alpha = 1$, the PVC is set so the small-amplitude peaks influence less in its estimation. To minimize the effects of false tracks, the weighting factor would naturally be set to a value close to 1, and empirical tests indicate that setting α between 0.7 and 1 achieve satisfying results.

The curve $\tilde{f}'[b]$ may present undesired small irregularities, which are caused by frequency inaccuracies and the tracking of non-tonal peaks which do not correspond to frequency components. Hence, this curve is smoothed by a moving average filter,

$$\tilde{f}[b] = \frac{1}{N_{MA}} \sum_{i=0}^{N_{MA}-1} \tilde{f}'[b+i], \quad (14)$$

where N_{MA} is the order of the filter, and the final PVC is then the vector \tilde{f} , whose components are $\tilde{f}[b]$, for $m = 1, \dots, N_B$, with the latter term being the total number of blocks.

It is then expected that the pitch variation curve of a signal exhibits values around zero. For a better interpretation of this curve, it is interesting to normalize it, which is realized by shifting its average to 1. This is a better notation when it is necessary to adopt a frequency reference. For example, curves related to parts in which there was no speed variations now exhibit values around 1, indicating that the frequencies of their tracks have all been multiplied by 1. If in a frame transition there is a deviation of 0.6%, this now is represented in the curve as 1.006, that is, all frequencies in this transition on average were multiplied by 1.006.

6. NON-UNIFORM RESAMPLING

In this section, it is described how non-uniform sampling rate conversion can be employed from the PVC to compensate speed variations in digital audio signals.

When an audio signal is played back with a different rate from that which was originally sampled, the perceived pitch is modified, as well as its time duration is distorted. It turns out, as explained in Section 1, that speed variations yield exactly pitch and time variations. Therefore sampling rate conversion is a suitable technique for compensating such defects in digital versions of degraded recordings.

Furthermore, since speed variations are not constant in time, this sampling rate conversion must be non-uniform, which can also be referred to as *time-varying resampling*. In [3] different methods for “wow” reduction are compared, and the sinc-based interpolation [19] achieved less distortions in reconstructed signals. Therefore, an algorithm that receives as input the digital version of the degraded signal and its PVC, and its output the reconstructed signal using the sinc-based time-varying resampling was implemented.

The sinc-based interpolation is closely related to the analog interpretation of resampling, which is the reconstruction of the continuous version from the discrete signal, followed by resampling it with the new desired rate. The expression for the converted signal $x_{rec}[n']$ with new rate F'_s using the sinc-based resampling is given by [19]:

$$x_{rec}[n'] = \sum_{n=-N_T}^{N_T} x[n] \operatorname{sinc}\left(\pi\left(\frac{n'}{f_r} - n\right)\right), \quad (15)$$

where $f_r = F'_s/F_s$ is the resampling factor, i.e. the ratio between the desired and original sampling rates, and $2N_T + 1$ is the number of samples used in the interpolation. The reconstruction of a sample is illustrated in the Figure 3.

6.1. Implementation

Since the pitch variation curve \tilde{f} from Eq. 14 is normalized to 1, each of its elements correspond exactly to the term f_r in Eq. 15. However, each element is associated to a block of the signal, which was segmented during the STFT procedure. Hence the transition between blocks must be modeled sample by sample.

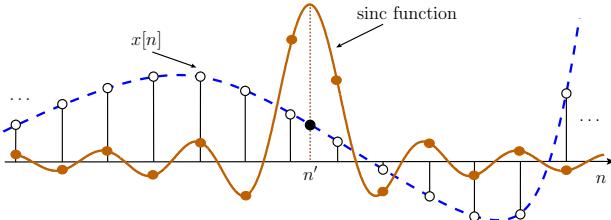


Figure 3: Illustration of the sinc-based interpolation method.

The following steps summarize the non-uniform resampling procedure that was implemented in this work, given a digital audio signal and its pitch variation curve:

1. Firstly, each element $\tilde{f}[b]$ of the vector \tilde{f} is associated to the central sample of its respective block;
2. Secondly, the number of samples which will be reconstructed between each pair of consecutive central samples of $\tilde{f}[b]$ and $\tilde{f}[b + 1]$ is calculated, so the transition between these central samples be smooth. This is obtained by realizing a linear interpolation to obtain the positions of each sample to be reconstructed (which can generically be represented by the sample in n' in Figure 3);
 - (a) This process inevitably involves numerical approximations, which in this context are translated into phase deviations. Such approximations must be taken into account, and therefore compensated in order to eliminate audible artifacts.
3. Finally, the expression in Eq. 15 is applied for each new sample.

It is also worth mentioning that the correction of speed variations can be performed offline, so N_T can be chosen large enough in order not to compromise the quality of the reconstructed signal. Experiments in [3] showed that using $N_T = 100$ achieved inaudible distortions. Moreover, our experiments with such number of samples or even less also did not degraded the signal.

7. RESULTS

This section presents the evaluation of the proposed algorithm. Several degraded signals were investigated, and two of them are shown in this paper, one from an artificially degraded recording, and one from a real recording. The audio signals presented here as well as all implemented codes in this work are available in [20]. All tests were realized with the same set of parameters, which are summarized in Table 1.

The first test was performed with an excerpt of orchestral music, containing long notes being played by bowed instruments. An artificial sinusoidal pitch variation was imposed into this signal and the proposed method was applied, so both true and estimated PVCs can be compared. Although the curves slightly differ in terms of their amplitudes, as can be seen in the first graph of Fig. 4, a satisfactory correspondence can be observed between them. As can be seen in the second graph of Fig. 4, the pitch variation curve of the resampled signal shows only slight variations, which can be explained by the amplitude difference mentioned before; however, informal listening tests reported no audible pitch variations. It is

Table 1: Parameters used to estimate the PVC.

Parameter	Value
N_W	4096 samples
N_H	256 samples
N_{FFT}	16384 samples
β	$1500/N_{FFT}$
T_{TH}	0.75
γ	1%
κ	0.6
D	5 blocks
E	10 blocks
α	0.8

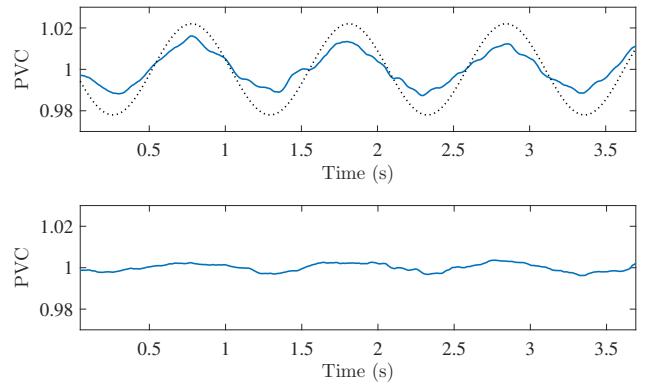


Figure 4: Restoration of the signal ‘orchestra’. In the first graph, the solid line and dotted line represent the estimated and true PVCs, respectively. The PVC of the restored signal is shown in the second graph.

also worth mentioning that there are no precise objective measurements to compare the original signal with the restored one.

The second test was realized on a piano recording which presents genuine speed variations. Figure 5 shows the PVCs of the degraded and restored signals, respectively. What sorts out from the graphs is that the proposed method estimated the shape of the variations with considerable accuracy, and therefore the result was considered satisfying. However, the PVC of the restored signal still presents some slight variations, but this can be explained by the inaccuracies of the sinusoidal analysis stage, more precisely during the partial tracking stage.

8. CONCLUSION

This work has presented a computationally efficient system which requires minimum user interaction for the estimation and correction of speed variations in the playback of musical recordings. The stage of estimating the pitch variation curve was based purely on sinusoidal analysis, and the good results indicated that the proposed framework can serve, for example, as the core of a more sophisticated and robust professional tool for audio restoration.

It can be said that the partial tracking stage appears as the less robust stage in the system, and it may have the biggest influence in the inaccuracies reported in Figures 4 and 5. Therefore, future works include the development of a more robust algorithm for peak tracking, possibly operating simultaneously with a group

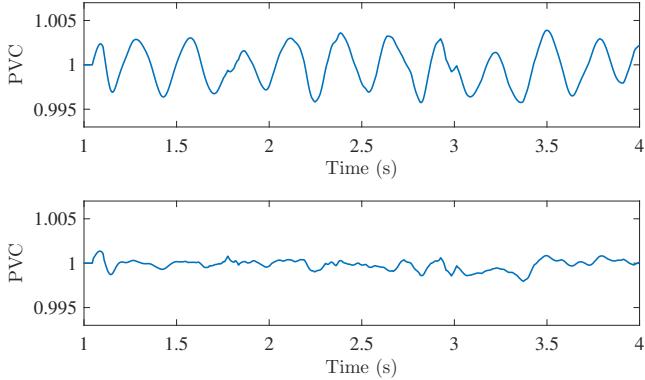


Figure 5: Restoration of the signal ‘piano’. The estimated PVC is depicted in the first graph, and the restored PVC is shown in the second one.

of blocks, as proposed in [17]. Other potential improvements to be implemented include the development of a better method for converting the set of tracks into the pitch variation curve, and the assessment of results via subjective tests.

As stated in [8], a fully automatic system, although tempting, is not feasible. Since the nature of pitch variations in old music recording is wide [5], this class of audio degradation requires a human operator for applying restoration algorithms. However, we believe that such systems can be minimally and friendly interactive, so non-professional users can restore their old domestic recordings by their own.

9. ACKNOWLEDGMENTS

The authors wish to express their sincere gratitude to Prof Luiz W. P. Biscainho for the motivation and ideas for this work. They also would like to thank the anonymous reviewers for their valuable comments and suggestions to improve the quality of the paper.

10. REFERENCES

- [1] U. R. Furst, “Periodic variations of pitch in sound reproduction by phonographs,” *Proceedings of the IRE*, vol. 34, no. 11, pp. 887–895, Nov 1946.
- [2] P. E. Axon and H. Davies, “A study of frequency fluctuations in sound recording and reproducing systems,” *Proceedings of the IEE - Part III: Radio and Communication Engineering*, vol. 96, no. 39, pp. 65–75, Jan 1949.
- [3] P. Maziewski, “Wow defect reduction based on interpolation techniques,” in *Proceedings of the 4th National Electronics Conference*, Darlowko Wschodnie, Poland, Jun 2005, pp. 481–486.
- [4] S. J. Godsill and P. J. W. Rayner, “The restoration of pitch variation defects in gramophone recordings,” in *Proceedings of IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*, Oct 1993, pp. 148–151.
- [5] S. J. Godsill, “Recursive restoration of pitch variation defects in musical recordings,” in *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP’94)*, Apr 1994, vol. 2, pp. 233–236.
- [6] J. Nichols, “An interactive pitch defect correction system for archival audio,” in *Proceedings of the AES 20th International Conference*, Budapest, Hungary, Oct 2001.
- [7] J. Nichols, “A high-performance, low-cost wax cylinder transcription system,” in *Proceedings of the AES 20th International Conference*, Budapest, Hungary, Oct 2001.
- [8] A. Czyzowski, A. Ciarkowski, A. Kaczmarek, J. Kotus, M. Kulesza, and P. Maziewski, “DSP techniques for determining wow distortion,” *Journal of the Audio Engineering Society*, vol. 55, no. 4, pp. 266–284, 2007.
- [9] J. Smith and X. Serra, “PARSHL: An analysis/synthesis program for non-harmonic sounds based on a sinusoidal representation,” in *International Computer Music Conference*, Urbana, Illinois, USA, Aug 1987, pp. 290–297.
- [10] P. A. A. Esquef and L. W. P. Biscainho, “Spectral-based analysis and synthesis of audio signals,” in *Advances in Audio and Speech Signal Processing: Technologies and Applications*, H. M. Pérez-Meana, Ed., chapter 3. Idea Group, Hershey, 2007.
- [11] S. Kraft, A. Lerch, and U. Zölzer, “The tonalness spectrum: feature-based estimation of tonal components,” in *Proceedings of the 16th International Conference on Digital Audio Effects (DAFx’14)*, Maynooth, Ireland, Sep 2014.
- [12] L. Nunes, L. Biscainho, and P. Esquef, “A database of partial tracks for evaluation of sinusoidal models,” in *Proceedings of the 13th International Conference on Digital Audio Effects (DAFx’10)*, Graz, Austria, Sep 2010.
- [13] N. Laurenti, G. De Poli, and D. Montagner, “A nonlinear method for stochastic spectrum estimation in the modeling of musical sounds,” *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 15, no. 2, pp. 531–541, Feb 2007.
- [14] D. J. Thomson, “Spectrum estimation and harmonic analysis,” *Proceedings of the IEEE*, vol. 70, no. 9, pp. 1055–1096, Sep 1982.
- [15] S. Kraft and U. Zölzer, “Polyphonic pitch detection by matching spectral and autocorrelation peaks,” in *Proceedings of the 23rd European Signal Processing Conference (EUSIPCO’15)*, Nice, France, Aug 2015, pp. 1301–1305.
- [16] R. McAulay and T. Quatieri, “Speech analysis/synthesis based on a sinusoidal representation,” *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 34, no. 4, pp. 744–754, Aug 1986.
- [17] M. Lagrange, S. Marchand, and J. B. Rault, “Using linear prediction to enhance the tracking of partials,” in *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP’04)*, May 2004, vol. 4, pp. 241–244.
- [18] L. Nunes, R. Merched, and L. Biscainho, “Recursive least-squares estimation of the evolution of partials in sinusoidal analysis,” in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP’07)*, April 2007, vol. 1, pp. 253–256.
- [19] F. Marvasti, *Nonuniform Sampling Theory and Practice*, Kluwer Academic Publishers, New York, USA, 2001.
- [20] WOW, “WOW companion webpage.” Available at <http://www.smt.ufrj.br/~luis.carvalho/WOW/>, 2017.

GRADIENT CONVERSION BETWEEN TIME AND FREQUENCY DOMAINS USING WIRTINGER CALCULUS

Hugo Caracalla

Analysis-Synthesis Team

IRCAM

Paris, France

hugo.caracalla@ircam.fr

Axel Roebel

Analysis-Synthesis Team

IRCAM

Paris, France

axel.roebel@ircam.fr

ABSTRACT

Gradient-based optimizations are commonly found in areas where Fourier transforms are used, such as in audio signal processing. This paper presents a new method of converting any gradient of a cost function with respect to a signal into, or from, a gradient with respect to the spectrum of this signal: thus, it allows the gradient descent to be performed indiscriminately in time or frequency domain. For efficiency purposes, and because the gradient of a real function with respect to a complex signal does not formally exist, this work is performed using Wirtinger calculus. An application to sound texture synthesis then experimentally validates this gradient conversion.

1. INTRODUCTION

Mathematical optimization is a recurring theme in audio signal processing: many sound synthesis algorithms, for instance physics-based synthesis [1] or sound texture synthesis [2, 3], require the solving of non-linear equations, which in turn becomes a function optimization problem. Formally, this means seeking the minimum of a real-valued cost function \mathcal{C} over the ensemble of its inputs.

From here on several solvers exist, including the widely used gradient descent algorithms. As the name implies, this kind of algorithm requires the computation of the gradient of \mathcal{C} to iteratively progress toward a minimum. But a problem arises if the parameters of \mathcal{C} are complex-valued: from the Cauchy-Riemann equations follows that a real-valued non-constant function with complex parameters is not differentiable. This means that if the parameters of our cost function \mathcal{C} are complex-valued, for example when the cost is evaluated from a spectrum, the gradient of \mathcal{C} does not exist.

Recently, this situation was encountered in the context of sound texture synthesis. In this synthesis algorithm the sub-bands envelopes of a white noise signal are adapted so that a set of their statistics reach some desired values, previously extracted from a sound texture. Current implementations of this algorithm either perform the optimization over each sub-band envelope individually while performing additional steps in parallel to compute a corresponding time signal [2], or perform it over the magnitude of the short-term Fourier transform of the white noise signal, combined with a phase retrieval algorithm allowing to recreate the synthesized time signal [3]. But both methods require additional steps to recreate the outputted signal, be it reconstruction from the sub-bands or phase retrieval, which in turn tend to damage the optimization performed over the sub-bands envelopes.

Such a problem could be avoided altogether if the optimization was performed directly over the base time signal, although this would mean that the gradient of the cost function would have to be

calculated with respect to said time signal. Since this cost is defined over the envelopes of the sub-band signals statistics, computing its gradient with respect to those envelopes is usually straightforward, but converting this gradient in a gradient with respect to the base time signal means going back up the whole envelope extraction process. Because this procedure involves manipulating spectra and because the sub-band signals are complex-valued, this situation typically falls in the case mentioned above where the formal gradients of the cost function with respect to those complex-valued signals do not exist.

Hence the goal of the work presented in this paper is twofold: establishing a relation between a gradient of a cost function with respect to a signal and with respect to its spectrum, all the while using a formalism that both allows the manipulation of those otherwise non-existent complex gradients and does so in the most efficient way possible.

The formalism chosen to be worked with is Wirtinger calculus, first introduced in [4], which offers an extension of complex differentiability in the form of Wirtinger derivatives and Wirtinger gradients. Even though it can be encountered in articles such as [5] (although without any mention of the name Wirtinger), it has been scarcely used in signal processing since then. Because this formalism is not well known, this paper starts off with an introduction to it, followed by our work on gradient conversion between time and frequency domains, and then the application and experimental validation of this conversion to the case of sound texture synthesis.

2. WIRTINGER CALCULUS AND NOTATIONS

2.1. General mathematical notations

Let us first introduce the mathematical notations that are used throughout this paper.

Arrays are denoted by bold lower case letters as \mathbf{c} , while their m -th element is denoted c_m .

For a differentiable function $f: \mathbb{R} \rightarrow \mathbb{C}$, its real derivative at $a \in \mathbb{R}$ is denoted:

$$\frac{\partial f}{\partial x}(a) \quad (1)$$

For a differentiable function $f: \mathbb{R}^M \rightarrow \mathbb{C}$, its real gradient at $\mathbf{a} \in \mathbb{R}^M$ is denoted:

$$\frac{\partial f}{\partial \mathbf{x}}(\mathbf{a}) = \left(\frac{\partial f}{\partial x_1}(\mathbf{a}), \frac{\partial f}{\partial x_2}(\mathbf{a}), \dots, \frac{\partial f}{\partial x_M}(\mathbf{a}) \right) \quad (2)$$

The discrete Fourier transform (DFT) is denoted by $\mathcal{F}: \mathbb{C}^M \rightarrow \mathbb{C}^N$, with $M \leq N$, while the inverse DFT (iDFT) is denoted by $\mathcal{F}^{-1}: \mathbb{C}^N \rightarrow \mathbb{C}^M$. For a vector $c \in \mathbb{C}^M$, its spectrum is denoted by a bold upper case letter $C \in \mathbb{C}^N$. As such we have:

$$C_n = \mathcal{F}(c)_n = \sum_{m \in [0, M-1]} c_m e^{-j \frac{2\pi m n}{N}} \quad (3)$$

And:

$$c_m = \mathcal{F}^{-1}(C)_m = \frac{1}{N} \sum_{n \in [0, N-1]} C_n e^{j \frac{2\pi m n}{N}} \quad (4)$$

2.2. Wirtinger Calculus

We now introduce Wirtinger calculus and summarize some of its more useful properties, as can be found in [5, 6, 7].

2.2.1. Wirtinger derivatives and gradients

Any complex number $c \in \mathbb{C}$ can be decomposed as $a + jb$ with $(a, b) \in \mathbb{R}^2$ its real and imaginary parts. Similarly, any function $f: \mathbb{C} \rightarrow \mathbb{C}$ can be considered as a function of $\mathbb{R}^2 \rightarrow \mathbb{C}$ with $f(c) = f(a, b)$. If it exists, the derivative of f at c with respect to the real part of its input is denoted by:

$$\frac{\partial f}{\partial x}(c) \quad (5)$$

This concords with our previous notations, since this derivative is also the real derivative of f when its domain is restrained to \mathbb{R} . If it exists, the derivative of f at c with respect to the imaginary part of its input is denoted by:

$$\frac{\partial f}{\partial y}(c) \quad (6)$$

If f is differentiable with respect to both the real and imaginary part of its input we call it differentiable in the real sense. This property is weaker than \mathbb{C} -differentiability since it lacks the Cauchy-Riemann conditions, but is sufficient when looking to optimize f since it means we could always manipulate it as a function of $\mathbb{R}^2 \rightarrow \mathbb{C}$, whose optimization does not require any differentiability over \mathbb{C} .

This is where Wirtinger calculus intervenes: it is a way of manipulating both partial derivatives of a function of $\mathbb{C} \rightarrow \mathbb{C}$ that is only differentiable in the real sense without going through the trouble of treating them individually. In addition to this, Wirtinger calculus acts as a bridge toward \mathbb{C} -differentiability since it overlaps with \mathbb{C} -derivation when the complex derivative exists.

For $f: \mathbb{C} \rightarrow \mathbb{C}$ differentiable in the real sense, its Wirtinger derivative (henceforth W -derivative) at $c \in \mathbb{C}$ is denoted by and defined as:

$$\frac{\partial f}{\partial z}(c) = \frac{1}{2} \left(\frac{\partial f}{\partial x}(c) - j \frac{\partial f}{\partial y}(c) \right) \quad (7)$$

While its conjugate Wirtinger derivative (henceforth W^* -derivative) is denoted by and defined as:

$$\frac{\partial f}{\partial z^*}(c) = \frac{1}{2} \left(\frac{\partial f}{\partial x}(c) + j \frac{\partial f}{\partial y}(c) \right) \quad (8)$$

Manipulating those two derivatives is equivalent to manipulating the two real partial derivatives, and we can with ease switch from

one expression to the other if need be: since the partial derivatives are enough to optimize a function, so is the Wirtinger derivative.

The case can then be extended to functions of several variables, such as arrays: if f denotes a function of $\mathbb{C}^M \rightarrow \mathbb{C}$ and is differentiable in the real sense, meaning here differentiable with respect to the real and imaginary parts of all of its inputs, we define the W and W^* -gradients of f similarly to their real counterpart:

$$\frac{\partial f}{\partial \mathbf{z}}(c) = \left(\frac{\partial f}{\partial z_1}(c), \frac{\partial f}{\partial z_2}(c), \dots, \frac{\partial f}{\partial z_M}(c) \right) \quad (9)$$

$$\frac{\partial f}{\partial \mathbf{z}^*}(c) = \left(\frac{\partial f}{\partial z_1^*}(c), \frac{\partial f}{\partial z_2^*}(c), \dots, \frac{\partial f}{\partial z_M^*}(c) \right) \quad (10)$$

Once more, and as shown in [6], in the case of a real-valued function (such as a cost function) knowing either the W or W^* -gradient of the function is sufficient to minimize it.

2.2.2. Link with \mathbb{C} -differentiability

If f denotes a function of $\mathbb{C}^M \rightarrow \mathbb{C}$ the following property holds:

$$f \text{ is } \mathbb{C}\text{-differentiable} \iff \begin{cases} f \text{ is differentiable in the real sense} \\ \frac{\partial f}{\partial \mathbf{z}^*} = \mathbf{0} \end{cases} \quad (11)$$

In the case where f is \mathbb{C} -differentiable, both its complex and W -gradients are equal: this is what makes of Wirtinger calculus a proper extension of \mathbb{C} -differentiability.

2.2.3. Linearity

The W and W^* -derivations are both linear, meaning for f and g two functions of $\mathbb{C}^M \rightarrow \mathbb{C}$ differentiable in the real sense and for $(\alpha, \beta) \in \mathbb{C}^2$:

$$\frac{\partial(\alpha f + \beta g)}{\partial z} = \alpha \frac{\partial f}{\partial z} + \beta \frac{\partial g}{\partial z} \quad (12)$$

$$\frac{\partial(\alpha f + \beta g)}{\partial z^*} = \alpha \frac{\partial f}{\partial z^*} + \beta \frac{\partial g}{\partial z^*} \quad (13)$$

2.2.4. Function composition

For f and g two functions of $\mathbb{C} \rightarrow \mathbb{C}$ differentiable in the real sense, the Wirtinger chain rule gives:

$$\frac{\partial f \circ g}{\partial z} = \left(\frac{\partial f}{\partial z} \circ g \right) \times \frac{\partial g}{\partial z} + \left(\frac{\partial f}{\partial z^*} \circ g \right) \times \frac{\partial g^*}{\partial z} \quad (14)$$

$$\frac{\partial f \circ g}{\partial z^*} = \left(\frac{\partial f}{\partial z} \circ g \right) \times \frac{\partial g}{\partial z^*} + \left(\frac{\partial f}{\partial z^*} \circ g \right) \times \frac{\partial g^*}{\partial z^*} \quad (15)$$

We now extend this in the case of functions of several variables: if this time f denotes a function of $\mathbb{C}^M \rightarrow \mathbb{C}$ and g denotes a function of $\mathbb{C}^N \rightarrow \mathbb{C}^M$, both being differentiable in the real sense, for $n \in [1, N]$ the chain rule gives:

$$\frac{\partial f \circ g}{\partial z_n} = \sum_{m \in [1, M]} \left(\frac{\partial f}{\partial z_m} \circ g \right) \times \frac{\partial g_m}{\partial z_n} + \left(\frac{\partial f}{\partial z_m^*} \circ g \right) \times \frac{\partial g_m^*}{\partial z_n} \quad (16)$$

$$\frac{\partial f \circ g}{\partial z_n^*} = \sum_{m \in [1, M]} \left(\frac{\partial f}{\partial z_m} \circ g \right) \times \frac{\partial g_m}{\partial z_n^*} + \left(\frac{\partial f}{\partial z_m^*} \circ g \right) \times \frac{\partial g_m^*}{\partial z_n^*} \quad (17)$$

2.2.5. Complex conjugate

If f denotes a function differentiable in the real sense, the following property holds:

$$\left(\frac{\partial f}{\partial \mathbf{z}}\right)^* = \frac{\partial f^*}{\partial \mathbf{z}^*} \quad (18)$$

This straightforwardly implies that if f is real-valued we have:

$$\left(\frac{\partial f}{\partial \mathbf{z}}\right)^* = \frac{\partial f}{\partial \mathbf{z}^*} \quad (19)$$

Meaning that in the case of functions with real output, such as cost functions, it is strictly equivalent to manipulate the W and the W^* -gradients.

3. CONVERSION BETWEEN TIME AND FREQUENCY DOMAINS

Those properties can now be put to use in order to convert the W -gradients of a real-valued cost function between time and frequency domains.

3.1. From time to frequency

Let us suppose a cost function \mathcal{E} , differentiable in the real sense and defined as:

$$\begin{aligned} \mathcal{E}: \mathbb{C}^M &\rightarrow \mathbb{R} \\ z &\mapsto \mathcal{E}(z) \end{aligned} \quad (20)$$

The value of the W -gradient of \mathcal{E} is supposed known at a given point $\mathbf{c} \in \mathbb{C}^M$. Our goal is now to evaluate this gradient at $\mathbf{C} \in \mathbb{C}^N$, the DFT of \mathbf{c} . More rigorously, this amounts to say that we wish to evaluate at \mathbf{C} the W -gradient of $\tilde{\mathcal{E}}$, defined as:

$$\begin{aligned} \tilde{\mathcal{E}}: \mathbb{C}^M &\rightarrow \mathbb{R} \\ z &\mapsto \mathcal{E}(\mathcal{F}^{-1}(z)) \end{aligned} \quad (21)$$

According to the chain rule of Wirtinger calculus stated in 2.2.4 we have for $n \in [0, N - 1]$:

$$\begin{aligned} \frac{\partial \tilde{\mathcal{E}}}{\partial z_n}(\mathbf{C}) &= \sum_{m \in [1, M]} \left(\frac{\partial \mathcal{E}}{\partial z_m}(\mathcal{F}^{-1}(\mathbf{C})) \right) \times \frac{\partial \mathcal{F}_m^{-1}}{\partial z_n}(\mathbf{C}) \\ &+ \left(\frac{\partial \mathcal{E}}{\partial z_m^*}(\mathcal{F}^{-1}(\mathbf{C})) \right) \times \frac{\partial (\mathcal{F}_m^{-1})^*}{\partial z_n}(\mathbf{C}) \end{aligned} \quad (22)$$

But \mathcal{F}^{-1} is \mathbb{C} -differentiable, meaning that according the property of Wirtinger calculus stated in 2.2.2 its W^* -gradient is null. From (18) then follows that the W -gradient of $(\mathcal{F}^{-1})^*$ is also null, which leaves us with:

$$\frac{\partial \tilde{\mathcal{E}}}{\partial z_n}(\mathbf{C}) = \sum_{m \in [1, M]} \frac{\partial \mathcal{E}}{\partial z_m}(\mathbf{c}) \times \frac{\partial \mathcal{F}_m^{-1}}{\partial z_n}(\mathbf{C}) \quad (23)$$

From (4) we easily derive:

$$\begin{aligned} \frac{\partial \mathcal{F}_m^{-1}}{\partial z_n}(\mathbf{C}) &= \frac{1}{N} \frac{\partial}{\partial z_n} \left(\sum_k z_k e^{j \frac{2\pi m k}{N}} \right) \Big|_{\mathbf{z}=\mathbf{C}} \\ &= \frac{1}{N} e^{j \frac{2\pi m n}{N}} \end{aligned} \quad (24)$$

Which re-injected in (23) gives:

$$\frac{\partial \tilde{\mathcal{E}}}{\partial z_n}(\mathbf{C}) = \frac{1}{N} \sum_{m \in [1, M]} \frac{\partial \mathcal{E}}{\partial z_m}(\mathbf{c}) e^{j \frac{2\pi m n}{N}} \quad (25)$$

$$= \frac{1}{N} \left[\sum_{m \in [1, M]} \left(\frac{\partial \mathcal{E}}{\partial z_m}(\mathbf{c}) \right)^* e^{-j \frac{2\pi m n}{N}} \right]^* \quad (26)$$

Here we recognize a DFT, leading to the expression of the whole W -gradient of $\tilde{\mathcal{E}}$ as:

$$\frac{\partial \tilde{\mathcal{E}}}{\partial \mathbf{z}}(\mathbf{C}) = \frac{1}{N} \mathcal{F} \left(\left[\frac{\partial \mathcal{E}}{\partial \mathbf{z}}(\mathbf{c}) \right]^* \right)^* \quad (27)$$

Please note that since $M \leq N$, the sum over m in (25) could not be interpreted as an inverse DFT, resulting in the identification of the conjugate of the DFT in (26).

But since \mathcal{E} (and $\tilde{\mathcal{E}}$) are real-valued functions, according to (19) the previous expression can be formulated in a cleaner way:

$$\frac{\partial \tilde{\mathcal{E}}}{\partial \mathbf{z}^*}(\mathbf{C}) = \frac{1}{N} \mathcal{F} \left(\frac{\partial \mathcal{E}}{\partial \mathbf{z}^*}(\mathbf{c}) \right) \quad (28)$$

In other words, knowing the W^* (or equivalently the W)-gradient of a cost function at a given point \mathbf{c} in time domain, it is possible to convert it over frequency domain to obtain the gradient at the spectrum \mathbf{C} .

3.2. From frequency to time

Alternatively, the expression (28) can also be reversed to give:

$$\frac{\partial \mathcal{E}}{\partial \mathbf{z}^*}(\mathbf{c}) = N \mathcal{F}^{-1} \left(\frac{\partial \tilde{\mathcal{E}}}{\partial \mathbf{z}^*}(\mathbf{C}) \right) \quad (29)$$

Which this time allows us to transition from the gradient of the error function at \mathbf{C} to the gradient at \mathbf{c} .

Thus, it is now possible to straightforwardly convert the gradient of a cost function between time and frequency domains: this also means that any gradient descent can be chosen to be performed on a signal or its spectrum, independently of which signal was used in cost computations. Additionally, and since this conversion is performed using only DFTs or iDFTs, we can also make full use of efficient Fourier transform algorithms such as the Fast Fourier Transform (FFT).

We now apply and experimentally validate the present results in a sound synthesis algorithm.

4. APPLICATION

As mentioned in the introduction, the conversion of a gradient between time and frequency domains can be put to use in the case of sound texture synthesis through statistics imposition.

During the synthesis algorithm a base signal such as a white noise is converted to frequency domain where it is filtered using a Fourier multiplier, then taken back to time domain to result in an analytic sub-band of the base signal. The goal of the algorithm is then to impose a given set of values to some selected statistics of those

sub-bands magnitudes. Since imposing the statistics via gradient descent directly over the sub-bands and then proceeding on recreating the corresponding time signal would damage the imposition, we seek to perform the gradient descent directly over the base time signal. This requires a conversion of the gradient of the cost function, easily defined with respect to the sub-bands, to a gradient with respect to the base signal.

The case can be formulated this way: we call \mathbf{s} a real-valued array of length M representing the base sound signal we wish to alter using gradient descent and \mathbf{S} its DFT of length N . \mathbf{S} is then windowed in frequency domain by a function \mathcal{G} defined as:

$$\begin{aligned}\mathcal{G}: \mathbb{C}^N &\rightarrow \mathbb{C}^N \\ \mathbf{S} &\mapsto \mathbf{W} \cdot \mathbf{S}\end{aligned}\quad (30)$$

With $\mathbf{W} \in \mathbb{C}^N$ the spectral weighting array used for band-filtering \mathbf{S} and \cdot the element-wise product. We denote the filtered spectrum $\mathbf{B} = \mathcal{G}(\mathbf{S})$, and $\mathbf{b} = \mathcal{F}^{-1}(\mathbf{B})$ its inverse DFT of length M : \mathbf{b} is thus the complex sub-band of \mathbf{s} centered around a frequency chosen via \mathbf{W} . In this example we want to impose a given value γ to the third order moment of the envelope of \mathbf{b} , so the cost function \mathcal{C} is chosen as the squared distance between the third order moment of the envelope of $|\mathbf{b}|$ the sub-band and γ :

$$\begin{aligned}\mathcal{C}: \mathbb{C}^M &\rightarrow \mathbb{R} \\ \mathbf{b} &\mapsto \left[\left(\sum_{k \in [0, M-1]} |b_k|^3 \right) - \gamma \right]^2\end{aligned}\quad (31)$$

For simplicity's sake we will consider the raw moment instead of the usual standardized moment that can be found in [2]. Using the rules of Wirtinger's calculus detailed in Section 2.2.4 we straightforwardly obtain the W^* -gradient of \mathcal{C} with respect to the sub-band at \mathbf{b} :

$$\frac{\partial \mathcal{C}}{\partial \mathbf{z}}(\mathbf{b}) = 3 \left[\left(\sum_{k \in [0, M-1]} |b_k|^3 \right) - \gamma \right] \mathbf{b} \cdot |\mathbf{b}| \quad (32)$$

But since the gradient descent is made over the base signal \mathbf{s} we need to convert the gradient at \mathbf{b} over a gradient at \mathbf{s} . Mathematically speaking, we wish to know the W^* -gradient of the function $\tilde{\mathcal{C}}$ defined as:

$$\begin{aligned}\tilde{\mathcal{C}}: \mathbb{C}^M &\rightarrow \mathbb{R} \\ \mathbf{s} &\mapsto \mathcal{C}(\mathcal{F}^{-1}(\mathcal{G}(\mathcal{F}(\mathbf{s}))))\end{aligned}\quad (33)$$

Since we know the gradient of \mathcal{C} , all we need to do is convert it to the frequency domain, compose it with \mathcal{G} and bring it back to time domain using the rules of Wirtinger calculus and time-frequency gradient conversion.

Using the time to frequency domain conversion expression in (28) we obtain the value of the W^* -gradient of $\mathcal{C} \circ \mathcal{F}^{-1}$ at \mathbf{B} as:

$$\frac{\partial (\mathcal{C} \circ \mathcal{F}^{-1})}{\partial \mathbf{z}^*}(\mathbf{B}) = \frac{1}{N} \mathcal{F} \left(\frac{\partial \mathcal{C}}{\partial \mathbf{z}^*}(\mathbf{b}) \right) \quad (34)$$

From here we need to obtain the gradient of $\mathcal{C} \circ \mathcal{F}^{-1} \circ \mathcal{G}$. Since \mathcal{G} is a simple element-wise product, and since as stated in section

2.2.3 Wirtinger derivation is linear, we directly obtain:

$$\frac{\partial (\mathcal{C} \circ \mathcal{F}^{-1} \circ \mathcal{G})}{\partial \mathbf{z}^*}(\mathbf{S}) = \mathbf{W} \cdot \frac{\partial (\mathcal{C} \circ \mathcal{F}^{-1})}{\partial \mathbf{z}^*}(\mathbf{B}) \quad (35)$$

All that is left now is the conversion from frequency to time domain to obtain the W^* -gradient at \mathbf{s} . Using the result in (29) gives:

$$\frac{\partial \tilde{\mathcal{C}}}{\partial \mathbf{z}^*}(\mathbf{s}) = N \mathcal{F}^{-1} \left(\frac{\partial (\mathcal{C} \circ \mathcal{F}^{-1} \circ \mathcal{G})}{\partial \mathbf{z}^*}(\mathbf{S}) \right) \quad (36)$$

Combining (34), (35) and (36) then gives:

$$\frac{\partial \tilde{\mathcal{C}}}{\partial \mathbf{z}^*}(\mathbf{s}) = \mathcal{F}^{-1} \left(\mathbf{W} \cdot \mathcal{F} \left(\frac{\partial \mathcal{C}}{\partial \mathbf{z}^*}(\mathbf{b}) \right) \right) \quad (37)$$

Using this relation we can now convert the W^* -gradient at \mathbf{b} expressed in (32) to a W^* -gradient at \mathbf{s} , the base signal.

In addition to this, since \mathbf{s} is a purely real-valued signal it is also straightforward to make use of the previous result to obtain the more common real gradient of $\tilde{\mathcal{C}}$ at \mathbf{s} . Indeed, from the definition of the W^* -derivative in (8) and since $\tilde{\mathcal{C}}$ is real-valued, we have that if $\mathbf{s} \in \mathbb{R}^M$:

$$\frac{\partial \tilde{\mathcal{C}}}{\partial \mathbf{x}}(\mathbf{s}) = 2\Re \left\{ \frac{\partial \tilde{\mathcal{C}}}{\partial \mathbf{z}^*}(\mathbf{s}) \right\} \quad (38)$$

Which results in the final expression of the real gradient of the cost function with respect to the real base time signal:

$$\frac{\partial \tilde{\mathcal{C}}}{\partial \mathbf{x}}(\mathbf{s}) = 6 \left[\left(\sum_{k \in [0, M-1]} |b_k|^3 \right) - \gamma \right] \Re \{ \mathcal{F}^{-1} (\mathbf{W} \cdot \mathcal{F}(\mathbf{b}, |\mathbf{b}|)) \} \quad (39)$$

We now have all the tools required to use a gradient descent algorithm over the base signal \mathbf{s} in order to minimize the cost function \mathcal{C} and thus impose a given statistic over a selected sub-band of it, without having to first perform it over the sub-band and then resort to a potentially damaging reconstruction.

Additionally, it is now possible to impose a given set of statistics to several sub-bands at once. Indeed, if we define a general cost function as the sum of several sub-bands cost functions, such as the one expressed in (31), then the linearity of derivation gives that the global cost gradient with respect to the base time signal is simply the sum of the sub-band cost gradients which we established in (39): using this global cost in the gradient descent will then tend to lower the sub-band cost functions, thus imposing the desired values to the statistics of each sub-bands at once.

So as to act both as an example and an experimental validation, such a simultaneous imposition was made over a 10 seconds-long white noise signal sampled at 48 kHz. The statistics chosen as goals are the raw third moments extracted from a rain sound texture over 3 sub-bands arbitrarily chosen at 20 Hz, 1765 Hz and 10.3 kHz, while the algorithm used is a conjugate gradient descent. The evolution over the iterations of the gradient descent of the cost functions, both global and band-wise, is plotted in Figure 1. Because the band-wise cost functions are decreasing during the gradient descent, the optimization successfully outputs a time signal possessing the exact statistics we meant to impose on it, without requiring a separate imposition for each sub-band nor a possibly harmful reconstruction of the time signal from the sub-bands signals.

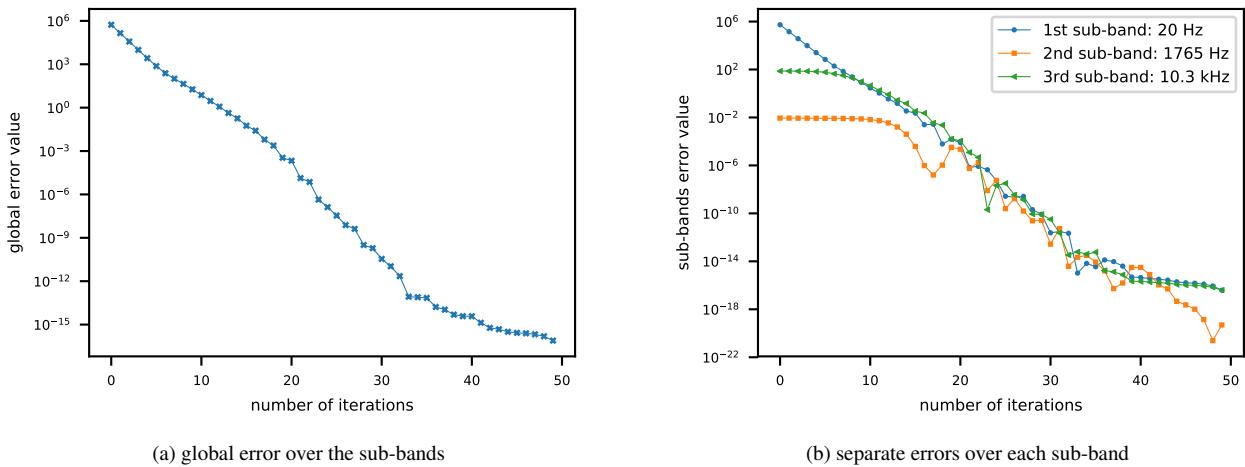


Figure 1: Value of the cost functions over the number of iterations made during the gradient descent. The total gradient is obtained by summing the gradients of 3 arbitrary sub-bands (20 Hz, 1765 Hz and 10.3 kHz) as expressed in (39)

5. CONCLUSION

By making use of Wirtinger formalism it is possible to link the gradient of a real-valued function with respect to a signal to the gradient with respect to its spectrum and transition between the two simply by use of a digital Fourier Transform and its inverse: this allows any gradient descent algorithm to be performed equivalently in time or frequency domain while avoiding any complication that may come from the complex non-differentiability of cost functions.

Put to use in sound texture synthesis this leads to a fully coherent approach to imposing arbitrary values to the statistics of the sub-bands of a signal, which is currently being investigated in order to establish an efficient sound texture synthesis algorithm.

6. REFERENCES

- [1] Stefan Bilbao, Alberto Torin, and Vasileios Chatzioannou, “Numerical modeling of collisions in musical instruments,” *Acta Acustica united with Acustica*, vol. 101, no. 1, pp. 155–173, 2015.
- [2] Josh H McDermott and Eero P Simoncelli, “Sound texture perception via statistics of the auditory periphery: Evidence from sound synthesis,” *Neuron*, vol. 71, no. 5, pp. 926–940, 2011.
- [3] Wei-Hsiang Liao, *Modelling and transformation of sound textures and environmental sounds*, Ph.D. thesis, Université Pierre et Marie Curie, 2015.
- [4] Wilhelm Wirtinger, “Zur formalen theorie der funktionen von mehr komplexen veränderlichen,” *Mathematische Annalen*, vol. 97, no. 1, pp. 357–375, 1927.
- [5] DH Brandwood, “A complex gradient operator and its application in adaptive array theory,” in *IEE Proceedings F-Communications, Radar and Signal Processing*. IET, 1983, vol. 130, pp. 11–16.

- [6] P Bouboulis, “Wirtinger’s calculus in general hilbert spaces,” *arXiv preprint arXiv:1005.5170*, 2010.
- [7] Robert FH Fischer, *Precoding and signal shaping for digital transmission*, John Wiley & Sons, 2005.

LIVE CONVOLUTION WITH TIME-VARIANT IMPULSE RESPONSE

Øyvind Brandtsegg *

Department of Music,
Norwegian University of Science and Technology
Trondheim, Norway
oyvind.brandtsegg@ntnu.no

Sigurd Sæue

Department of Music,
Norwegian University of Science and Technology
Trondheim, Norway
sigurd.saue@ntnu.no

ABSTRACT

This paper describes a method for doing convolution of two live signals, without the need to load a time-invariant impulse response (IR) prior to the convolution process. The method is based on stepwise replacement of the IR in a continuously running convolution process. It was developed in the context of creative live electronic music performance, but can be applied to more traditional use cases for convolution as well. The process allows parametrization of the convolution parameters, by way of real-time transformations of the IR, and as such can be used to build parametric convolution effects for audio mixing and spatialization as well.

1. INTRODUCTION

Convolution has been used for filtering, reverberation, spatialization and as a creative tool for cross-synthesis ([1], [2] and [3] to name a few). Common to most of them is that one of the inputs is a time-invariant impulse response (characterizing a filter, an acoustic space or similar), allocated and preprocessed prior to the convolution operation. Although developments have been made to make the process latency free (using a combination of partitioned and direct convolution [4]), the time-invariant nature of the impulse response (IR) has inhibited a parametric modulation of the process. Modifying the IR traditionally has implied the need to stop the audio processing, load the new IR, and then re-start processing using the updated IR. This paper will briefly present a context for the work before presenting the most common convolution methods. A number of observations along the way will motivate a strategy for convolution with time-variant impulse responses based on stepwise replacement of IR partitions. Finally we present a number of use cases for the method with focus on live performance.

2. CONTEXT

The current implementation was developed in the context of our work with cross-adaptive audio processing (see [5] and also the project blog <http://crossadaptive.hf.ntnu.no/>) and live processing (previous project blog at [6] and the released album "Evil Stone Circle" at [7]). Our previous efforts on live streaming convolution area are described in ([8], [9]). We also note that the area of flexible convolution processing is an active field of research in other environments (for example [10], in some respects also [11] and [12] due to the parametric approach to techniques closely related to convolution).

Our primary goal has been to enable the use of convolution as a creative tool for live electronic music performance, by allowing

ing two live sources to be convolved with each other in a streaming fashion. Our current implementation allows this kind of live streaming convolution with minimal buffering. As we shall see, this also allows for parametric transformations of the IR. Examples of useful transformations might be pitch shifting, time stretching, time reversal, filtering, all done in a time-variant manner.

3. CONVOLUTION METHODS

Convolution of two finite sequences $x(n)$ and $h(n)$ of length N is defined as:

$$y(n) = x(n) * h(n) = \sum_{k=0}^{N-1} x(n-k) h(k) \quad (1)$$

This is a direct, time-domain implementation of a FIR filter structure with filter length N . A few observations can be made at this stage:

- Filter length is the only parameter involved.
- There is no latency: $y(0) = x(0)h(0)$.
- The output length N_y is equal to $2N - 1$
- Computational complexity is of the order $O(N^2)$.

Eq. 1 can be interpreted as a weighted sum of N time-shifted copies of the input sequence $x(n)$. The weights are given by the coefficients of the filter $h(n)$. In acoustic simulations, a room impulse response is a typical example of such a filter, with wall reflections represented as non-zero coefficients. The result of running a source signal through this filter is an accumulation of delayed reflections, into what we recognize as a reverberant sound.

If instead the filter $h(n)$ is live-sampled from a musical performance, the filter coefficients will typically form continuous sequences of non-negligible values. Hence, when the sound of another musical instrument is convolved with this filter, the output will exhibit a dense texture of overlays, leading to a characteristic time smearing of the signals involved.

Figure 1 illustrates what happens when a simple sinusoidal signal is convolved with itself ($x(n) = h(n)$). As expected the output is stretched out to almost double length. What is more interesting is the ramp characteristics caused by the gradual overlap of the sequences $x(n)$ and $h(n)$ in eq. 1. The summation has a single term at $n = 0$, reaches a maximum of N terms at $n = N - 1$, and returns to a single term again at $n = 2N - 2$. This inherent "fade in/fade out"-property of convolution will be exploited in our approach.

The computational complexity of direct convolution will be prohibiting when filter length increases. We normally work with filters of duration 1-2 seconds or more, which can amount to 96000

* Thanks to NTNU for generously providing a sabbatical term, within which substantial portions of this research have been done

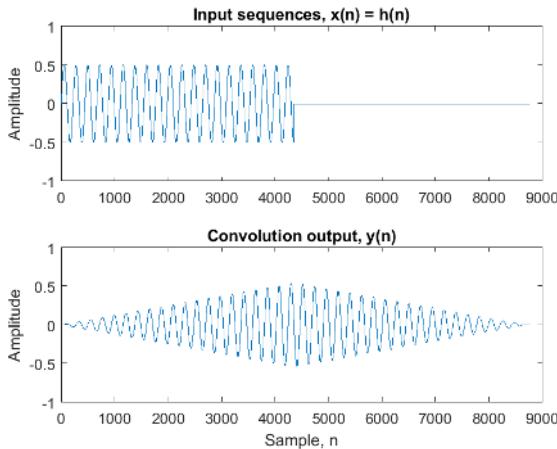


Figure 1: Convolving a time-limited sine sequence with itself

multiplications per output sample at 48 kHz. A far more efficient method takes advantage of the *circular convolution* property of the Discrete Fourier Transform (DFT) to perform convolution as multiplication in the frequency domain [13]:

$$y(n) = DFT^{-1}\{DFT\{x(n)\} \cdot DFT\{h(n)\}\} \quad (2)$$

where DFT and DFT^{-1} are the forward and inverse Discrete Fourier Transform, respectively. The Fast Fourier Transform (FFT) provides an efficient implementation. It is worth noting that circular convolution assumes periodicity of the input sequences. Fortunately it can be made applicable for linear filtering, provided that the FFT block size K satisfies the condition $K \geq M + N - 1$ where M, N are the lengths of the input sequences $x(n)$ and $h(n)$, respectively. If the condition does not hold there will be time aliasing [14]. The input sequences must be zero padded to length K prior to the FFT.

Equation 2 also calls for some observations:

- Computational complexity is reduced to $O(N \log N)$ (the complexity of the FFT).
- Latency has increased to the full filter length.

It is now obvious that convolution is simply multiplication in the frequency domain. From which we may further observe that:

- Output amplitude strongly depends on degree of frequency overlap between the two inputs. Hence convolution can be a challenge to work with in live performance due to lack of amplitude control.
- We may expect a relative loss of high frequency energy. For non-synthetic sounds the energy typically falls off in the high frequency range. When two such frequency spectra are multiplied the relative imbalance between high and low frequencies is magnified.

The increased latency is undesirable for real-time applications. *Partitioned convolution* reduces the latency by breaking up the inputs into smaller partitions [15]. Assume that the input sequences in eq. 1 are segmented into P partitions of uniform length $N_P = N/P$:

$$x(n) = [x_0(n), x_1(n), \dots, x_{P-1}(n)] \quad (3)$$

$$h(n) = [h_0(n), h_1(n), \dots, h_{P-1}(n)] \quad (4)$$

with:

$$x_i(n) = \begin{cases} x(n), & \text{if } n \in [i \cdot N_P, (i+1) \cdot N_P - 1] \\ 0, & \text{otherwise.} \end{cases} \quad (5)$$

and:

$$h_j(n) = \begin{cases} h(n), & \text{if } n \in [j \cdot N_P, (j+1) \cdot N_P - 1] \\ 0, & \text{otherwise.} \end{cases} \quad (6)$$

Then it can be proven from the linear and commutative properties of convolution that eq. 1 can be rewritten as:

$$y(n) = x(n) * h(n) = \sum_{i=0}^{P-1} \sum_{j=0}^{P-1} x_i(n) * h_j(n) \quad (7)$$

In the last part of equation 1 we notice that the indices of $x(n)$ and $h(n)$ in each term of the summation always add to n . This can be extended to the partitions $x_i(n)$ and $h_j(n)$ and allows us to deduce the interval of n where a particular partitioned convolution $x_i(n) * h_j(n)$ contributes to the output $y(n)$. If we define S_{ij} and E_{ij} as the starting and ending point of this interval, respectively, we get by adding the respective ranges in equations 5 and 6 [8]:

$$S_{ij} = (i+j) \cdot N_P \quad (8)$$

$$E_{ij} = (i+j+2) \cdot N_P - 2 \quad (9)$$

Now let's define the interval Y_k as:

$$Y_k = [k \cdot N_P, (k+1) \cdot N_P - 1] \quad (10)$$

If we compare equations 8, 9 and 10, we see that the partitioned convolution $x_i(n) * h_j(n)$ only contributes to the output $y(n)$ for n in the intervals Y_{i+j} and Y_{i+j+1} . A trivial example of the computation with only $P = 3$ partitions is shown in Figure 2 (the sample index n is omitted for simplicity).

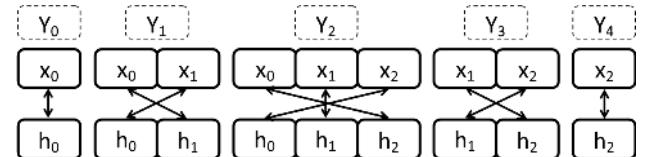


Figure 2: Example of partitioned convolution with 3 partitions [8]

Each partitioned convolution can be performed as multiplication in the frequency domain (equation 2). Partition size should be adjusted to make an optimal compromise between computational efficiency and latency. From this implementation we observe:

- Latency is reduced to the partition size, N_P .
- The partitions $x_i(n)$ and $h_j(n)$ contributes to output interval Y_k only if $i, j \leq k$.
- The partitions $x_i(n)$ and $h_j(n)$ contributes to output interval Y_{P+k} only if $i, j \geq k^1$.

The latter two observations play an important role for our streaming approach.

It should also be added that techniques combining partitioned and direct-form convolution can eliminate processing latency entirely [4]. Wefers [14] provide an excellent, updated review of convolution algorithms in general, with a particular focus on partitioned convolution.

¹We will later show that the inequality can be expressed as $i, j > k$ when implementation is taken into account.

4. RESEARCH GOALS AND PREVIOUS WORK

Traditionally convolution is an asymmetric operation where the two inputs have different status: a continuous stream of input samples $x(n)$ and an impulse response (IR) $h(n)$ of finite length N . Typically the latter is a relatively short segment, a time-invariant representation of a system (e.g. a filter or a room response) onto which the input signal is applied.

Instead we are searching for more flexible tools for musical interplay and have previously presented a number of strategies for dynamic convolution [9]. Our aim has been to:

- Attain dynamic parametric control over the convolution process in order to increase playability.
- Investigate methods to avoid or control dense and smeared output
- Provide the ability to use two live audio sources as inputs to a continuous real-time convolution process.
- Provide the ability to update/change the impulse responses in real-time without glitches.

With existing tools we haven't been able to get around the time-invariant nature of the impulse response in an efficient way. In order to make dynamic updates of the IR during convolution without audible artifacts, we had to use two (or more) concurrent convolution processes and then crossfade between them whenever the IR should be modified. The IR update could be triggered explicitly, at regular intervals or based on the dynamics of the input signal (i.e. transient detection). Every update of the IR triggered a reinitialization of the convolution process.

We have also done work on live convolution of two audio signals of equal status: neither signal has a subordinate status as filter for the other [8]. Instead both are continuous audio streams segmented at intervals triggered by transient detection. Each pair of segments are convolved in a separate process. With frequent triggering the number of concurrent processes could grow substantially due to the long convolution tail ($P - 1$ partitions) of each process.

5. TIME-VARIANT IMPULSE RESPONSE

In this paper we present a simple, but efficient implementation of convolution with a time-variant impulse response $h(n)$ of finite length N . In this case the coefficients of $h(n)$ are no longer constant throughout the convolution process. At a point n_T in time the current impulse response, denoted $h_A(n)$ is updated with a new filter $h_B(n)$.

Wefers and Vorländer [16] points at two possible solutions to time-varying FIR filtering: instantaneous switching or crossfading. The former can be expressed as:

$$y_n = \begin{cases} x(n) * h_A(n), & \text{if } n < n_T \\ x(n) * h_B(n), & \text{if } n \geq n_T \end{cases} \quad (11)$$

It is a cheap implementation, but the hard switching is known to cause audible artifacts from discontinuities in the output waveform.

The second option, crossfading, avoids these artifacts by filtering $x(n)$ with both impulse responses, $h_A(n)$ and $h_B(n)$, in a transition interval and then crossfade between the two outputs to

smooth out discontinuities. The disadvantage of this method (assuming fast convolution in the frequency domain) is the added cost of an extra spectral convolution and inverse transform, as well as the operations consumed by the crossfade. A modified approach with crossfading in the frequency domain [16] saves the extra inverse transform.

Our goal is to be able to dynamically update the impulse response without reinitializations and crossfares of parallel convolution processes. In the following we assume a uniformly partitioned convolution scheme implemented with FFTs and multiplication in the frequency domain. The key to our approach is the inherent "fade in/fade out"-characteristic of convolution (illustrated in figure 1) and the previously noted properties of partitioned convolution:

Property 1. *The partitions $x_i(n)$ and $h_j(n)$ contributes to output interval Y_k only if $i, j \leq k$.*

An immediate consequence of Property 1 is that we can load the impulse response partition by partition in parallel with the input. A fully loaded IR is not necessary to initiate the convolution process since the later partitions initially do not contribute to the convolution output. This drastically reduces the latency e.g. for application of live sampled impulse responses. Convolution can start during sampling, as soon as a single partition of the IR is available. In addition the computational load associated with FFT calculations on the IR is conveniently spread out in time, hence avoiding bursts of CPU usage when loading long impulse responses.

Property 2. *The partitions $x_i(n)$ and $h_j(n)$ contributes to output interval Y_{P+k} only if $i, j \geq k$.*

From Property 2 it is clear that we also can *unload* the impulse response partition by partition without audible artifacts, even while the convolution process is running. The reason being that the earlier partitions no longer contribute to the output. Unloading a partition simply means clearing it to zero, and it should progress in the same order as the loading process.

The shortest possible load/unload interval is a single partition, which is actually equivalent to segmenting out a single partition of the *input signal* and convolve it with the full impulse response. Figure 3 illustrates the effect.

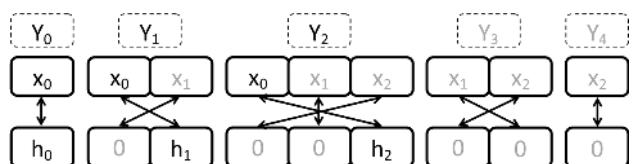


Figure 3: Example of minimal load/unload interval

The combination of the two strategies, stepwise loading and unloading, naturally leads to *stepwise replacement* of the impulse response. If we return to the direct form we can express the convolution with stepwise IR replacement starting at time n_T as:

$$y_n = \sum_{k=\max\{0, n-n_T+1\}}^{N-1} x(n-k) h_A(k) + \sum_{k=0}^{\min\{N-1, n-n_T\}} x(n-k) h_B(k) \quad (12)$$

For $n < n_T$ equation 12 reduces to:

$$y_n = \sum_{k=0}^{N-1} x(n-k) h_A(k) \quad (13)$$

For $n > n_T + N - 1$ it reduces to:

$$y_n = \sum_{k=0}^{N-1} x(n-k) h_B(k) \quad (14)$$

In the transition interval $n \in [n_T, n_T + N - 2]$ the filter coefficients of $h_A(n)$ are replaced one by one with the coefficients of $h_B(n)$, beginning with the first coefficient $h_A(0)$. This transition interval ensures a smooth transition between filters since the convolution tail of filter $h_A(n)$ is allowed to fade out properly after time n_T while the new filter $h_B(n)$ fades in. It is important to note that there is no extra cost, since it is solved by gradual coefficient replacement, and the total number of additions in equation 12 is equal to $N - 1$ at every step n . At the same time it avoids the artifacts caused by instantaneous switching (eq. 11).

The same logic holds for partitioned convolution where the filter is replaced partition by partition instead. The only restriction is that the replacement time n_T must be at a partition border:

$$n_T = k * N_P \text{ for } k = 0, 1, 2, \dots \quad (15)$$

At any time during the running convolution process we can trigger an update of the IR and start filling in new partitions from the beginning of the IR buffer. Figure 4 shows a simplified example of the procedure, where a filter update is triggered at the beginning of time interval Y_1 . The impulse response partitions $H_{A,k}$ is replaced by $H_{B,k}$. In a transition period equal to $P - 1$ (where P is the number of IR partitions) the output is a mix of two convolution processes.

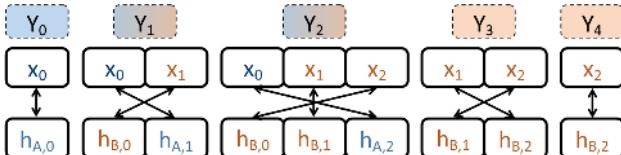


Figure 4: Example of dynamic IR replacement starting at time interval Y_1

It should be apparent that triggering a new IR update amounts to a segmentation of the input signal. No input partitions buffered before the trigger (x_0 in the figure) will be convolved with the new IR, and similarly, no input partitions buffered after the trigger (x_1 and x_2 in the figure) will be convolved with the old IR. To avoid discontinuities in the IR and hence audible clicks in the output signal, the impulse response buffer should always be completely refilled when updated (all N partitions).

In contrast to our earlier attempts and also to the crossfade methods suggested by [16], the proposed method does not add any extra computational cost. Impulse response partitions are simply replaced in the right order. A possible disadvantage is that the length of the transition interval is fixed to the filter length N and not available for user control.

6. THE IMPLEMENTATION

We have implemented convolution with time-variant IR as a plugin opcode titled *liveconv* written in C for the audio programming language Csound². A Csound opcode normally provides three programming components: An internal data structure, an initializing function and a processing function [17]. The internal data structure is allocated in the initialization function, including all dynamic memory. In *liveconv* partition length and impulse response table are specified during this step. The content of the IR table may be updated at any time, but not its memory location or size.

The starting point for our implementation is a previous opcode, *ftconv*³, that implements uniformly partitioned convolution with a fixed IR. The partition length must be an integer power of two: $N_P = 2^k$, and the FFT block size N_B is twice the partition length: $N_B = 2N_P$. The inputs partitions $x_i(n)$ and $h_j(n)$ are padded with N_P zeros and thereby satisfies the condition for circular convolution without time-aliasing.

To ensure correct real-time processing the implementation employs the Overlap-Add scheme (OLA): The convolution produces overlapping output segments of length N_B , which is twice the partition length N_P . Consequently, half the output samples produced by the partitioned convolution $x_i(n) * h_j(n)$ at interval Y_{i+j} must be stored and added to the output of the next interval, Y_{j+i+1} . It is worth noting that since this contribution is buffered with the output of interval Y_{i+j} , it is not necessary to recalculate the contributing convolution during interval Y_{j+i+1} . For that reason the final inequality of Property 2 can be adjusted from \geq to $>$: The partitions $x_i(n)$ and $h_j(n)$ contributes to output interval Y_{P+k} only if $i, j > k$.

The Overlap-Save scheme (OLS) is found to compute more efficiently than OLA [14]. It remains to be ascertained if our partition replacement scheme is applicable to OLS without modifications. The integration with other convolution algorithms such as direct convolution in the time domain and non-uniform partitioned convolution should also be verified.

The outline of the processing function is as follows (details on OLA are omitted):

- Check the update control signal. If set to "load", prepare to reload the IR. If set to "unload", prepare to unload the IR. A data structure maintains control of each load/unload process. In theory there could be a new process for every partition.
- Read audio inputs into an internal circular buffer in chunks given by the control rate of Csound ($ksmps$ is the number of samples processed during each pass of the processing function).
- When an entire partition is filled with input data:

²More information and links to source code at <http://csound.github.io/>. Documentation for the *liveconv* opcode can be found at <http://csound.github.io/docs/manual/liveconv.html>.

³Documentation at <http://csound.github.io/docs/manual/ftconv.html>.

- Calculate the FFT of the input partition
- For every running *load* process fetch a specified partition from the IR table and calculate its FFT. Note that several parts of the IR may be updated in parallel.
- Do complex multiplication of input and IR buffers in the frequency domain
- Calculate the inverse FFT of the result and write to output
- For every running *unload* process clear a specified partition to zero.
- Increment load/unload processes to prepare for the next partition. If a process has completed its pass through the IR table, it is set inactive.

No assumptions have to be made on the impulse responses involved. The load process can be signaled as soon as one *ksmps* chunk of new data has been filled into the IR table.

In order to clarify the behavior of time-variant convolution with partition replacement, we will compare it with two overlapping time-invariant convolutions. Figure 5 shows the input signals. On top are the two impulse responses, $h_A(n)$ and $h_B(n)$. They are both 1 second excerpts of speech sampled at 44,100 Hz ($N = 44100$). At the bottom is the input source signal $x(n)$, which is a 10 second excerpt of baroque chamber music, also sampled at 44,100 Hz.

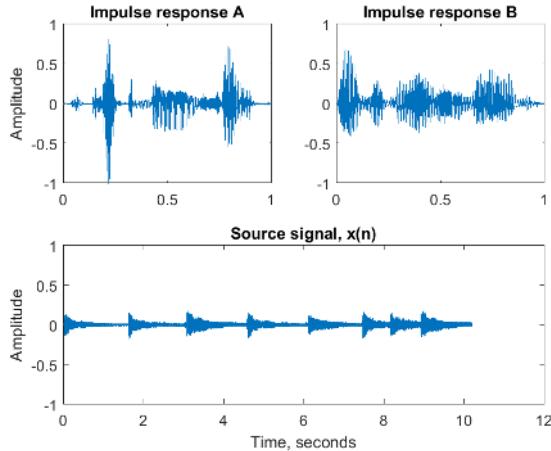


Figure 5: Convolution demonstration. Top: The two impulse responses, $h_A(n)$ and $h_B(n)$. Bottom: The input signal $x(n)$

In the following all convolution operations are running with partition length $N_P = 1024$ and FFT block size $N_B = 2048$. At time $n_T \approx 4.5$ seconds, a switch between impulse responses $h_A(n)$ and $h_B(n)$ is initiated (the exact time is slightly less in order to align it with the partition border).

First we show the output of two separate time-invariant convolution processes, using traditional partitioned convolution (figure 6). The input signal is split in two non-overlapping segments at the transition time n_T . The first segment defined by $n < n_T$ is convolved with impulse response $h_A(n)$ only and the output is shown immediately below (Convolution 1). The second input segment defined by $n \geq n_T$ follows next. It is convolved with

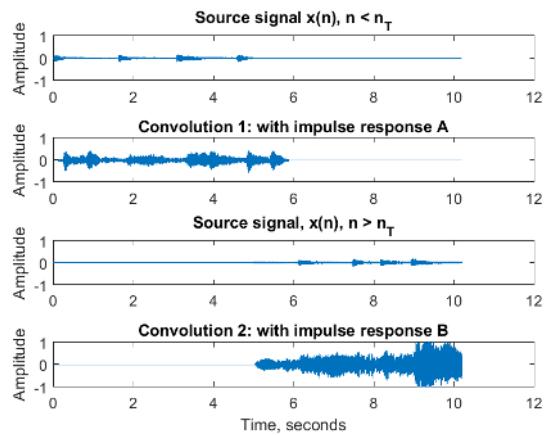


Figure 6: Convolution demonstration: Two overlapping time-invariant convolutions. Upper pair: Source signal segment for $n < n_T$ followed by the convolution with impulse response $h_A(n)$. Lower pair: Source signal segment for $n \geq n_T$ followed by the convolution with impulse response $h_B(n)$.

impulse response $h_B(n)$ only and the output is shown at the bottom (Convolution 2). Notice that the two outputs overlap in a time interval of length comparable to the impulse response.

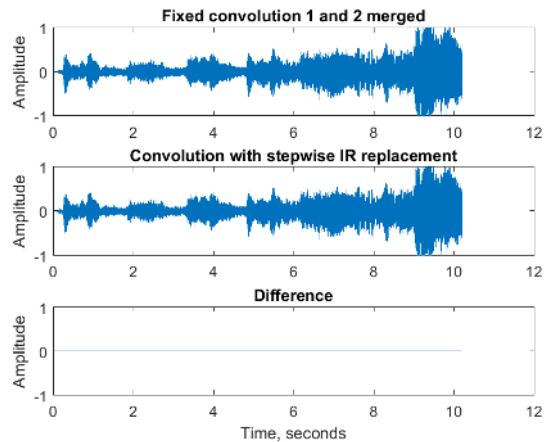


Figure 7: Convolution demonstration: Comparing time-variant method with two overlapping time-invariant convolutions. Top: The merged result of the two time-invariant convolutions in figure 6. Middle: The output when using stepwise IR replacement. Bottom: Difference between the two.

If we merge together the two convolution outputs in figure 6 we get the signal shown at the top of figure 7. If, on the other hand, we run the entire operation as a single time-variant convolution on the original signal $x(n)$, where the impulse response is stepwise replaced beginning at time n_T , we get the output signal shown in the middle of figure 7. The difference between the two signals are displayed at the bottom. It is uniform and for all practical purposes negligible. An estimate of the signal-to-noise ratio (SNR) returns 67.2 dB. From this demonstration it should be clear that

convolution with stepwise IR replacement preserves the behavior of time-invariant convolution, while at the same time allowing filter updates within a single process at no extra cost.

A special case appears when the IR's are sampled from the same continuous audio stream, one partition apart. Then we get a result similar to cross-synthesis, pairwise multiplication of partitions from two parallel audio streams, but still with the effect of long convolution filters. It should be added that there are more efficient ways to implement this particular condition, but that will be deferred to another paper.

The interval between IR updates is provided as a parameter available for performance control, which increases the playability of convolution. There is still a risk of huge dynamic variations due to varying degrees of frequency overlap between input signal and impulse response. The housekeeping scheme introduced to maintain the IR update processes could be exploited for smarter amplitude scaling. This is ongoing work. We would also like to look at integration of some of the extended convolution techniques proposed by Donahue & al [10].

7. USE CASE: PLUGIN FOR LIVE PERFORMANCE

A dedicated VST plugin has been implemented to show the use of the new convolution methods, built around the *liveconv* opcode. Even though the incremental IR update allows for a vast array of application areas, we have initially focused on realtime convolution of two live signals. As an indication of its primary use, we've named it "Live convolver". The plugin is implemented with a "dual mono" input signal flow, as shown in figure 8, allowing it to be used on a stereo track of a DAW. The *left* stereo signal will be used to record the IR, while the *right* stereo signal will be used as input to the convolution process.

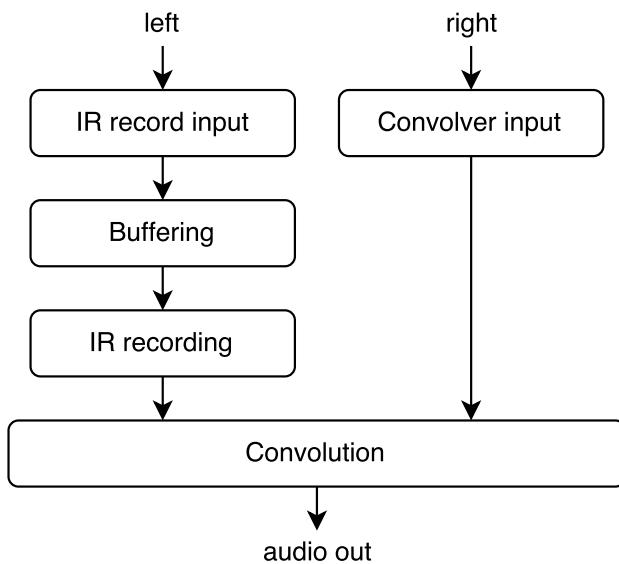


Figure 8: Plugin signal flow overview

The signal on the *IR record input* channel is continuously written to a circular buffer. When we want to replace the current IR, we read from this buffer and replace the IR partition by partition. The main reason for the circular buffer is to enable transformation

(for example time reversal) of the audio before making the IR. As an attempt to visualize *when* the IR is taken from, we use a circular colouring scheme to display the circular input buffer. We also represent the IR using the same colours. Time (of the input buffer) is thus represented by colour. As the color of *now* continuously and gradually changes from red to green to blue, it is possible to identify *time lag* as an azimuthal distance on the color circle⁴. Figure 9 shows the plugin gui, with a representation of the coloured input buffer and a live sampled IR. Similarly, a time reversed IR is shown in figure 10. Note the direction of color change (green to red) of the reversed IR as compared with the color change in the normal (forward) IR (blue to red).

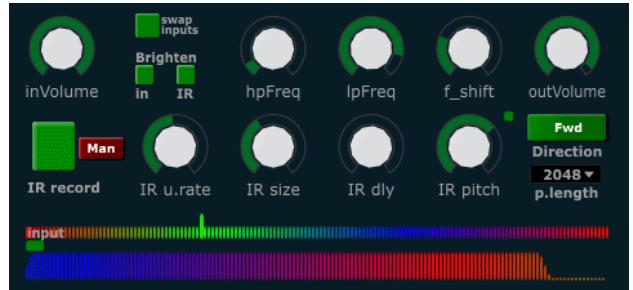


Figure 9: Liveconvolver plugin GUI

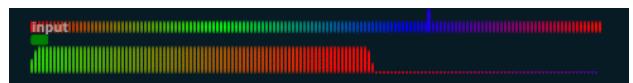


Figure 10: Visualization of time reversed IR

The plugin has controls for manual triggering of the IR recording, or the IR can be automatically updated by a periodic trigger (using *IR update rate* and *IR size controls*). Methods for triggering IR update via transient detection has also been implemented. Pitch modification and time reversal is available by dedicated gui controls. In addition, we have simple lowpass and highpass filtering, as this can be very useful for quickly fixing spectral problems of using convolution in a live setting. Since convolution can lead to a certain loss of high frequency content, we also have the option of "brightening" each input by applying a low-q high shelf filter. Finally, there is a significant potential for audio feedback using convolution in a live setting, when the IR is sampled in the same room where the convolver output is played back. To alleviate this risk, we have implemented a frequency shifter with relatively small amounts of shift as suggested by [18]. The amount of frequency shift is also controllable from the gui. By shifting the convolver output with just a few Hz, the feedback potential is significantly reduced.

8. USE CASE: LIVE PERFORMANCES

The liveconvolver plugin has been used for several studio sessions and performances from late 2016 onwards. Reports and sound examples from some of these experimental sessions can be found at [19] and [20]. These reports also contain reflections on the performative difference experienced in the roles of *recording the IR*

⁴https://en.wikipedia.org/wiki/Color_wheel

and *playing through it*. It is notable that a difference is perceived, since in theory the mathematical output of the convolution process is identical regardless of which one signal is used as the IR. The buffering process allows continuous updates to the IR with minimal latency, and several methods for triggering the IR update has been explored. This should ideally facilitate a seamless merging of the two input signals. However, the IR update still needs to be triggered, and the IR will be invariant between triggered updates. In this respect, the instrument providing the source for the IR will be subject to recording, and the live input to the convolution process is allowed a continuous and seamless flow. It is natural for a performer to be acutely aware of the distinction between being recorded and playing live. Similar distinctions can assumably be made in the context of all forms of live sampling performance, and in many cases of live processing as an instrumental practice. The direct temporal control of the musical energy resides primarily with the instrument *playing through the effects processing*, and to a lesser degree with the instrumental process *creating the effects process* (recording the IR in our case here).

9. OTHER USE CASES, FUTURE WORK

The flexibility gained by our implementation allows parametric control of convolution also in more traditional effects processing applications. One could easily envision a parametric convolution reverb with continuous pitch and filter modulation for example. Parametric modulation of the IR can be done either by applying transformations on the audio being recorded to the IR, or directly on the spectral data. Such changes to the IR could have been done with traditional convolution techniques too, by preparing a set of transformed IR's and crossfading between them. The possibilities inherent in the incremental IR update as we have described allows direct parametric experimentation, and thus is much more immediate. It also allows for automated time-variant modulations (using LFO's and random modulators), all without introducing artifacts due to IR updates.

10. CONCLUSIONS

We have shown a technique for incremental update of the impulse response for convolution purposes. The technique provides time-variant filtering by doing a continuous update of the IR from a live input signal. It also opens up possibilities for direct parametric control of the convolution process and as such enhancing the general flexibility of the technique. We have implemented a simple VST plugin as proof of concept of the live streaming convolution, and documented some practical musical exploration of its use. Further applications within more traditional uses of convolution has also been suggested.

11. ACKNOWLEDGMENTS

We would like to thank the Norwegian Artistic Research Programme for support of the research project "Cross-adaptive audio processing as musical intervention", within which the research presented here has been done. Thanks also to the University of California, and performers Kyle Motl and Jordan Morton for involvement in the practical experimentation sessions.

12. REFERENCES

- [1] Mark Dolson, “Recent advances in *musique concrète* at CARL,” in *Proceedings of the 1985 International Computer Music Conference, ICMC 1985, Burnaby, BC, Canada, August 19-22, 1985*, 1985.
- [2] Curtis Roads, “Musical sound transformation by convolution,” in *Opening a New Horizon: Proceedings of the 1993 International Computer Music Conference, ICMC 1993, Tokio, Japan, September 10-15, 1993*, 1993.
- [3] Trond Engum, “Real-time control and creative convolution,” in *11th International Conference on New Interfaces for Musical Expression, NIME 2011, Oslo, Norway, May 30 - June 1, 2011*, 2011, pp. 519–522.
- [4] William G. Gardner, “Efficient convolution without input-output delay,” *Journal of the Audio Engineering Society*, vol. 43, no. 3, pp. 127, 1995.
- [5] Øyvind Brandtsegg, “A toolkit for experimentation with signal interaction,” in *Proceedings of the 18th International Conference on Digital Audio Effects (DAFx-15)*, 2015, pp. 42–48.
- [6] Øyvind Brandtsegg, Trond Engum, Andreas Bergslund, Tone Aase, Carl Haakon Waadeland, Bernt Isak Wærstad, and Sigurd Saue, “T-temp communication and interplay in an electronically based ensemble,” <https://www.researchcatalogue.net/view/48123/48124/10/10>, 2013.
- [7] Øyvind Brandtsegg, Trond Engum, Tone Aase, Carl Haakon Waadeland, and Bernt Isak Wærstad, “Evil stone circle,” <https://www.cdbaby.com/cd/temptrondheimelectroacou>, 2015.
- [8] Lars Eri Myhre, Antoine H Bardoz, Sigurd Saue, Øyvind Brandtsegg, and Jan Tro, “Cross convolution of live audio signals for musical applications,” in *International Symposium on Computer Music Multidisciplinary Research*, 2013, pp. 878–885.
- [9] Øyvind Brandtsegg and Sigurd Saue, “Experiments with dynamic convolution techniques in live performance,” in *Linux Audio Conference*, 2013.
- [10] Chris Donahue, Tome Erbe, and Miller Puckette, “Extended convolution techniques for cross-synthesis,” in *Proceedings of the International Computer Music Conference 2016*, 2016, pp. 249–252.
- [11] Jonathan S Abel, Sean Coffin, and Kyle S Spratt, “A modal architecture for artificial reverberation,” *Journal of the Acoustical Society of America*, vol. 134, no. 5, pp. 4220–4220, 2013.
- [12] Jonathan S Abel and Kurt James Werner, “Distortion and pitch processing using a modal reverberator architecture,” in *International Conference on Digital Audio Effects (DAFx-15)*, Trondheim, Norway, November/2015 2015, ;
- [13] H.J. Nussbaumer, *Fast Fourier Transform and convolution algorithms*, Springer, 1982.
- [14] Frank Wefers, *Partitioned convolution algorithms for real-time auralization*, Logos Verlag Berlin GmbH, 2015.
- [15] Thomas G. Stockham, “High-speed convolution and correlation,” in *Proceedings of the April 26-28, 1966, Spring joint computer conference*, 1966, pp. 229–233.

- [16] Frank Wefers and Michael Vorländer, “Efficient time-varying fir filtering using crossfading implemented in the dft domain,” in *Forum Acousticum. European Acoustics Association*, 2014.
- [17] Victor Lazzarini, “Extensions to the csound language: from user-defined to plugin opcodes and beyond,” in *Proceedings of the 3rd Linux Audio Conference*, 2005.
- [18] Carlos Vila, “Digital frequency shifting for electroacoustic feedback suppression,” in *Audio Engineering Society Convention 118*, May 2005.
- [19] Øyvind Brandtsegg and Kyle Motl, “Session ucسد 14. februar 2017,” <http://crossadaptive.hf.ntnu.no/index.php/2017/02/15/session-ucsd-14-februar-2017/>, 2017.
- [20] Øyvind Brandtsegg and Jordan Morton, “Convolution experiments with jordan morton,” <http://crossadaptive.hf.ntnu.no/index.php/2017/03/01/convolution-experiments-with-jordan-morton/>, 2017.

MODAL AUDIO EFFECTS: A CARILLON CASE STUDY

Elliot Kermit Canfield-Dafilou

Center for Computer Research in Music and Acoustics,
Stanford University, Stanford, CA 94305 USA
kermit@ccrma.stanford.edu

Kurt James Werner

The Sonic Arts Research Centre
Queen's University, Belfast, Northern Ireland
k.werner@qub.ac.uk

ABSTRACT

Modal representations—decomposing the resonances of objects into their vibrational modes has historically been a powerful tool for studying and synthesizing the sounds of physical objects, but it also provides a flexible framework for abstract sound synthesis. In this paper, we demonstrate a variety of musically relevant ways to modify the model upon resynthesis employing a carillon model as a case study. Using a set of audio recordings of the sixty bells of the Robert and Ann Lurie Carillon recorded at the University of Michigan, we present a modal analysis of these recordings, in which we decompose the sound of each bell into a sum of decaying sinusoids. Each sinusoid is characterized by a modal frequency, exponential decay rate, and initial complex amplitude. This analysis yields insight into the timbre of each individual bell as well as the entire carillon as an ensemble. It also yields a powerful parametric synthesis model for reproducing bell sounds and bell-based audio effects.

1. INTRODUCTION

The modal approach conceives of resonant objects and rooms by their modes of vibration, where each mode is characterized by a frequency, decay rate, and complex amplitude [1]. Historically, this concept has informed acoustic analysis of both interiors [2] and musical instruments [3], and has informed digital audio synthesis techniques including the Functional Transformation Method [4], MODALYS [5], MOSAIC [6], and advanced numerical modeling of strings [7–9] and bridges [10]. Recently, modal analysis of rooms has informed a new family of artificial reverberation algorithms called “modal reverb,” where a room’s modal response is synthesized directly as a sum of parallel filters [11, 12]. Modifications to the basic modal reverb algorithm have been used to produce novel abstract audio effects based on distortion, pitch, and time-scale modification [13] as well as an abstract Hammond-organ-based reverberation algorithm [14].

In this paper, we show that extensions to a modal *synthesis* algorithm based on analysis of recordings can also be used to produce interesting and musically useful abstract audio synthesis algorithms. We use recordings of 60 bells of the Robert and Ann Lurie Carillon at the University of Michigan as a case study of this approach [15]. Because bells are struck by a metal clapper, the driving force is impulsive. It excites a large number of inharmonic modes to resonate and decay exponentially.¹ A modal representation provides an intuitive interpretation for bell-modeling. Moreover, the model is well suited for manipulations allowing us to augment bell-sound synthesis for extended audio processing.

The University of Michigan released the audio recordings of one of their carillons as part of an initiative to promote contem-

porary composers to write electroacoustic music for their carillon. While the samples could be manipulated in a variety of ways, a parametric model of the carillon is useful for producing an extended range of bell-based sounds and effects. Our modal analysis yields insight into the timbre of each individual bells as well as the entire carillon as an ensemble, and provides a powerful parametric synthesis model.

Carillons date back to the middle of the seventeenth century. A carillon is a musical instrument consisting of at least 23 bells that are tuned chromatically and played with a baton keyboard interface [16].² Carillons are usually held in towers and are the second largest instrument following the pipe organ. Over the last 350 years, bell casting technology has improved and musical sensibilities have changed. The Lurie Carillon consists of sixty Royal Eijsbouts bells that were cast in 1995–6 in the Netherlands [17, 18]. It is one of the largest, heaviest, and youngest carillons in the United States.

Carillons and bells in general have been widely studied by acoustic researchers. Some are interested in the physics of bell sound production [3, 19]. Others have investigated the perception of strike tones [20]. Of particular relevance to this work, [21–25] have made measurements of carillons and investigated the tuning of carillon bells. While some researchers start with physical measurements or models of bells for finite element analysis, we are performing our analysis on single recordings of each bell.

In §2 we discuss the process extracting the modal frequencies, decay rates, and initial complex amplitudes. Next, §3 presents an analysis of the harmonic structure and tuning of the Lurie Carillon, discusses methods for using the analysis data to retune the carillon in the resynthesis process, and proposes novel audio effects that make use of the modal data as a framework. Finally, §4 offers some concluding thoughts.

2. ANALYSIS

There are a number of ways to estimate modal parameters, including Prony’s method [26] and the matrix pencil [27]. In this paper, we use a technique based on successive estimation of frequencies, then decay rates, then complex amplitudes, which uses well-known and fundamental signal processing tools like the FFT, bandpass filtering, and linear least-squares estimation. Similar approaches are used to estimate parameters for modal reverb [12] and other bell studies [21].

²An instrument consisting of fewer than 23 bells is called a chime and a grand carillon requires at least 47 bells.

¹In this paper, we use the words mode and partial interchangeably.

2.1. Modal Approach

We use modal analysis to represent each bell of the Lurie Carillon as a sum of exponentially decaying sinusoids

$$\mathbf{x} = \sum_{m=1}^M a_m e^{j2\pi\omega_m t} e^{-t/\tau_m}, \quad (1)$$

where a_m is the complex amplitude, ω_m the frequency, and τ_m the decay rate for each mode m . An analysis block diagram can be seen in Fig. 1, and the steps for estimating the parameters a , ω , and τ are as follows:

1. Perform peak picking in the frequency domain
2. Form band-pass filters around each peak
3. Compute Root-Mean-Squared (RMS) energy envelopes of the band-passed signals
4. Estimate the decay rate on the energy envelopes
5. Estimate the initial complex amplitudes

2.2. Estimating Frequency

The first step in our analysis is to estimate the modal frequencies of each bell. We use a peak picking algorithm in the frequency domain to identify candidate frequencies. Before taking the Fourier Transform for each bell, we high-pass filter the time domain signal half an octave below the hum tone for that bell (see Table 1 for a detailed list of carillon bell modes). The carillon is located outdoors in a high-noise environment and this reduces the likelihood of picking spurious peaks that are simply background noise. Additionally, we use a portion of the time domain bell recording beginning 10ms after the onset so the bell's noisy transient does not produce a large number of false peaks in the FFT. We use a length 2^{14} Hann window, which produces side-lobes that roll off approximately 18dB per octave.

Our algorithm identifies peaks around points in the frequency domain signal where the slope changes sign from both sides. We then integrate the energy in the FFT bins surrounding the identified peaks and discard peaks that have low power or fall too close to one another. Finally, we pick the N highest candidate peaks that are above some absolute threshold (set by inspection for each bell). For the lowest bells, we estimate around 50 modes and for the highest bells we estimated fewer than ten.

While longer-length FFTs provide better frequency resolution, the fact that we are estimating decaying sinusoids runs counter to this argument. With a long FFT and modes that decay quickly, we will amplify noise that occurs later in the recording. Therefore, the signal to noise ratio is best at the beginning of the recordings. Experimentally, we found 2^{14} samples to be an ideal length across the sixty bells of the carillon, and Fig. 2 shows the results of the peak picking algorithm for several bells.

2.3. Estimating Exponential Decay

We use each frequency found in §2.2 as the center frequency for a fourth-order Butterworth band-pass filter. We find the energy envelope for each partial by averaging the band-pass filtered signals using a 10ms RMS procedure. We then perform a linear fit to the amplitude envelope using least squares to estimate the decay rate of each partial. The region over which the linear fit is performed

Table 1: Partial name and intervalic relationship to the fundamental (prime)

Partial Name	Partial Interval
Hum	Octave (below)
Prime	Fundamental
Tierce	Minor Third
Quint	Perfect Fifth
Nominal	Octave
Deciem	Major Third
Undeciem	Fourth
Duodeciem	Twelfth
Double Octave	Octave
Upper Undeciem	Upper Fourth
Upper Sixth	Major Sixth
Triple Octave	Octave

was found by hand as the bell recordings had large variance of partial-signal-level and noise floor. The result of the slope fitting can be seen in Fig. 3.

2.4. Estimating Complex Amplitude

Once we have estimated the frequency and decay rate of each mode, we estimate the initial amplitude of each partial required to reconstruct the original bell recording. To do this, we form a matrix where each column holds each partial independently as in

$$\mathbf{M} = \begin{bmatrix} 1 & \dots & 1 \\ e^{(j\omega_1 - \tau_1)} & \dots & e^{(j\omega_M - \tau_M)} \\ \vdots & \ddots & \vdots \\ e^{(j\omega_1 - \tau_1)T} & \dots & e^{(j\omega_M - \tau_M)T} \end{bmatrix}, \quad (2)$$

where ω_m are the frequencies, τ_m the decay rates, and T is the length of the time vector. We use least squares to find the complex amplitudes

$$\mathbf{a} = (\mathbf{M}^\top \mathbf{M})^{-1} \mathbf{M}^\top \mathbf{x}, \quad (3)$$

where \mathbf{x} is the original bell recording and \mathbf{a} the vector of complex amplitudes.

2.5. Results

As a result of our analysis, we have estimated frequencies, decay rates, and initial complex amplitudes necessary to model the bells as they were recorded. Fig. 4 plots these parameters for three bells throughout the range of the instrument.

3. RESYNTHESIS

We can resynthesize a “noiseless” copy of the original carillon recordings by directly plugging in the estimated amplitudes, frequencies, and decay rates, into Eq. (1). We can also implement the bell as a filter with the transfer function

$$\frac{X(z)}{U(z)} = \sum_{m=1}^M \frac{2\Re\{a_m\} - 2e^{-1/\tau_m} \Re\{a_m e^{-j\omega_m}\} z^{-1}}{1 - 2e^{-1/\tau_m} \cos(\omega_m) z^{-1} + e^{-2/\tau_m} z^{-2}}, \quad (4)$$

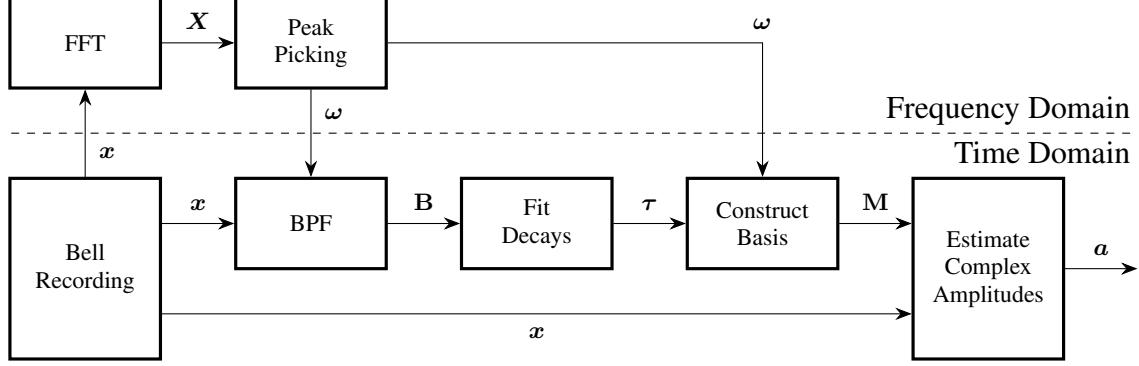


Figure 1: Block diagram detailing the analysis process for finding the modal frequencies (ω), decay rates (τ), and complex amplitudes (a).

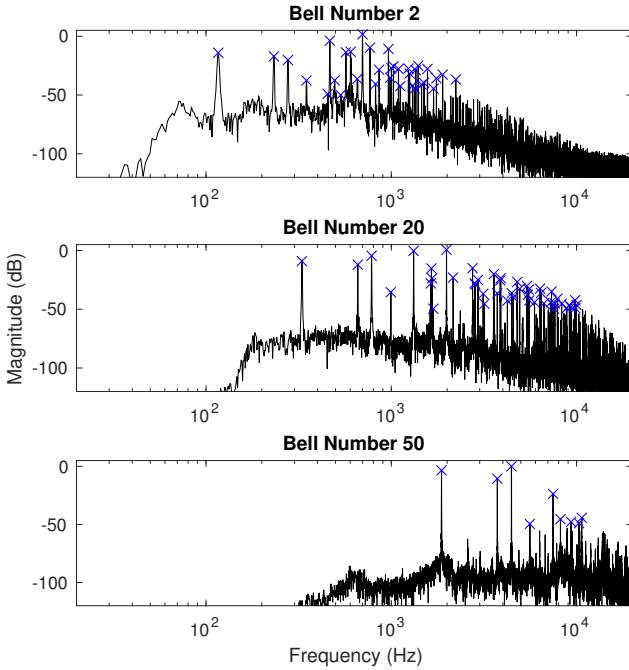


Figure 2: Frequency spectrum of three Bells from the Lurie Carillon showing the results of the peak picking algorithm (in blue).

where U is the input. This is the same as a parallel sum of M standard second order (biquad) transfer functions

$$\frac{X(z)}{U(z)} = \sum_{m=1}^M \frac{\beta_{0,m} + \beta_{1,m}z^{-1} + \beta_{2,m}z^{-2}}{1 + \alpha_{1,m}z^{-1} + \alpha_{2,m}z^{-2}} \quad (5)$$

with coefficients

$$\begin{aligned} \beta_{0,m} &= 2\Re\{a_m\} \\ \beta_{1,m} &= -2e^{-1/\tau_m}\Re\{a_m e^{-j\omega_m}\} \quad \alpha_{1,m} = -2e^{-1/\tau_m}\cos(\omega_m) \\ \beta_{2,m} &= 0 \quad \alpha_{2,m} = e^{-2/\tau_m}. \end{aligned}$$

If U is an impulse, we reconstruct the sound of the bell, and Eq. (4) is equivalent to Eq. (1). Both synthesis models take the form of summing modes as seen in Fig. 5.

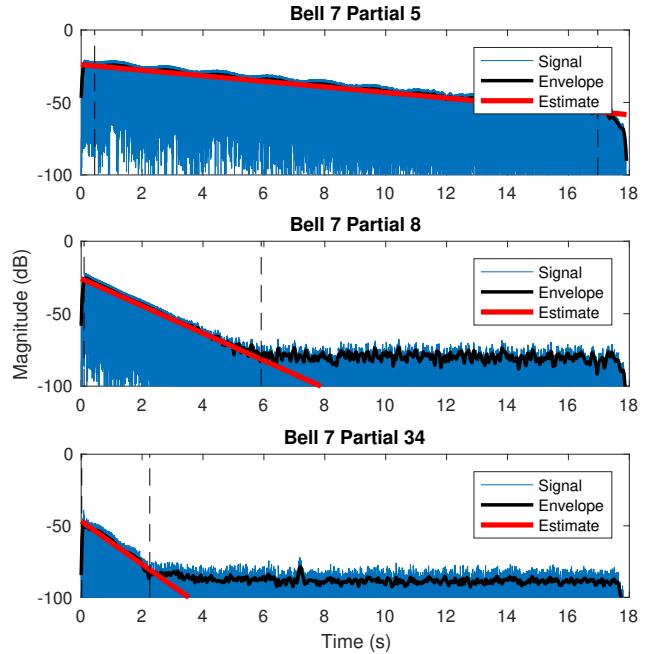


Figure 3: Estimated decay rates for three partials of bell seven. Each plot shows the recorded bell signal band-pass filtered around the partial frequency (blue), the RMS smoothed energy envelope (black), and the estimated decay rate (red). Dotted lines show the region over which the linear fit was performed.

We find that thirty modes for a low-pitched bell and ten for a high-pitched bell do a reasonable job reproducing the sound of the recordings. Using the modal data extracted from the recordings of the Lurie Bells, we find the correlation coefficient between the recorded bells and synthesized bells was on average 0.837 with a standard deviation of 0.117. Fig. 6 shows the result of this resynthesis for one of the bells of the Lurie Carillon. One major advantage to the resynthesis is that the real bells ring for a significantly long time. The recordings faded out once the bell is indistinguishable from the noise floor. The resynthesis does not have this limitation and is therefore more impervious to amplitude modifications.

As a result of our analysis, we have a parameterized model rather than simply a means for synthesizing the sound of the orig-

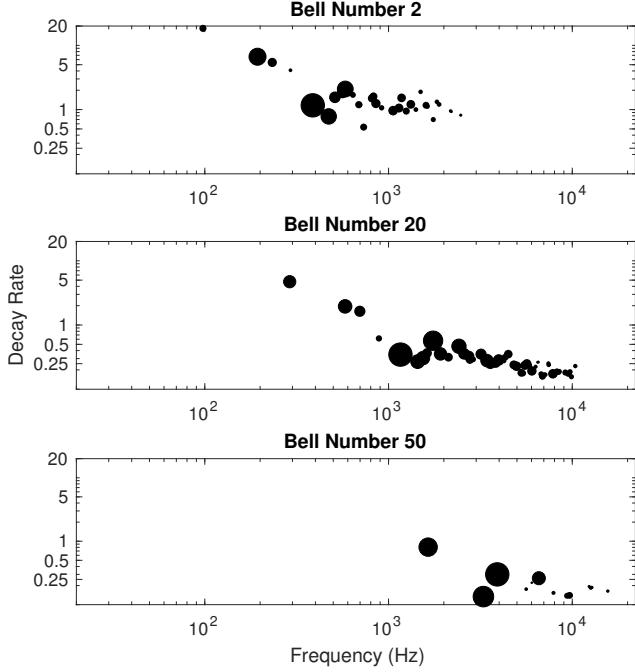


Figure 4: Frequency vs decay rate for three analyzed bells with marker diameter showing each partial’s amplitude.

inal recordings. In addition to providing a way to study the harmonic structure and tuning of the Lurie Carillon, we can manipulate the model to synthesize bell sounds with different characteristics or bell-based audio effects. The analysis/resynthesis diagram can be seen in Fig. 7. The modifications we make in the resynthesis primarily involve scaling and adding offsets to the estimated frequencies, amplitudes, and decay rates, as well as the use of external control data. Some bell related processing is described below. Audio examples and code to produce them can be found at <https://ccrma.stanford.edu/~kermit/website/bells.html>.

3.1. Computer Aided Electronic Bell Foundry

Carillons bells are designed to function together as a cohesive instrument, however the range of the instrument, the process of casting bronze bells, and the way the instrument wears when played make this challenging. The Lurie Carillon has sixty bells that span five octaves. The largest bells weigh up to 5000kg while the smallest weigh only 5kg. Naturally this means the bells across the instrument are cast with different shapes and it takes skill and patience to tune the instrument as a whole. Furthermore, the bells are often exposed to extreme weather conditions as they are located in semi-open spaces atop towers. Over time the bells do not necessarily wear in the same way due to how often each is struck and its independent location within the belfry. Last, bell foundry technology and playing techniques have developed over time [28]. Bells cast by each manufacture have subtle differences, and while most modern instruments are tuned to equal temperament, this is not the case for older instruments.

For these reasons, we propose a collection of manipulations using the modal data from the Lurie Carillon to modify the sound

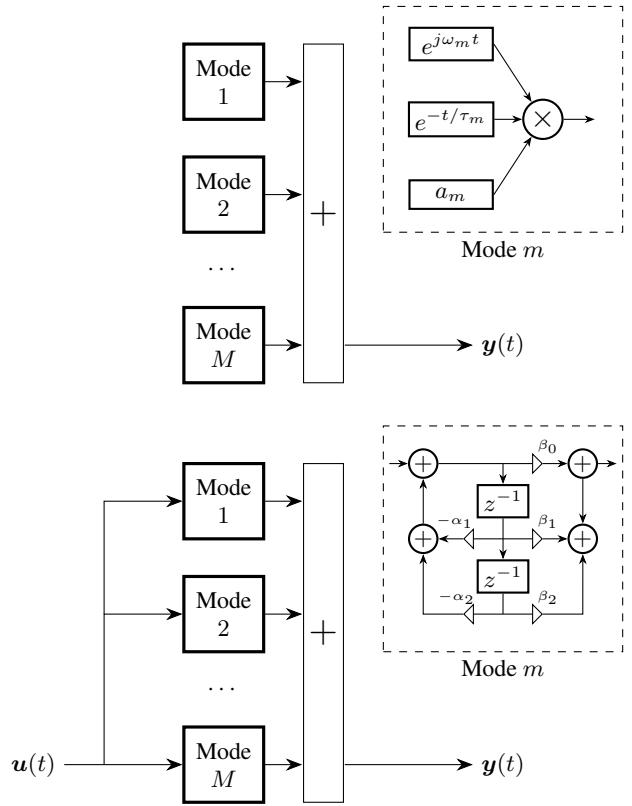


Figure 5: Direct resynthesis (top) and biquad (bottom) models.

in ways that could be relevant to carillonists and musicologists. Fig. 8 shows all the bells of the Lurie Carillon, pitch-shifted to share a common fundamental. From inspection, one can see that the lowest two partials are in tune across the whole instrument but the quint (fourth partial) starts going sharp to the point that it is almost half an octave sharp in the highest bells. Additionally, the highest bells have partials that are significantly flatter than the other bells. Even so, the bells of the instrument sound well matched lending precedence to the fact that one cannot use a single bell and pitch shift it up and down to synthesize bells at all pitch-heights.

3.1.1. Fixing Irregularities Due to Wear

Tuning carillon bells is an arduous process. The procedure often involves turning the bells on a lathe to evenly remove material at specific points on the bell to sharpen specific partials. Modern technology has improved bell tuners’ ability to make carillon bells sound more homogeneous, however, material can only be removed. This makes it impossible to flatten the pitch of a bell. Furthermore, even though the bells are designed to sound like the rest of the bells of a specific instrument, the nature of bell casting makes this idealism impossible. With our analysis of the full set of bells from the Lurie Carillon, we can average the modal parameters of several consecutive bells in order to smooth out any irregularities caused by a poorly tuned bell.

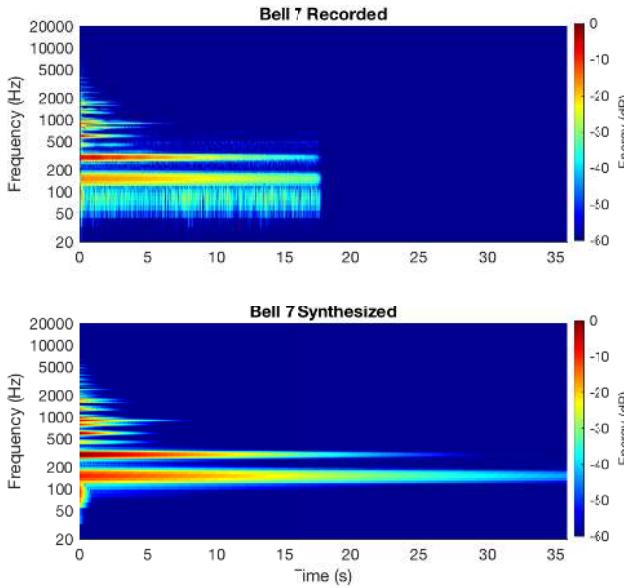


Figure 6: Spectrograms showing original recording and resynthesis of bell number seven.

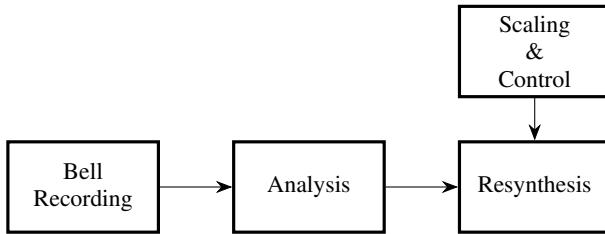


Figure 7: Analysis/Resynthesis block diagram.

3.1.2. Retuning the Carillon

In addition to making the carillon bells sound more consistent with one another, we can also retune the entire carillon. While modern instruments are tuned in equal temperament, there is an enormous repertoire that was written for mean-tone and other tuning schemes. Some music was written to be played in mean-tone temperament and does not sound as the composer intended when played in equal temperament. With the modal data, we can make two types of adjustments. First, we can resynthesize a set of carillon bell samples with the same characteristics as the Lurie Carillon but shift the frequencies of each bell’s partials such that the fundamentals are tuned to a scheme other than the current equally tempered bells. Second, we can modify the tuning within each bell, correcting irregularities and ill-tuned partials.

3.1.3. Extending the Range of the Carillon

The Lurie Carillon consists of sixty bells, however there are larger instruments. For example, the Laura Spelman Rockefeller Memorial Carillon at the Riverside Church in New York City has a carillon with 74 bells [29]. Carillons this large are rare as they are expensive to construct and the towers that hold them must be able

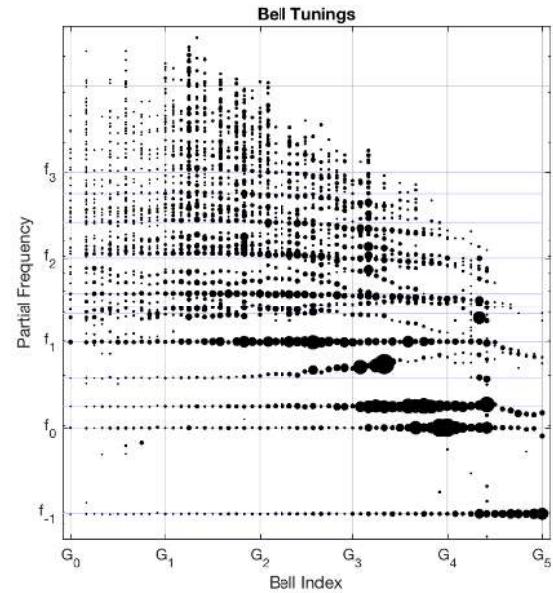


Figure 8: Estimated parameters of all the bells of the Lurie Carillon with common fundamental. Marker diameter corresponds to bell-power integrated over the first second. The measured partial frequencies are overlaid on the theoretical 12-tone equal temperament tunings (blue lines).

to support the immense weight of such a heavy instrument. Moreover, the Lurie Carillon, like many instruments, is missing the second largest bell for the above mentioned reasons and the fact that it is rarely needed to perform the standard carillon literature. By extrapolating from the measured modal data, we can virtually extend the range of the instrument to incorporate missing bells and ones higher or lower in pitch than the physical bells. Additionally, we can interpolate between bells to produce notes other than the 12-tone equally tempered pitches. This allows us to synthesize, for example, a set of quarter-tone bells that would be useful to contemporary composers, but would be impractical to cast otherwise.

3.1.4. Adapting the Decay Rate

The tradition of writing and performing carillon music has naturally progressed alongside the technical developments of bell casting and carillon construction. In many ways, the style of carillon music has worked around physical limitations of the acoustic instrument. One fundamental property of a carillon is that the higher pitched bells ring for a shorter amount of time than the lower ones. A Dutch technique was developed for playing slower tempo pieces that utilizes playing tremolo and trill gestures in the high bells to mimic a sustain [30]. Some carillonists find this technique displeasing to the ear. To overcome the fact that the high bells decay much faster than the lower bells, we can modify the decay rates of the higher bells so they last a similar length as the lower bells. This would allow one to play legato pieces and circumvent the physical limitation inherent in the physical instrument. We can also render the low bells with a quicker decay rate so that they can be played faster without sounding muddy to their natural, long decay rate.

3.1.5. Major Third Bells

Until recently (1980s), the spectrum produced by most bells has prominently featured a minor third.³ This pronounced harmonic structure caused bell-to-bell harmonies to often have a "rough" sound due to the clashing of the tunings. Due primarily to the immense construction cost, there are very few major-third carillons in existence. It is rather humorous, but carillonists are used to the sound of old bells and mostly prefer the traditional sound while modern composers say they prefer major-third bells for major-key pieces and minor-third bells for minor-key pieces [30]. It is easy for us to resynthesize the Lurie Carillon samples with the partial related to the minor third modulated up a half-step.

3.2. Electroacoustic Modifications

In addition to manipulating the modal parameters and synthesizing exponentially decaying bell sounds, we can produce a host of time-varying bell effects that could be useful to electroacoustic composers.

3.2.1. Adapt the Decay Rates

A limitation that contemporary composers writing for carillon often encounter is the speed at which the low bells can be re-struck. Physically, there is a minimum time necessary to move the clapper into the bell. Ignoring that, the long ring time of the lowest bells makes the sound of re-striking the bell blend together. The articulations become indistinguishable from the previous strike's sustain. If one simply scales the decay rates of all the partials of a bell, the resulting sound appears to be low-pass filtered as the high frequencies would decay quickly. To improve the result, we propose scaling the decay rates nonlinearly to only shorten the ring time of the lowest frequencies. This affords a composer the ability to re-trigger the low bells quickly without the transients becoming blurred out and without the whole bell sounding filtered.

3.2.2. Add Arbitrary Amplitude Envelopes

Typically, bells are characterized by an exponentially decaying energy envelope. In resynthesis, we can use the estimated frequencies and replace the amplitude envelopes with arbitrary functions (e.g., ADSRs). In effect, this allows us to allude to the harmony of the carillon bells without producing sounds that decay exponentially like the bells.

3.2.3. Spectral Morphing

In his magnum opus, *Mortuos Plango, Vivos Voco*, Jonathan Harvey uses the spectrum of the tenor bell from the Winchester Cathedral as the basis for the harmony and structure of his piece [31]. Using the modal data, we can morph between various bells by connecting the partials together with glissandi. In Harvey's piece, the composer resynthesizes the individual partials of the bell using a pitch-shifted sample of a boy's voice. We too can use the extracted frequencies for each partial to control other musical parameters, such as the rate at which a buffer is read, the pitch of a synthesizer, or the center frequency for a filter.

³Unlike the harmonic spectrum produced by most musical instruments.

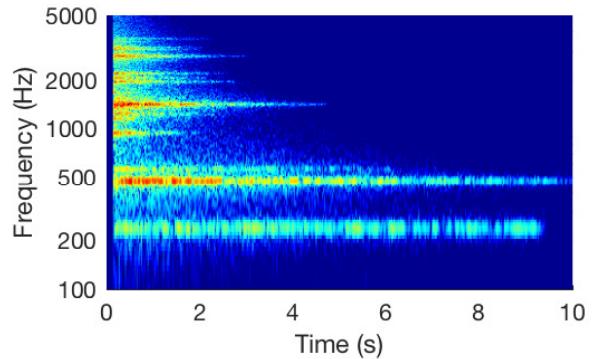


Figure 9: Spectrogram of exponentially-enveloped Gaussian noise passed through band-pass filters centered around the estimated modal frequencies.

3.2.4. Bells as Reverb

Our default synthesis model resynthesizes each partial as an exponentially decaying sinusoid. Instead, we can pass Gaussian noise through band-pass filters, each centered around the bell-partial frequencies. We can additionally control the *Q* factor of the filters in order to control the bandwidth of the noise at each partial. Alone, this resynthesis creates an airy sound effect that has bell-like characteristics (see Fig. 9), but we can also use it as a reverberation effect applied to other sounds.

3.2.5. Add Modulation and Doublets

Natural sounding bells have beating due to minor imperfections. Across the measured bells, there is a high variance for how much beating there is (both how different the frequencies are and how strongly the two modes differ in amplitude and phase). In the resynthesis, we can specifically control how much beating occurs on a partial-by-partial basis. We can achieve this in two ways. First, for each measured partial we can synthesize two partials that have small differences in the modal parameters. Second, we can apply a low frequency oscillator (LFO) to each partial to apply amplitude or frequency modulation. Both approaches allow us to synthesize natural sounding bell sounds. By applying severe detuning or modulation, we can create additional effects (e.g., vibrato, tremolo, FM distortion, etc). Moreover, we can change this effects dynamically. For example, we can have the sound of a bell that has no modulation, one second into the decay apply vibrato, and two seconds into the decay increase the modulation depth and speed to create distortion.

3.2.6. Spatialization and Time Modifications

Since we have control of each partial, we can modulate their spatial positioning and onset timing independently. Thus we can deconstruct the bell into an arpeggio, randomize the entrance time for each partial, or spread the bell across the stereo field.

3.3. Summary

Fig. 10 shows a full sound example using several of the techniques described above. At the beginning, the bell is synthesized three times with each partial entering at a different time. Over the first

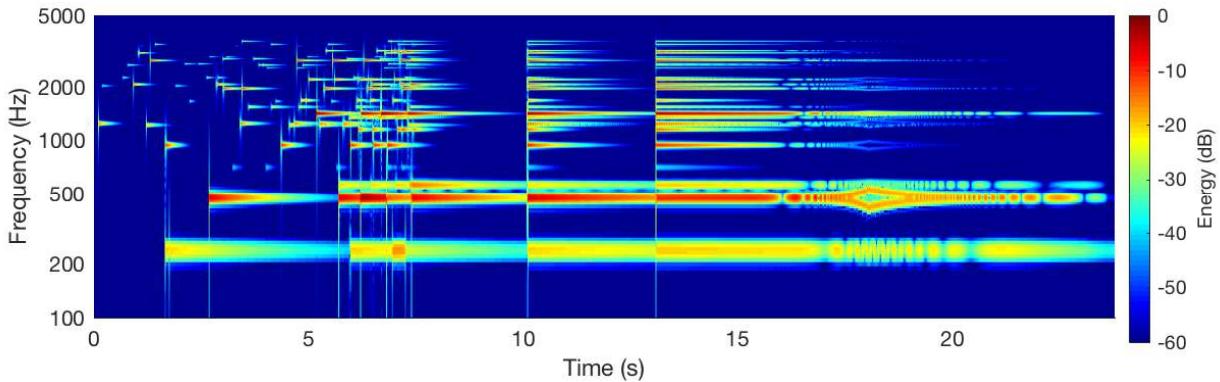


Figure 10: Spectrogram of a through composed example demonstrating scaling and modulation effects.

eight seconds, the timing between each partial’s entrance decreases and the decay rates for each partial is slowly lengthened. At ten seconds, the unmodified bell is sounded. The bell is struck once more around second twelve with a long decay and a modulation applied on each partial. The modulation depth is increased until a frequency modulation effect is apparent before slowing back to a vibrato.

Overall, this is not a comprehensive list of modal resynthesis techniques. In fact, this exploration just shows that the range of sonic possibilities is enormous, even when constrained by a simple model consisting of frequency, decay rate, and initial amplitude.

4. CONCLUSIONS

In this paper we demonstrate how a modal model for sound synthesis can be manipulated to achieve a wide range of musical effects. Using the carillon as a case study, we provide an analysis of the University of Michigan’s Lurie Carillon. By modeling each bell as a sum of decaying sinusoids at each modal frequency, we provide a versatile model that lends itself well to a host of audio manipulations consisting primarily of scaling and shifting the parameters. As a tool for composers, we hope this analysis of the Lurie Carillon will help encourage new electroacoustic music for carillon. The modal data generated from analyzing the samples from [15] is freely available under a creative commons license and can be downloaded at <https://ccrma.stanford.edu/~kermit/website/bells.html>.

The method used in this paper has a limited ability to resolve very closely spaced partials, known as “doublets” in the bell context. Future work will attempt to estimate doublet parameters using nonlinear optimization. An alternate approach to modeling doublet behavior is shown in [32, 33], where groups of closely-spaced modes are approximated using ARMA modeling.

5. ACKNOWLEDGMENTS

The authors thank Jonathan Abel for his helpful comments.

6. REFERENCES

- [1] Stefan Bilbao, *Numerical Sound Synthesis*, Wiley, 2009.

- [2] 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, July 2012.
- [3] Tomas Rossing and Robert Perrin, “Vibrations of bells,” *Applied Acoustics*, vol. 20, pp. 41–70, 1987.
- [4] Lutz Trautmann and Rudolf Rabenstein, *Digital Sound Synthesis by Physical Modeling Using the Functional Transform Method*, Springer, New York, 1st edition, 2003.
- [5] Gerhard Eckel, Francisco Iovino, and René Caussé, “Sound synthesis by physical modelling with Modalys,” in *Proceedings of the International Symposium on Musical Acoustics*, Dourdan, France, 1995.
- [6] Joseph Derek Morrison and Jean-Marie Adrien, “MOSAIC: A framework for modal synthesis,” *Computer Music Journal*, vol. 17, no. 1, pp. 45–56, Spring 1993.
- [7] Maarten van Walstijn and Jamie Bridges, “Simulation of distributed contact in string instruments: A modal expansion approach,” in *Proceedings of the 24th European Signal Processing Conference (EUSIPCO)*, Budapest, Hungary, August 29 – September 2 2016, pp. 1023–7.
- [8] Maarten van Walstijn, Jamie Bridges, and Sandor Mehés, “A real-time synthesis oriented tanpura model,” in *Proceedings of the 19th International Conference on Digital Audio Effects (DAFx-16)*, Brno, Czech Republic, September 5–9 2016, pp. 175–82.
- [9] Clara Issanchou, Stefan Bilbao, Cyril Touze, and Olivier Doare, “A modal approach to the numerical simulation of a string vibrating against an obstacle: Applications to sound synthesis,” in *Proceedings of the 19th International Conference on Digital Audio Effects (DAFx-16)*, Brno, Czech Republic, September 5–9 2016.
- [10] Esteban Maestre, Gary P. Scavone, and Julius O. Smith, “Design of recursive digital filters in parallel form by linearly constrained pole optimization,” *IEEE Signal Processing Letters*, vol. 23, no. 11, pp. 1547–50, 2016.
- [11] Jonathan S. Abel, Sean Coffin, and Kyle S Spratt, “A modal architecture for artificial reverberation,” *The Journal of the Acoustical Society of America*, vol. 134, no. 5, pp. 4220, 2013.

- [12] 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 Convention of the Audio Engineering Society*, Los Angeles, CA, October 9–12 2014.
- [13] 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)*, Trondheim, Norway, November 30 – December 3 2015.
- [14] Kurt James Werner and Jonathan S. Abel, “Modal processor effects inspired by Hammond tonewheel organs,” *Applied Sciences*, vol. 6, no. 7, pp. 1421–48, 2016, Article #185.
- [15] Isaac Levine, Ashton Baker, Rowena Ng, Rachael Park, Anjana Rajagopal, Tiffany Ng, and John Granzow, “Download Lurie carillon samples,” <https://goblubellows.wordpress.com/2016/09/21/lurie-carillon-samples/>.
- [16] “World carillon federation,” <http://www.carillon.org/>.
- [17] “Lurie carillon,” https://www.music.umich.edu/about/facilities/north_campus/lurie/lurie.htm.
- [18] “Ann & Robert H. Lurie Carillon,” <http://www.towerbells.org/data/MIANNAU2.HTM>.
- [19] Robert Perrin and T. Charnley, “A comparative study of the normal modes of various modern bells,” *Journal of Sound and Vibration*, vol. 117, no. 3, pp. 411–420, 1987.
- [20] William Hibbert, *The Quantification of Strike Pitch and Pitch Shifts in Church Bells*, Ph.D. thesis, The Open University, 2008.
- [21] Xavier Boutillon and Bertrand David, “Assessing tuning and damping of historical carillon bells and their changes through restoration,” *Applied Acoustics*, vol. 64, pp. 901–10, 2001.
- [22] Vincent Debut, Miguel Carvalho, and Jose Antunes, “An objective approach for assessing the tuning properties of historical carillons,” in *Proceedings of the Stockholm Music Acoustics Conference*, Stockholm, Sweden, July 2013.
- [23] Wiegman van Heuven, *Acoustical measurements on church-bells and carillons*, Ph.D. thesis, De Gebroeders van Cleef, The Hague, Netherlands, 1949.
- [24] Andre Lehr, “The system of the Hemony-carillons tuning,” *Acustica*, vol. 3, pp. 101–104, 1951.
- [25] Albrecht Schneider and Marc Leman, *Studies in Musical Acoustics and Psychoacoustics*, chapter Sound, Pitches and Tuning of a Historic Carillon, pp. 247–98, Springer, 2016.
- [26] T. W. Parks and C. S. Burrus, *Digital Filter Design*, Wiley-Interscience, New York, NY, USA, 1987.
- [27] Y. Hua and T. K. Sarkar, “Matrix pencil method for estimating parameters of exponentially damped/undamped sinusoids in noise,” *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 38, no. 5, pp. 814–24, May 1990.
- [28] Andre Lehr, “Contemporary Dutch bell-founding art,” *Netherlands Acoustical Society*, , no. 7, pp. 20–49, 1965.
- [29] “The Riverside Church carillon,” www.trcnyc.org/events/carillon?/.
- [30] Brian Swager, *A History of the Carillon: Its origins, Development, and Evolution as a Musical Instrument*, Ph.D. thesis, Indiana University, 1993.
- [31] Jonathan Harvey, “Mortuos Plango, Vivos Voco: A realization at IRCAM,” *Computer Music Journal*, vol. 5, no. 47, 1981.
- [32] Matti Karjalainen, Paulo A. A. 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. 1012–29, December 2002.
- [33] 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 (DAFx-02)*, Hamburg, Germany, September 25–28 2002, pp. 181–6.

LP-BLIT: BANDLIMITED IMPULSE TRAIN SYNTHESIS OF LOWPASS-FILTERED WAVEFORMS

Sebastian Kraft, Udo Zölzer

Department of Signal Processing and Communications
Helmut-Schmidt-University
Hamburg, Germany
sebastian.kraft@hsu-hh.de

ABSTRACT

Using bandlimited impulse train (BLIT) synthesis, it is possible to generate waveforms with a configurable number of harmonics with an equal amplitude. In contrast to the sinc-pulse, which is typically used for bandlimiting in BLIT and only allows to set the cutoff frequency, a Hammerich pulse can be tuned by two independent parameters for cutoff frequency and stop band roll-off. Replacing the perfect lowpass sinc-pulse in BLIT with a Hammerich pulse, it is possible to directly synthesise a multitude of signals with an adjustable lowpass spectrum.

1. INTRODUCTION

Subtractive sound synthesis with analogue synthesisers requires generic and spectrally rich harmonic waveforms. Traditionally, these are rectangular, triangular and sawtooth waveforms. Their flat spectrum is successively shaped by filters until it matches the expectations of the musician. These filters usually have a low-pass characteristic with a variable cutoff frequency, adjustable resonance peak and a slope of 12 or 24 dB per octave. One major problem of digital subtractive synthesis is the fact that trivial implementations of the required basic waveforms leads to massive aliasing. The creation of bandlimited oscillators has been an active research topic ever since digital sound synthesis became of interest. An extensive summary and discussion of various methods can be found in [1] and [2]. The latter also introduced the PolyBLEP approach (detailed in [3]) which today is a standard method to create high quality bandlimited waveforms due to its low computational cost and easy implementation.

Subtractive synthesis is a straightforward approach when building analogue synthesisers. Today, many digital synthesisers mimic an analogue subtractive workflow, mainly due to the fact that musicians have been used to it for decades and the use of filters to shape the sound is at the same time intuitive, simple and versatile. However, for the performing musician the exact synthesis method does not matter. It is more important that the synthesiser allows to create musically sounding signals which can be intuitively controlled by only a few but powerful parameters [4]. From an engineering point of view it does not make sense to put a lot of effort into the generation of aliasing-free waveforms with harmonics up to half the sampling frequency and then to remove most of the high frequency content with a lowpass filter. A method to directly synthesise the desired signal spectra would alleviate the effort for anti-aliasing strategies as such a signal will contain less high frequency content from the beginning. Additive synthesis as a discrete summation of amplitude-weighted sine waves would offer full control to the creation process of perfectly bandlimited signals but due to the resulting computational complexity it is rarely used in practice.

The discrete summation formula (DSF) from Moorer [5] are a set of closed form solutions to discrete harmonic series (infinite or finite). They allow a direct synthesis of harmonic signals with a specified number of partials and an exponentially increasing or decreasing spectral envelope. By the combination of differently parametrised DSF one can create further complex spectral envelopes. Although the DSF are more efficient than a discrete additive synthesis they still require a considerable amount of trigonometric function evaluations.

Bandlimited impulse train (BLIT) synthesis [1] is another approach to create lowpass signals and relies on the fact that an impulse train exhibits a flat spectrum with an infinite amount of harmonics. When the impulse train is convolved with a sinc-function, a perfect bandlimited signal will be obtained, whereas the frequency of the sinc-pulse determines the cutoff frequency. In practical applications, the infinite length sinc-function has to be windowed and limited to a reasonable length and the computationally expensive convolution is replaced by a summation of overlapping sinc-pulses with a pulse distance proportional to the fundamental frequency (sum of windowed sinc-function BLIT synthesis [1]). A windowed sinc-function will lose its perfect lowpass characteristic and stop-band ripple occurs. The selection and parametrisation of the window and its length influences the final lowpass shape [6]. BLIT synthesis was further developed and optimised in various aspects (e.g. in [7, 8]), but the created signals have been always used as a bandlimited input for subtractive workflows. To the knowledge of the authors, a direct BLIT synthesis of signals with real-time configurable lowpass spectra has not been investigated so far.

The Hammerich pulse [9] was introduced as a pulse shape filter [10] for transmission systems and its spectral shape can be tuned by two independent parameters for cutoff frequency and stop band roll-off. Replacing the sinc-pulses in BLIT with Hammerich pulses allows to directly synthesise signals with adjustable low-pass spectra. The fundamental frequency, cutoff frequency and stop band roll-off are monotonic parameters and can be modulated smoothly. In a sound synthesis application, the few parameters and inherent limitation to lowpass spectra reduces complexity in the user interface and offers the musician an intuitive and less technical access. Nevertheless, a wide variety of sounds similar to subtractive synthesisers can be created without additional filtering. Moreover, unique sounds can be generated for example by modulating the filter roll-off which would not be possible with classical analogue synthesisers.

The idea to create a versatile oscillator with Hammerich pulse shapes was already briefly described in [11]. In this paper, the focus will be on a detailed discussion of such a lowpass bandlimited impulse train (LP-BLIT) oscillator in the context of sound synthe-

sis applications. Throughout the following Section 2, the principle of BLIT synthesis and the integration of the Hammerich pulse will be explained. In Section 3 we will give some examples how to create more complex spectra by combination of several oscillator outputs, leaky integration and parameter modulation before Section 4 provides a short discussion and conclusion.

2. IMPULSE TRAIN SYNTHESIS

A continuous-time pulse train with an inter-pulse distance T_0

$$d(t) = \sum_{l=-\infty}^{\infty} \delta(t - l T_0) \quad (1)$$

exhibits a spectrum

$$D(j\omega) = \sum_{l=-\infty}^{\infty} \delta(\omega - l \omega_0), \quad \omega_0 = \frac{2\pi}{T_0} \quad (2)$$

with harmonics at multiples of ω_0 . The fact that the harmonic interval ω_0 is directly related to T_0 can be utilised to synthesise a signal with an infinite amount of harmonics and a fundamental frequency $F_0 = 1/T_0$. Sampling of $d(t)$ for digital implementations would lead to massive aliasing due to its infinite bandwidth. Applying an ideal lowpass filter with a cutoff frequency $\omega_c = 2\pi f_c$

$$h(t) = \frac{\sin(\omega_c t)}{\omega_c t} \leftrightarrow H(j\omega) = \text{rect}\left(\frac{\omega}{2\omega_c}\right) \quad (3)$$

results in a bandlimited signal

$$x(t) = d(t) * h(t) \quad (4)$$

which can then be sampled at a sample rate $f_s > 2f_c$ without aliasing [1]. In the frequency domain, the convolution from Eq. 4 will create a harmonic spectrum

$$\begin{aligned} X(j\omega) &= H(j\omega) \cdot D(j\omega) \\ &= H(j\omega) \cdot \left(\sum_{l=-\infty}^{\infty} \delta(j\omega - l \omega_0) \right) \end{aligned} \quad (5)$$

weighted with the spectral shape H of the filter. By replacing or modifying the filter, it is possible to synthesise any bandlimited harmonic signal with an arbitrary spectral shape.

Instead of a computationally expensive time-domain convolution, a direct summation of time-shifted impulse responses

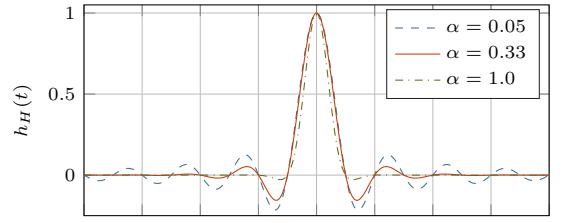
$$x(t) = d(t) * h(t) = \sum_{l=-\infty}^{\infty} h(t - l T_0) \quad (6)$$

will lead to an identical result. This is particularly useful if the impulse response can be simply calculated with a closed-form equation as it is the case with the sinc-function.

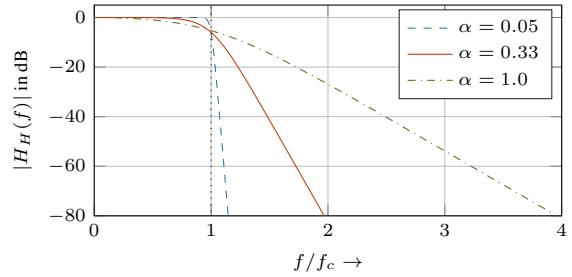
Other pulses with a lowpass characteristic, as for example the raised cosine pulse, offer a more detailed adjustment of their frequency response than the sinc-function. The Hammerich pulse was introduced in [10] as a pulse shape filter for transmission systems. Its impulse and frequency response is given by

$$h_H(t) = \frac{\alpha \cdot \sin(\omega_c t)}{\sinh(\alpha \cdot \omega_c t)} \quad (7)$$

$$H_H(j\omega) = \frac{1}{4f_c} \left[1 - \tanh\left(\frac{|\omega| - \omega_c}{4\alpha f_c}\right) \right] \quad (8)$$



(a) Time domain



(b) Frequency domain

Figure 1: Hammerich pulse for various values of α .

and permits an intuitive adjustment of the lowpass characteristic by two independent parameters for cutoff frequency ω_c and stop band slope α . Reasonable parameter ranges are $0 < \alpha < 10$ and $\omega_c > 2\pi F_0$. For small values of α , the impulse response

$$\lim_{\alpha \rightarrow 0} h_H(t) = \frac{\sin(\omega_c t)}{\omega_c t} = \text{sinc}(\omega_c t) \quad (9)$$

converges towards a sinc-function and for $\alpha \rightarrow \infty$ the resulting stop band slope as well as the pulse width converges to zero. Exemplary impulse and frequency responses are depicted in Fig. 1 for selected values of α . The -6 dB point is fairly accurate set by ω_c , whereas the linear slope beyond this point is only controlled by α . Figure 2 depicts a time-domain pulse train after convolution with a Hammerich impulse response together with the corresponding spectrum. The parameters of the Hammerich filter were chosen as $f_c = 4F_0$ and $\alpha = 0.4$. All harmonics follow the shape of the pulse spectrum as it was expected based on Eq. 5.

2.1. Discrete-time implementation with finite pulse length

By sampling the Hammerich pulse from Eq. 7 with a sample rate f_s , the discrete-time pulse

$$h_H(n) = \frac{\alpha \cdot \sin(\Omega_c n)}{\sinh(\alpha \cdot \Omega_c n)}, \quad \Omega_c = 2\pi \frac{f_c}{f_s}, \quad (10)$$

is obtained. The cutoff frequency $f_c = N_h \cdot F_0$ can also be expressed as a multiple of the fundamental frequency, whereas $N_h \geq 1$ determines the number of harmonics before the filter starts to roll off. This yields a pulse

$$h_H(n) = \frac{\alpha \cdot \sin(N_h \cdot \Omega_0 n)}{\sinh(\alpha \cdot N_h \cdot \Omega_0 n)}, \quad \Omega_0 = 2\pi \frac{F_0}{f_s}, \quad (11)$$

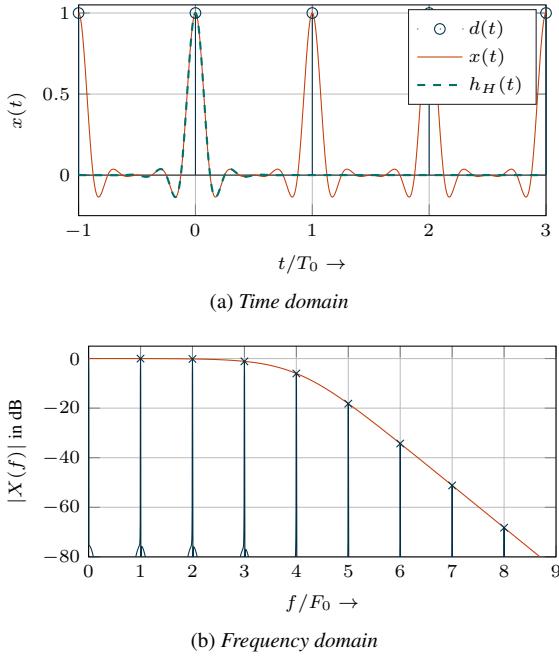


Figure 2: Pulse train convolved with a Hammerich pulse ($f_c = 4 F_0$ and $\alpha = 0.4$).

with parameters for the number of harmonics, filter slope and fundamental frequency being directly accessible from a synthesiser application.

Two facts have to be considered for a discrete-time implementation compared to the continuous-time derivation in the previous section. First, the theoretically infinite length of the pulse has to be limited in order to avoid an infinite amount of overlapping pulses in the sum from Eq. 6. This limitation of the impulse length is equivalent to a windowing of the impulse response and leads to a distortion of the pulse spectrum. The shape of the resulting error can be optimised by applying a smooth window to both ends of the pulse [6]. The number of overlapping pulses is a trade-off between computational complexity and how close the actual spectral envelope will match the theoretic spectrum given in Eq. 8. Second, even if the cutoff frequency of the Hammerich pulse is below $f_s/2$, aliasing may occur due to the flat spectral roll-off after f_c depending on the actual selection of α . Hence, it is necessary to limit the combination of the parameters N_h and α such that the resulting stop band attenuation in Eq. 8 falls below an acceptable level at half the sample rate.

Regarding computational complexity, the calculation of the sine and hyperbolic sine in the Hammerich pulse is the limiting factor. Both functions could be approximated by the Taylor series

$$\sin(x) = x - \frac{x^3}{3!} + \frac{x^5}{5!} + \dots = \sum_{k=0}^{K-1} \frac{(-1)^k \cdot x^{2k+1}}{(2k+1)!} \quad (12)$$

$$\sinh(x) = x + \frac{x^3}{3!} + \frac{x^5}{5!} + \dots = \sum_{k=0}^{K-1} \frac{x^{2k+1}}{(2k+1)!} \quad (13)$$

where the number of evaluated terms K determines the accuracy. For the periodic sine, the argument has to be wrapped to a range $x \in [-\pi/2, \pi/2]$ to minimize the required order of the Taylor

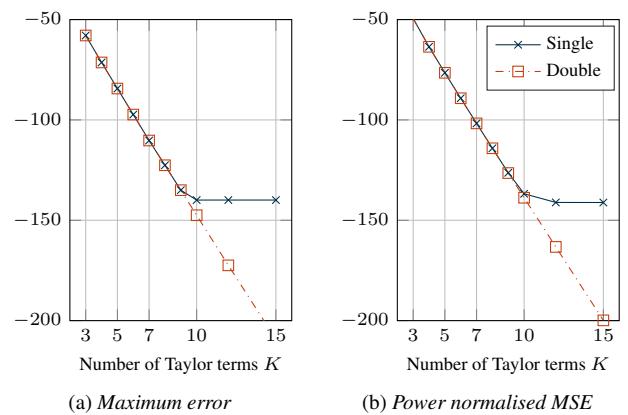


Figure 3: Error after the approximation of $h_H(n)$ with K Taylor series terms for single and double precision floating point implementations.

polynomial. The inverse factorial can be calculated in advance and using the Horner scheme for the evaluation of the polynomial, only $K - 1$ additions and multiplications are required for each Taylor series. The standard Matlab sin and sinh implementations and the single and double precision Taylor approximations were compared for the calculation of a Hammerich pulse. The respective maximum error as well as the power normalised mean square error are shown in Fig 3. Based on these results it appears that 7 terms are already sufficient to achieve an error below -100 dB and for more than 10 terms the numerical resolution limit of single precision floating point numbers will be reached.

3. SOUND SYNTHESIS EXAMPLES

3.1. Combination of oscillators

Recursive filters are usually used in subtractive synthesis due to computational constraints but will lead to a frequency-dependent phase shift of each harmonic and a predictable combination of several oscillators without unwanted partial cancellations is difficult. The pulse shaping in BLIT corresponds to a linear-phase FIR low-pass filtering, hence all harmonics are still in phase after the filtering and it is straightforward to add or subtract multiple oscillator outputs in a predictable manner to create more complex spectra.

Let us define a signal $x_i(n)$ that has a fundamental frequency $F_{0i} = i \cdot F_0$ which is an integer multiple of another signal $x(n)$ but both share the same spectral envelope. In this case, the difference between $x(n)$ and $x_i(n)$

$$\begin{aligned} x_D(n) &= x(n) - \frac{x_i(n)}{i} \\ &= d(n) * h_H(n) - \frac{d_i(n)}{i} * h_H(n) \\ &= \left[d(n) - \frac{d_i(n)}{i} \right] * h_H(n) \end{aligned} \quad (14)$$

yields a signal where every i -th harmonic is cancelled. For $i = 2$, which is equivalent to using a bipolar impulse train as source signal [1], a signal $x_O(n)$ with only odd harmonics remains (Fig. 4 b).

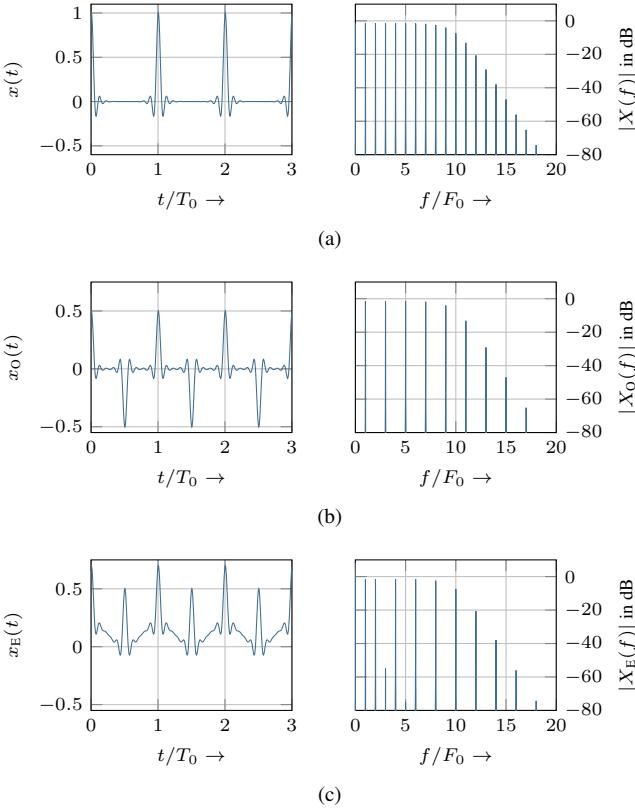


Figure 4: Unipolar pulse train with full number of harmonics (a), bipolar pulse train with only odd harmonics (b) and signal with even harmonics (c) as a sum of two different unipolar pulse trains.

Using Hammerich pulses with different parameters for each oscillator offers further possibilities. A signal with even harmonics

$$x_E(n) = x_1(n) + \frac{d_2(n)}{2} * h_H(n), \quad (15)$$

as depicted in Fig. 4 b), can be constructed from the sum of one pulse train $d_2(n)$ with twice the fundamental frequency and arbitrary filter and another single harmonic signal $x_1(n)$ and fundamental frequency F_0 .

3.2. Standard waveforms

It was shown in [1] that the standard waveform (rectangular, sawtooth and triangular) can be created with a simple integration of impulse train signals. In our case, a lowpass filtered sawtooth

$$x_{\text{saw}}(n) = x(n) * h_I(n) \quad (16)$$

is obtained by convolving a bandlimited impulse train signal $x(n)$ with an integrator impulse response $h_I(n)$. To avoid accumulation of an error constant in the integration, it is usually recommended to use a leaky integrator. A second order leaky integrator with zero DC gain was proposed by [12]

$$H_I(z) = \pi \frac{\gamma + 1}{2} \left(\frac{1 - z^{-1}}{1 - 2\gamma z^{-1} + \gamma^2 z^{-2}} \right) \quad (17)$$

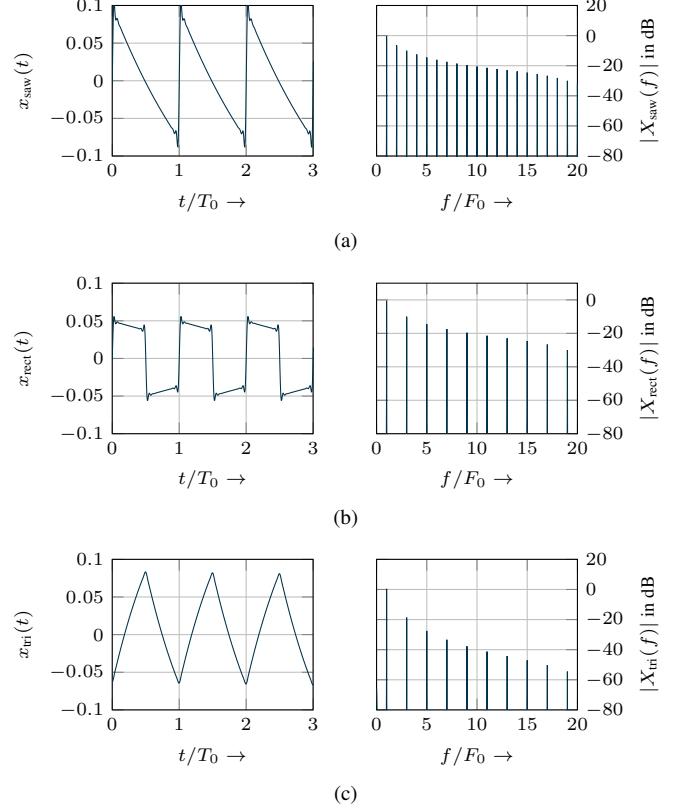


Figure 5: Sawtooth (a), rectangular (b) and triangular waveforms (c) created by a combination of oscillators and leaky integration.

and consists of a cascaded first order leaky integrator and a one-pole highpass. The parameter $\gamma = \exp(2\pi f_{cI}/f_s)$ defines the cut-off frequency of the highpass (typically $f_{cI} < 20$ Hz) and thereby the crossover point between leaky and non-leaky integration. The rectangular waveform

$$x_{\text{rect}}(n) = x_O(n) * h_I(n) \quad (18)$$

is obtained by leaky integration of a signal with only odd harmonics. Finally, the triangular signal

$$x_{\text{tri}}(n) = x_{\text{rect}}(n) * h_I(n) = x_O(n) * h_I(n) * h_I(n) \quad (19)$$

is an integrated rectangular signal, or two-times integrated signal with odd harmonics. Figure 5 depicts exemplary bandlimited sawtooth, rectangular and triangular signals which were obtained by leaky integration.

3.3. Modulation

All pulse parameters can be directly modulated in a sound synthesis application. Figure 6 a) visualises a fundamental frequency sweep ranging from 20 Hz up to 7 kHz with $N_H = 5$ and $\alpha = 0.8$ at a sample rate of 44.1 kHz. Aliasing is kept at a low level by constantly checking and limiting the parameters N_H and α in dependency of the current fundamental frequency. A step-wise modulation of the number of harmonics is shown in Fig. 6 b).

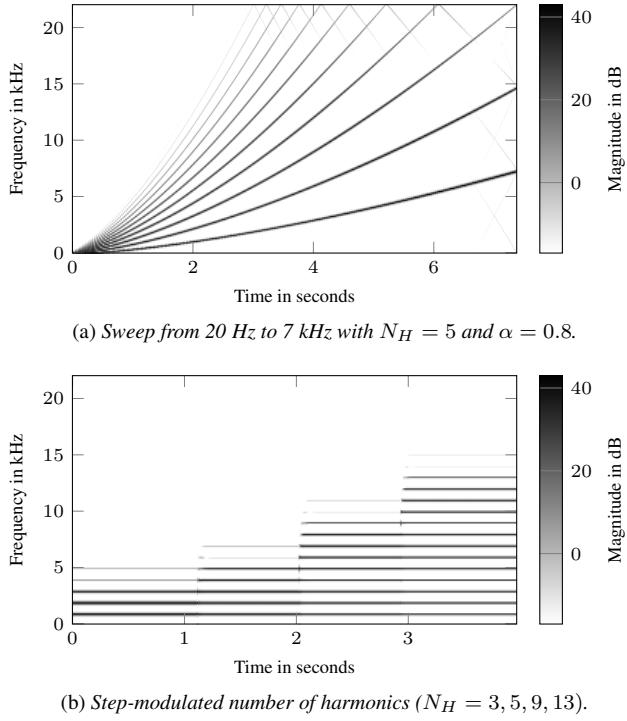


Figure 6: Example spectrograms showing a modulation of the fundamental frequency and number of harmonics.

4. CONCLUSION

Usually, in bandlimited impulse train (BLIT) synthesis, sinc-pulses are used to filter a pulse train and to obtain a spectrum with a defined number of harmonics of equal magnitude. In this paper it was proposed to replace the sinc-pulse with a Hammerich pulse as its spectral shape can be directly controlled by two independent parameters for cutoff frequency and filter roll-off. The closed form equation for the Hammerich pulse can be evaluated per sample, does not require the creation of a wavetable and an immediate modulation of the pulse parameters is possible. As all harmonics in a pulse are in phase, differently configured oscillators can be easily combined to create more complex spectral shapes. It was shown how to synthesise spectra with odd or even harmonics and together with a leaky integrator, various standard waveforms (rectangular, triangular, sawtooth) can be created. The Hammerich pulse considerably expands the BLIT principle to become a full-featured synthesis procedure and despite the restriction to lowpass spectra, a wide variety of useful sounds and waveforms can be created without additional filtering.

The possibilities and limitations of this new waveform generation algorithm still have to be explored in practical musical applications. First tests with a real-time modulation of the pulse parameters were quite promising. In particular the simple interface with only a few but very expressive parameters supports an intuitive and creative workflow. For the future it might be in particular interesting to find further pulse shapes which can be calculated and parametrised in a similar fashion as the Hammerich pulse but exhibit a different frequency response, e.g. highpass, bandpass or resonant lowpass.

5. ACKNOWLEDGMENTS

The authors would like to thank the anonymous reviewers for their extensive and valuable feedback which helped to improve this paper in many aspects.

6. REFERENCES

- [1] Tim Stilson and Julius Smith, “Alias-free digital synthesis of classic analog waveforms,” in *Proc. of the Int. Computer Music Conference (ICMC)*, 1996.
- [2] Vesa Välimäki and Antti Huovilainen, “Antialiasing oscillators in subtractive synthesis,” *IEEE Signal Processing Magazine*, vol. 24, no. 2, pp. 116–125, 2007.
- [3] Vesa Välimäki, Jussi Pekonen, and Juhani Nam, “Perceptually informed synthesis of bandlimited classical waveforms using integrated polynomial interpolation,” *The Journal of the Acoustical Society of America*, vol. 131, no. 1, pp. 974, 2012.
- [4] John Lazzaro and John Wawrzynek, “Subtractive Synthesis without Filters,” in *Audio Anecdotes II - Tools, Tips, and Techniques for Digital Audio*, pp. 55–64. 2004.
- [5] James A. Moorer, “The Synthesis of Complex Audio Spectra by Means of Discrete Summation Formulas,” *Journal of the Audio Engineering Society*, vol. 24, no. 9, pp. 717–727, 1976.
- [6] Jussi Pekonen, Juhani Nam, Julius O. Smith, Jonathan S. Abel, and Vesa Välimäki, “On Minimizing the Look-Up Table Size in Quasi-Bandlimited Classical Waveform Oscillators,” in *Proc. of the 13th Int. Conference on Digital Audio Effects*, 2010.
- [7] Juhani Nam, Jonathan S. Abel, and Julius O. Smith, “Efficient Antialiasing Oscillator Algorithms Using Low-Order Fractional Delay Filters,” *IEEE Transactions on Audio, Speech and Language Processing*, vol. 18, no. 4, pp. 773–785, 2010.
- [8] Stéphan Tassart, “Band-limited impulse train generation using sampled infinite impulse responses of analog filters,” *IEEE Transactions on Audio, Speech and Language Processing*, vol. 21, no. 3, pp. 488–497, 2013.
- [9] Edwin Hammerich, “A Generalized Sampling Theorem for Frequency Localized Signals,” *Sampling Theory in Signal and Image Processing*, vol. 8, no. 2, pp. 127–146, 2007.
- [10] Edwin Hammerich, “Design of Pulse Shapes and Digital Filters Based on Gaussian Functions,” 2009.
- [11] Udo Zölzer, “Pitch-based digital audio effects,” in *Proc. of the 5th Int. Symposium on Communications, Control and Signal Processing (ISCCSP)*, 2012.
- [12] Eli Brandt, “Hard Sync without Aliasing,” in *Proc. of the Int. Computer Music Conference (ICMC)*, 2001.

REDRESSING WARPED WAVELETS AND OTHER SIMILAR WARPED TIME-SOMETHING REPRESENTATIONS

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 and frequency warping provide effective methods for fitting signal representations to desired physical or psychoacoustic characteristics. However, warping in one of the variables, e.g. frequency, disrupts the organization of the representation with respect to the conjugate variable, e.g. time. In recent papers we have considered methods to eliminate or mitigate the dispersion introduced by warping in time frequency representations and Gabor frames. To this purpose, we introduced redressing methods consisting in further warping with respect to the transformed variables. These methods proved not only useful for the visualization of the transform but also to simplify the computation of the transform in terms of shifted precomputed warped elements, without the need for warping in the computation of the transform. In other linear representations, such as time-scale, warping generally modifies the transform operators, making visualization less informative and computation more difficult. Sound signal representations almost invariably need time as one of the coordinates in view of the fact that we normally wish to follow the time evolution of features and characteristics. In this paper we devise methods for the redressing of dispersion introduced by warping in wavelet transforms and in other expansions where time-shift plays a role.

1. INTRODUCTION

Linear representations play a central role in sound synthesis, digital audio effects, feature extraction, coding and music information retrieval. In most of these representations, one of the coordinates in the transformed domain is time-shift, as in the case of time-frequency representations [1], e.g., based on the STFT (Short-Time Fourier Transform, also known as phase VOCODER) or as in the case of time-scale representations based on the WT (Wavelet Transform) [2, 3]. Other examples, based on different signal transformations such as the MDCT (Modified Discrete Cosine Transform) [4, 5], windowed Hankel (Fourier-Bessel) [6, 7], to name a few, are available.

In this paper we are mostly concerned with audio signal representations that are exact, i.e., for which an inverse or pseudo-inverse exists such that perfect reconstruction of the original signal is possible from the analysis data. Other analysis-only methods, such as the constant Q transformation [8] may enjoy easier computation but may introduce data loss or have inefficient reconstruction algorithms so they are not directly suitable for the synthesis from analysis.

Generally speaking, mathematical transforms operate on a one-dimensional audio signal and produce a multidimensional representation. For example the STFT yields a 2D signal representation in which one coordinate can be interpreted as time (more

precisely time-shift) and another coordinate as frequency, so that the variation of the frequency spectrum in time can be visualized and used for feature detection such as instantaneous frequency or amplitude envelopes of the partials.

Linear signal transformations are computed by taking the scalar product of the signal with a transform nucleus function [9] $K_{\alpha_1, \dots, \alpha_N}(t)$, where $\alpha_1, \dots, \alpha_N$ are the N transform domain coordinates [2]. For example, in the STFT, the nucleus is a modulated window where α_1 is associated to time-shift and α_2 to frequency.

When time-shift is one of the transform coordinates, in a large class of linear 2D representations the transform nucleus is computed by operating on a test function g , e.g., the prototype analysis window or the analysis mother wavelet. In order to formalize the generation of the transform nucleus of this type, one can introduce a time-shift operator \mathbf{T}_τ , whose action on a time signal s is defined as follows:

$$[\mathbf{T}_\tau s](t) = s(t - \tau), \quad (1)$$

and another parametric linear operator \mathbf{O}_σ . The corresponding transform nucleus can be then written as

$$K_{\tau, \sigma}(t) = [\mathbf{T}_\tau \mathbf{O}_\sigma g](t) \quad (2)$$

and the linear signal transformation computed as follows:

$$S(\tau, \sigma) = \langle \mathbf{T}_\tau \mathbf{O}_\sigma g, s \rangle, \quad (3)$$

where $\langle f, g \rangle$ denotes the scalar product in the signal space¹. We call “time-something” the generic representation whose representative elements are obtained by cascading time-shift with another operator \mathbf{O} acting on a prototype function g .

In the STFT based representations the operator \mathbf{O} corresponds to modulation

$$[\mathbf{M}_\nu s](t) = e^{j2\pi\nu t} s(t), \quad (4)$$

while in the WT based representations the operator \mathbf{O} corresponds to dilation

$$[\mathbf{D}_a s](t) = \frac{1}{\sqrt{a}} s\left(\frac{t}{a}\right), \quad (5)$$

which we wrote here as a unitary operator.

Notice that in some definitions the time-shift and the \mathbf{O} operators appear in reverse order with respect to that in (3). However, with the introduction of a phase change or of a modification of the transform coordinates, in most useful cases one can redefine the transformation as in (3). For example, in the STFT, we have:

$$[\mathbf{T}_\tau \mathbf{M}_\nu g](t) = e^{j2\pi\nu(t-\tau)} g(t - \tau) = e^{-j2\pi\nu\tau} [\mathbf{M}_\nu \mathbf{T}_\tau g](t). \quad (6)$$

¹Typically this is the space $\mathbb{L}^2(\mathbb{R})$ of square integrable functions (finite energy signals) on the real line, however, the choice might depend on the signal representation we are considering.

The phase factor $e^{-j2\pi\nu\tau}$ is irrelevant, it cancels out when analysis and synthesis is performed and does not need to be included in the computation. For the wavelet transform, one can show that:

$$\mathbf{D}_a \mathbf{T}_{\tau/\alpha} = \mathbf{T}_\tau \mathbf{D}_a, \quad (7)$$

with obvious reinterpretation of scaled time-shift τ/α .

Most of the mentioned representations can be studied in terms of the time and frequency spreading of the transform nucleus

$$K_{\tau,\sigma}(t) = [\mathbf{T}_\tau \mathbf{O}_\sigma g](t), \quad (8)$$

as the parameters τ and σ , which are the coordinates of the transformed domain, vary. Directly or indirectly (via the relationship between the coordinate σ and the frequency), this leads to a tiling of the time-frequency plane, in which the specific form of the time-frequency tiles carries a loose meaning in terms of the signal components that get detected or trapped in a certain time-frequency zone.

However, the tiling is often dictated by mathematical simplifications for the definition of the operations used in generating the transform nucleus. Thus, for example, in the STFT based time-frequency representations time and frequency resolutions are both uniform and in the time-scale WT based representations time-shift is uniform in each scale. Also, in the WT, scale and time are interrelated so that the uncertainty product of the representative elements is constant throughout the tiling. In the applications often one would like to achieve higher degree of flexibility in allocating the time-frequency spread of the representative elements in order to produce a tiling prescribed, e.g., by psychoacoustic or physical characteristics.

In previous work, frequency warping has been considered in conjunction with signal representations in order to increase the flexibility and adaptivity of the analysis-synthesis scheme [10, 11, 12, 13, 14]. Since frequency warping introduces dispersion in time-localization, methods for reducing or eliminating this effect are desired. In recent work [15, 16, 17], we proposed techniques, the so called redressing methods, that are directly suitable for the STFT based time-frequency representation once the warping map is defined. These basically consist in the application of suitable operators aimed at linearizing the phase of the dispersive delays resulting from the application of warping to the time-shift operator.

Since, over the continuum, a 2D representation of a 1D signal is generally redundant, sampling can be introduced by selecting a grid of points in the transform domain. Provided that certain conditions on the grid and on the window or wavelet are satisfied, the original signal can be reconstructed from the transform samples evaluated at the grid points only. This leads to a signal expansion in terms of a set of functions constituting a frame or otherwise to a bi-orthogonal / orthogonal and complete set for the signal space.

Sampling introduces some complications in the application of the redressing methods and may impose limitations on the choice of the window in generalized Gabor frames considered in [15, 16, 17]. There, discrete-time frequency warping operators were applied in the sampled time STFT domain in order to partially eliminate dispersion. The results show that for bandlimited windows, the dispersion elimination procedure can be made exact and leads to the same results as ad-hoc methods for the construction of non-stationary Gabor frames [18, 19, 20].

Transform domain discrete-time frequency warping also leads to approximated algorithms for the computation of the generalized

warped Gabor frames [21, 22, 23]. On the other hand, computation of discrete-time frequency warping as a digital audio effect can be eased by the use of the STFT [24].

In this paper we extend our methods to other generic time-something representations of which wavelet expansions are an example.

The paper is organized as follows: in Section 2 we discuss the concept of unitary equivalence, applying it to unitary warping in Section 3. In Section 4 we discuss redressing methods for the generic unsampled and sampled time-something representations, with examples to the wavelet expansion. 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/>.

2. UNITARY EQUIVALENCE

In order to generalize the linear signal representations and be able to adapt the representing elements to important physical or perceptual characteristics, one can resort to further transformation using invertible operators. This is shown in Figure 1 where the analysis and synthesis blocks allow for perfect reconstruction. However, in the scheme in the figure, the synthesis of the signal s is achieved by further operating with an invertible operator \mathbf{U} . In other words, if in the original analysis-synthesis scheme the signal was synthesized in terms of a continuous or discrete set of functions $\psi_\alpha(t)$, in the new scheme the signal is synthesized in terms of the functions $[\mathbf{U}\psi_\alpha](t)$, where α is either the variable or the index of superposition for the synthesis, e.g.,

$$s(t) = \sum_{\alpha} C_{\alpha} [\mathbf{U}\psi_{\alpha}](t), \quad (9)$$

in which C_{α} denote the coefficients of the expansion. Here, in order to simplify the notation, in the variable α we incorporate all of the transform coordinates $\alpha_1, \dots, \alpha_N$, such as time-shift and frequency or time-shift and scale.

Clearly, in order to provide the correct analysis algorithm, in the new scheme one needs to operate on the signal with the inverse operator \mathbf{U}^{-1} and obtain the analysis coefficients C_{α} in (9) from the signal $[\mathbf{U}^{-1}s](t)$. However, in general, perfect reconstruction is not guaranteed and one needs to prove properties like the norm convergence of the expansion on the right hand side of (9) to the signal on the left hand side.

Perfect reconstruction is guaranteed, with no further proof, if the operator \mathbf{U} is unitary, i.e., if \mathbf{U} is a bounded linear operator satisfying

$$\mathbf{U}\mathbf{U}^{\dagger} = \mathbf{U}^{\dagger}\mathbf{U} = \mathbf{I}, \quad (10)$$

where \mathbf{I} is the identity operator and \mathbf{U}^{\dagger} represents the adjoint of \mathbf{U} , i.e., the operator satisfying

$$\langle \mathbf{U}f, g \rangle = \langle f, \mathbf{U}^{\dagger}g \rangle \quad (11)$$

for any pair of functions f and g belonging to the signal space. In this case, the operator \mathbf{U}^{\dagger} is both the left and right inverse of \mathbf{U} and \mathbf{U} realizes a surjective isometry in the function space. Therefore, properties such as orthogonality, norm and norm convergence are preserved, so one can simply invoke unitary equivalence to verify perfect reconstruction [11].

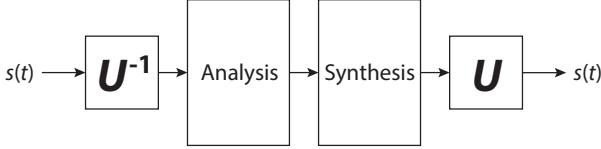


Figure 1: Inserting invertible operators in the analysis-synthesis path of a signal representation

The simple insertion of the invertible operator \mathbf{U} and its inverse in the analysis-synthesis path could, however, lead to dramatic consequences in the interpretation of the transform coordinates. In fact, suppose that the nucleus of the original transform is given as in (8), then the new nucleus is:

$$K_{\tau,\sigma}(t) = [\mathbf{U}\mathbf{T}_\tau \mathbf{O}_\sigma g](t), \quad (12)$$

By inserting neutral pairs $\mathbf{U}^{-1}\mathbf{U}$ in (12), one obtains [11]:

$$K_{\tau,\sigma}(t) = [\mathbf{U}\mathbf{T}_\tau \mathbf{U}^{-1} \mathbf{U}\mathbf{O}_\sigma \mathbf{U}^{-1} \mathbf{U}g](t), \quad (13)$$

which shows that the new “time” and the new “O” operators are obtained by similarity transformations, i.e., $\mathbf{U}\mathbf{T}_\tau \mathbf{U}^{-1}$ and $\mathbf{U}\mathbf{O}_\sigma \mathbf{U}^{-1}$, respectively, which obviously changes the meaning of the corresponding variables.

3. ALTERNATIVE UNITARY EQUIVALENT REPRESENTATIONS OBTAINED BY WARPING

Appealing signal transformations that can be cascaded to a given analysis-synthesis scheme consist of time or frequency warping, which provide new signal representations that are unitary equivalent to well-known ones, as discussed in the previous section [10, 11, 12, 13, 14].

In this section we illustrate the fundamental notions of warping in connection with time-something representations. As these apply with same formalism to both continuous and discrete (sampled) representations, we will review these concepts without specifying the nature of the representation.

In order to make the concepts more concrete, in Section 3.2 we provide an example of warping map for changing the scale factor in complete wavelet sets obtained by sampling in time-scale.

In Section 4 we provide a solution for the redressing of the warped time-shift operator, in order to eliminate or reduce dispersion resulting from frequency warping. Redressing proves easy in the case of continuous (with respect to time-shift) representations. It requires a bit more ingenuity when sampling is introduced in the transformed space.

By duality arguments, the principles illustrated in this paper also apply to the time dispersion of frequency localization when time warping is employed.

3.1. Warping Operators

Warping operators perform function composition in order to remap the abscissa. Thus, a time warping operator \mathbf{W}_γ remaps the time axis by means of the following action:

$$s_{tw} = \mathbf{W}_\gamma s = \mathbf{C}_\gamma s = s \circ \gamma, \quad (14)$$

where s_{tw} is the time-warped version of the signal s , γ is the time warping map, \mathbf{C}_γ is the composition by γ operator and \circ denotes function composition.

One can show that the composition operator \mathbf{C}_γ is bounded on $\mathbb{L}_2(\mathbb{R})$ if the map is a monotonic function and $1/\gamma'$ is essentially bounded, where γ' is the first derivative of the map, i.e., γ' must be essentially bounded from below. This, together with more general and necessary and sufficient conditions for the continuity of the composition operator, which require the absolute continuity of the measure introduced by the map and the essential boundedness of the Radon-Nikodym derivative, can be found in [25, 26].

Similarly, a frequency warping operator $\mathbf{W}_{\tilde{\theta}}$ is completely characterized by a function composition operator \mathbf{W}_θ in the frequency domain:

$$\hat{s}_{fw} = \widehat{\mathbf{W}}_{\tilde{\theta}} s = \widehat{\mathbf{W}}_{\tilde{\theta}} \hat{s} = \mathbf{C}_\theta \hat{s} = \hat{s} \circ \theta, \quad (15)$$

where θ is the frequency warping map, which transforms the Fourier transform $\hat{s} = \mathcal{F}s$ of a signal s into the Fourier transform $\hat{s}_{fw} = \mathcal{F}s_{fw}$ of another signal s_{fw} , the frequency warped version of the signal s , where \mathcal{F} is the Fourier transform operator and the hat over a symbol denotes the Fourier transformed quantity (signal or operator). We affix the \sim symbol over the map θ as a reminder that the map operates in the frequency domain. Accordingly, we have $\mathbf{W}_{\tilde{\theta}} = \mathcal{F}^{-1} \widehat{\mathbf{W}}_{\tilde{\theta}} \mathcal{F} = \mathcal{F}^{-1} \mathbf{C}_\theta \mathcal{F}$.

For the case of frequency warping, in order to transform real signals to real signals, one has to constrain the map θ to have the odd parity

$$\theta(-\nu) = -\theta(\nu). \quad (16)$$

This way, positive (negative) frequencies are mapped to positive (negative) frequencies. In general, this is not a big constraint as the frequency band allocation of the representation is usually symmetric (same bandwidth in positive and negative frequencies).

In this paper, we consider warping operators that are invertible. One can show [27] that the composition operator (14) is invertible if and only if the map γ is invertible and the composition operator $\mathbf{C}_{\gamma^{-1}}$ based on the inverse map γ^{-1} is bounded on $\mathbb{L}_2(\mathbb{R})$. If the map γ is a monotonic function then also its inverse γ^{-1} is monotonic. Then, a sufficient condition for the invertibility of the composition operator is that also the first derivative of γ^{-1} is essentially bounded from below. Theorem 3 in [27] also shows that in the monotonic case the operator \mathbf{C}_γ is invertible if and only if γ' is essentially bounded and is absolutely continuous on the finite intervals in \mathbb{R} .

If the warping map is almost everywhere strictly increasing, one-to-one and differentiable then a unitary form of the warping operator can be defined by amplitude scaling, as given by the square root of the magnitude derivative of the map (dilation function). For example, a unitary frequency warping operator $\mathbf{U}_{\tilde{\theta}}$ has frequency domain action

$$\hat{s}_{fw}(\nu) = \left[\widehat{\mathbf{U}}_{\tilde{\theta}} s \right] (\nu) = \sqrt{\left| \frac{d\theta}{d\nu} \right|} \hat{s}(\theta(\nu)), \quad (17)$$

where ν denotes frequency. We assume henceforth that all warping maps are almost everywhere increasing so that the magnitude sign can be dropped from the derivative under the square root.

The unitary warping operator (17) is a special case of weighted composition operator [28]. It turns out that the weight function – the derivative of the map in our case – helps releasing some constraints on the maps that guarantee the continuity of the weighted composition operator. A classical example [28] is the map $\gamma(x) = x^2$ in the space $\mathbb{L}^2([0, 1])$. The map does not define a bounded composition operator (γ' is not bounded from below).

However, with the weight $\sqrt{\gamma'(x)} = \sqrt{2x}$ one obtains a bounded (unitary) weighted composition operator. More general conditions for the definition of unitary weighted composition operators can be found in [29].

Incidentally, note that the dilation operator introduced in (5) is a particular case of unitary time warping operator \mathbf{U}_γ where the warping map $\gamma(t) = t/a$ is linear. By the Fourier scaling theorem, this is also equivalent to a frequency warping unitary operator $\mathbf{U}_{\tilde{\theta}}$ with map $\theta(\nu) = a\nu$.

When a unitary frequency warping operator $\mathbf{U}_{\tilde{\theta}}$ acts as modifier of a time-shift based signal representation, the time-shift operator is affected by a similarity transformation as in (13). By a calculation similar to the one conducted in the Appendix one can conclude that

$$[\mathbf{U}_{\tilde{\theta}} \mathbf{T}_\tau \mathbf{U}_{\tilde{\theta}}^{-1} s](t) = \int_{-\infty}^{+\infty} d\nu e^{j2\pi\nu t} e^{-j2\pi\theta(\nu)\tau} \hat{s}(\nu), \quad (18)$$

which shows how, by the effect of warping, the delay τ converts into a dispersive delay represented, in the frequency domain, by the factor $e^{-j2\pi\theta(\nu)\tau}$.

3.2. Example: Warping Map for Change of Scale in Sets of Wavelet

Sets of wavelets for signal expansions are obtained by choosing a grid of points for sampling the WT in time-scale space of the type $(a^n m, a^n)$, where m and n are integers and a is a constant scale factor. It is well known that the dyadic scheme, which fixes $a = 2$ and octave band resolution, is the easiest way to generate orthogonal and complete sets of wavelets. For sound and music processing scopes, it is highly interesting to improve the frequency resolution, for example to half-tones. One can achieve this by means of a warping map θ that allows for a change of the scale factor from a to $b < a$. In other words, we are requiring the map to satisfy:

$$a\theta(\nu) = \theta(b\nu), \quad (19)$$

so that, by repeated applications of (19) one can show that

$$\sqrt{\frac{d\theta}{d\nu}} \sqrt{a^n} \hat{\psi}(a^n \theta(\nu)) = \sqrt{b^n} \sqrt{\frac{d\theta}{d\nu}} \hat{\psi}(\theta(b^n \nu)) \quad (20)$$

where $\hat{\psi}$ is the Fourier transform of a scale- a mother wavelet. This shows that the frequency warped wavelet $\hat{\psi}$, whose Fourier transform is

$$\hat{\psi}(\nu) = \sqrt{\frac{d\theta}{d\nu}} \hat{\psi}(\theta(\nu)) \quad (21)$$

properly scales through powers of b to the warped versions of the a -scaled versions of the original mother wavelet ψ :

$$\sqrt{b^n} \hat{\psi}(b^n \nu) = \sqrt{\frac{d\theta}{d\nu}} \sqrt{a^n} \hat{\psi}(a^n \theta(\nu)). \quad (22)$$

A particular map that satisfies the generalized homogeneity condition (19) is given by the exp-log function:

$$\theta(\nu) = C \operatorname{sgn}(\nu) a^{\log_b |\nu|}, \quad (23)$$

where $\operatorname{sgn}(\nu)$ is the signum function and C is an arbitrary constant that can be set, for example, by constraining the warping map to fix a specific frequency. This capability is important in sampled

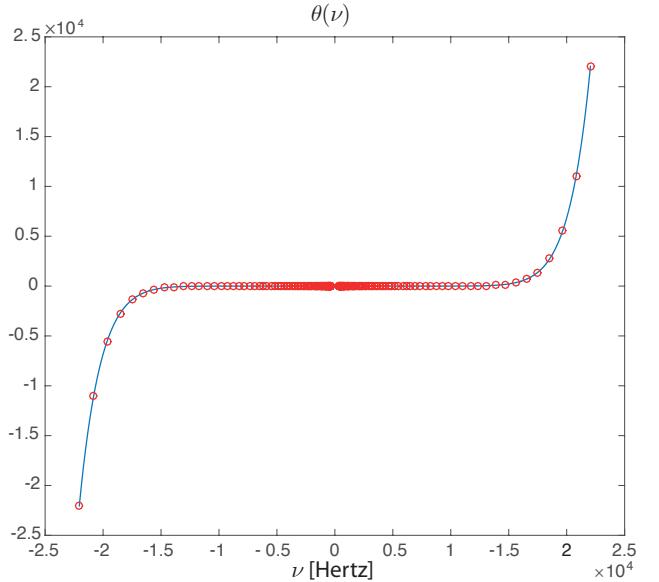


Figure 2: Frequency warping map to convert octave-band to halftone-band wavelets. The circles in the plot denote the halftone mapping points.

systems in order to preserve the sampling rate, in which case we set:

$$C = \nu_N a^{-\log_b \nu_N}, \quad (24)$$

where ν_N is the Nyquist frequency, so that $\theta(\nu_N) = \nu_N$ and the overall bandwidth is preserved.

Notice that for the map in (23) we have:

$$\frac{d\theta}{d\nu} = C \log_b(a) \frac{a^{\log_b |\nu|}}{|\nu|}, \quad (25)$$

so that, for the case of interest where $a > b > 1$, which corresponds to higher frequency resolution warped wavelets, we have $\lim_{\nu \rightarrow 0} \theta'(\nu) = 0$, so that θ' is not essentially bounded from below and therefore the corresponding composition operator may not be bounded. However, in representations for sound and music signals, we are not interested in extreme accuracy around zero frequency. In order to obtain a bounded composition operator, one can therefore replace the warping map (23) with one that is identical to it except that, on a small interval around zero frequency, the original map is replaced by a linear segment continuously going from 0 to the value of the original map at the extremes of the interval. Warping with this map will still yield an exact representation, with the desired frequency band allocation.

A map suitable for the conversion of octave-band wavelets into halftone-band wavelets is shown in Figure 2. On a log-log scale the map would appear as linear. The Fourier transforms of the corresponding frequency warped wavelets based on an octave-band pruned tree-structured FIR Daubechies filter bank [3] of order 31 are shown in Figure 3.

The dispersion pattern of 1/3-octave resolution frequency warped wavelets is shown in Figure 4, where each area delimited by the band edge limits (horizontal lines) and two adjacent group delay curves roughly represents the time-frequency resolution or, more precisely, the uncertainty zone of the corresponding

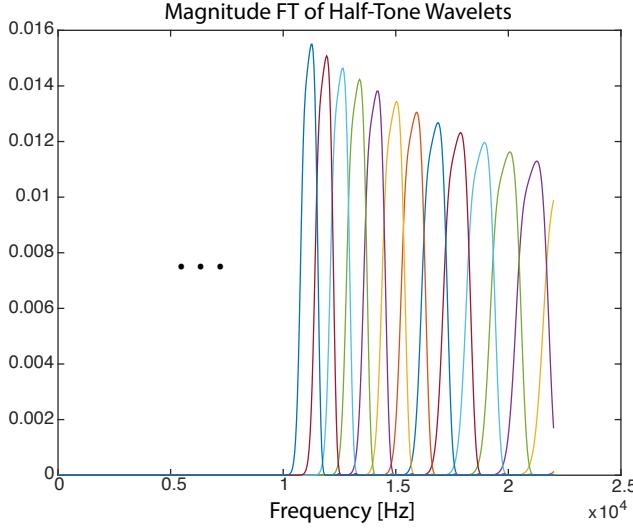


Figure 3: Magnitude Fourier transform of frequency warped wavelets with halftone frequency resolution.

wavelet. The fact that the time boundaries are curved is not good news for the time localization of signal components or features. Fortunately, redressing methods, which we will illustrate in the remainder of the paper, are available for the elimination or reduction of dispersion.

4. REDRESSING

As shown in Section 3.1, frequency warping results in a similarity transformation of the time-shift operator, which introduces frequency dependency, i.e., the dispersive time-shift in (18). Once established, the result proven in the Appendix that

$$\mathbf{U}_{\tilde{\theta}}^{(\tau)} \mathbf{U}_{\tilde{\theta}} \mathbf{T}_\tau = \mathbf{T}_\tau \mathbf{W}_{\tilde{\theta}}, \quad (26)$$

provides a simple way to counteract dispersion.

Here the operator $\mathbf{U}_{\tilde{\theta}}^{(\tau)}$ is a genuine warping operator as discussed in Section 3.1, however, it operates in the transform domain, i.e., with respect to the time-shift coordinate instead of the signal's time. Basically, by performing a supplementary frequency warping operation, denoted by $\mathbf{U}_{\tilde{\theta}}^{(\tau)}$, with respect to the time-shift coordinate in the transform domain, one obtains a commutation rule of the time-shift operator with the frequency warping operator. The latter occurs in both its unitary $\mathbf{U}_{\tilde{\theta}}$ and non-unitary $\mathbf{W}_{\tilde{\theta}}$ versions in the commutation rule (26).

Equation (26) shows that the pure time-shifting of the warped nucleus of the transform leads to a perfect reconstruction scheme, which is equivalent to operating in both signal and transform domain with unitary operators. The insertion of signal frequency warping operators $\mathbf{U}_{\tilde{\theta}}$ and $\mathbf{U}_{\tilde{\theta}}^\dagger$ for frequency remapping and frequency warping operators $\mathbf{U}_{\tilde{\theta}}^{(\tau)}$ and $\mathbf{U}_{\tilde{\theta}}^{(\tau)\dagger}$ in the transform domain for the computation of a time-something representation is shown in Figure 5.

We remark that the scheme illustrated in the picture is a conceptual one, as redressing may provide a great simplification, which makes the online computation of the warping operators unnecessary. In fact, once we apply the redressing operator $\mathbf{U}_{\tilde{\theta}}^{(\tau)}$,

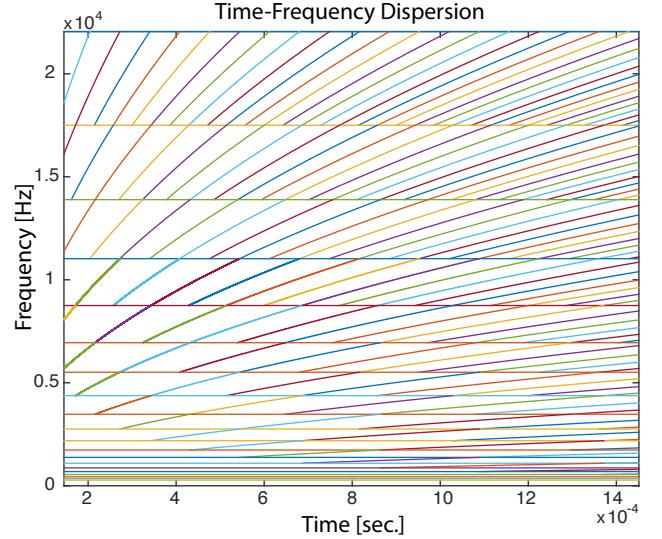


Figure 4: Uncertainty zones delimited by the group delay dispersion profiles of 1/3-octave warped wavelets without redressing.

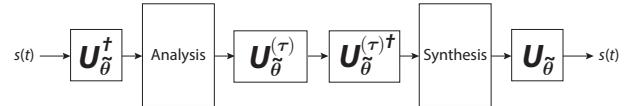


Figure 5: Inserting frequency warping operators for both frequency remapping and redressing in the analysis-synthesis path of a signal representation

the frequency warped nucleus of the transform may become shift-invariant or approximately so. Thus, in the computation of the transform, one can compute the scalar products of the signal with time-shifted versions of the frequency warped modulated prototype window or dilated mother wavelet.

4.1. Redressing Sampled Time-Something Representations

In most time-something representations the time-shift, as well as other variables, can be sampled, while preserving the capability of perfectly reconstructing the signal. Sampling in time-frequency the STFT leads to the concept of Gabor frames [30], while sampling the WT leads to orthogonal / biorthogonal wavelets or to wavelet frames [2, 3]. However, since frequency warping does not commute with the time sampling operator, the redressing procedure does not carry over to the sampled transforms in an exact way.

Once time sampling is performed in the transform domain, the time-shift operator \mathbf{T}_τ becomes \mathbf{T}_{nT} . Thus, we cannot apply the operator $\mathbf{U}_{\tilde{\theta}}^{(\tau)}$ with respect to the variable τ in order to eliminate the dispersive delays as we did in (26). Instead, we can try to apply a discrete version of frequency warping operator acting on sampled time, i.e. on the index n .

A discrete-time frequency warping operator can be built from an almost everywhere differentiable warping map ϑ that is one-to-one and onto $[-\frac{1}{2}, +\frac{1}{2}]$. In this case, one can form an orthonormal

basis of $\ell^2(\mathbb{Z})$ as follows:

$$\mu_m(n) = \int_{-\frac{1}{2}}^{+\frac{1}{2}} \sqrt{\frac{d\vartheta}{d\nu}} e^{j2\pi(n\vartheta(\nu)-m\nu)} d\nu, \quad (27)$$

where $n, m \in \mathbb{Z}$ (see [31, 32, 14, 12, 13]).

The map ϑ can be extended over the entire real axis as congruent modulo 1 to a 1-periodic function. One can always write the map in terms of a 1-periodic function plus an integer staircase function, so that

$$\vartheta(\nu + k) = \vartheta(\nu) + k \quad (28)$$

for any $k \in \mathbb{Z}$.

Similarly to the continuous counterpart, one can define the action of the discrete-time unitary warping operator $\mathbf{V}_{\tilde{\vartheta}}$ on any sequence $\{x(n)\}$ in $\ell^2(\mathbb{Z})$ as follows:

$$[\mathbf{V}_{\tilde{\vartheta}}x](m) = \langle \mu_m, x \rangle_{\ell^2(\mathbb{Z})}. \quad (29)$$

Given $\tilde{x}(m) = [\mathbf{V}_{\tilde{\vartheta}}x](m)$, in the frequency domain one obtains

$$\hat{\tilde{x}}(\nu) = \sqrt{\frac{d\vartheta}{d\nu}} \hat{x}(\vartheta(\nu)), \quad (30)$$

where here the $\hat{\cdot}$ symbol denotes discrete-time Fourier transform. The sequences $\overline{\eta_m(n)}$, where the overbar symbol \overline{x} denotes complex conjugation, define the nucleus of the inverse unitary frequency warping $\ell^2(\mathbb{Z})$ operator $\mathbf{V}_{\tilde{\vartheta}^{-1}} = \mathbf{V}_{\tilde{\vartheta}}^\dagger$. where $\eta_m(n) = \mu_n(m)$.

Discrete-time frequency warping operators were applied in the sampled time STFT domain in order to partially eliminate dispersion in [15, 16, 17]. We showed that complete elimination of dispersion, i.e., full linearization of the non-linear phase resulting from the application of frequency warping to the time-shift operator is possible in the case of bandlimited analysis window. In the more general case of non-bandlimited window we showed that one can resort to high-performance approximations [23, 21].

In order to extend the redressing methods to other time-something representations, we assume that also the other transform domain coordinate(s) are being sampled. For example, in the case of dyadic wavelets, the basis elements are obtained by time-shifting a scaled version of the mother wavelet $\psi_r(t) = \sqrt{2^{-r}} \psi\left(\frac{t}{2^r}\right)$, in which the scale variable has been exponentially sampled to powers of 2.

In order to simplify our notation, we assume that the redressing method is to be applied to basis or frame elements denoted by ψ_r , which are indexed just by one index r like in the wavelet example. Moreover, we assume for sake of generality that the time-shift factor T_r depends on the index r . Thus, redressing can be achieved by means of a collection of discrete-time frequency warping operators $\mathbf{V}_{\tilde{\vartheta}_r}^{(m)}$, each acting on the frequency warped time-shifted representation element of index r :

$$\tilde{\psi}_{r,m} = \mathbf{V}_{\tilde{\vartheta}_r}^{(m)} \mathbf{U}_{\tilde{\vartheta}} \mathbf{T}_{mT_r} \psi_r, \quad (31)$$

where the superscript (m) in the discrete-time warping operator is a reminder that it operates with respect to the time-shift index m .

Exploiting the odd parity (16) of the $\theta(\nu)$ map and (28), one can show that (31) becomes:

$$\tilde{\psi}_{r,m}(t) = \int_{-\infty}^{+\infty} d\nu A_r(\nu) e^{j2\pi(\theta^{-1}(\varphi_r(\nu))t-m\nu)} \hat{\psi}_r(\varphi_r(\nu)), \quad (32)$$

where

$$\begin{aligned} \varphi_r(\nu) &= \frac{\vartheta_r(\nu)}{T_r} \\ A_r(\nu) &= \sqrt{\frac{d\theta^{-1}(\varphi_r(\nu))}{T_r d\nu}}. \end{aligned} \quad (33)$$

From the phase in (32) one can conclude that if, for some constant $\tilde{T}_r > 0$, we had

$$\theta^{-1}(\varphi_r(\nu)) = \frac{\nu}{\tilde{T}_r} \quad (34)$$

then, since by the invertibility of the map θ the last equation is equivalent to

$$\varphi_r(\nu \tilde{T}_r) = \theta(\nu), \quad (35)$$

we would have

$$\tilde{\psi}_{r,m}(t) = \sqrt{\frac{\tilde{T}_r}{T_r}} \int_{-\infty}^{+\infty} d\nu e^{j2\pi\nu(t-m\tilde{T}_r)} \hat{\psi}_r(\theta(\nu)). \quad (36)$$

This shows that, for each r , the redressed warped representation elements could be obtained from the non-unitarily warped original elements just by altered time-shift $\mathbf{T}_{m\tilde{T}_r}$. This would be the discrete counterpart of the unsampled result described in the previous section. However, as we previously remarked, since the maps ϑ_r are constrained to be congruent modulo 1 to a 1-periodic function with odd parity, their shape can only be arbitrarily assigned on a band of width 1/2.

Collecting (33) and (35) together we have the following condition on the discrete-time warping map ϑ :

$$\vartheta(\nu \tilde{T}_r) = T_r \theta(\nu). \quad (37)$$

Generally, condition (37) cannot be satisfied everywhere but only in a small bandwidth of size $1/2\tilde{T}_r$. However, \tilde{T}_r is a design parameter, which can be selected at will, with the trade-off that it also affects the final sampling rate of the transform. If the warped wavelet, whose Fourier transform is $\hat{\psi}_r(\theta(\nu))$, is strictly bandlimited to a sufficiently small band, then it is only necessary to satisfy (37) in its bandwidth. In the general case, one can satisfy (37) within the essential bandwidth of the warped wavelet. We remark that if the complete chain of warping operators defining the warped transform and the redressing method is computed, the partially redressed warped transform has perfect reconstruction.

As a design example, in the dyadic wavelet case choose $T_r = 2^r T$, where $1/T$ is a reference frequency given by the upper cut-off frequency of the mother wavelet. In designing warped wavelet transforms one generally desires to improve the octave band frequency resolution of the dyadic wavelets. Changing to smaller bandwidths for the warped wavelets, and suitably choosing \tilde{T}_r , results in a rescaling of the frequency interval of the ϑ_r map for the linearization of the phase within the essential bandwidths of the wavelets, i.e., where the wavelets have essentially non-zero magnitude Fourier transform. This is shown in Figure 6 for a warping map carrying dyadic octave-band wavelets to 1/3 octave resolution. In this example, with reference to the map in Section 3.2, we chose $a = 2$ and $b = 2^{1/3}$, which provides warped wavelets with bandwidths $(b-1)/2Tb^r$, with cutoff frequencies $1/2Tb^r$, $r = 1, 2, \dots$. Thus, one can select $\tilde{T}_r = Tb^r/(b-1)$ in order to define the branch of the map ϑ_r according to (37), shown, for the case $r = 2$, in the highlighted (red) portion of the map in Figure 6. For any r , this choice defines ϑ_r on a width 1/2 frequency band

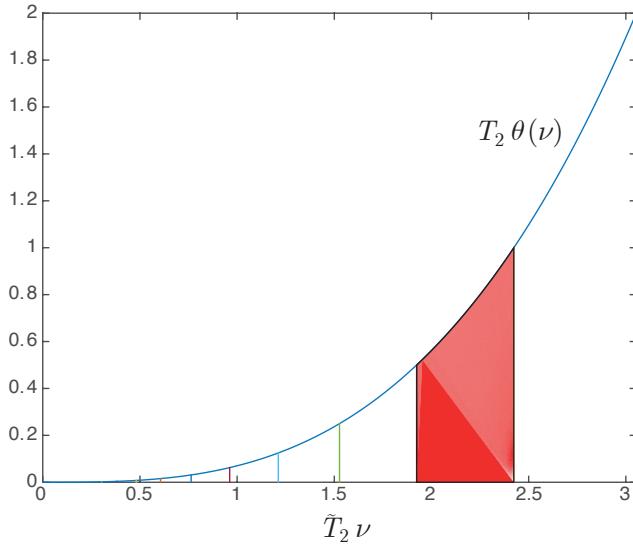


Figure 6: Segment of the map ϑ_2 for the redressing of 1/3-octave warped wavelets obtained from dyadic ones.

with total variation 1/2. The remaining part of each map ϑ_r is then extended by odd symmetry about 0 frequency and modulo 1 periodicity. Actually, in this specific example, in view of property (19), the maps are the same for any r .

Of course, in general, the detailed approximation margins depend on the particular time-something representation at hand. However, the flexibility in the choice of the sampling of the time-shift variable in the transform domain can lead to good results as in the case of generalized Gabor frames [23, 21].

5. CONCLUSIONS

In this paper we have revisited recently introduced redressing methods for mitigating dispersion in warped signal representations, with the objective of generalizing the procedure to generic time-something representations like wavelet expansions.

We showed that redressing methods can be extended to generalized settings in both unsampled and sampled transform domains, introducing notation to simplify their application to signal analysis-synthesis scheme.

The phase linearization techniques lead to approximated schemes for the computation of the warped and redressed signal representations, where one can pre-warp the analysis window or mother wavelet and compute the transform without the need for further warping.

6. REFERENCES

- [1] K. Gröchenig, *Foundations of Time-Frequency Analysis*, Applied and Numerical Harmonic Analysis. Birkhäuser Boston, 2001.
- [2] Stephane Mallat, *A Wavelet Tour of Signal Processing, Third Edition: The Sparse Way*, Academic Press, 3rd edition, 2008.
- [3] Ingrid Daubechies, *Ten Lectures on Wavelets*, Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 1992.
- [4] J. P. Princen, A. W. Johnson, and A. B. Bradley, “Subband transform coding using filter bank designs based on time domain aliasing cancellation,” in *IEEE Proc. Intl. Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 1987, pp. 2161–2164.
- [5] H. S. Malvar, “Modulated QMF filter banks with perfect reconstruction,” *Electronics Letters*, vol. 26, no. 13, pp. 906–907, June 1990.
- [6] S. Ghobber and S. Omri, “Time-frequency concentration of the windowed Hankel transform,” *Integral Transforms and Special Functions*, vol. 25, no. 6, pp. 481–496, 2014.
- [7] C. Baccar, N.B. Hamadi, and H. Herch, “Time-frequency analysis of localization operators associated to the windowed Hankel transform,” *Integral Transforms and Special Functions*, vol. 27, no. 3, pp. 245–258, 2016.
- [8] J. Brown, “Calculation of a constant Q spectral transform,” *Journal of the Acoustical Society of America*, vol. 89, no. 1, pp. 425–434, January 1991.
- [9] Y.A. Abramovich and C.D. Aliprantis, *An Invitation to Operator Theory*, vol. 50 of *Graduate studies in mathematics*, American Mathematical Society, 2002.
- [10] 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.
- [11] 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.
- [12] 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.
- [13] 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.
- [14] G. Evangelista, “Dyadic Warped Wavelets,” *Advances in Imaging and Electron Physics*, vol. 117, pp. 73–171, Apr. 2001.
- [15] G. Evangelista, M. Dörfler, and E. Matusiak, “Phase vocoders with arbitrary frequency band selection,” in *Proceedings of the 9th Sound and Music Computing Conference*, Copenhagen, Denmark, 2012, pp. 442–449.
- [16] G. Evangelista, M. Dörfler, and E. Matusiak, “Arbitrary phase vocoders by means of warping,” *Music/Technology*, vol. 7, no. 0, 2013.
- [17] G. Evangelista, “Warped Frames: dispersive vs. non-dispersive sampling,” in *Proceedings of the Sound and Music Computing Conference (SMC-SMAC-2013)*, Stockholm, Sweden, 2013, pp. 553–560.
- [18] P. Balazs, M. Dörfler, F. Jaitlet, 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.

- [19] G. A. Velasco, N. Holighaus, M. Dörfler, and T. Grill, “Constructing an invertible constant-Q transform with non-stationary Gabor frames,” in *Proceedings of the Digital Audio Effects Conference (DAFx-11)*, Paris, France, 2011, pp. 93–99.
- [20] N. Holighaus and C. Wiesmeyr, “Construction of warped time-frequency representations on nonuniform frequency scales, Part I: Frames,” *ArXiv e-prints*, Sept. 2014.
- [21] T. Mejstrik and G. Evangelista, “Estimates of the Reconstruction Error in Partially Redressed Warped Frames Expansions,” in *Proceedings of the Digital Audio Effects Conference (DAFx-16)*, Brno, Czech Republic, 2016, pp. 9–16.
- [22] T. Mejstrik, “Real Time Computation of Redressed Frequency Warped Gabor Expansion,” Master thesis, University of Music and Performing Arts Vienna, tomm-sch.com/science.php, 2015.
- [23] G. Evangelista, “Approximations for Online Computation of Redressed Frequency Warped Vocoders,” in *Proceedings of the Digital Audio Effects Conference (DAFx-14)*, Erlangen, Germany, 2014, pp. 1–7.
- [24] 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.
- [25] R.K. Singh and Ashok Kumar, “Characterizations of invertible, unitary, and normal composition operators,” *Bulletin of the Australian Mathematical Society*, vol. 19, pp. 81–95, 1978.
- [26] R.K. Singh and J.S. Manhas, *Composition Operators on Function Spaces*, vol. 179, North Holland, 1993.
- [27] R.K. Singh, “Invertible composition operators on $\mathbb{L}^2(\lambda)$,” *Proceedings of the American Mathematical Society*, vol. 56, no. 1, pp. 127–129, April 1976.
- [28] James T. Campbell and James E. Jamison, “On some classes of weighted composition operators,” *Glasgow Mathematical Journal*, vol. 32, no. 1, pp. 87–94, 1990.
- [29] V.K. Pandey, *Weighted Composition Operators and Representations of Transformation Groups*, Ph.D. thesis, University of Lucknow, India, 2015.
- [30] O. Christensen, “Frames and pseudo-inverses,” *Journal of Mathematical Analysis and Applications*, vol. 195, no. 2, pp. 401–414, 1995.
- [31] P. W. Broome, “Discrete orthonormal sequences,” *Journal of the ACM*, vol. 12, no. 2, pp. 151–168, Apr. 1965.
- [32] L. Knockaert, “On Orthonormal Muntz-Laguerre Filters,” *IEEE Transactions on Signal Processing*, vol. 49, no. 4, pp. 790–793, apr 2001.

7. APPENDIX: PROOF THAT $\mathbf{U}_{\tilde{\theta}}^{(\tau)} \mathbf{U}_{\tilde{\theta}} \mathbf{T}_\tau = \mathbf{T}_\tau \mathbf{W}_{\tilde{\theta}}$

In this Appendix we prove an important result on which the redressing methods for the warped time-shift operators are based. We show that the application of the same warping operator $\mathbf{U}_{\tilde{\theta}}$ but with respect to time-shift τ instead of time t , which we denote by $\mathbf{U}_{\tilde{\theta}}^{(\tau)}$, results in the commutation of the warping operator $\mathbf{U}_{\tilde{\theta}}$ with the time-shift operator \mathbf{T}_τ in its non-unitary form.

We are going to prove our result under the assumption that the map θ is almost everywhere increasing and has odd parity:

$$\theta(-\nu) = -\theta(\nu). \quad (38)$$

In this case, the first derivative

$$\theta'(\nu) = \frac{d\theta}{d\nu} \quad (39)$$

is almost everywhere positive and has even parity:

$$\theta'(-\nu) = \theta'(\nu). \quad (40)$$

From (17) we have:

$$[\mathbf{U}_{\tilde{\theta}} s](t) = \int_{-\infty}^{+\infty} d\alpha K_\theta(t, \alpha) s(\alpha) \quad (41)$$

where

$$\begin{aligned} K_\theta(t, \alpha) &= \int_{-\infty}^{+\infty} d\nu \sqrt{\frac{d\theta}{d\nu}} e^{j2\pi(\nu t - \theta(\nu)\alpha)} \\ &= \int_{-\infty}^{+\infty} d\nu \sqrt{\frac{d\theta^{-1}}{d\nu}} e^{j2\pi(\theta^{-1}(\nu)t - \nu\alpha)} \end{aligned} \quad (42)$$

where we have used the fact that the map θ is invertible almost everywhere, so that

$$\theta^{-1}(\theta(\nu)) = \nu \quad (43)$$

far almost all $\nu \in \mathbb{R}$, from which it follows that

$$1 = \frac{d[\theta^{-1}(\theta(f))]}{df} = \frac{d\theta^{-1}}{d\alpha} \Big|_{\alpha=\theta(f)} \frac{d\theta}{df} \quad (44)$$

almost everywhere.

Thus, given any finite energy signal $s(t)$, we have

$$\begin{aligned} [\mathbf{U}_{\tilde{\theta}}^{(\tau)} \mathbf{U}_{\tilde{\theta}} \mathbf{T}_\tau s](t) &= \\ &= \int_{-\infty}^{+\infty} d\alpha K_\theta(\tau, \alpha) \int_{-\infty}^{+\infty} d\beta K_\theta(t, \beta) s(\beta - \alpha) \\ &= \int_{-\infty}^{+\infty} d\alpha K_\theta(\tau, \alpha) \int_{-\infty}^{+\infty} d\eta K_\theta(t, \eta + \alpha) s(\eta) \end{aligned} \quad (45)$$

Using (42), by direct calculation, it is easy to show that

$$\int_{-\infty}^{+\infty} d\alpha K_\theta(\tau, \alpha) K_\theta(t, \eta + \alpha) = L_\theta(t - \tau, \eta), \quad (46)$$

where $L_\theta(t, \alpha)$ is the nucleus of the non-unitary warping operator $\mathbf{W}_{\tilde{\theta}}$, i.e.,

$$\begin{aligned} L_\theta(t, \alpha) &= \int_{-\infty}^{+\infty} d\nu e^{j2\pi(\nu t - \theta(\nu)\alpha)} \\ &= \int_{-\infty}^{+\infty} d\nu \frac{d\theta^{-1}}{d\nu} e^{j2\pi(\theta^{-1}(\nu)t - \nu\alpha)}. \end{aligned} \quad (47)$$

Thus, (45) becomes

$$\begin{aligned} [\mathbf{U}_{\tilde{\theta}}^{(\tau)} \mathbf{U}_{\tilde{\theta}} \mathbf{T}_\tau s](t) &= \int_{-\infty}^{+\infty} d\eta L_\theta(t - \tau, \eta) s(\eta) \\ &= [\mathbf{T}_\tau \mathbf{W}_{\tilde{\theta}} s](t), \end{aligned} \quad (48)$$

which is what we needed to prove.

REDS: A NEW ASYMMETRIC ATOM FOR SPARSE AUDIO DECOMPOSITION AND SOUND SYNTHESIS

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

In this paper, we introduce a function designed specifically for sparse audio representations. A progression in the selection of dictionary elements (*atoms*) to sparsely represent audio has occurred: starting with symmetric atoms, then to damped sinusoid and hybrid atoms, and finally to the re-appropriation of the gammatone (GT) and formant-wave-function (FOF) into atoms. These asymmetric atoms have already shown promise in sparse decomposition applications, where they prove to be highly correlated with natural sounds and musical audio, but since neither was originally designed for this application their utility remains limited.

An in-depth comparison of each existing function was conducted based on application specific criteria. A directed design process was completed to create a new atom, the *ramped exponentially damped sinusoid* (REDS), that satisfies all desired properties: the REDS can adapt to a wide range of audio signal features and has good mathematical properties that enable efficient sparse decompositions and synthesis. Moreover, the REDS is proven to be approximately equal to the previous functions under some common conditions.

1. INTRODUCTION

A sparse synthesis model suggests that a signal $\mathbf{s} \in \mathbb{R}^n$ may be represented by a linear combination of a few elements (*atoms*) from dictionary $\mathbf{D} \in \mathbb{R}^{n \times m}$: $\mathbf{s} = \mathbf{D}\mathbf{v}$, where $\mathbf{v} \in \mathbb{R}^m$ is the signal's sparse representation [1] [2]. Decomposing a signal with few elements implies, informally, great meaning is assigned to those elements. On the other hand, creating complex sounds with a few additions provides major efficiency improvements over the alternative (non-sparse) methods. Source-filter synthesis exemplifies a sparse synthesis model because a few waveforms are summed to create a complex sound (typically, a vocal sound) [3]. In either case, a sparse synthesis application requires a dictionary that includes easily controllable atoms capable of representing a wide range of signal content.

Knowledge of salient audio signal features can help guide dictionary design: they are asymmetric in time (short attack and long decay) and usually have time-varying frequency content [4]. Thus, a time-frequency structured signal model that is asymmetric in time (e.g., a damped sinusoid) is appropriate. However, the damped sinusoidal model [5] does not have a smooth attack while real signals almost always do. A compromise involves building a heterogeneous dictionary that includes symmetric atoms (e.g., Gabor atoms) and damped sinusoid atoms. Heterogeneous dictionaries must be indexed by more data, however, because each atom class within the dictionary will have a unique parameter set. More importantly, decomposing asymmetric signal content with a finite number of symmetric atoms will either lead to a non-sparse solution or pre-echo (*dark energy*) [6].

A better approach is to design a homogeneous dictionary (contains a single atom class), wherein the atoms are exponentially damped sinusoids with an attack envelope. Currently, only two functions common in literature have assumed this atomic role: the formant-wave-function [3] [7] (used in audio synthesis) and the gammatone (used in perceptual audio coding) [8] [9]. Matching Pursuit Toolkit (MPTK) supports the use of either function as a dictionary atom [10]. Neither function was designed to be optimized for the task of sparse audio decomposition, though, and they both suffer from limited parameter adaptability.

In this paper, we introduce a new asymmetric atom that is better suited for the sparse synthesis model. A theoretical and practical comparison of existing atom models is presented to consolidate knowledge and highlight their relative strengths and limitations. Our points of comparison reflect the qualities that we seek in a model: ability to match diverse signal behavior (especially transients), and good mathematical properties. Some of the desired mathematical properties include having a concentrated spectrum and an analytic inner product formula. Detailed criteria explanations and justifications reside in a following section.

The paper is structured as follows. Section 2 details the desired atom properties and form. Section 3 includes an analysis and comparison of existing functions. The new atomic model is introduced in Section 4. Section 5 overviews some new atomic model sparse synthesis applications. Fi-

* Sound Processing and Control Laboratory

† Centre for Interdisciplinary Research in Music Media and Technology

nally, in Section 6 we reflect on the future work intended for a sparse audio decomposition system using the new atomic model.

2. ATOM PROPERTIES

2.1. General Form

We generalize the form of a causal asymmetric atom as

$$x[n] = E[n]e^{i\omega_c n}, \quad (1)$$

where

$$E[n] = A[n]e^{-\alpha n}u[n] \quad (2)$$

is the atom's envelope, $\alpha \in \mathbb{R}_{\geq 0}$ is the damping factor, $\omega_c = 2\pi f_c$ is the normalized angular frequency of oscillation ($0 \leq f_c \leq \frac{1}{2}$), $u[n]$ is the unit step function, n is discrete time, and $A[n]$ is an attack envelope that distinguishes each atom ($A[n] \in \mathbb{R}_{\geq 0} \forall n \in \mathbb{N}$). We introduce the atoms as discrete time signals because, in practice, dictionaries are composed of finite sampled (discretized) atoms. We establish mathematical properties of the atoms (e.g., their derivatives) from their continuous time counterparts.

In the literature, the formant-wave-function and gammatone are typically defined with real sinusoids rather than complex ones. We choose to adopt a complex form for mathematical ease of manipulation and concise representation in transform domains. Moreover, it is necessary to parametrize phase for real valued but not for complex valued atoms: expansion coefficients from a signal decomposition using complex atoms will be complex and will thus provide both the magnitude and phase [5].

2.2. Desired Atom Properties

Our comparison criteria are grouped into three categories: time-frequency properties, control & flexibility, and algorithmic efficiency.

2.2.1. Time-Frequency Properties

A dictionary of atoms with varying degrees of *time and frequency concentration* is important for creating a sparse representation overall. For example, a sustained piano note begins with a short attack, which is best represented with concentrated time (spread frequency) resolution, followed by a long decay, which requires an atom with long time support and a concentrated spectrum. Multi-resolution analysis involves decomposing a signal onto a set of analyzing functions whose time-frequency tiling is non-uniform [11] [12]. We are going one step further by considering that some sounds require excellent time localization in the transient region and concentrated frequency resolution in the decay region. We aim at representing both regions with atoms

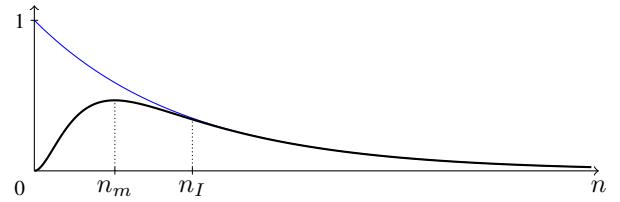


Figure 1: An example envelope of the form (2) overlaid with an exponential envelope (blue), where n_I is the influence time and n_m is the time location of the envelope maximum.

whose envelopes are closer to those of natural sounds. We quantify concentration in time and frequency by the time spread, T , and frequency spread, B , respectively [13]. The Heisenberg-Gabor inequality states $BT \geq 1$.

Moreover, we prefer an atom that has a *unimodal spectrum*: a spectrum $X(\omega)$ is unimodal if $|X(\omega)|$ is monotonically increasing for $\omega \leq \omega_c$ and monotonically decreasing for $\omega \geq \omega_c$. A function that is truncated in time with a rectangular window admits a non-unimodal spectrum because the truncation is equivalent to convolving the spectrum with a sinc function whose oscillations introduce multiple local maxima/minima [14]. Multiple local maxima/minima in the spectrum can complicate spectral parameter estimation. We prefer an infinitely differentiable atom (i.e., of class C^∞ , as defined in [15]) because its spectrum is unimodal.

2.2.2. Control & Flexibility

We modulate the damped sinusoid with A to enhance the atom's adaptability to natural sounds. A damped sinusoid's damping factor α indirectly controls its T and B . Smoothing the damped exponential's initial discontinuity with A concentrates its frequency localization in exchange for a more spread time localization. We want a parametrization of A that enables precise control over its time and frequency characteristics, controllability being an essential aspect of audio synthesis. Furthermore, the attack portion of an audio signal often contains dense spectral content that allows humans to characterize its source [16].

Influence time has a major effect on the atom's overall perceived sound as it controls the degree to which the initial discontinuity is smoothed [16]. We define influence time n_I as the duration that A influences the atom: n_I is the largest value of n for which $e^{-\alpha n}(A[n] - 1) > \delta$ is true (in this paper $\delta = .001$, see Figure 1). The effects of varying influence time are intuitively linked to T and B . In the frequency domain, influence time mostly controls the spectral envelope far from its center frequency (*skirt width* as defined in [3]). Increasing influence time spreads the atom's time localization and concentrates its spectrum.

An important quantity to compare between the atoms is

the time $\Delta_I = n_I - n_m$, where n_m is the time location of E 's maximum. n_m is often called a temporal envelope's attack time in sound synthesis [17]. We find n_m by setting E 's continuous time derivative equal to zero and solving for n . For a continuous E whose $\alpha > 0$, n_m precedes n_I (i.e., A influences E even after n_m). To compare atoms along this criteria, we equalize their n_m values then compare their Δ_I values. Δ_I indicates the amount of influence that varying the skirt width will have on the bandwidth. We prefer an atom with a small Δ_I value because its 3 dB bandwidth (set through α) is not affected much by the structure of A . An envelope with a small Δ_I also reflects those produced by many acoustic instruments: an exciter increases the system's energy and then releases (at n_I), which results in a freely decaying resonance.

We do not want to complicate the definition of the atom when modulating the damped sinusoid by A either; we encourage *time-domain simplicity*. The damped sinusoid's simple definition enables us to solve for its parameters algebraically. Classic parametric estimation techniques can be used to adapt the damped sinusoid to an arbitrary signal [18]. We want to retain these desirable properties even after introducing A . An atom's time-domain simplicity will depend on how its A marries with the complex damped sinusoid. Finally, after modulating the damped sinusoid with A , we want the atom's envelope to match well with those in actual musical signals.

2.2.3. Algorithmic Efficiency

Fast algorithms are one of the focuses of sparse representations research, as they aim to make sparse decomposition processes more tractable. Amid publications dedicated to creating faster algorithms, some reported techniques have become widely adopted [10]. Specifically, certain analytic formulas are known to increase the algorithm speed because they avoid some of the algorithm's most time consuming numerical calculations (e.g., the inner product).

An envelope shape that enables the inner product of two atoms to be expressed as an analytic formula is required for a fast matching pursuit algorithm [1]. A summary of the relevant algorithm steps are included for justification.

In matching pursuit, a dictionary \mathbf{D} is compared with a signal s by calculating and storing the inner product $\langle s, g_\gamma \rangle$ of each atom g_γ and the signal. The atom that forms the largest inner product g_q is picked as the signal's best fit. Then $\langle g_q, g_\gamma \rangle$ is computed and subtracted from $\langle s, g_\gamma \rangle$. This continues until some stopping criteria is met.

Dictionary inner products can be calculated and stored once when the dictionary is static. However, when atom parameters are refined within the iterative loop these inner products cannot be precomputed and, therefore, must be computed at each iteration. Numerical calculations of many

inner products at every iteration prohibit speed. Analytic formulas make the process tractable.

Another way to increase the efficiency of a sparse decomposition program is to use parametric atoms, then refine atom parameters using an estimator. Finding a more adapted atom at every iteration may require less iterations overall. Developing parametric estimation techniques sometimes relies on having analytic discrete Fourier transform (DFT) formula. For example, in derivative methods, two spectra are divided to solve for one or more variables [18]. We include each atom's analytic DFT formula in [19] and [20].

Finally, [5] explains how recursion may be exploited to calculate the convolution of damped sinusoidal atoms with a signal: since the impulse response of a complex one-pole filter is a damped complex exponential sinusoid, a recursive filter can efficiently calculate the correlation. We provide each atom's Z-transform to indicate its *causal filter simplicity* and therefore practicality for calculating the correlation. Besides, the Z-transform is useful for source-filter synthesis and auditory filtering.

3. EXISTING FUNCTIONS

3.1. Damped Sinusoid

3.1.1. Background

The damped sinusoid (DS) is essential in audio as it represents a vibrating mode of a resonant structure. The use of a DS model in the context of analysis dates back to Prony's method [21], according to our knowledge, and was the first asymmetric atom used in the context of sparse representations [5].

3.1.2. Properties

Staying with the predefined generic atom expression (1):

$$A_{DS}[n] = 1 \quad (3)$$

and thus $n_m = n_I = 0$. Its continuous-time Fourier transform is well known,

$$X_{DS}(\omega) = \frac{1}{\alpha + i(\omega - \omega_c)} \quad (4)$$

as is the DFT [20], and finally, the Z-transform,

$$X_{DS}[z] = \frac{1}{1 - e^{-\alpha+i\omega_c} z^{-1}} \quad (5)$$

The DS' spectrum is unimodal but not concentrated.

3.2. Gammatone

3.2.1. Background

Auditory filter models are designed to emulate cochlea processing and are central to applications like perceptual audio

coding, where auditory filters are used to determine which sounds should be coded or not according to auditory masking principles. Auditory filter modeling has a variety of applications in bio-mechanics and psychoacoustic research.

The most popular auditory filter model is the gammatone (GT) filter due to its heritage and simple time domain expression. Originally described in 1960 as a fitting function for basilar displacement in the human ear [8], the gammatone filter was later found to precisely describe human auditory filters, as proven from psychoacoustic data [22]. [9] shows that atoms learned optimally from speech and natural sounds resemble gammatones. Designing gammatone filters remains a focus in audio signal processing [23].

More recently, filter models closely related to the gammatone filter have been proposed, such as the all-pass gammatone filter and the cascade family [24]. Added features of these variants do not overlap with our criteria so they are not included for comparison.

3.2.2. Properties

We assign the gammatone as the prototypical auditory filter model. A single variable polynomial envelope function shapes the gammatone:

$$A_{GT}[n] = n^p \quad (6)$$

In literature, $p + 1$ is called the filter order. A_{GT} does not converge (its derivative is strictly positive), and thus admits the largest Δ_I in this study, as $n_m = \frac{p}{\alpha}$ and $n_I > 2n_m$. No part of the gammatone is, strictly speaking, a freely decaying sinusoid (excluding when $p = 0$, in which case it is a DS), though it asymptotically approaches a DS as $n \rightarrow \infty$.

We demonstrate the filter order's effect by applying the Fourier transform frequency differentiation property to express its spectrum parametrized by p :

$$X_{GT}(\omega) = \frac{p!}{(\alpha + i(\omega - \omega_c))^{p+1}} \quad (7)$$

From its frequency representation, we see that the filter order determines the denominator polynomial order. Finally, referencing the convolution property of the Fourier transform, the gammatone impulse response is a DS convolved with itself p times.

Frequency spread B decreases with respect to the model order, while the time spread T increases. A gammatone of order four ($p = 3$) correlates best with auditory models [23]. The gammatone's spectrum is unimodal and concentrated.

The attack envelope is not parametrized, and therefore cannot be controlled independently of α . After setting p , controlling the atom is solely through α and ω_c . Influence time (or skirt width) is not directly controllable, so one cannot tune the atom to have time concentration in exchange

for frequency spread. Thus, the adaptability of this model to a range of sound signal behavior is limited.

We establish an analytic formula for the gammatone's Z-transform that supports an arbitrary integer $p > 0$:

$$X_{GT}[z] = \frac{\sum_{r=1}^p \binom{p}{r-1} a^r}{(1-a)^{p+1}} \quad (8)$$

where $a = e^{-\alpha+i\omega_c} z^{-1}$, and the Eulerian number $\binom{p}{r-1} = \sum_{j=0}^r (-1)^j \binom{p+1}{j} (r-j)^p$. The gammatone's analytic inner product formula is complicated and described in [20].

3.3. Formant-Wave-Function

3.3.1. Background

In the source-filter model, an output sound signal is considered to be produced by an excitation function sent into a (resonant) filter, referred to as a source-filter pair [3]. Most acoustic instruments involve an exciter, either forced or free, and a resonator [4]. When an instrument's exciter and resonator are independent, or have only a small effect on one another, its sound production mechanism may be described sufficiently with a source-filter model. An example is the voice production system, where excitations produced by glottal pulses are filtered within the vocal tract.

Source-filter synthesis involves sending an excitation function through one or more resonant filters in parallel. The filters are typically one or two pole and defined by their auto-regressive filter coefficients. The excitation function can be an impulse, but is more often an impulse smoothed by a window to emulate natural excitation. The window shape effects the transient portion of the time-domain output from the system, and the skirts of the spectral envelope. The filter coefficients control the shape of the spectral envelope near the resonant peak.

Time-domain formant-wave-function synthesis describes the output of the source-filter model by a single function in the time domain. The amplitude envelope of the function is designed to generically match the output envelope of a source-filter pair: a damped exponential (filter) for which the initial discontinuity is smoothed (excitation). The advantage of this approach is twofold: direct parametrization of the spectral envelope, and efficient synthesis by table lookup [3].

Creating a sustained sound (e.g., a voice) from this model involves filtering a sparse excitation signal made of a (possibly) periodic sequence of short duration signals. Likewise, synthesizing a percussive sound (e.g., a piano) involves summing the output of several resonant filters with comparably long decay times from a single excitation. In the time-domain method, this means that the model synthesizes a signal s as a linear combination of time-shifted resonant filter impulse responses (i.e., time-frequency atoms). Formally,

we express this as $s[n] = \sum h_\lambda[n - \tau]v_{\lambda,\tau} = \mathbf{D}\mathbf{v}$, where \mathbf{D} is a dictionary of atoms $h_{\lambda,\tau}[n] = h_\lambda[n - \tau]$ that are indexed by λ and time shift τ , and \mathbf{v} contains their amplitude coefficients $v_{\lambda,\tau}$. Thus, the source-filter model is a sparse synthesis representation.

3.3.2. Properties

The formant-wave-function (FOF) is ubiquitous with time-domain wave-function synthesis. It was introduced for its desirable properties: a concentrated spectral envelope that can be controlled rather precisely using two parameters. The FOF's A is defined as:

$$A_{FOF}[n] = \begin{cases} \frac{1}{2}(1 - \cos(n\beta)) & \text{for } 0 \leq n \leq \frac{\pi}{\beta}, \\ 1 & \text{for } \frac{\pi}{\beta} < n. \end{cases} \quad (9)$$

where $\beta \in \mathbb{R}_{>0}$ controls influence time. Decreasing β increases influence time, $n_I \approx \frac{\pi}{\beta}$, and the time location of the maximum,

$$n_m = \frac{1}{\beta} \cos^{-1} \left(\frac{\alpha^2 - \beta^2}{\alpha^2 + \beta^2} \right) \quad (10)$$

Δ_I and $\frac{\alpha}{\beta}$ are positively correlated.

A raised cosine is an excellent attack shape in terms of concentration, however, since it is piecewise (its value must be held at one after half of a period) some other design criteria suffer.

$$X_{FOF}(\omega) = \frac{\beta^2}{2} \frac{1 + e^{-\frac{\pi}{\beta}(\alpha+i(\omega-\omega_c))}}{(\alpha+i(\omega-\omega_c))((\alpha+i(\omega-\omega_c))^2 + \beta^2)} \quad (11)$$

The FOF's spectrum is non-unimodal when the piecewise transition occurs within the window of observation. Moreover, it is difficult to estimate the FOF's parameters and its analytic inner product formula is complicated [7].

We establish the FOF's DFT and Z-transform by converting the cosine function into a sum of complex exponentials and using the linear property:

$$X_{FOF}[z] = \frac{1}{2} \frac{1+a^{N_1}}{1-a} - \frac{1}{4} \left(\frac{1-(ae^{i\beta})^{N_1}}{1-ae^{i\beta}} + \frac{1-(ae^{-i\beta})^{N_1}}{1-ae^{-i\beta}} \right) \quad (12)$$

where $a = e^{-\alpha+i\omega_c} z^{-1}$, and $N_1 = [\frac{\pi}{\beta}]$. From (12), we see that a FOF filter may be implemented as a sum of three complex pole-zero filters. The time-varying input delay complicates controlling attack shape.

3.4. Connecting FOF to Gammatone

Applying the small angle theorem to A_{FOF} reveals a relation between the FOF and GT:

$$\lim_{\beta \rightarrow 0} \frac{2}{\beta^2} (1 - \cos(\beta n)) = n^2 \quad (13)$$

We establish that a FOF and gammatone of $p = 2$ are approximately equal when $\beta = \frac{1}{4}\alpha\sqrt{12\epsilon}$, where ϵ is the approximation error (see [20] for proof).

3.5. Recapitulation

Each existing function has several desired properties missing. While the gammatone's unimodal frequency spectrum and time-domain simplicity are appealing, expressing its DFT and inner product is complicated. Most importantly, without a parameter to control influence time, the gammatone is not flexible enough to sparsely represent a variety of signal features. On the other hand, the FOF's attack function enables precise control over its spectral envelope, however, its piecewise construction is problematic: spectral ripples result from a truncation in time, refining its parameters is difficult, and its frequency, Z-transform, and inner product expressions are complicated.

3.6. Towards a New Atom

The starting goal of this paper was to design a $C^\infty A$ that is similar to A_{FOF} . While piecewise construction is the reason for the FOF's shortcomings, approximating the raised cosine with a C^∞ function does not necessarily improve the situation because many functions admit complicated frequency-domain and Z-domain formulas once a unit step is introduced. For example, $e^{-\beta n^2}$ has a compact bell shape that seems to be, at first inspection, a good candidate to replace the raised cosine. However, when a unit step is introduced, it admits a non-algebraic Fourier transform expression (a special function defines the imaginary part). Many bell-shaped functions have the same problem (e.g., $\tanh(\beta n)^2$).

On the other hand, there are A options that are simple but have Δ_I that are large compared to the FOF for equal n_m . In fact, any C^∞ function will have a larger Δ_I than the FOF's for equal n_m . Therefore, our goal became more specific: define a $C^\infty A$ that admits simple mathematical expressions when married with a complex damped exponential, and whose Δ_I is close to that of the FOF's for equal n_m . After an exhaustive search, we resolved that designing a function to satisfy all of the design criteria is difficult.

4. THE NEW ASYMMETRIC ATOM

We have designed a new atom specifically for sparse audio representations. All the aforementioned criteria were in mind when constructing this new atom.

4.1. Background

To reflect generality, we call the new atom the *ramped exponentially damped sinusoid* (REDS). Identically to existing source-filter and auditory filter models, a complex exponentially damped sinusoid defines the atom's decay section. A binomial with one exponential term shapes the atom's attack envelope. By defining the atom as a sum of exponentials (see (16)), all the desired mathematical properties

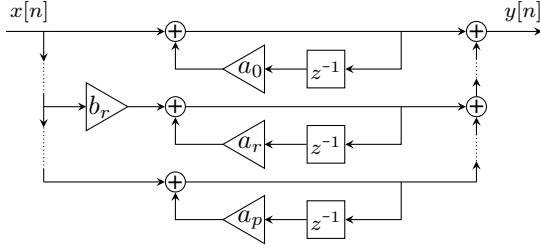


Figure 2: REDS filter diagram, where $a_r = e^{-\alpha - r\beta + i\omega_c}$ and $b_r = (-1)^r \binom{p}{r}$.

are achieved. The main idea is that the linear property of the Fourier transform and Z-transform can be exploited and each exponential has a transform that is simple and well known.

4.2. Properties

We define REDS concisely in the time-domain by expressing $A_{REDS}[n]$ polynomially as $(1 - e^{-\beta n})^p$:

$$x[n] = (1 - e^{-\beta n})^p e^{n(-\alpha + i\omega_c)} u[n] \quad (14)$$

where β controls the influence time (or skirt width) and $p+1$ is the order.

$$n_m = \frac{1}{\beta} \log(1 + \frac{p\beta}{\alpha}) \quad (15)$$

and $n_I \approx -\frac{1}{\beta} \log(1 - (1 - \delta)^{1/p})$, where δ is the same as in Section 2.2.2.

Like in the gammatone model, order is often constant within an application: we may choose the order, for example, to match with auditory data or to approximate a frame condition [23]. Given that the order is a constant, the number of control parameters and their effect are the same as the FOF. To summarize, the REDS parameter set is a conflation of the source-filter and auditory filter models.

We express the REDS in binomial form to reveal its sum of exponentials construction:

$$x[n] = \sum_{r=0}^p (-1)^r \binom{p}{r} e^{n(-\alpha - r\beta + i\omega_c)} u[n] \quad (16)$$

Considering the Fourier transform linear property, we readily find from (16) the Fourier transform of REDS:

$$X_{REDS}(\omega) = \sum_{r=0}^p (-1)^r \binom{p}{r} \frac{1}{\alpha + r\beta + i(\omega - \omega_c)} \quad (17)$$

and the analytic DFT [20]. Finally, we apply the linear property to retrieve the Z-transform:

$$X_{REDS}[z] = \sum_{r=0}^p (-1)^r \binom{p}{r} \frac{1}{1 - e^{-\alpha - r\beta + i\omega_c} z^{-1}} \quad (18)$$

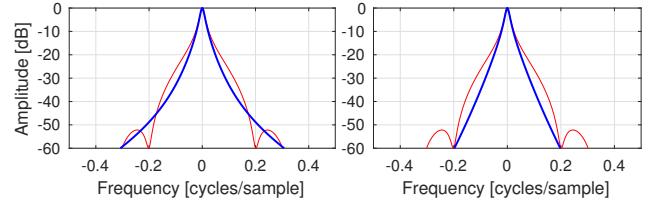


Figure 3: $X_{FOF}[\omega_k]$ (red) and $X_{RED[p]}[\omega_k]$ (blue) for fixed β , where $p = 2$ on the left, $p = 10$ on the right, and $\alpha = .05$.

A sum of $p+1$ complex one-pole filters in parallel will thus output a REDS (see Figure 2).

The REDS has a concentrated and unimodal spectrum. Similarly to the FOF, it is possible to precisely control the REDS' spectrum: by varying β one may exchange concentration in time for frequency, and vice versa. The FOF has greater time concentration than the REDS because the raised cosine attack function has a fast uniform transition from zero to one, while the REDS attack envelope is bell-shaped. Formally, $n_{I_{REDS}} > n_{I_{FOF}}$ when $n_{m_{REDS}} = n_{m_{FOF}}$. The REDS' spectral concentration surpasses the FOF's as p increases (see Figure 3).

We established analytic inner product and convolution formulas for two REDS atoms that support the case when atoms have different lengths N (see [20]). Considering that these formulas for Gabor atoms and FOFs provide an efficiency boost in existing programs [7], and the REDS formulas are simpler than those, we assume that using the formulas are more efficient than numerical computations.

4.3. Connection to Existing Functions

A REDS is approximately equal to a GT when β is very small:

$$\lim_{\beta \rightarrow 0} \frac{1}{\beta^p} (1 - e^{-\beta n})^p = n^p \quad (19)$$

We establish that a REDS and a GT are approximately equal when $\beta = \epsilon \alpha p^{-2}$, where ϵ is the approximation error (see [20] for proof). This is important because the REDS filter requires fewer mathematical operations per sample than the gammatone filter. In practice, the perceptual difference between the two is negligible when $\epsilon < .001$, which corresponds to a signal-to-noise ratio between the two atoms greater than 60 dB.

Furthermore, by (13), $A_{REDS}[n] \approx 2A_{FOF}[n]$ when $p = 2$ and their β values are $\frac{1}{4}\epsilon\alpha$.

5. APPLICATIONS

This section includes two sparse synthesis examples: REDS source-filter synthesis, and musical audio decomposition.

5.1. Synthesis

Sparse synthesis via the source-filter model generally involves sending a short excitation periodically through one or more filters, typically one per formant. In this example we checked the ability of the REDS to synthesize a vowel sound /i/ with 5 REDS filters tuned by parameters provided in [3], see Table 1. We set $p = 2$ (see (19)) for better comparison with FOF, which have demonstrated their aptitude for singing voice synthesis. After normalization (see normalization factor in [19] or [20]), a gain G ($0 \leq G \leq 1$) tunes the filter output amplitude. Results show the quality of the synthesized sound as well as the ability to match the spectral envelopes, even in the valleys, mainly controlled by β (see [19]).

Table 1: REDS filter settings for synthesizing a vowel, where ν_c is center frequency in Hz and $p = 2$.

	ν_c	260	1764	2510	3100	3600
α	.005	.006	.006	.009	.011	
β	.018	.059	.034	.011	.008	
G	1.0	.501	.447	.316	.056	

5.2. Decomposition

We decomposed a set of real audio signals using a standard matching pursuit algorithm¹. We selected the audio signal set to reflect a range of the source-filter model: it includes a vocal sound (sustained, relatively high damping and smooth attack per atom), a vibraphone (not-sustained, made of low damping and short attack per atom), and a violin (intermediate situation).

We created damped sinusoid dictionaries to fit with each signal's content. Then we made dictionaries for each atom class by modulating the damped sinusoid dictionaries with a set of each atom's $A[n]$. We superimposed each selected atom's Wigner-Ville distribution to show their time-frequency footprint, such as in [6] & [7].

We can represent a signal as time-varying sinusoidal trajectories per the additive model, or as filtered excitation sequences per the source-filter model, by decomposing it onto a dictionary of REDS atoms with constrained damping factors. We chose to demonstrate the ability of the REDS to analyze the signal set from the source-filter viewpoint. For the singing voice, if the dictionary contained atoms with small damping (long time support) then the selected atoms would represent the sinusoidal partials of the signal. We set the damping to be high and in doing so, successfully extracted the excitation sequence of atoms whose spectral con-

¹Standard in the sense that the dictionary was static and it did not involve fast algorithms or parameter refinements. We implemented the algorithm based on [1].

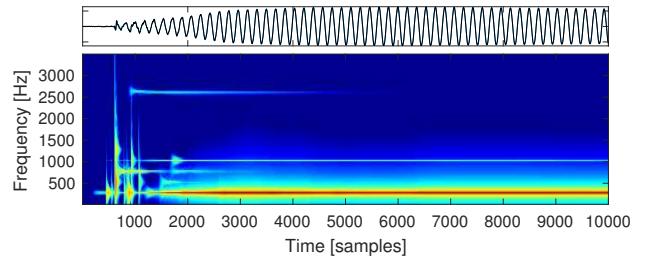


Figure 4: Sparse representation of a vibraphone (transient part shown) from a decomposition onto 50 REDS atoms.

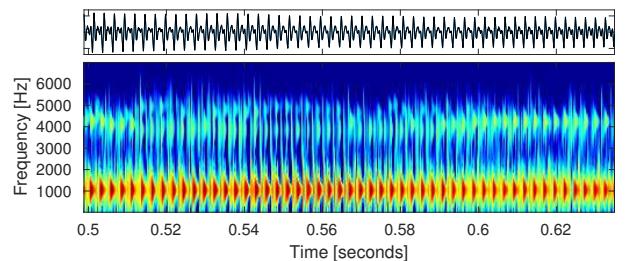


Figure 5: Source-filter sparse representation of a singing voice. Atom spacing expands/contracts reflecting vibrato.

Table 2: Decomposition results. Each audio signal is sampled at 44100 Hz. d_s is the signal's duration in seconds.

d_s	Atoms	N	SNR (dB)			
			DS	GT	FOF	REDS
Vocal	10^4	2^8	30.7	35.9	37.8	38.8
Violin	10^4	2^9	20.0	14.6	27.7	28.0
Vibes	50	2^{17}	17.6	32.1	36.9	37.1

tent represented vocal formants rather than the sinusoidal partials (see Figure 5). Regarding the vibraphone, we created a dictionary whose damped sinusoids had large time support with low decay rates.

For each test, the REDS dictionaries provided higher SNR values for the same number of iterations, see Table 2. For the singing voice, the gammatone and REDS were close in performance because the formant time-domain envelopes had very smooth attacks. REDS matched the vibraphone's envelope tightly, while the gammatone caused pre-echo because of its greater amount of symmetry (see [19]). The reconstructed signal from the REDS decomposition had an SNR of 38.8 dB, and consisted of 50 atoms (.04% of the signal length) (see Figure 4). Our companion website includes audio files for each signal approximation and further details the decomposition study results [19].

6. CONCLUSION

We have introduced a new function called REDS that can sparsely represent audio. Through the comparison of functions previously used in sparse representation contexts, we highlighted the most important features for this new function to embody (see Table 3). We have started researching an efficient sparse audio decomposition system that exploits the good properties of the new asymmetric atom.

Since the mathematical properties of the REDS enable efficient implementations of filter banks, source-filter synthesis, and audio coding, the REDS has potential to be used in many audio signal processing fields.

Table 3: Comparison results.

Criteria	DS	GT	FOF	REDS
Concentrated Spectrum	—	✓	✓	✓
Unimodal Spectrum	✓	✓	—	✓
Influence Time Control	—	—	✓	✓
Time-Domain Simplicity	✓	✓	—	✓
Causal Filter Simplicity	✓	✓	—	✓
Inner Product Simplicity	✓	—	—	✓

7. REFERENCES

- [1] S. Mallat and Z. Zhang, “Matching pursuits with time-frequency dictionaries,” *IEEE Trans. Signal Process.*, vol. 41, no. 12, pp. 3397–3415, Dec. 1993.
- [2] P. Balazs, M. Doerfler, M. Kowalski, and B. Torresani, “Adapted and adaptive linear time-frequency representations: A synthesis point of view,” *IEEE Signal Process. Magazine*, vol. 30, no. 6, pp. 20–31, Nov. 2013.
- [3] X. Rodet, *Time-domain formant-wave-function synthesis*, chapter 4-Speech Synthesis, pp. 429–441, J.C. Simon, Ed. New York, 1980.
- [4] N. Fletcher and T. Rossing, *The Physics of Musical Instruments*, Springer New York, 2nd edition, 1998.
- [5] M. Goodwin, “Matching pursuit with damped sinusoids,” in *IEEE Proc. Int. Conf. Acoust. Speech Signal Process. (ICASSP)*, Munich, Germany, Apr. 1997, vol. 3, pp. 2037–2040.
- [6] B. L. Sturm, C. Roads, A. McLeran, and J. J. Shynk, “Analysis, visualization, and transformation of audio signals using dictionary-based methods,” *J. New Music Research*, vol. 38, no. 4, pp. 325–341, 2009.
- [7] 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.
- [8] J. L. Flanagan, “Models for approximating basilar membrane displacement,” *Bell System Technical Journal*, vol. 39, no. 5, pp. 1163–1191, 1960.
- [9] E. Smith and M. Lewicki, “Efficient auditory coding,” *Nature*, vol. 439, no. 7079, pp. 978–982, Feb. 2006.
- [10] S. Krstulovic and R. Gribonval, “MPTK: Matching pursuit made tractable,” in *IEEE Proc. Int. Conf. Acoust. Speech Signal Process. (ICASSP)*, May 2006, vol. 3, pp. 496–499.
- [11] S. Mallat, *A Wavelet Tour of Signal Processing*, Academic Press, 3rd edition, 2009.
- [12] P. Balazs, M. Dörfler, F. Jaillet, N. Holighaus, and G. Velasco, “Theory, implementation and applications of nonstationary gabor frames,” *J. Comput. Appl. Math*, vol. 236, no. 6, pp. 1481 – 1496, 2011.
- [13] P. Flandrin, *Time-Frequency / Time-Scale Analysis*, vol. 10 of *Wavelet analysis and its applications*, chapter 1, pp. 9 – 47, Academic Press, 1999.
- [14] F. J. Harris, “On the use of windows for harmonic analysis with the discrete Fourier transform,” *Proceedings of the IEEE*, vol. 66, no. 1, pp. 51–83, Jan. 1978.
- [15] F. W. Warner, *Foundations of Differentiable Manifolds and Lie Groups*, Springer New York, 1983.
- [16] A.S. Bregman, *Auditory Scene Analysis*, MIT Press, Cambridge, MA, 1990.
- [17] M. Mathews and J. Miller, *The technology of computer music*, M.I.T. Press, Cambridge, Mass., 1969.
- [18] S. Marchand and P. Depalle, “Generalization of the derivative analysis method to non-stationary sinusoidal modeling,” in *Proc. of the 11th Int. Conf. on Digital Audio Effects (DAFx-08)*, Espoo, Finland, Sep. 2008.
- [19] J. Neri, “REDS website,” <http://www.music.mcgill.ca/~julian/dafx17>, 2017.
- [20] J. Neri, “Sparse representations of audio signals with asymmetric atoms,” M.A. thesis, McGill University, Montréal, Canada, 2017.
- [21] G. M. Riche de Prony, “Essai expérimental et analytique sur les lois de la dilatabilité de fluides élastiques,” *Journal de l’École Polytechnique*, vol. 1, no. 22, pp. 24–76, 1795.
- [22] R. Patterson, I. Nimmo-Smith, J. Holdsworth, and P. Rice, “An efficient auditory filterbank based on the gammatone function,” *Paper presented at a meeting of the IOC Speech Group on Auditory Modelling at RSRE*, Dec. 1987.
- [23] S. Strahl and A. Mertins, “Analysis and design of gammatone signal models,” *J. Acoust. Soc. Am.*, vol. 126, no. 5, pp. 2379–2389, Nov. 2009.
- [24] R. Lyon, A. Katsiamis, and E. Drakakis, “History and future of auditory filter models,” in *Proc. IEEE Int. Conf. Circuits and Systems (ISCAS)*, Aug. 2010, pp. 3809–3812.

HARMONIC-PERCUSSIVE SOUND SEPARATION USING RHYTHMIC INFORMATION FROM NON-NEGATIVE MATRIX FACTORIZATION IN SINGLE-CHANNEL MUSIC RECORDINGS

F.J. Canadas-Quesada¹, D. Fitzgerald², P. Vera-Candeas¹, N. Ruiz-Reyes¹ *

¹ Telecommunication Engineering Department, Higher Polytechnic School of Linares, University of Jaen, Jaen, Spain

² Cork School of Music, Cork Institute of Technology, Cork, Ireland

fcanadas@ujaen.es, Derry.Fitzgerald@cit.ie, pvera@ujaen.es, nicolas@ujaen.es

ABSTRACT

This paper proposes a novel method for separating harmonic and percussive sounds in single-channel music recordings. Standard non-negative matrix factorization (NMF) is used to obtain the activations of the most representative patterns active in the mixture. The basic idea is to classify automatically those activations that exhibit rhythmic and non-rhythmic patterns. We assume that percussive sounds are modeled by those activations that exhibit a rhythmic pattern. However, harmonic and vocal sounds are modeled by those activations that exhibit a less rhythmic pattern. The classification of the harmonic or percussive NMF activations is performed using a recursive process based on successive correlations applied to the activations. Specifically, promising results are obtained when a sound is classified as percussive through the identification of a set of peaks in the output of the fourth correlation. The reason is because harmonic sounds tend to be represented by one valley in a half-cycle waveform at the output of the fourth correlation. Evaluation shows that the proposed method provides competitive results compared to other reference state-of-the-art methods. Some audio examples are available to illustrate the separation performance of the proposed method.

1. INTRODUCTION

Harmonic (pitched instruments) and percussive (drums) sound separation is still an unsolved problem in music signal processing and machine learning. It can be applied to Music Information Retrieval (MIR) in two ways. From a percussive point of view, it can enhance tasks such as, onset detection and tempo estimation. From a harmonic point of view, it can improve other tasks such as, score alignment, multi-pitch and melody estimation, chord detection or vocal extraction.

Rhythm can be considered of core importance in most music, and it is often provided by percussive sounds. Although there are some harmonic instruments (e.g., bass guitar) that show repetitive temporal behavior, in this paper, we assume that percussive sounds (repetitive) exhibit a more rhythmic pattern compared to harmonic sounds (non-repetitive). In this manner, rhythmic information could be useful to discriminate percussive and harmonic sounds in an acoustic mixture in the same manner that a non-trained listener can effortlessly discriminate between them.

In recent years, several approaches have been applied in the field of harmonic-percussive sound separation. Most of these approaches utilize the anisotropy of harmonic and percussive sounds, that is, percussive sounds have a structure that is vertically smooth

in frequency, whereas harmonic sounds are temporally stable and have a structure that is horizontally smooth in time. Anisotropy is applied in a maximum a posteriori (MAP) framework in [1]. Furthermore, the concept of anisotropy is applied using median filtering assuming that harmonics and percussive onsets can respectively be considered outliers in a temporal or frequency slice of a spectrogram [2]. In [3], a non-negative matrix partial co-factorization is presented that forces some portion of bases to be associated with drums only. Kim et.al [4] develop an extension of non-negative matrix factorization (NMF) using temporal repeatability of the rhythmic sound sources among segments of a mixture. Canadas et.al [5] propose an unsupervised NMF integrating spectro-temporal features, such as anisotropic smoothness or time-frequency sparseness, into the factorization process. Kernel additive modeling (KAM) is used to separate sound sources assuming that individual time-frequency bins are close in value to other bins nearby in the spectrogram where nearness is defined through a source-specific proximity kernel [6]. Driedger et. al [7] enforce the components to be clearly harmonic or percussive by exploiting a third residual component that captures the sounds that lie in between the clearly harmonic and percussive sounds. Park and Lee [8] include sparsity and harmonicity constraints in a NMF approach which uses a generalized Dirichlet prior. In [9], the concept of percussive anisotropy is used assuming that the percussive chroma clearly shows an energy distribution which is approximately flat.

One of the main goals of this paper is to determine if only the use of rhythmic information can provide reliable information to discriminate between harmonic and percussive sources. In this paper, we propose a method to separate harmonic and percussive sounds only analyzing the temporal information contained in the activations obtained from non-negative matrix factorization. The basic idea is to classify automatically those activations that exhibit rhythmic (percussive) and non-rhythmic (harmonic and vocal) patterns, assuming that a rhythmic pattern models repetitive events typically shown by percussive sounds and a non-rhythmic pattern models non-repetitive events typically shown by harmonic or vocal sounds. Specifically, the proposed method uses a recursive process based on successive correlations applied to the NMF activations. As shown later, a percussive sound is characterized by a set of peaks at the output of the fourth correlation but a harmonic sound tends to be represented by one valley in a half-cycle waveform at the output of the fourth correlation. Some of the advantages of our proposal are (i) simplicity; (ii) no prior information about the spectral content of the musical instruments and (iii) no prior training.

The remainder of this paper is organized as follows. Section 2 introduces briefly the mathematical background related to the standard NMF. In Section 3, the proposed method is detailed. Sec-

* This work was supported by the Spanish Ministry of Economy and Competitiveness under Project TEC2015-67387-C4-2-R

tion 4 optimizes and evaluates the separation performance of the proposed method compared to reference state-of-the-art methods. Finally, conclusions and future work are presented in Section 5.

2. NON-NEGATIVE MATRIX FACTORIZATION

Standard (unconstrained) non-negative matrix factorization (NMF) [10] attempts to obtain a parts-based representation of the most representative objects in a matrix by imposing non-negative constraints. The basic concept of NMF can be expressed as $X_{F,T} \approx \hat{X}_{F,T} = W_{F,K}H_{K,T}$ where the mixture is modelled as the linear combination of K components. Specifically, $X_{F,T}$ represents the magnitude spectrogram of the mixture, where $f = 1, \dots, F$ denotes the frequency bin and $t = 1, \dots, T$ is the time frame, $\hat{X}_{F,T}$ is the estimated matrix, $W_{F,K}$ is the basis matrix whose columns are the basis functions (or spectral patterns) and $H_{K,T}$ is the activation matrix for the basis functions. The rank or number of components K is generally chosen such that $FK + KT \ll FT$ in order to reduce the dimensions of the data. The factorization is obtained by minimizing a cost function $D(X|\hat{X})$ defined as,

$$D(X|\hat{X}) = \sum_{f=1}^F \sum_{t=1}^T d(X_{f,t}|\hat{X}_{f,t}) \quad (1)$$

where $d(a|b)$ is a function of two scalar variables. In this work, we used the generalized Kullback-Leibler divergence $D(X|\hat{X}) = D_{KL}(X|\hat{X})$ because it has been successfully applied in the field of sound source separation [11] [12] [13],

$$D_{KL}(X|\hat{X}) = \left(X \odot \log \left(X \oslash \hat{X} \right) \right) - X + \hat{X} \quad (2)$$

where \odot is the element-wise multiplication and \oslash is the element-wise division.

The cost function $D_{KL}(X|\hat{X})$ is minimized using an iterative algorithm based on multiplicative update rules and the non-negativity of the bases and the activations is ensured. In this manner, the multiplicative update rule for an arbitrary scalar parameter Z is computed as follows,

$$Z \leftarrow Z \odot \left(\left[\frac{\partial D_{KL}(X|\hat{X})}{\partial Z} \right]^- \oslash \left[\frac{\partial D_{KL}(X|\hat{X})}{\partial Z} \right]^+ \right) \quad (3)$$

3. PROPOSED METHOD

The proposed method uses standard NMF to separate percussive sounds $x_p(t)$ from harmonic sounds $x_h(t)$ in single-channel music mixtures $x(t)$. The magnitude spectrogram X of a mixture $x(t)$, calculated from the magnitude of the short-time Fourier transform (STFT) using a N -sample hamming window $w(n)$ and a J -sample hop size, is composed of F frequency bins and T frames. We assume that percussive and harmonic sounds are mixed in an approximately linear manner, that is, $x(t) = x_p(t) + x_h(t)$ in the time domain or $X = X_p + X_h$ in the magnitude frequency domain. As a result, X can be factorized into two separated spectrograms, \hat{X}_p (an estimated spectrogram only composed of percussive sounds) and \hat{X}_h (an estimated spectrogram only composed of harmonic sounds),

$$X = X_p + X_h = [W_p \quad W_h] \begin{bmatrix} H_p \\ H_h \end{bmatrix} \quad (4)$$

where W_p, H_p are the original percussive bases and activations; W_h, H_h are the original harmonic bases and activations. All the previous data are non-negative matrices. The flowchart of the proposed method is shown in Figure 1.

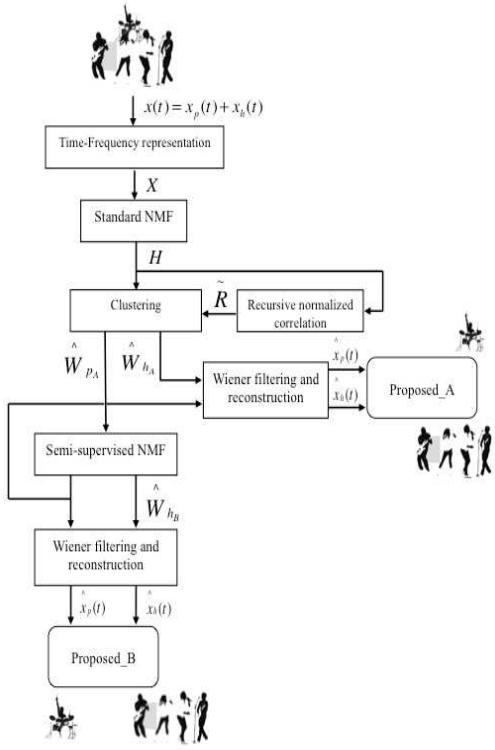


Figure 1: Flowchart of the proposed method for the task of harmonic-percussive sound separation in single-channel music recordings.

3.1. Obtaining activations

Standard NMF is applied to the magnitude spectrogram X using the cost function $D_{KL}(X|\hat{X})$ previously mentioned in section 2. The update rules are defined as follows,

$$H \leftarrow H \odot \left(\left(W^T (X \oslash \hat{X}) \right) \oslash \left(W^T \mathbf{1}_{F,T} \right) \right) \quad (5)$$

$$W \leftarrow W \odot \left(\left((X \oslash \hat{X}) H^T \right) \oslash \left(\mathbf{1}_{F,T} H^T \right) \right) \quad (6)$$

where W and H are initialized as random positive matrices, $\mathbf{1}_{F,T}$ represents a matrix of all-ones composed of F rows and T columns and T is the transpose operator. Note that in this paper we normalise each i^{th} basis function using the L^2 -norm, that is, $\tilde{W}_i = \frac{W_i}{\|W_i\|_2}$, being $\|\tilde{W}_i\|_2 = 1.0$.

Standard NMF can only ensure convergence to local minima, which enables the reconstruction of the mixture but cannot distinguish by itself if the i^{th} component represents a percussive or harmonic sound.

3.2. Recursive normalized correlation and clustering

Our main contribution attempts to discriminate harmonic and percussive sounds only analyzing the activations H . The basic idea is to classify automatically those activations that exhibit rhythmic and non-rhythmic patterns. We assume that a rhythmic pattern models repetitive events typically associated with percussive sounds. A non-rhythmic pattern is assumed as a non-repetitive event typically shown by harmonic or vocal sounds.

We develop a recursive process, based on the normalized unbiased correlation $\tilde{R}^L(\tau)$ with order L , to identify rhythmic and non-rhythmic patterns. The normalized unbiased correlation $\tilde{R}^L(\tau)$ is computed using as input the signal $I(t)$ as shown in eq (7). We define the order L as the number of times that the normalized unbiased correlation is computed using the recursive process. In this manner, $I(t)=H(t)$ for $L=0$, $I(t)=\tilde{R}^0(\tau)$ for $L=1$, $I(t)=\tilde{R}^1(\tau)$ for $L=2$, etc. In a recursive way, the output of the current order L will be the input of the next order $L+1$. Specifically, the normalized unbiased correlation $\tilde{R}^L(\tau)$ is computed in eq. (8),

$$R^L(\tau) = \frac{1}{T-\tau} \sum_{t=0}^{T-1-\tau} I(t)I(t+\tau), \tau = 0, 1, 2, \dots, T-1 \quad (7)$$

$$\tilde{R}^L(\tau) = \frac{R^L(\tau)}{\|R^L(\tau)\|_2} \quad (8)$$

The analysis of $\tilde{R}^L(\tau)$ indicates that $\tilde{R}^4(\tau)$ provides reliable information to discriminate harmonic and percussive sounds as can be observed in Figure 2 and Figure 4. Figure 2 and Figure 4 show the matrix H of activations of a music excerpt composed of harmonic and percussive sounds. It can be observed that the components 4, 5, 9, 14 and 17 of Figure 2 and the components 14, 15, 17 and 18 of Figure 4 represents predominant percussive sounds modeled by rhythmic patterns. Both Figure 3 and Figure 5 show that $\tilde{R}^L(\tau)$ is still showing a set of peaks even as the order L increases when a percussive sound is analyzed. However, this does not occur when analyzing harmonic sounds since these tend to be represented using only one valley in a half-cycle waveform when the order L is increased. It can be observed that $\tilde{R}^4(\tau)$ is optimal (see section 4.4) to discriminate between percussive and harmonic sounds because $\tilde{R}^4(\tau)$ clearly shows a set of peaks considering percussive sounds and only one valley in a half-cycle waveform considering harmonic sounds. As a result, $\tilde{R}^{L>4}(\tau)$ could model harmonic sounds as percussive sounds because $\tilde{R}^{L>4}(\tau)$ can represent harmonic sounds with more than one peak as shown in Figures 3(b), 3(d), 3(f) and 3(h) and Figures 5(b), 5(d), 5(f) and 5(h). However, $\tilde{R}^{L>4}(\tau)$ could model percussive sounds as harmonic sounds because $\tilde{R}^{L>4}(\tau)$ tends to remove most of the peaks as shown in Figure 3(k) and Figure 3(m) and Figure 5(k) and Figure 5(m).

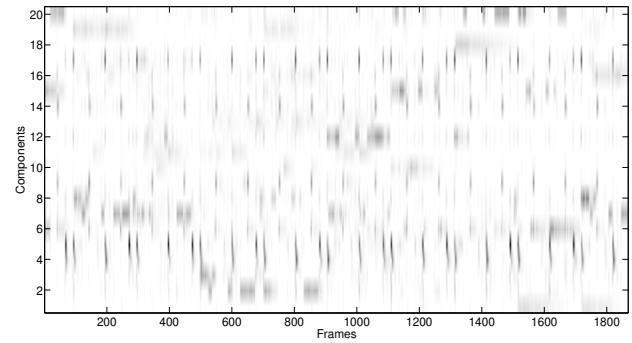


Figure 2: Activations H from the excerpt 'Hotel California' (Table 2), using $K=20$ components.

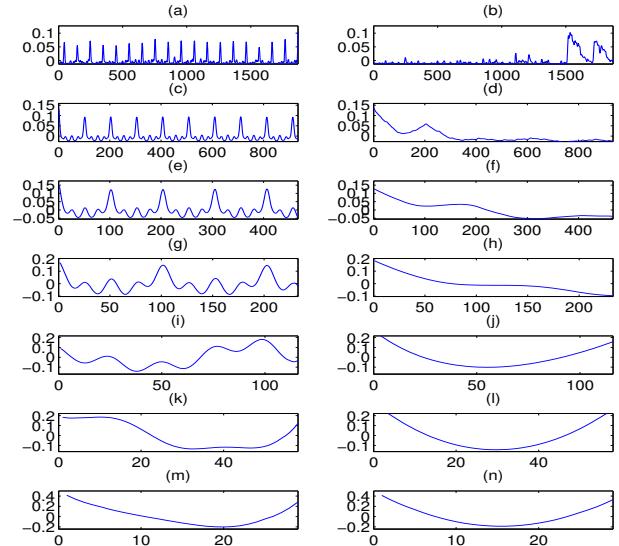


Figure 3: The column of the left represents the percussive component 14 of Figure 2: (a) $\tilde{R}^0(\tau)$, (c) $\tilde{R}^1(\tau)$, (e) $\tilde{R}^2(\tau)$, (g) $\tilde{R}^3(\tau)$, (i) $\tilde{R}^4(\tau)$, (k) $\tilde{R}^5(\tau)$, (m) $\tilde{R}^6(\tau)$. The column of the right represents the harmonic component 1 of Figure 2: (b) $\tilde{R}^0(\tau)$, (d) $\tilde{R}^1(\tau)$, (f) $\tilde{R}^2(\tau)$, (h) $\tilde{R}^3(\tau)$, (j) $\tilde{R}^4(\tau)$, (l) $\tilde{R}^5(\tau)$, (n) $\tilde{R}^6(\tau)$.

Based on the above observation, we use $\tilde{R}^4(\tau)$ as the basis for automatically discriminating between percussive and harmonic sounds. A component, obtained from NMF decomposition, is classified as percussive if a set of N_p or more peaks are found at the output of the $\tilde{R}^4(\tau)$. Preliminary results indicated that the best separation performance was obtained when $N_p \geq 2$. In any other case, a component is classified as harmonic.

We have developed two approaches based on the classification of the rhythmic activations as shown in Figure 1. In the first approach called Proposed_A, \hat{W}_{p_A} and \hat{H}_{p_A} represent the estimated bases and activations classified as percussive. However, \hat{W}_{h_A} and \hat{H}_{h_A} represent the estimated bases and activations clas-

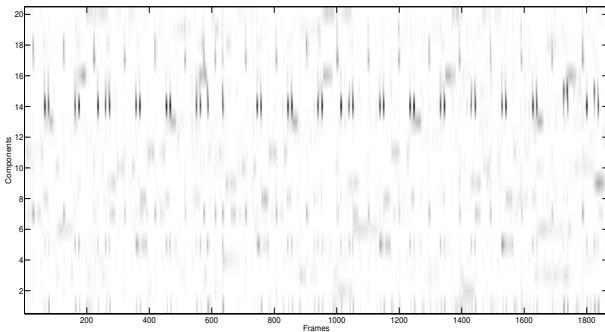


Figure 4: Activations H from the excerpt 'So lonely' (Table 1), using $K=20$ components.

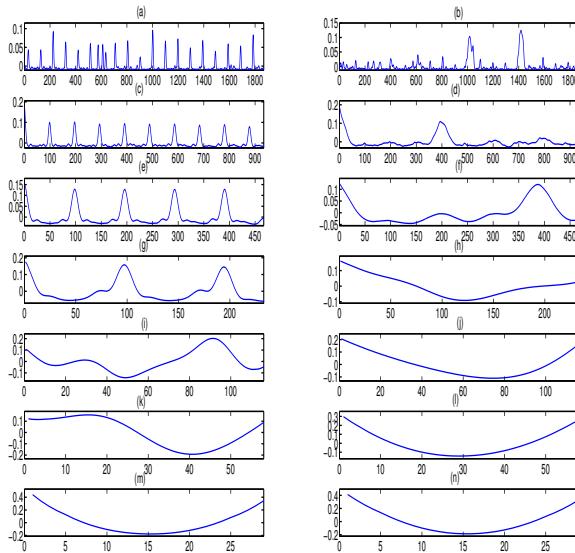


Figure 5: The column of the left represents the percussive component 17 of Figure 4: (a) $\tilde{R}^0(\tau)$, (c) $\tilde{R}^1(\tau)$, (e) $\tilde{R}^2(\tau)$, (g) $\tilde{R}^3(\tau)$, (i) $\tilde{R}^4(\tau)$, (k) $\tilde{R}^5(\tau)$, (m) $\tilde{R}^6(\tau)$. The column of the right represents the harmonic component 2 of Figure 4: (b) $\tilde{R}^0(\tau)$, (d) $\tilde{R}^1(\tau)$, (f) $\tilde{R}^2(\tau)$, (h) $\tilde{R}^3(\tau)$, (j) $\tilde{R}^4(\tau)$, (l) $\tilde{R}^5(\tau)$, (n) $\tilde{R}^6(\tau)$.

sified as harmonic. Specifically, $\hat{W}_p = \hat{W}_{p_A}$ and $\hat{H}_p = \hat{H}_{p_A}$, $\hat{W}_h = \hat{W}_{h_A}$, $\hat{H}_h = \hat{H}_{h_A}$

In the second approach called Proposed_B, a semi-supervised NMF, based on the Kullback-Liebler divergence, is used. The estimated percussive bases \hat{W}_{p_B} are fixed to $\hat{W}_p = \hat{W}_{p_B} = \hat{W}_{p_A}$ and not updated. However, the estimated harmonic bases and the estimated percussive and harmonic activations are initialized $\hat{W}_h = \hat{W}_{h_B} = \hat{W}_{h_A}$, $\hat{H}_p = \hat{H}_{p_B} = \hat{H}_{p_A}$ and $\hat{H}_h = \hat{H}_{h_B} = \hat{H}_{h_A}$ and updated in the factorization process.

3.3. Reconstruction and Wiener filtering

Performing each approach, the separated percussive and harmonic signals $\hat{x}_p(t)$, $\hat{x}_h(t)$ are synthesized using the magnitude spectrogram, that is, $\hat{X}_p = \hat{W}_p \hat{H}_p$ and $\hat{X}_h = \hat{W}_h \hat{H}_h$.

If the power spectral density (PSD) of the estimated signals are denoted as $|\hat{X}_p|^2$ and $|\hat{X}_h|^2$, respectively, then each ideally estimated source $\hat{x}_p(t)$ or $\hat{x}_h(t)$ can be estimated from the mixture $x(t)$ using a generalized time-frequency mask over the STFT domain. To ensure that the reconstruction process is conservative, a Wiener filtering has been used as in [5]. In this manner, a percussive or harmonic Wiener mask represents the relative percussive or harmonic energy contribution of each type of sound with respect to the energy of the mixture defined as follows,

$$\hat{X}_p = \left(|\hat{X}_p|^2 \oslash \left(|\hat{X}_p|^2 + |\hat{X}_h|^2 \right) \right) \odot X \quad (9)$$

$$\hat{X}_h = \left(|\hat{X}_h|^2 \oslash \left(|\hat{X}_p|^2 + |\hat{X}_h|^2 \right) \right) \odot X \quad (10)$$

The estimated percussive and harmonic signals $\hat{x}_p(t)$, $\hat{x}_h(t)$ are obtained computing the inverse overlap-add STFT from the final estimated percussive and harmonic magnitude spectrograms \hat{X}_p , \hat{X}_h using the phase spectrogram of the mixture.

4. EXPERIMENTAL RESULTS

4.1. Data and metrics

Evaluation has been performed using two databases *DBO* and *DBT*. Both databases are composed of single-channel real-world music excerpts taken from the Guitar Hero game [14] [15] as can be seen in Table 1 and Table 2. Each excerpt has a duration about 30 seconds and it was converted from stereo to mono and sampled at $f_s = 16$ kHz.

The database *DBO* has been used in the optimization and the database *DBT* has been used in testing. Note that the database used in the optimization is not the same as that used in the testing to validate the results. The subscript ph is related to percussive and harmonic instrumental sounds without adding the original vocal sounds. In this case, each percussive signal $x_p(t)$ is composed of percussive sounds (drums) and each harmonic signal $x_h(t)$ is composed of harmonic instrumental sounds. However, the subscript phv is related to percussive, harmonic and vocal sounds. As a result, each percussive signal $x_p(t)$ is composed of percussive sounds (drums) and each harmonic signal $x_h(t)$ is composed of harmonic and vocal sounds.

Table 1: Title and artist of the excerpts of the databases *DBO_{ph}* and *DBO_{phv}*

TITLE	ARTIST
Are you gonna go my way	Lenny Kravitz
Feel the pain	Dinosaur Jr
Kick out the James	MCS's Wayne Krame
One way or another	Blondie
In my place	Coldplay
Livin' on a prayer	Bon Jovi
No one to depend on	Santana
So lonely	The police
Song 2	Blur

Table 2: Title and artist of the excerpts of the databases DBT_{ph} and DBT_{phv}

TITLE	ARTIST
Hollywood Nights	Bob Seger & The Silver Bullet Band
Hotel California	Eagles
Hurts So Good	John Mellencamp
La Bamba	Los Lobos
Make It Wit Chu	Queens Of The Stone Age
Ring of Fire	Johnny Cash
Rooftops	Lost prophets
Sultans of Swing	Dire Straits
Under Pressure	Queen

The assessment of the performance of the proposed method has been performed using the metrics Source to Distortion Ratio (SDR), Source to Interference Ratio (SIR) and Source to Artifacts Ratio (SAR) [16] [17] which are widely used in the field of sound source separation. Specifically, SDR provides information on the overall quality of the separation process. SIR is a measure of the presence of percussive sounds in the harmonic signal and vice versa. SAR provides information on the artifacts in the separated signal from separation and/or resynthesis. Higher values of these ratios indicate better separation quality. More details can be found in [16].

4.2. Setup

An initial evaluation has been performed taking into account the computation of the STFT in order to optimize the frame size $N = (1024, 2048 \text{ and } 4096 \text{ samples})$ using the sampling rate f_s previously mentioned. Preliminary results indicated that the best separation performance was achieved using $(N,J)=(2048,256)$ samples.

A random initialization of the matrices W and H was used and the convergence of the NMF decomposition was evaluated using $Niter$ iterations. Due to the fact that standard NMF is not guaranteed to find a global minimum, the performance of the proposed method depends on the initial values W and H obtaining different results. For this reason, we have repeated three times for each excerpt and the results in the paper are averaged values.

4.3. State-of-the-art methods

Two reference state-of-the-art percussive and harmonic separation methods have been used to evaluate the proposed method: HPSS [1] and MFS [2]. These were both implemented for the evaluation of this paper. The ideal separation, called Oracle, is provided to better compare the quality of the proposed methods. The Oracle separation has been computed using the ideal soft masks extracted from the original percussive and harmonic signals applied to the input mixture.

4.4. Optimization

An optimization of the parameters K , $Niter$ and L is performed in the databases DBO_{ph} and DBO_{phv} as shown in Figure 6 and Figure 7. For this purpose, a hyperparametric analysis is applied to each parameter of the proposed method as occurs in [5]. In this work, $K=(5, 10, 20, 30, 50, 100, 150, 200)$, $Niter=(10, 20, 30, 50, 100, 150)$ and $L=(0, 1, 2, 3, 4, 5, 6)$.

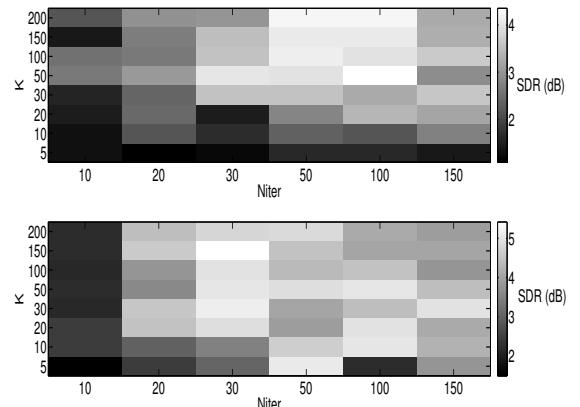


Figure 6: Optimization of the parameters K and $Niter$ (using $L=4$) jointly averaging percussive and harmonic SDR. (top) evaluating the database DBO_{ph} , (bottom) evaluating the database DBO_{phv}

Figure 6(top) shows that the parameters $K=50$ and $Niter=100$ maximizes the average percussive and harmonic SDR only considering mixtures composed of percussive and harmonic sounds without vocal sounds. It can be observed that using $K<20$ and $Niter<30$ provides the worst separation performance of the method. Results suggest that using a small number of components does not allow sufficient separation of repetitive and non-repetitive elements, thereby causing poor separation quality. Further, a small number of iterations does not allow to converge the NMF decomposition.

Figure 6(bottom) shows that $K=150$ and $Niter=30$ maximize the average percussive and harmonic SDR only considering mixtures composed of percussive, harmonic and vocal sounds. Figure 6(bottom) shows that a higher number of components is necessary to obtain the highest SDR. The effect of adding vocal sounds indicates the presence of a higher variety of spectral patterns so the proposed method needs a higher number of components to represent the mixture adequately..

Using the optimal parameters in the databases DBO_{ph} and DBO_{phv} , the optimization of the parameter L is shown in Figure 7. Comparing the separation performance of the parameter L , $L=4$ provides the highest robustness to discriminate harmonic and percussive sounds evaluating audio mixtures with or without vocal sounds. The principal reason for this is that $\tilde{R}^4(\tau)$ retains sufficient peaks in the repetitive basis functions to allow discrimination from the non-repetitive basis functions which tend to have a single valley at $L=4$.

4.5. Results

Figure 8(a) and Figure 8(b) show SDR, SIR and SAR results evaluating the databases DBT_{ph} and DBT_{phv} for the proposed method and the reference state-of-the-art methods. Each box represents nine data points, one for each excerpt of the test database. The lower and upper lines of each box show the 25th and 75th percentiles for the database. The line in the middle of each box represents the median value of the dataset. The left (blue), center (red) and right (black) boxes are related to the estimated percussive, har-

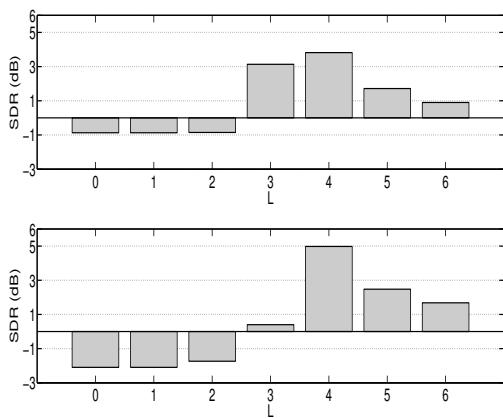


Figure 7: Optimization of the parameters L jointly averaging percussive and harmonic SDR. (top) evaluating the database DBO_{ph} using the optimal parameters $K=50$ and $Niter=100$; (bottom) evaluating the database DBO_{phv} using the optimal parameters $K=150$ and $Niter=30$

monic and average between percussive and harmonic signals.

Figure 8(a) indicates that MFS and the proposed method outperform the separation performance of HPSS in percussive and harmonic SDR but HPSS can be considered as competitive method. Although MFS and Proposed_B exhibit a similar behavior in percussive SDR, MFS ($SDR = 5.9dB$) is slightly better than Proposed_B ($SDR = 5.0dB$) in harmonic SDR. However, Proposed_B ($SIR = 9.8dB$) is slightly better than MFS ($SIR = 9.1dB$) considering the average between percussive and harmonic SIR. HPSS obtains the highest percussive SIR at the expense of introducing a high amount of artifacts, as shown by having the worst percussive and harmonic SAR. This does not occur with either MFS or the proposed methods. Further, HPSS loses most of the transients associated with the beginning of the harmonic sounds, thereby obtaining low harmonic SDR compared to the other methods. Although Proposed_A and Proposed_B show similar separation performance, it seems that Proposed_B is slightly better than Proposed_A. This can be explained because the semi-supervised NMF, initialized with the harmonic bases from Proposed_A, tends to converge to a better solution, obtaining a higher quality reconstruction of the estimated harmonic signals. Informal listening tests suggest that the proposed methods achieves higher polyphonic richness of the harmonic sounds compared to HPSS and MFS because it is able to capture most of the onsets of the harmonic sounds, such as onsets played by bass guitar or lead guitar. Nevertheless, a weakness of the proposed method is that it classifies as a percussive sound those harmonic sounds, e.g., bass guitar, that exhibit a very strong rhythmic pattern, especially if the bass guitar is playing a repeated note which has been factorized together with a part of a percussive sound in the same NMF component.

Figure 8(b) shows the separation performance for the database DBT_{phv} . Comparing with Figure 8(a), it can be observed that the addition of vocal sounds significantly worsens the SDR, SIR and SAR obtained by HPSS and MFS. However, the proposed methods still perform strongly, even with the addition of vocals. Results indicate that Proposed_A and Proposed_B show similar or even better SDR, SIR and SAR results compared to their separa-

tion performance in Figure 8(a). The evaluation metrics indicate that the proposed methods offer a more robust performance, especially when comparing their percussive and harmonic SDR and SIR results. Specifically, the proposed methods remove most of the vocal sounds from the estimated percussive signal, while HPSS and MFS do not. In some excerpts, this fact implies that the vocal sounds are unintelligible in the estimated harmonic signal. The promising behavior of the proposed methods with respect to vocal sounds can be explained by the fact that vocal sounds tend to exhibit less repetition, both in terms of melody and in terms of modulations than other sound sources.

5. CONCLUSIONS AND FUTURE WORK

This paper presents a novel approach for separating harmonic and percussive sounds using only temporal information extracted from activations by means of non-negative matrix factorization. The basic idea is to classify automatically those activations that exhibit rhythmic and non-rhythmic patterns. We assume that a rhythmic pattern models repetitive events which are typically associated with percussive sounds. However, a non-rhythmic pattern models non-repetitive events typically associated with harmonic or vocal sounds. Some of the advantages of this approach are (i) simplicity; (ii) no prior information about the spectral content of the musical instruments and (iii) no prior training.

Evaluating instrumental mixtures without vocals, results indicate that the proposed methods obtain promising audio separation. The performance of Proposed_B is slightly better than Proposed_A because the update of the semi-supervised NMF allows convergence to a better solution which provides higher quality reconstruction of the estimated harmonic signals. Moreover, our approach obtains higher polyphonic richness of the harmonic sounds because it captures most of the onsets of the harmonic sounds. However, a weakness of the proposed method is that highly repetitive harmonic instruments can occasionally be classified as percussive.

Evaluating instrumental mixtures containing vocals, results show that the proposed method gives a more robust performance both in percussive and harmonic SDR and SIR compared to the reference state-of-the-art methods. The proposed method extracts most of the vocal sounds from the estimated percussive signal unlike the other reference state-of-the-art methods evaluated.

Future work will be focused on three directions. Firstly, we will try to improve the audio quality of the estimated sources using another time-frequency representation that provides a better resolution in the low frequency bands. Secondly, we will attempt to remove residual harmonic sounds that have been factorized in percussive NMF components. Thirdly, we will address how to separate harmonic sounds, e.g. bass guitar, that show temporally repetitive characteristics from percussive sounds which have been factorized in the same NMF component, by looking into other ways of extracting the periodicity of the activations.

6. REFERENCES

- [1] N. Ono, K. Miyamoto, J. Le Roux, H. Kameoka, and S. Sagayama, “Separation of a monaural audio signal into harmonic/percussive components by complementary diffusion on spectrogram,” in *Proceedings of the European Signal Processing Conference (EUSIPCO)*, 2008, pp. 25–29.

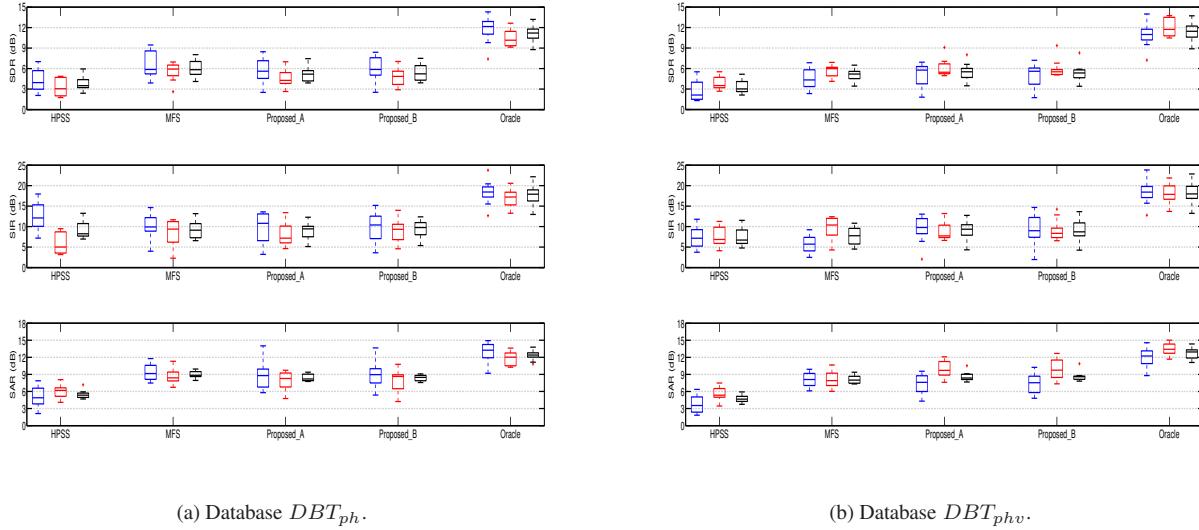


Figure 8: SDR, SIR and SAR results of the proposed methods and the state-of-the-art methods. The left (blue), center (red) and right (black) boxes are related to the estimated percussive, harmonic and average between percussive and harmonic signals.

- [2] D. Fitzgerald, “Harmonic-percussive separation using median filtering,” in *Proceedings of Digital Audio Effects (DAFx)*, 2010.
- [3] J. Yoo, M. Kim, K. Kang, and S. Choi, “Nonnegative matrix partial co-factorization for drum source separation,” in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 2010, pp. 1942–1945.
- [4] M. Kim, J. Yoo, K. Kang, and S. Choi, “Blind rhythmic source separation: Nonnegativity and repeatability,” in *IEEE International Conference on Acoustics Speech and Signal Processing (ICASSP)*, 2010.
- [5] F. Canadas, P. Vera, N. Ruiz, J. Carabias, and P. Cabanas, “Percussive-harmonic sound separation by non-negative matrix factorization with smoothness-sparseness constraints,” *Journal on Audio, Speech, and Music Processing*, vol. 2014, no. 26, pp. 1–17, 2014.
- [6] D. Fitzgerald, A. Liukus, Z. Rafi, B. Pardo, and L. Daudet, “Harmonic-percussive separation using kernel additive modelling,” in *25th IET Irish Signals and Systems Conference*, 2014.
- [7] J. Driedger, M. Muller, and S. Disch, “Extending harmonic-percussive separation of audio signals,” in *15th International Society for Music Information Retrieval Conference (ISMIR)*, 2014.
- [8] J. Park and K. Lee, “Harmonic-percussive source separation using harmonicity and sparsity constraints,” in *16th International Society for Music Information Retrieval Conference (ISMIR)*, 2015.
- [9] F. Canadas-Quesada, P. Vera-Candeas, N. Ruiz-Reyes, A. Munoz-Montoro, and F. Bris-Penalver, “A method to separate musical percussive sounds using chroma spectral flatness,” in *The First International Conference on Advances in Signal, Image and Video Processing (SIGNAL)*, 2016.
- [10] D. Lee and S. Seung, “Algorithms for non-negative matrix factorization,” in *Proceedings of Advances in Neural Inf. Process. System*, 2000, pp. 556–562.
- [11] S. Raczyński, N. Ono, and S. Sagayama, “Multipitch analysis with harmonic nonnegative matrix approximation,” in *8th International Conference on Music Information Retrieval (ISMIR)*, 2007.
- [12] C. Févotte, N. Bertin, and J. Durrieu, “Nonnegative matrix factorization with the itakura-saito divergence with application to music analysis,” *Neural Computation*, vol. 21, no. 3, pp. 793–830, 2009.
- [13] B. Zhu, W. Li, R. Li, and X. Xue, “Multi-stage non-negative matrix factorization for monaural singing voice separation,” *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 21, no. 10, pp. 2096–2107, 2013.
- [14] Activision Neversoft, “Guitar hero 5,” <http://gh5.guitarhero.com>, September 2009.
- [15] Activision Neversoft, “Guitar hero world tour,” <http://worldtour.guitarhero.com/us/>, November 2008.
- [16] C. Févotte, R. Gribonval, and E. Vincent, “Bss_eval toolbox user guide - revision 2.0,” in *Technical report 1706, IRISA*, 2005.
- [17] E. Vincent, C. Févotte, and R. Gribonval, “Performance measurement in blind audio source separation,” *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 14, no. 4, pp. 1462–1469, 2006.

ITERATIVE STRUCTURED SHRINKAGE ALGORITHMS FOR STATIONARY/TRANSIENT AUDIO SEPARATION

Kai Siedenburg* and Simon Doclo

Dept. of Medical Physics and Acoustics
Cluster of Excellence Hearing4All
University of Oldenburg, Germany
kai.siedenburg@uni-oldenburg.de
simon.doclo@uni-oldenburg.de

ABSTRACT

In this paper, we present novel strategies for stationary/transient signal separation in audio signals in order to exploit the basic observation that stationary components are sparse in frequency and persistent over time whereas transients are sparse in time and persistent across frequency. We utilize a multi-resolution STFT approach which allows to define structured shrinkage operators to tune into the characteristic spectrotemporal shapes of the stationary and transient signal layers. Structure is incorporated by considering the energy of time-frequency neighbourhoods or modulation spectrum regions instead of individual STFT coefficients, and shrinkage operators are employed in a dual-layered Iterated Shrinkage/Thresholding Algorithm (ISTA) framework. We further propose a novel iterative scheme, *Iterative Cross-Shrinkage* (ICS). In experiments using artificial test signals, ICS clearly outperforms the dual-layered ISTA and yields particularly good results in conjunction with a dynamic update of the shrinkage thresholds. The application of the novel algorithms to recordings from acoustic musical instruments provides perceptually convincing separation of transients.

1. INTRODUCTION

Among the primitives that constitute music signals, quasi-stationary sinusoidal components and short-lived transients are of prime importance. This becomes intuitively clear when considering spectrograms of music signals, which oftentimes feature horizontal and vertical lines, corresponding to stationary components that are sparse in frequency and persistent over time and transient components that are sparse in time and persistent over frequency. Whereas modeling sinusoidal components is a well-established field [1], transient estimation as such remains relatively unexplored, despite its numerous applications. Examples include audio restoration where short clicks and crackles must be removed from signals [2], beat tracking where the availability of a transient layer may improve onset detection algorithms [3], or psychoacoustics where robust transient separation could allow for refined investigations into the role of acoustic features in musical timbre perception [4]. The goal of this study is to exploit the distinctive properties of *sparsity* and *persistence* in order to propose robust schemes for stationary/transient separation.

It is important to note that stationary/transient separation is a different if not more fine-grained problem than drum sepa-

tion or harmonic/percussive separation [5, 6]. Although sounds from drums and percussive instruments are mostly impulsively excited and often inharmonic, percussive sounds may comprise components that extend over similar time scales as those of non-percussive musical instruments (think of the sustained and tonal portions of a snare drum or an open bass-drum sound). This distinction also illustrates the problem that transients are notoriously hard to define in semantic terms, because defining features such as short-livedness and stochastic nature can be easily contested in the context of complex audio mixtures. Consequently, ground truth for the stationary/transient separation task is only available for synthesized test signals.

In order to individually characterize stationary and transient components, our approach is related to several studies using *multi-layered* (or *hybrid*) audio representations that decompose the signal with at least two distinct dictionaries [7]. Early research used orthogonal bases [8]. More recently, a combination of a Modified Discrete Cosine Transform (MDCT) and a wavelet basis was proposed in [9, 10, 11], and Févotte and colleagues further modelled dependencies between coefficients of dual-layered MDCT expansions using a Bayesian framework for the simultaneous estimation of both layers [12]. Our current approach has the same goals as the aforementioned work, utilizing sparsity and inter-coefficient dependencies, but takes a different formal pathway. Here we follow up on the well-known Iterative Shrinkage/Thresholding Algorithm (ISTA) for solving ℓ_1 -regularized minimization problems [13], generalized to multi-frame mixed-norm regularization in [14, 15]. Further work extended the involved shrinkage operators with neighborhood-weighting in order to take into account the correlation between adjacent time-frequency coefficients [16, 17]. However, the structured shrinkage framework has not yet been applied to dual-layer signal decomposition with redundant STFT dictionaries.

The goal of the current study is to explore several strategies for structured shrinkage as part of a dual-layered framework with STFT dictionaries of different resolutions. Within each layer, the structured shrinkage operators are tailored towards the distinct spectrotemporal orientations of the stationary and transient components. In order to increase separation robustness, we further propose a novel iteration scheme and update rule for the threshold parameters. In Sec. 2, we provide a framework that allows us to formulate the structured shrinkage operators. Iteration rules are described in Sec. 3, before a comprehensive evaluation of the algorithmic variants using both artificial test signals as well as recordings is provided in Sec. 4 and Sec. 5.

* This work was supported by the European Union's Seventh Framework Programme (FP7/2007-2013) under Grant ITN-GA-2012-316969 and by a postdoc grant from the University of Oldenburg.

2. STRUCTURED SHRINKAGE

This section provides a background on structured shrinkage. We outline the basic signal model in Sec. 2.1, before Sec. 2.2 reviews the idea of extending shrinkage operators with neighbourhood weighting, and Sec. 2.3 reformulates this as an operation in the modulation domain.

2.1. Formal framework

We assume the observed time-domain audio signal $\mathbf{y} \in \mathbb{R}^M$ of length M to be an additive combination of a stationary layer \mathbf{s}_S and a transient layer \mathbf{s}_T . We further posit that each layer can be sparsely represented using appropriately chosen time-frequency dictionaries Φ and Ψ (practically realized below via STFTs with long and short analysis window lengths, respectively). That is, the layers are of the form $\mathbf{s}_S = \Phi\boldsymbol{\alpha}$ and $\mathbf{s}_T = \Psi\boldsymbol{\beta}$ with only few non-zero elements in the coefficient vectors $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$. Specifically, the layer \mathbf{s}_S (analogous for \mathbf{s}_T) can be written as

$$\mathbf{s}_S = \Phi\boldsymbol{\alpha} = \sum_{\gamma} \alpha_{\gamma} \varphi_{\gamma} \quad (1)$$

where for the sake of notational convenience $\gamma = (k, l)$ denotes a double labelling with $k = 1, \dots, K$ and $l = 1, \dots, L$ as frequency and time indices, respectively. The collection of time-localized atoms $\varphi_{\gamma} \in \mathbb{R}^M$ constitute the dictionary $\Phi \in \mathbb{R}^{(M, K \times L)}$, and α_{γ} are the STFT expansion coefficients. Here we use regular STFTs Φ and Ψ with perfect reconstruction, corresponding to one form of tight Gabor dictionaries [18].

Because there is no access to the separate layers, the general problem is to estimate the coefficients $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ from an additive mixture with Gaussian white noise \mathbf{e} (e.g., corresponding to measurement error):

$$\mathbf{y} = \mathbf{s}_S + \mathbf{s}_T + \mathbf{e} = \Phi\boldsymbol{\alpha} + \Psi\boldsymbol{\beta} + \mathbf{e}, \quad (2)$$

which can be stated as a dual-layer sparse regression problem [14],

$$\min_{\boldsymbol{\alpha}, \boldsymbol{\beta}} \left\{ \frac{1}{2} \|\mathbf{y} - \Phi\boldsymbol{\alpha} - \Psi\boldsymbol{\beta}\|_2^2 + \lambda \|\boldsymbol{\alpha}\|_1 + \mu \|\boldsymbol{\beta}\|_1 \right\}, \quad (3)$$

with sparsity parameters $\lambda, \mu > 0$. The solution of this problem is approximated by the *Iterative Shrinkage/Thresholding Algorithm* (ISTA) [13, 19], and more specifically its multi-layered extension [14, 15]:

$$\begin{cases} \boldsymbol{\alpha}^{(n+1)} = \mathbb{S}_{\lambda}(\boldsymbol{\alpha}^{(n)} - \Phi^*(\mathbf{y} - \Phi\boldsymbol{\alpha}^{(n)} - \Psi\boldsymbol{\beta}^{(n)})) \\ \boldsymbol{\beta}^{(n+1)} = \mathbb{S}_{\mu}(\boldsymbol{\beta}^{(n)} - \Psi^*(\mathbf{y} - \Phi\boldsymbol{\alpha}^{(n)} - \Psi\boldsymbol{\beta}^{(n)})), \end{cases} \quad (4)$$

with iteration index n and initializations $\boldsymbol{\alpha}^{(0)} = \boldsymbol{\beta}^{(0)} = 0$. Here and in the following, the notation $(x)_+ = \max(x, 0)$ denotes the positive part of any $x \in \mathbb{R}$ and all operations are understood component-wise, i.e., per time-frequency index $\gamma = (k, l)$. For a complex-valued vector $\boldsymbol{\alpha}$, the operator \mathbb{S} then refers to a shrinkage operation that is usually called *soft-thresholding*,

$$\mathbb{S}_{\lambda}(\boldsymbol{\alpha}) = \begin{cases} e^{i \arg \boldsymbol{\alpha}} (|\boldsymbol{\alpha}| - \lambda) : |\boldsymbol{\alpha}| \geq \lambda \\ 0 : |\boldsymbol{\alpha}| < \lambda \end{cases} = \boldsymbol{\alpha} \left(1 - \frac{\lambda}{|\boldsymbol{\alpha}|} \right)_+, \quad (5)$$

Whereas the optimization problem in (3) together with ISTA in (4) provide a theoretical footing for sparse multilayer decomposition, this approach has several drawbacks in practical situations.

Most importantly, the independent handling of time-frequency coefficients does not utilize the full gamut of structure of stationary and transient layers. This is where the idea of *social sparsity* becomes useful.

2.2. Structured shrinkage via neighbourhoods

The approach of social sparsity extends classical shrinkage operators by the aspect of neighbourhood dependencies, yielding solutions of low computational cost that respect structural dependencies but are not strictly attached to known minimization functionals any more [17]. By generalizing the soft-thresholding operator in (5), we here focus on shrinkage operators of the form,

$$\mathbb{S}_{\lambda, *}(\boldsymbol{\alpha}) = \boldsymbol{\alpha} \left(1 - \left[\frac{\lambda}{\|\boldsymbol{\alpha}\|_*} \right]^\tau \right)_+ \quad (6)$$

The placeholder $*$ refers to a norm that allows to take into account neighbourhood structures and thus helps to orient the shrinkage operator towards stationary or transient components (e.g., by letting neighbourhoods extend across time for stationary components or frequency for transient components). By choosing a vector of non-negative time-frequency neighbourhood weights $\mathbf{w} = w_{\gamma, \gamma'}$, we can define the neighbourhood-based norm as

$$\|\boldsymbol{\alpha}\|_* = (\|\boldsymbol{\alpha}\|_*)_ \gamma = \sqrt{\sum_{\gamma'} w_{\gamma, \gamma'} |\alpha_{\gamma'}|^2} \quad (7)$$

In practice we use sliding neighbourhoods, i.e., $w_{\gamma, \gamma'} = w_{0, \gamma - \gamma'}$, such that the norm $\|\cdot\|_*$ can be efficiently computed via convolution:

$$\begin{aligned} \|\boldsymbol{\alpha}\|_*^2 &= \sum_{\gamma'} w_{\gamma, \gamma'} |\alpha_{\gamma'}|^2 \\ &= \sum_{\gamma'} w_{0, \gamma - \gamma'} |\alpha_{\gamma'}|^2 \\ &= (\mathbf{w} * |\boldsymbol{\alpha}|^2) \gamma \end{aligned} \quad (8)$$

The generic choice $\tau = 1$ leads to the *Windowed Group Lasso* [16], which constitutes a natural extension of the classic *Least Absolute Shrinkage/Selection Operator* (LASSO) [20], for which $\|\cdot\|_* = |\cdot|$. Here, we focus on $\tau = 2$, which withdraws less energy from the signal compared to $\tau = 1$ and has been called *empirical Wiener* operator [21] or non-negative garotte shrinkage [22]. For single-layered expansions, previous research has shown that both the inclusion of neighbourhoods that extend across a few coefficients in time and the choice of $\tau = 2$ significantly improve audio noise removal [23] and declipping [24].

2.3. Structured shrinkage via modulation-filtering

Instead of defining the neighbourhood weights directly, they can also be defined in terms of their effect on the modulation spectrum of the shrinkage operation in (6) [25]. Let \mathcal{F}_2 denote the two-dimensional discrete Fourier transform on $\mathbb{C}^{K \times L}$, then (8) can be directly reformulated as

$$\|\boldsymbol{\alpha}\|_*^2 = \left[\mathcal{F}_2^{-1} \left(\mathcal{F}_2(\mathbf{w}) \cdot \mathcal{F}_2(|\boldsymbol{\alpha}|^2) \right) \right] \quad (9)$$

The x-axis of the resulting modulation spectrum $\mathcal{F}_2(|\boldsymbol{\alpha}|^2)$ corresponds to temporal modulation measured in Hz, the y-axis to spectral modulation with unit cycles per Hz. This means the choice of the neighborhood \mathbf{w} is equivalent to the choice of desired temporal and spectral modulation frequencies to be captured by the

modulation filter $\mathbf{W} := \mathcal{F}_2(\mathbf{w})$. For example, the usage of a rectangular neighbourhood corresponds to modulation filtering with a two-dimensional sinc-function centered at zero, i.e., a form of modulation low-pass filtering.

We here follow previous suggestions [26] and use an additional log nonlinearity for computing the modulation spectrum:

$$\|\boldsymbol{\alpha}\|_{\sim} := \exp \left(\mathcal{F}_2^{-1} \left[\mathbf{W} \cdot \mathcal{F}_2 \left(\log(|\boldsymbol{\alpha}| + \kappa) \right) \right] \right) - \kappa \quad (10)$$

with a compression constant $\kappa = 1$. The log nonlinearity may be justified by its resemblance to cepstral analysis, potentially separating the contributions of a signal's source and filter into additive contributions [26]. In conclusion, instead of shrinking coefficients in dependence of their neighbourhood's energy, an alternative perspective is to shrink these coefficients according to the energy retained by the modulation filter.

3. ALGORITHMS FOR STATIONARY/TRANSIENT SEPARATION

3.1. Iterative shrinkage/thresholding and cross-shrinkage

The point of departure of this study was to use structured shrinkage operators in iterative schemes such as the multilayered ISTA (4) for stationary/transient separation. Although a fast version of this algorithm, FISTA, has been proposed in [19], we could not observe improvements over ISTA for the considered application, and thus here only report on the regular ISTA.

When using ISTA in practice, we often encountered the problem that the transient layer was swallowed by the stationary layer, unless tedious tuning of the thresholds λ and μ was undertaken. For that reason, we also explored a related scheme, which follows a simple rationale: Assuming the estimate of the stationary layer ($\boldsymbol{\alpha}$) is accurate, the residual $\mathbf{y} - \Phi\boldsymbol{\alpha} = \Psi\boldsymbol{\beta} + \mathbf{e}$ mainly comprises components from the transient layer ($\boldsymbol{\beta}$) and thus allows for a more precise estimation of $\boldsymbol{\beta}$ than from the mixture. Due to the iterative estimation from the residual of the respective alternate layer, this yields a novel scheme which we call *Iterative Cross-Shrinkage* (ICS):

$$\begin{cases} \boldsymbol{\alpha}^{(n+1)} &= \mathbb{S}_{\lambda}(\Phi^*(\mathbf{y} - \Psi\boldsymbol{\beta}^{(n)})) \\ \boldsymbol{\beta}^{(n+1)} &= \mathbb{S}_{\mu}(\Psi^*(\mathbf{y} - \Phi\boldsymbol{\alpha}^{(n)})) \end{cases} \quad (11)$$

with initializations $\boldsymbol{\alpha}^{(0)} = \boldsymbol{\beta}^{(0)} = 0$. Essentially, this corresponds to ISTA with zero contribution of the respective previous iterate of each layer (e.g., with $\boldsymbol{\alpha}^{(n)}$ set to zero in the estimation of $\boldsymbol{\alpha}^{(n+1)}$).

3.2. Choice of thresholds

The right selection of the thresholds λ and μ is critical in applications. In scenarios with strong additive noise, the optimal thresholds naturally depend on the noise level [27]. In the stationary/transient separation scenario, however, additive noise does not play a similarly as crucial role such that alternative strategies for selecting the thresholds can be sought. In addition to using fixed thresholds, we here explored so-called *warm-start* strategies [28, 24]. These strategies start out conservatively with relatively high thresholds which are successively reduced and thus allow for more liberal estimates towards the end.

In the first warm-start strategy, we chose $\lambda^{(n)} = \lambda$ and $\mu^{(n)} = \mu$ as piece-wise constant sequences that decreased after every 10th iteration. Specifically, the thresholds were set equal to the $P\%$

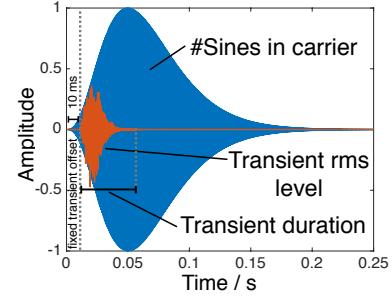


Figure 1: Exemplary test signal. Transients varied according to duration and level. Stationary parts varied by the number of sinusoidal components of their carrier.

quantiles q_P of the distribution of the magnitudes of the initial analysis coefficients of the respective layer, i.e., $\lambda^{(n)} = q_P(|\Phi^*\mathbf{y}|)$ and $\mu^{(n)} = q_P(|\Psi^*\mathbf{y}|)$. P was set equal to 99% and linearly decreased after every 10th iteration by around 2.11 percentage points, reaching 80% at iteration no. 91 (yielding 100 iterations in total). See Fig. 4 (right panel) for an example trajectory.

A shortcoming of this strategy is that it is based on an inherently imperfect estimate of the magnitude distribution of the expansion coefficients of individual layers, because $|\Phi^*\mathbf{y}| = |\Phi^*\mathbf{s}_S + \Phi^*\mathbf{s}_{ST}|$ obviously contains both layers. Say we were aiming to adjust the threshold λ according to the true underlying magnitude distribution $q_P(|\Phi^*\mathbf{s}_S|)$, it should be beneficial to update this threshold according to presumably more precise estimates of the individual layers that only arise at later iterations. For that reason, we also tested a second warm-start strategy where the thresholds were adjusted dynamically. That means, $\lambda^{(n)}$ and $\mu^{(n)}$ were updated after every 10th step according to the magnitude of the argument of the shrinkage operators in ISTA (4) and ICS (11). For ICS, for instance, this would correspond to $\lambda^{(n)} = q_P(|\Phi^*(\mathbf{y} - \Psi\boldsymbol{\beta}^{(n)})|)$. As before, P started at 99% and linearly decreased after every 10th iteration to reach 80% for the last 10 iterations (see Fig. 4).

4. EXPERIMENTS WITH SYNTHESIZED SOUNDS

In this section, we describe simulation experiments with signals for which the stationary and transient parts were artificially synthesized. We tested the ISTA and ICS algorithms in conjunction with three types of threshold selection strategies (cf., Sec. 3.2) and three ways of incorporating inter-coefficient structure (cf., Sec. 2.2 and 2.3).

4.1. Test stimuli and factors

We synthesized a set of fixed-frequency sinusoids and Gamma-shaped noise bursts as test components. For the stationary component, sinusoidal frequencies were randomly and uniformly chosen between 100 and 10,000 Hz and shaped with an Gamma-type amplitude envelope of 245 ms effective duration. The transient components were generated with white Gaussian noise, which was shaped by a much shorter Gamma-envelope. Specifically, we varied three factors (using four levels for each of them):

- i) The transient Gamma-shaped white noise bursts had effective durations between 4.9 and 49 ms.

- ii) The level of the transient relative to the stationary component was adjusted between -30 and 0 dB.
- iii) For the stationary component, harmonic tone complexes comprised 1 to 50 sinusoids (i.e., corresponding to the sparsity of the stationary signal).

The Gamma-envelope was of the form

$$e(t) = t^{(Q-1)} \exp(-2\pi b t),$$

where the order $Q = 4$ was fixed and t denotes time in seconds. The scale parameter b was set as $b = Q - 1/(2\pi\eta)$, where η is the underlying Gamma distribution's mode (or the tone's rise time). From this, it can be inferred that the effective duration of any resulting sound (with a threshold of -60 dB) amounts to 4.9η .

Transients were delayed by 10 ms in order to ensure even for short transients a significant overlap with the energy of the slower rising envelope of the stationary components (i.e., to prohibit the possibility of overly simplistic separation). The additive combination of the stationary and transient components was used in conjunction with a white Gaussian noise floor at a signal-to-noise ratio of 40 dB. Throughout all numerical simulations reported in this paper, audio signals were sampled at 44.1 kHz. Fig. 1 shows an illustration of the stationary and transient components of a test signal.

4.2. Algorithm settings

We chose Gabor dictionaries with a tight Hann (raised cosine) window and a hop size of a quarter of the window length. In order to capture stationary components, Φ was chosen with a window length of 2048 samples (46 ms). For capturing transient components, Ψ was chosen with a window length of 128 samples (3 ms). All simulations were performed with shrinkage exponent $\tau = 2$.

As outlined in Sec. 3.2, three strategies for choosing the thresholds $\lambda^{(n)}$ and $\mu^{(n)}$ were considered: i) A fixed threshold at the 80% quantile of each layer's initial analysis coefficients (denoted as *fix*), ii) a sequence of quantiles, linearly decreasing from the 99% quantile to the 80% quantile of the initial analysis coefficients (*quant*), and iii) dynamically decreasing thresholds (*dyn*).

The utility of exploiting inter-coefficient structure was investigated by comparing neighbourhood-based shrinkage (denoted as *Neigh.*) and modulation-based shrinkage (*Modul.*) to shrinkage with independent-coefficients (*Indep.*). For shrinkage of the stationary layer \mathbb{S}_λ , neighbourhoods comprised two coefficients forward and two coefficients backward of the centre coefficient. For shrinkage of the transient layer \mathbb{S}_μ , the neighbourhood extended for three coefficients above and three coefficient below the centre coefficient. For modulation-based shrinkage, we chose W as a separable two-dimensional Gaussian distribution centred at the origin of the modulation spectrum. In order to tune in on spectral information for shrinkage of the stationary layer, the Gaussian's standard deviations were set to (1, 0.1) for the spectral scale and temporal rate axes, respectively, and the distribution was evaluated across the range $[-1, 1] \times [-1, 1]$. For the transient layer, we chose the reverse settings (0.1, 1), mainly tuning in on temporal information.

4.3. Results

We measured the performance in terms of the estimation accuracy of the 100th iterate of the above presented algorithms using the

signal-to-distortion ratio (SDR). For the stationary layer \mathbf{s}_S , for instance, this corresponds to

$$\text{SDR}(\mathbf{s}_S, \Phi\hat{\alpha}) = 20 \log_{10} \left(\frac{\|\mathbf{s}_S\|}{\|\mathbf{s}_S - \Phi\hat{\alpha}\|} \right).$$

We present results in terms of the SDR improvement compared to the unprocessed signal, i.e., $\Delta\text{SDR} = \text{SDR}(\mathbf{s}_S, \Phi\hat{\alpha}) - \text{SDR}(\mathbf{s}_S, y)$.

Fig. 2 shows the mean ΔSDR values for all $2 \times 3 \times 3$ algorithmic variants, averaged across the three stimulus factors of transient duration, level, and number of sinusoids in the stationary carrier signal (each with four levels, such that every data point corresponds to a mean across 64 test signals). Both for the stationary and transient components, the ISTA and ICS methods yield similar performance in conjunction with fixed thresholds. At the same time, fixed thresholds yield negative ΔSDR values for the stationary layer, which indicates that this strategy has significant shortcomings. Whereas ISTA works best in conjunction with the quantile-based update of the thresholds and fails for the dynamic update, ICS seems to particularly profit from this latter strategy. Furthermore, ΔSDR is generally higher with neighbourhoods compared to the independent handling of coefficients and best performance is reached for modulation filtering.

Overall, the figure clearly illustrates the superior performance of ICS *dyn* with gains of around 10 dB SDR compared to the best-performing variant of ISTA (*quant*) for both stationary and transient layers. Compared to the initial reference algorithm—the dual-layered ISTA with independent coefficients originally proposed in [14, 15]—we were thus able to achieve improvements of more than 20 dB SDR.

In order to additionally compare our algorithm to an orthogonal transform, we used the dual-layered expansion of the Modified Discrete Cosine Transform, which also serves as a dual-layer decomposition example of the Large Time Frequency Analysis Toolbox [29]. This configuration achieved an average $\Delta\text{SDR}_S = -2.8$ and $\Delta\text{SDR}_T = 4.5$, thus markedly worse for transient estimation compared to the most rudimentary variant of our Gabor-dictionary-based ISTA approach with fixed thresholds and independent coefficients.

Fig. 3 depicts a more detailed picture of the performance of the three best-performing algorithmic variants: ISTA and ICS with quantile-based thresholds, and ICS with dynamic threshold update. These figures demonstrate that performance drops as the durations of the transients grow, which is expected, given that the differences in time-scale of stationary and transient components increasingly vanish. Yet, most algorithmic variants are still able to achieve a substantial SDR improvement even for the longest transients of 49 ms, which span more than a whole Gabor atom of the stationary layer (46 ms length). The dependency of ΔSDR on the transient level appears to be fairly linear. As expected, the biggest improvements of the stationary ΔSDR occur for conditions with strongest transients, and vice versa for the transient ΔSDR . Finally, when it comes to the number of sinusoidal components in the stationary carrier signal, that is, its sparsity, the dynamic ICS appears to be particularly robust, whereas the rest of the algorithms yields a sharp drop in performance already for four sinusoids.

Regarding the iterative behavior of the algorithms, Fig. 4 shows an example of the three best-performing algorithmic variants (using modulation filtering) across iterations as well as the corresponding evolution of thresholds (rightmost panel). It is clearly

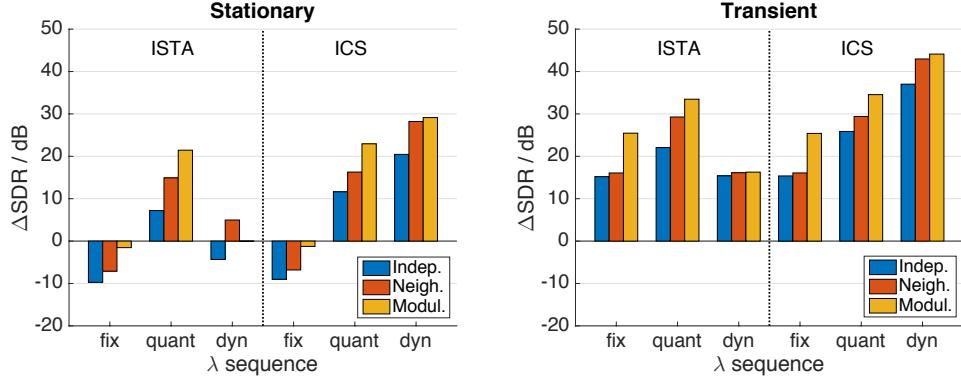


Figure 2: Mean SDRs improvement across all acoustic conditions for different algorithms (ISTA, ICS), different shrinkage operators (Indep., Neigh., Modul.), and different threshold update strategies (fix, quant, dyn).

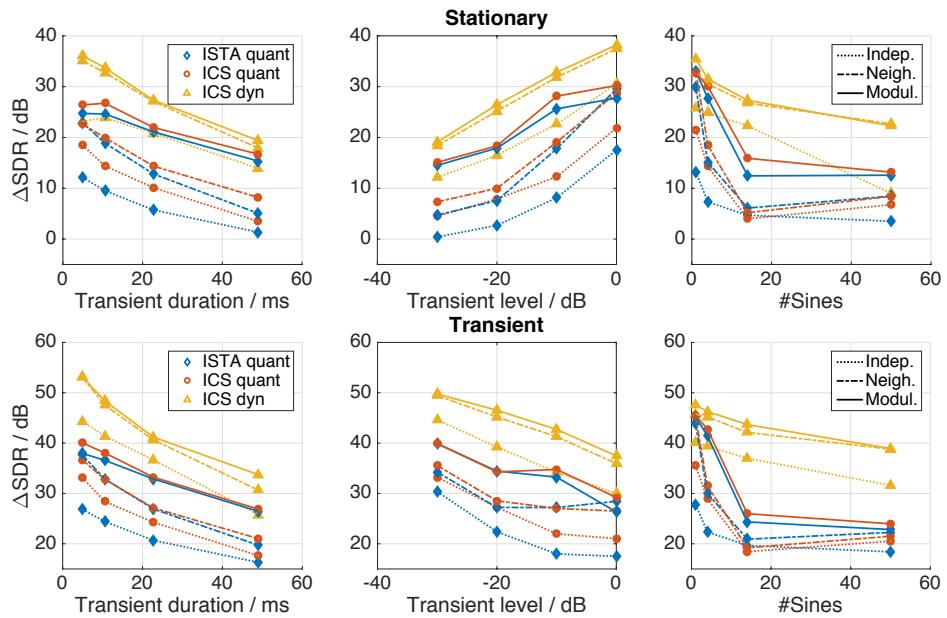


Figure 3: SDR improvement for estimates of stationary and transient components as a function of the algorithm type (indicated by symbols and line color, see left legend) and inter-coefficient structure (indicated by linetype, see right-hand-side legend).

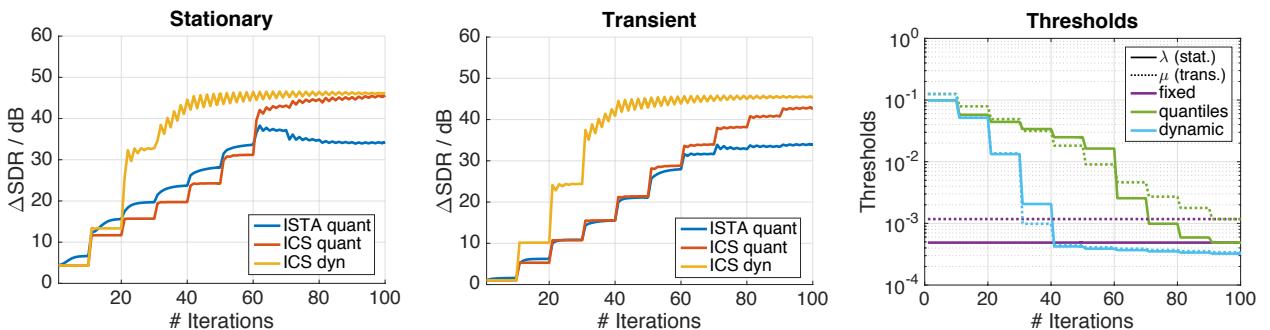


Figure 4: Exemplary iteration. Left and center panel: SDR improvement across 100 iterations for modulation filtering and operator types (see color legend). The sound's stationary part comprised 4 sinusoids and the transient was of 10 ms duration at an RMS level of 0 dB. Rightmost panel: Evolution of thresholds of stationary components (solid lines) and transient components (dotted lines). The threshold update is color-coded, see legend.

visible how after every 10 iterations, decreases of thresholds are accompanied by increases in Δ SDR. Notably, the ICS with dynamic threshold update appears to converge much faster, likely due to a steeper slope of thresholds across iterations for the dynamic update. Also note that the dynamic threshold update reaches the same absolute threshold values for both layers, which stands in contrast to the other two methods for which the 80% quantile of the stationary layer is much smaller compared to that of the transient layer.

Due to their rather heuristic nature, our findings motivate further mathematical and experimental inquiries into the formal roots of the proposed algorithms. For instance, it is currently unclear why ISTA appears to be incompatible with the dynamic threshold update whereas ICS profits substantially from it. It is also questionable whether this gain in performance (from ICS *quant* to ICS *dyn*) is due to the update's actual adaptivity or solely based on a steeper decrease of thresholds over iterations. Finally, it is left completely open whether it could be beneficial to replace the STFT dictionary with short window lengths by a wavelet basis, which may be better suited to account for the stochastic nature of transients [10]. On the other hand, the proposed algorithms appear to be robust enough already in order to be useful for stationary/transient separation in practical situations, as described in the next section.

5. EXAMPLES WITH RECORDED AUDIO

We considered natural recorded instrumental tones produced by a violoncello, a vibraphone, and a harpsichord. Each sound was of 500 ms duration, fundamental frequency 311 Hz, and of equal perceptual loudness as used in [30]. Due to its continuous mode of excitation, the violoncello is a quasi-harmonic sound without marked attack transient, yet with low-energy noise components stemming from the bow. As an impulsively excited sound, the vibraphone features strong attack transient at its onset. Interestingly, the harpsichord comprises one transient component at its onset, but also one at the release of its hopper while other sinusoidal components sustain. Fig. 5 presents the waveform of these three sounds in sequence, their spectrograms, as well as the separation provided by ICS with dynamic threshold update and modulation filtering.¹

For this example, the stationary layer of the ISTA algorithm swallows the transient layers, i.e., the separation fails. On the contrary, the ICS algorithm provides non-zero estimates of transients for all three sounds, even for the onset noise components of the cello bow. As visible in the figure, the vibraphone contains the strongest transient component, which is clearly separated from the remaining sinusoidal components. Finally, the two transients of the harpsichord sound are well separated, even though they fully overlap in time. The latter sound once again illustrates that separation performance is not at all relying on temporal separation, but on distinct spectrotemporal shapes of stationary and transient signal components.

6. CONCLUSION

In this paper, we presented novel strategies for stationary/transient signal separation. Several shrinkage operators were defined by

¹These and additional audio examples can be accessed via <http://www.uni-oldenburg.de/en/mediphysics-acoustics/sigproc/research/audio-demos/>.

considering either the energy of time-frequency neighbourhoods or modulation spectrum regions instead of individual coefficients. These shrinkage operators were specifically tuned to the presumed sparsity and persistence properties of stationary and transient components, exploiting the basic observation that stationary components are sparse in frequency and persistent over time, and vice versa for transients. This step extends the usage of structured shrinkage operators to the context of dual-layer decomposition. We also proposed a novel iteration scheme, *Iterative Cross-Shrinkage*, which appears to work particularly well in conjunction with a dynamic update of the thresholds. In experiments with artificial test signals, the proposed scheme improved stationary/transient separation by surprisingly large margins by about 10 dB SDR compared to the dual-layered ISTA with neighbourhood/modulation persistence. Compared to the dual-layered ISTA with independent coefficients, we were able to achieve improvements of more than 20 dB SDR. In addition, the application of Iterative Cross-Shrinkage to recorded sounds from acoustic musical instruments provided a perceptually convincing separation of transients.

7. REFERENCES

- [1] J. W. Beauchamp, “Analysis and synthesis of musical instrument sounds,” in *Analysis, Synthesis, and Perception of Musical Sounds*, J. W. Beauchamp, Ed. Springer, 2007, pp. 1–89.
- [2] S. J. Godsill and P. J. Rayner, *Digital audio restoration*. New York, NY: Springer, 2002.
- [3] M. Müller, D. P. Ellis, A. Klapuri, and G. Richard, “Signal processing for music analysis,” *IEEE Journal of Selected Topics in Signal Processing*, vol. 5, no. 6, pp. 1088–1110, 2011.
- [4] S. Handel, “Timbre perception and auditory object identification,” in *Hearing*, ser. Handbook of Perception and Cognition, B. C. Moore, Ed. San Diego, CA: Academic Press, 1995, vol. 2, pp. 425–461.
- [5] C. Laroche, M. Kowalski, H. Papadopoulos, and G. Richard, “A structured nonnegative matrix factorization for source separation,” in *Proc. of the 23rd European Signal Processing Conference (EUSIPCO)*. Nice, France: IEEE, 2015, pp. 2033–2037.
- [6] C. Laroche, H. Papadopoulos, M. Kowalski, and G. Richard, “Genre specific dictionaries for harmonic/percussive source separation,” in *Proc. of the 17th International Society for Music Information Retrieval Conference (ISMIR)*, New York, NY, 2016.
- [7] M. D. Plumley, 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.
- [8] G. Evangelista, “Pitch-synchronous wavelet representations of speech and music signals,” *IEEE Transactions on Signal Processing*, vol. 41, no. 12, pp. 3313–3330, 1993.
- [9] P. Polotti and G. Evangelista, “Analysis and synthesis of pseudo-periodic 1/f-like noise by means of wavelets with applications to digital audio,” *EURASIP Journal on Applied Signal Processing*, no. 1, pp. 1–14, 2001.

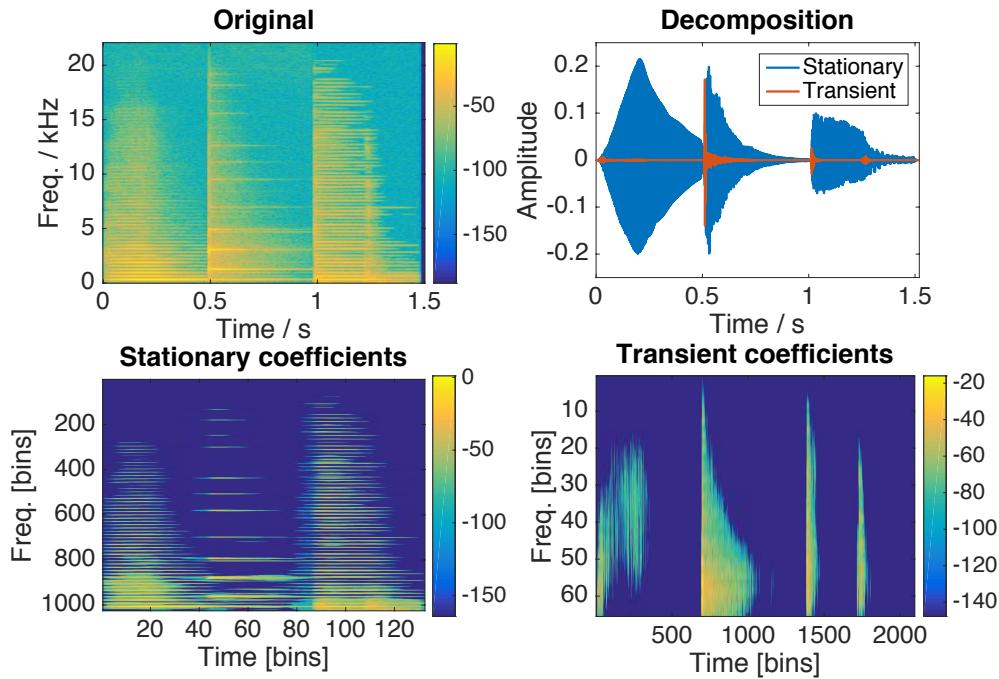


Figure 5: The ICS algorithm with dynamic threshold update and modulation filtering for a sequence of three recorded musical instrument tones (from left to right: violoncello, vibraphone, harpsichord). Top left figure shows the original signal spectrogram. Top right figure shows the time-domain representation of the separation. Bottom left and right figures show the estimated magnitude coefficients for the stationary (left) and transient (right) components. Axes units are in bins to highlight the different time-frequency resolutions.

- [10] L. Daudet and B. Torr  sani, “Hybrid representations for audiophonic signal encoding,” *Signal Processing*, vol. 82, no. 11, pp. 1595–1617, 2002.
- [11] S. Molla and B. Torresani, “A hybrid scheme for encoding audio signal using hidden markov models of waveforms,” *Applied and Computational Harmonic Analysis*, vol. 18, pp. 137–166, 2005.
- [12] C. F  votte, B. Torresani, L. Daudet, and S. J. Godsill, “Sparse linear regression with structured priors and application to denoising of musical audio,” *IEEE Transactions on Audio, Speech and Language Processing*, vol. 16, no. 1, pp. 174–185, 2008.
- [13] I. Daubechies, M. Defrise, and C. De Mol, “An iterative thresholding algorithm for linear inverse problems with a sparsity constraint,” *Communications on Pure and Applied Mathematics*, vol. 57, no. 11, pp. 1413–1457, 2004.
- [14] G. Teschke, “Multi-frame representations in linear inverse problems with mixed multi-constraints,” *Applied and Computational Harmonic Analysis*, vol. 22, pp. 43–60, 2007.
- [15] M. Kowalski, “Sparse regression using mixed norms,” *Applied and Computational Harmonic Analysis*, vol. 27, no. 3, pp. 303 – 324, 2009.
- [16] M. Kowalski and B. Torresani, “Sparsity and persistence: mixed norms provide simple signal models with dependent coefficients,” *Signal, Image and Video Processing*, vol. 3, no. 3, pp. 251–264, 2008a.
- [17] M. Kowalski, K. Siedenburg, and M. D  rfler, “Social sparsity! Neighborhood systems enrich structured shrinkage operators,” *IEEE Transactions on Signal Processing*, vol. 61, no. 10, pp. 2498–2511, 2013.
- [18] M. D  rfler, “Time-frequency analysis for music signals a mathematical approach,” *Journal of New Music Research*, vol. 30, no. 1, pp. 3–12, 2001.
- [19] A. Beck and M. Teboulle, “A fast iterative shrinkage-thresholding algorithm for linear inverse problems,” *SIAM, Journal of Imaging Sciences*, vol. 2, no. 1, pp. 183–202, 2009.
- [20] R. Tibshirani, “Regression shrinkage and selection via the lasso,” *Journal of the Royal Statistical Society. Series B (Statistical Methodology)*, vol. 58, no. 1, pp. 267–288, 1996.
- [21] S. P. Ghosh, A. M. Sayeed, and R. G. Baraniuk, “Improved wavelet denoising via empirical Wiener filtering,” in *Proceedings of SPIE*, vol. 3169. San Diego, CA, 1997, pp. 389–399.
- [22] A. Antoniadis *et al.*, “Wavelet methods in statistics: Some recent developments and their applications,” *Statistics Surveys*, vol. 1, pp. 16–55, 2007.
- [23] K. Siedenburg and M. D  rfler, “Persistent time-frequency shrinkage for audio denoising,” *Journal of the Audio Engineering Society (AES)*, vol. 61, no. 1/2, pp. 29–38, 2013.
- [24] K. Siedenburg, M. Kowalski, M. D  rfler *et al.*, “Audio de-clipping with social sparsity,” in *Proc. of the IEEE Interna-*

tional Conference on Acoustics, Speech and Signal Processing, Florence, Italy, 2014, pp. 1577–1581.

- [25] K. Siedenburg and P. Depalle, “Modulation filtering for structured time-frequency estimation of audio signals,” in *Proc. of the IEEE International Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*, New Paltz, NY, USA, 2013, pp. 1–4.
- [26] T. M. Elliott and F. E. Theunissen, “The modulation transfer function for speech intelligibility,” *PLoS Computational Biology*, vol. 5, no. 3, p. e1000302, 2009.
- [27] K. Siedenburg, “Persistent empirical Wiener estimation with adaptive threshold selection for audio denoising,” in *Proceedings of the 9th Sound and Music Computing Conference*, Copenhagen, DK, 2012.
- [28] I. Loris, “On the performance of algorithms for the minimization of ℓ_1 -penalized functionals,” *Inverse Problems*, vol. 25, 2009.
- [29] P. L. Søndergaard, B. Torrésani, and P. Balazs, “The Linear Time Frequency Analysis Toolbox,” *International Journal of Wavelets, Multiresolution Analysis and Information Processing*, vol. 10, no. 4, 2012.
- [30] K. Siedenburg, K. Jones-Mollerup, and S. McAdams, “Acoustic and categorical dissimilarity of musical timbre: Evidence from asymmetries between acoustic and chimeric sounds,” *Frontiers in Psychology*, vol. 6, no. 1977, p. doi: 10.3389/fpsyg.2015.01977, 2016.

AN EXPLORATIVE STRING-BRIDGE-PLATE MODEL WITH TUNABLE PARAMETERS

Maarten van Walstijn, Sandor Mehes

Sonic Arts Research Centre

School of Electronics, Electrical Engineering, and Computer Science

Queen's University Belfast, UK

{m.vanwalstijn, smehes01}@qub.ac.uk

ABSTRACT

The virtual exploration of the domain of mechano-acoustically produced sound and music is a long-held aspiration of physical modelling. A physics-based algorithm developed for this purpose combined with an interface can be referred to as a virtual-acoustic instrument; its design, formulation, implementation, and control are subject to a mix of technical and aesthetic criteria, including sonic complexity, versatility, modal accuracy, and computational efficiency. This paper reports on the development of one such system, based on simulating the vibrations of a string and a plate coupled via a (nonlinear) bridge element. Attention is given to formulating and implementing the numerical algorithm such that any of its parameters can be adjusted in real-time, thus facilitating musician-friendly exploration of the parameter space and offering novel possibilities regarding gestural control. Simulation results are presented exemplifying the sonic potential of the string-bridge-plate model (including bridge rattling and buzzing), and details regarding efficiency, real-time implementation and control interface development are discussed.

1. INTRODUCTION

Physical modelling studies often focus on the simulation and virtualisation of a specific musical instrument or class thereof [1]. Progress in this field has, however, also been driven by the prospect of new variations of virtual-acoustic instruments. Going beyond the aim of faithful imitation, the challenge then shifts towards design and control, seeking physical model configurations, implementations, and user interfaces that facilitate creative exploration of the domain of mechanically/acoustically plausible sounds. Notable past efforts in this area include several early physical modelling software environments such as CORDIS [2], MOSAIC [3], and TAO [4]. These systems and more recent variations of the same concept (see, e.g. [5, 6, 7, 8]) facilitate the construction of new instruments by connecting either elementary masses or distributed objects (e.g. strings, membranes), usually via spring elements.

Beyond such modularity, the user invariably faces the task of learning to navigate the parameter space of the specified configuration. Even though the parameters are physical and therefore intuitive, this tends to be a formidable exercise for all but the simplest systems, especially so if it has to be performed off-line. A principal motivation behind the present study is to devise virtual-acoustic instruments that facilitate this learning and exploration process. This is pursued here by developing a specific physical configuration (hence de-emphasising modularity) that allows the user to perform design and control tasks via on-line tuning of any of the model parameters with instant aural feedback.

The first challenge that arises from this objective is one of design, i.e. determining what kind of physical model configuration is appropriate as a testbed. Drawing inspiration from several relevant DAFX studies [9, 5, 8, 10] and partly building on earlier ideas [11, 12], the proposed model takes the form of a string and a plate connected by a parameterised bridge element, with a local damper fitted on the string (see Fig. 1). The bridge can be parametrically configured to simulate different types of linear and nonlinear coupling, including mass-like behaviour, spring stiffening and contact phenomena (i.e. rattling and buzzing).

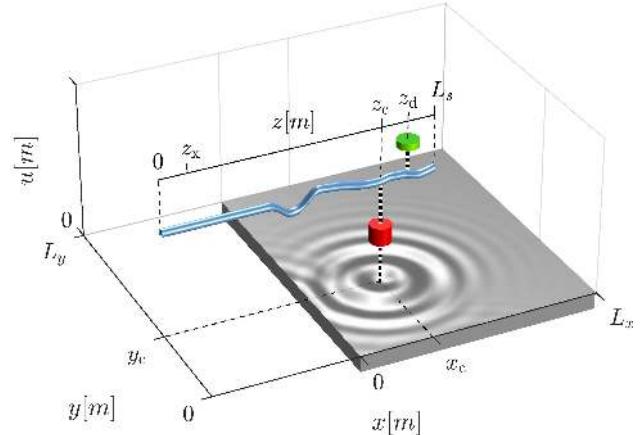


Figure 1: *Geometry of the string-bridge-plate model. The red cylinder represents the bridge mass, and the green disk indicates where the string damper is located.*

The second, more technical challenge - the addressing of which the bulk of this paper is devoted to - consists in the derivation of a computationally robust and sufficiently efficient numerical formulation that supports on-line parameter adjustment, and can be scaled to common hardware for real-time operation. The approach taken to address this can be summarised as follows. The system equations of the string-bridge-plate model are set out in § 2. An unconditionally stable and modally exact numerical formulation is then developed in § 3. This avoids complications and restrictions regarding on-line variation of some of the model parameters that would arise with more general spatio-temporal discretisation. Similar to the Port-Hamiltonian approach [13], discretisation is performed over a first-order form, which in the presence of non-smooth forces can help avoid spurious oscillations [14]. The proposed model is computationally robust due to (a) its provable stability, (b) the provable uniqueness and demonstrable convergence of the solution to the nonlinear equation that has to be found iteratively at each discrete-time instant, and (c) the empirically set constraints on each of the tunable parameters. The latter is needed

to limit the number of required nonlinear solver iterations. The model's sonic potential is exemplified with numerical experiments in § 4, which also reports on real-time implementation aspects and on preliminary findings with the initial control configuration. Finally, § 5 offers concluding remarks and future perspectives.

2. STRING-BRIDGE-PLATE MODEL

2.1. System Equations

Consider transverse vibrations (u) of a stiff string and a thin rectangular plate, both under simply supported boundary conditions, coupled via a bridge element of a mass m_b with parameterised spring elements, and with local damping applied at $z = z_d$ along the string axis (see Fig. 1). The string is characterised by its length L_s , mass density ρ_s , cross-sectional area A_s , Young's modulus E_s , moment of inertia I_s , and damping factor γ_s of yet to be determined wave-number dependency. The plate is of dimensions $L_x \times L_y \times h_p$, mass density ρ_p , and experiences damping according to γ_p . Two of its material properties are encapsulated in the parameter $G_p = (E_p h_p^3) / (12(1 - \nu_p^2))$, where ν_p is the Poisson ratio. The dynamics of the coupled system are governed by

$$\rho_s A_s \frac{\partial^2 u_s}{\partial t^2} = T_s \frac{\partial^2 u_s}{\partial z^2} - E_s I_s \frac{\partial^4 u_s}{\partial z^4} - \gamma_s \frac{\partial u_s}{\partial t} + \psi_c(z) F_1(t) + \psi_x(z) F_x(t) + \psi_d(z) F_d(t), \quad (1)$$

$$m_b \frac{\partial^2 u_b}{\partial t^2} = -r_b \frac{\partial u_b}{\partial t} - F_1(t) + F_2(t) + F_b(t), \quad (2)$$

$$\rho_p h_p \frac{\partial^2 u_p}{\partial t^2} = -G_p \nabla^4 u_p - \gamma_p \frac{\partial u_p}{\partial t} - \Psi_c(x, y) F_2(t), \quad (3)$$

where, for $\kappa = c, x, d$, $\psi_\kappa(z) = \delta(z - z_\kappa)$ and $\Psi(x, y) = \delta(x - x_c, y - y_c)$ are single-point spatial distributions, and $\nabla^4 = (\partial^4 / \partial x^4 + \partial^4 / \partial y^4)$ is a biharmonic operator. It is readily seen that (1) models a beam rather than a string when $T_s \ll E_s I_s$.

The system is brought into vibration by $F_x(t)$, which excites the string, and/or $F_b(t)$ which drives the bridge mass; bridge damping is controlled with r_b . The damper force - which allows suppression of string oscillations on either side of the connection point - is $F_d(t) = -r_d \frac{\partial u_d}{\partial t}$, where $u_d(t) = u_s(z_d, t)$. The spring connection forces F_1 and F_2 are functions of the inter-object distances $u_1(t) = u_b(t) - u_s(z_c, t)$ and $u_2(t) = u_p(x_c, y_c, t) - u_b(t)$ as follows ($\ell = 1, 2$):

$$F_\ell(t) = k_L u_\ell(t) + k_\ell^+ [u_\ell(t)]^\alpha - k_\ell^- [-u_\ell(t)]^\alpha, \quad (4)$$

where $[u] \triangleq \max(0, u)$, k_L , k_ℓ^+ , and k_ℓ^- are stiffness coefficients, and $\alpha \geq 1$ is a power law exponent. The specific form of (4) includes various types of linear and nonlinear restoring-force based connections, allowing to parametrically remove or add the pushing or pulling force of each of the springs through adjusting k_ℓ^+ , and k_ℓ^- , respectively; Fig. 2 shows a few example force-distance curves. Modelling of the bridge in this manner facilitates the simulation of a variety of connection configurations, four example cases of which are shown in Fig. 3. Note that while the bridge mass cannot be set to zero in our model, it can nonetheless be made negligible by making it very small compared with the string mass. For audio output, we define the plate momentum at K pickup positions $(x_{a,k}, y_{a,k})$:

$$p_{a,k}(t) = \rho_p h_p \frac{\partial}{\partial t} u_p(x_{a,k}, y_{a,k}, t). \quad (5)$$

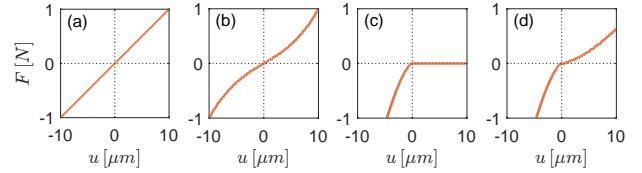


Figure 2: Examples of spring force-distance curves. (a): linear spring [$k_L = 10^5$, $k_\ell^\pm = 0$]. (b): stiffening spring [$k_L = 6 \times 10^4$, $k_\ell^+ = k_\ell^- = 4 \times 10^{14}$, $\alpha = 3$]. (c): one-sided spring [$k_L = 0$, $k_\ell^+ = 0$, $k_\ell^- = 1 \times 10^8$, $\alpha = 1.5$]. (d): asymmetric spring [$k_L = 0$, $k_\ell^+ = 1 \times 10^8$, $k_\ell^- = 2 \times 10^7$, $\alpha = 1.5$].

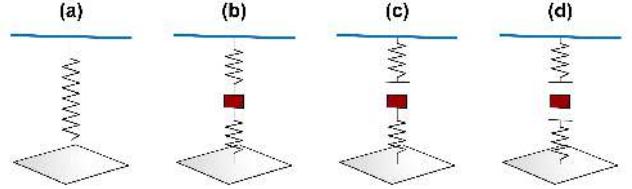


Figure 3: Bridge configuration examples. (a): massless bridge [$m_b \ll \rho_s A_s L_s$, $k_\ell^\pm = k_L$]. (b): mass-spring bridge [$k_\ell^\pm = k_L$]. (c): 'flat' bridge [$k_L = 0$, $k_\ell^+, k_\ell^- > 0$, $k_\ell^- = 0$]. (d): rattling bridge [$k_L = 0$, $k_\ell^+, k_\ell^- > 0$, $k_\ell^- = k_\ell^+$].

2.2. Modal Expansion

The displacement of each of the two distributed linear sub-systems can be written as a modal expansion:

$$u_s(z, t) = \sum_{i=1}^{M_s} v_{s,i}(z) \bar{u}_{s,i}(t), \quad (6)$$

$$u_p(x, y, t) = \sum_{i=1}^{M_x} \sum_{j=1}^{M_y} v_{p,i,j}(x, y) \bar{u}_{p,i,j}(t), \quad (7)$$

where

$$v_{s,i} = \sin(\beta_i x), \quad v_{p,i,j}(x, y) = \sin(\beta_{x,i} x) \sin(\beta_{y,j} y), \quad (8)$$

are the respective mode shape functions under simply supported boundary conditions. The wave numbers are $\beta_i = i\pi/L_s$ for the string/beam and $\beta_{x,i} = i\pi/L_x$, $\beta_{y,j} = j\pi/L_y$ for the plate (the overall wave number defined as $\beta_{i,j} = \sqrt{\beta_{x,i}^2 + \beta_{y,j}^2}$). For both the string ($\kappa = s$) and the plate ($\kappa = p$), the modal displacements (\bar{u}_κ) in (6) and (7) are governed by second-order differential equations of the form

$$m_\kappa \frac{\partial^2 \bar{u}_{\kappa,l}}{\partial t^2} = -k_{\kappa,l} \bar{u}_{\kappa,l}(t) - r_{\kappa,i} \frac{\partial \bar{u}_{\kappa,l}}{\partial t} + \bar{F}_{\kappa,l}(t), \quad (9)$$

where we have introduced a new modal index l , which maps as $l = i$ for the string and as $l = (M_y - 1)i + j$ for the plate, and where the respective modal parameters are

$$m_s = \frac{\rho_s A_s L_s}{2}, \quad k_{s,l} = \frac{L_s}{2} (E_s I_s \beta_l^4 + T_s \beta_l^2), \quad r_{s,l} = \frac{L_s}{2} \gamma_s(\beta_l), \quad (10)$$

$$m_p = \frac{\rho_p h_p L_x L_y}{4}, \quad k_{p,l} = \frac{L_x L_y G_p}{4} \beta_l^4, \quad r_{p,l} = \frac{L_x L_y}{4} \gamma_p(\beta_l), \quad (11)$$

The modal frequencies are $\omega_{\kappa,l} = \sqrt{k_{\kappa,l}/m_{\kappa} - \zeta_{\kappa,l}^2}$, and the modal attenuation rates are parameterised here in the convenient low-parameter frequency-dependent form

$$\zeta_{\kappa,\ell} = \frac{r_{\kappa}}{2m_{\kappa}} = \sigma_{\kappa,0} + \sigma_{\kappa,1}\beta_{\ell} + \sigma_{\kappa,3}\beta_{\ell}^3, \quad (12)$$

which retrospectively defines the damping parameters γ_s and γ_p in equations (1) and (3) as wave-number dependent. For the two sub-systems, the modal force term in (9) is

$$\begin{aligned} \bar{F}_{s,l}(t) &= \int_0^{L_s} v_{s,l}(z) \left[\psi_c(z) F_1(t) + \psi_x(z) F_x(t) + \psi_d(z) F_d(t) \right] dz \\ &= g_{s,l} F_1(t) + g_{x,l} F_x(t) + g_{d,l} F_d(t), \end{aligned} \quad (13)$$

$$\begin{aligned} \bar{F}_{p,l}(t) &= - \int_0^{L_x} \int_0^{L_y} v_{p,l}(x, y) \psi(x, y) F_2(t) dy dx \\ &= -g_{p,l} F_2(t), \end{aligned} \quad (14)$$

where (for $\kappa = e, d$)

$$g_{s,l} = v_{s,l}(z_c), \quad g_{\kappa,l} = v_{s,l}(z_{\kappa}), \quad g_{p,l} = v_{p,l}(x_c, y_c). \quad (15)$$

For audio output, the plate momenta are modally expanded as

$$p_{a,k} = p_p h_p \sum_{l=1}^{M_p} g_{a,k,l} \frac{\partial \bar{u}_{p,l}}{\partial t}, \quad (16)$$

where $g_{a,k,l} = v_{p,l}(x_{a,k}, y_{a,k})$ and $M_p = M_x M_y$.

3. NUMERICAL FORMULATION

3.1. Discretisation in Time

The differential equations (9) and (2) are first re-written in first-order form:

$$\begin{aligned} \frac{\partial \bar{u}_{s,l}}{\partial t} &= \frac{\bar{p}_{s,l}}{m_s}, \quad \frac{\partial \bar{p}_{s,l}}{\partial t} = -k_{s,l} \bar{u}_{s,l} - r_{s,l} \frac{\partial \bar{u}_{s,l}}{\partial t} \\ &\quad + g_{s,l} F_1 + g_{x,l} F_x + g_{d,l} F_d, \end{aligned} \quad (17)$$

$$\frac{\partial u_b}{\partial t} = \frac{p_b}{m_b}, \quad \frac{\partial p_b}{\partial t} = -r_b \frac{\partial u_b}{\partial t} - F_1 + F_2 + F_b, \quad (18)$$

$$\frac{\partial \bar{u}_{p,l}}{\partial t} = \frac{\bar{p}_{p,l}}{m_p}, \quad \frac{\partial \bar{p}_{p,l}}{\partial t} = -k_{p,l} \bar{u}_{p,l} - r_{p,l} \frac{\partial \bar{u}_{p,l}}{\partial t} - g_{p,l} F_2, \quad (19)$$

where p_{κ} denotes momentum. Gridding time as $u^n \hat{=} u(n\Delta_t)$, the following *sum* and *difference* operators

$$\mu u = u^{n+1} + u^n, \quad \delta u = u^{n+1} - u^n, \quad (20)$$

are then employed to discretise these equations at $t = (n+\frac{1}{2})\Delta_t$, which (with $F_x^{n+\frac{1}{2}} = \frac{1}{2}\mu F_x$, $F_b^{n+\frac{1}{2}} = \frac{1}{2}\mu F_b$) yields

$$\begin{aligned} \frac{\delta \bar{u}_{s,l}}{\Delta t} &= \frac{\mu \bar{p}_{s,l}}{2m_s}, \quad \frac{\delta \bar{p}_{s,l}}{\Delta t} = -k_{s,l}^* \frac{\mu \bar{u}_{s,l}}{2} - r_{s,l}^* \frac{\delta \bar{u}_{s,l}}{\Delta t} \\ &\quad + g_{s,l}^* F_1^{n+\frac{1}{2}} + g_{x,l}^* F_x^{n+\frac{1}{2}} + g_{d,l}^* F_d^{n+\frac{1}{2}}, \end{aligned} \quad (21)$$

$$\frac{\delta u_b}{\Delta t} = \frac{\mu p_b}{2m_b}, \quad \frac{\delta p_b}{\Delta t} = -r_b \frac{\delta u_b}{\Delta t} - F_1^{n+\frac{1}{2}} + F_2^{n+\frac{1}{2}} + F_b^{n+\frac{1}{2}}, \quad (22)$$

$$\frac{\delta \bar{u}_{p,l}}{\Delta t} = \frac{\mu \bar{p}_{p,l}}{2m_p}, \quad \frac{\delta \bar{p}_{p,l}}{\Delta t} = -k_{p,l}^* \frac{\mu \bar{u}_{p,l}}{2} - r_{p,l}^* \frac{\delta \bar{u}_{p,l}}{\Delta t} - g_{p,l}^* F_2^{n+\frac{1}{2}}. \quad (23)$$

The damper and spring connection forces ($\ell = 1, 2$) are discretised as

$$F_d^{n+\frac{1}{2}} = -r_d \frac{\delta u_d}{\Delta t}, \quad F_{\ell}^{n+\frac{1}{2}} = k_L \frac{\mu u_{\ell}}{2} + \frac{\delta V_{\ell}}{\delta u_{\ell}}, \quad (24)$$

the latter utilising the nonlinear spring element potentials

$$V_{\ell}(u_{\ell}) = \left(\frac{k_{\ell}^+}{\alpha + 1} \right) [u_{\ell}]^{\alpha+1} + \left(\frac{k_{\ell}^-}{\alpha + 1} \right) [-u_{\ell}]^{\alpha+1}. \quad (25)$$

Note that in (21) and (23) the modal elasticity and damping constants of the string and plate have been substituted as follows:

$$k_{\kappa,l} \rightarrow k_{\kappa,l}^* = \frac{4m_{\kappa} a_{\kappa,l}^*}{\Delta_t^2}, \quad r_{\kappa,l} \rightarrow r_{\kappa,l}^* = \frac{2m_{\kappa} b_{\kappa,l}^*}{\Delta_t}. \quad (26)$$

where, with $R_{\kappa,l} = \exp(-\zeta_{\kappa,l} \Delta_t)$ and $\Omega_{\kappa,l} = \cos(\omega_{\kappa,l} \Delta_t)$,

$$a_{\kappa,l}^* = \frac{1 - 2R_{\kappa,l} \Omega_{\kappa,l} + R_{\kappa,l}^2}{1 + 2R_{\kappa,l} \Omega_{\kappa,l} + R_{\kappa,l}^2}, \quad b_{\kappa,l}^* = \frac{2(1 - R_{\kappa,l}^2)}{1 + 2R_{\kappa,l} \Omega_{\kappa,l} + R_{\kappa,l}^2}. \quad (27)$$

As explained in previous work [12], this eliminates numerical dispersion and attenuation at the sub-system resonance frequencies. In addition, each modal weight $g_{\kappa,l}$ ($\kappa = x, s, p$) has been substituted in (21-23) with $g_{\kappa,l}^* = W_r(\frac{1}{2}\omega_{\kappa,l}/\pi) g_{\kappa,l}$, where

$$W_r(f) = \begin{cases} 1 & : f < f_r \\ (f_n - f)/(f_n - f_r) & : f_r \leq f < f_n \\ 0 & : f \geq f_n \end{cases} \quad (28)$$

is a frequency window in which $f_n = \frac{1}{2}\Delta_t^{-1}$ is the Nyquist frequency and $f_r < f_n$ is the highest mode frequency to be rendered in full amplitude. This allows variation of system parameters over time without mode aliasing and - control rate permitting - without causing significant discontinuities in the output signal. For audio rates of 44.1 kHz or higher, a sensible choice is to set $f_r = 20$ kHz.

3.2. A Vector-Matrix Update Form

Stacking all modal states of the string and plate in column vectors ($\bar{\mathbf{u}}_s^n$, $\bar{\mathbf{q}}_s^n$) and ($\bar{\mathbf{u}}_p^n$, $\bar{\mathbf{q}}_p^n$), the system modal state vectors can be defined as

$$\bar{\mathbf{u}}^n = [(\bar{\mathbf{u}}_s^n)^T, u_b, (\bar{\mathbf{u}}_p^n)^T]^T, \quad \bar{\mathbf{q}}^n = [(\bar{\mathbf{q}}_s^n)^T, q_b, (\bar{\mathbf{q}}_p^n)^T]^T, \quad (29)$$

where $\bar{q}_{\kappa,l} = (\Delta_t/(2m_{\kappa}))\bar{p}_{\kappa,l}$ is a convenient change of variable. The complete discrete-time system can then be written

$$\delta \bar{\mathbf{u}} = \mu \bar{\mathbf{q}}, \quad (30)$$

$$\delta \bar{\mathbf{q}} = -(\mathbf{A}\mu + \mathbf{B}\delta) \bar{\mathbf{u}} + \Xi \left(\mathbf{G}_c \mathbf{F}^{n+\frac{1}{2}} + \mathbf{G}_e \mathbf{F}_e^{n+\frac{1}{2}} + \mathbf{G}_d F_d^{n+\frac{1}{2}} \right), \quad (31)$$

where $\mathbf{F}^{n+\frac{1}{2}} = [F_1^{n+\frac{1}{2}} \ F_2^{n+\frac{1}{2}}]^T$ and $\mathbf{F}_e^{n+\frac{1}{2}} = [F_x^{n+\frac{1}{2}} \ F_b^{n+\frac{1}{2}}]^T$ are force vectors and where the matrices

$$\mathbf{A} = \begin{bmatrix} \mathbf{A}_s & \mathbf{0}_s & \mathbf{0}_{sp} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0}_{ps} & \mathbf{0}_p & \mathbf{A}_p \end{bmatrix}, \quad \mathbf{B} = \begin{bmatrix} \mathbf{B}_s & \mathbf{0}_s & \mathbf{0}_{sp} \\ \mathbf{0} & b_b & \mathbf{0} \\ \mathbf{0}_{ps} & \mathbf{0}_p & \mathbf{B}_p \end{bmatrix}, \quad (32)$$

contain submatrices \mathbf{A}_{κ} and \mathbf{B}_{κ} that are $M_{\kappa} \times M_{\kappa}$ diagonal matrices with diagonal elements $A_{\kappa,l,l} = a_{\kappa,l}^*$ and $B_{\kappa,l,l} = b_{\kappa,l}^*$,

respectively, and with $b_b = (r_b \Delta_t) / (2m_b)$. The other matrices in (31) are

$$\begin{aligned}\mathbf{G}_c &= \begin{bmatrix} \mathbf{g}_s & \mathbf{0}_s \\ -1 & 1 \\ \mathbf{0}_p & -\mathbf{g}_p \end{bmatrix}, \quad \mathbf{G}_e = \begin{bmatrix} \mathbf{g}_x & \mathbf{0}_s \\ 0 & 1 \\ \mathbf{0}_p & \mathbf{0}_p \end{bmatrix}, \\ \mathbf{G}_d &= \begin{bmatrix} \mathbf{g}_d \\ 0 \\ \mathbf{0}_p \end{bmatrix}, \quad \boldsymbol{\Xi} = \begin{bmatrix} \xi_s \mathbf{I}_s & \mathbf{0}_s & \mathbf{0}_{sp} \\ 0 & \xi_b & 0 \\ \mathbf{0}_{ps} & \mathbf{0}_p & \xi_p \mathbf{I}_p \end{bmatrix},\end{aligned}\quad (33)$$

where matrices $\mathbf{G}_{c,e,d}$ feature modal column vectors \mathbf{g}_κ which have M_κ elements defined by (15), $\xi_\kappa = \Delta_t^2 / (2m_\kappa)$, and where \mathbf{I}_κ are $M_\kappa \times M_\kappa$ identity matrices. By setting $\bar{s} = \delta \bar{u} = \mu \bar{q}$ from (30), equation (31) can be reworked into

$$\bar{s} = \bar{e} + \mathbf{H}_c \mathbf{F}^{n+\frac{1}{2}} + \mathbf{H}_d F_d^{n+\frac{1}{2}}, \quad (34)$$

where

$$\bar{e} = 2\mathbf{C}(\bar{q}^n - \mathbf{A}\bar{u}^n) + \mathbf{H}_e \mathbf{F}_e^{n+\frac{1}{2}}, \quad (35)$$

and

$$\begin{aligned}\mathbf{H}_c &= \begin{bmatrix} \mathbf{h}_s & \mathbf{0}_s \\ -\xi_b & \xi_b \\ \mathbf{0}_p & -\mathbf{h}_p \end{bmatrix}, \quad \mathbf{H}_e = \begin{bmatrix} \mathbf{h}_x & \mathbf{0}_s \\ 0 & \xi_b \\ \mathbf{0}_p & \mathbf{0}_p \end{bmatrix}, \\ \mathbf{H}_d &= \begin{bmatrix} \mathbf{h}_d \\ 0 \\ \mathbf{0}_p \end{bmatrix}, \quad \mathbf{C} = \begin{bmatrix} \mathbf{C}_s & \mathbf{0}_s & \mathbf{0}_{sp} \\ 0 & (1+b_p)^{-1} & 0 \\ \mathbf{0}_{ps} & \mathbf{0}_p & \mathbf{C}_p \end{bmatrix},\end{aligned}\quad (36)$$

with, for $\kappa = s, p$, we have $\mathbf{C}_\kappa = (\mathbf{I}_\kappa + \mathbf{A}_\kappa + \mathbf{B}_\kappa)^{-1}$, and

$$\mathbf{h}_s = \xi_s \mathbf{C}_s \mathbf{g}_s, \quad \mathbf{h}_p = \xi_p \mathbf{C}_p \mathbf{g}_p, \quad \mathbf{h}_d = \xi_d \mathbf{C}_d \mathbf{g}_d, \quad \mathbf{h}_x = \xi_x \mathbf{C}_x \mathbf{g}_x, \quad (37)$$

Once known, the step vector \bar{s} is employed to update the modal states with

$$\bar{u}^{n+1} = \bar{s} + \bar{u}^n, \quad \bar{q}^{n+1} = \bar{s} - \bar{q}^n, \quad (38)$$

after which the audio output signal vector can be computed as

$$\mathbf{p}_a^n = \mathbf{W}_a \mathbf{q}_p^n, \quad (39)$$

where $\mathbf{W}_a = (2m_p)\Delta_t^{-1} [\mathbf{g}_{a,1}, \mathbf{g}_{a,2}, \dots, \mathbf{g}_{a,K}]^\top$ and where $\mathbf{g}_{a,k}$ is the plate modal column vector for the pickup position $(x_{a,k}, y_{a,k})$. Note that all of the $M \times M$ matrices ($\mathbf{A}, \mathbf{B}, \mathbf{C}$ and $\boldsymbol{\Xi}$) featured above, with $M = M_s + M_p + 1$, are diagonal, and as such can be replaced with $M \times 1$ vectors, with the calculation in (35) being achieved using elementwise vector multiplication.

3.3. Solving for the Step Vector

To find a way to compute the step vector \bar{s} , we first eliminate the damper force from the system. From (24), one may write

$$F_d^{n+\frac{1}{2}} = -r_d \Delta_t^{-1} s_d, \quad (40)$$

where $s_d = \delta u_d$, which (from multiplying (34) with \mathbf{G}_d^\top) also equals

$$s_d = \mathbf{G}_d^\top (\bar{e} + \mathbf{H}_c \mathbf{F}^{n+\frac{1}{2}}) + \mathbf{G}_d^\top \mathbf{H}_d F_d^{n+\frac{1}{2}}. \quad (41)$$

Eliminating both s_d and $F_d^{n+\frac{1}{2}}$ then allows re-writing (34) as

$$\bar{s} = \bar{w} + \Phi_c \mathbf{F}^{n+\frac{1}{2}}, \quad (42)$$

where $\bar{w} = \bar{e} - \theta_d \mathbf{H}_d \mathbf{G}_d^\top \bar{e}$ and $\Phi_c = \mathbf{H}_c - \theta_d \mathbf{H}_d \mathbf{G}_d^\top \mathbf{H}_c$, with $\theta_d = r_d / (r_d \mathbf{G}_d^\top \mathbf{H}_d + \Delta_t)$. Next, the modal coordinate equation (42) is transformed into an equation in the spring connection coordinates by pre-multiplying (42) with $-\mathbf{G}_c^\top$, which gives

$$\mathbf{M} \mathbf{F}^{n+\frac{1}{2}} = \mathbf{w} - \mathbf{s}, \quad (43)$$

where $\mathbf{w} = -\mathbf{G}_c^\top \bar{w}$, $\mathbf{s} = \delta \mathbf{u}$ (with $\mathbf{u}^n = [u_1^n, u_2^n]^\top$), and

$$\mathbf{M} = \mathbf{G}_c^\top \boldsymbol{\Phi}_c = \begin{bmatrix} (\xi_b + \phi_s) & -\xi_b \\ -\xi_b & (\xi_b + \phi_p) \end{bmatrix}, \quad (44)$$

with $\phi_s = \mathbf{g}_s^\top \mathbf{h}_s - \theta_d \mathbf{g}_s^\top \mathbf{h}_d \mathbf{g}_d^\top \mathbf{h}_s$ and $\phi_p = \mathbf{g}_p^\top \mathbf{h}_p$. From the second equation in (24), one may write

$$\mathbf{F}^{n+\frac{1}{2}} = \boldsymbol{\lambda}(\mathbf{s}) + k_L \left(\frac{1}{2} \mathbf{s} + \mathbf{u}^n \right), \quad (45)$$

where $\boldsymbol{\lambda}(\mathbf{s}) = [\lambda_1(s_1), \lambda_2(s_2)]^\top$ with, for $\ell = 1, 2$

$$\lambda_\ell(s_\ell) = \frac{V_\ell(s_\ell + u_\ell^n) - V_\ell(u_\ell^n)}{s_\ell}. \quad (46)$$

After substituting (45) into (43), a system of two coupled nonlinear simultaneous equations is obtained:

$$\mathbf{M} \boldsymbol{\lambda}(\mathbf{s}) + \left(\mathbf{I} + \frac{1}{2} k_L \mathbf{M} \right) \mathbf{s} + \boldsymbol{\iota} = \mathbf{0}, \quad (47)$$

where $\boldsymbol{\iota} = k_L \mathbf{M} \mathbf{u}^n - \mathbf{w}$. These can be solved iteratively using Newton's method. Once \mathbf{s} is solved, the spring forces are updated from (43), after which the step vector \bar{s} is calculated with (42). Besides updating the states with (38) and the output with (39), it is also required that the connection displacement vector is updated as $\mathbf{u}^{n+1} = \mathbf{s} + \mathbf{u}^n$.

3.4. Uniqueness and Convergence

Defining $\lambda'_\ell = \frac{\partial \lambda_\ell}{\partial s_\ell} \geq 0$, the Jacobian to be used for solving (47) is

$$\mathbf{J} = \begin{bmatrix} 1 + (\xi_b + \phi_s) \left(\lambda'_1 + \frac{k_L}{2} \right) & -\xi_b \left(\lambda'_2 + \frac{k_L}{2} \right) \\ -\xi_b \lambda'_1 + \left(\frac{k_L}{2} \right) & 1 + (\xi_b + \phi_p) \left(\lambda'_2 + \frac{k_L}{2} \right) \end{bmatrix} \quad (48)$$

which is clearly positive definite for any positive bridge mass (meaning ξ_b is finite), hence (47) has a unique root \mathbf{s}^* . \mathbf{J} is also a so-called M-matrix, which is a condition for global convergence [15]. From the fact that $\lambda_\ell(s_\ell)$ has a single inflection point (i.e. concave for $s_\ell < 0$ and convex for $s_\ell > 0$), it then follows that there is guaranteed monotonic convergence from any \mathbf{s} for which each of the elements s_ℓ has the same sign as the corresponding element of the actual root and satisfies $|s_\ell| \geq |s_\ell^*|$ (see [15], p. 112-113). Empirically observing that the solver generally reaches this condition from any starting position then suggests global convergence. Such robustness indeed appears to hold, as extensive testing with randomised starting points has indicated that the solver generally converges (no cases of non-convergence observed for the tested parameters sets).

What is, however, not guaranteed without any further conditions is that the iterative solver converges within a fixed number of iterations. The iteration count in fact depends not only on the initial guess, but also on the driving signal amplitude and all system parameters. A practical way to try and keep the iteration count under control is by empirically setting constraints on the parameters (see Table 1) and applying a limiter on the input signal.

3.5. Stability

The system energy at time instant $t = n\Delta_t$ is the sum of the modal energies of the string and plate plus the kinetic energy of the bridge mass and the potential energies in the linear and nonlinear spring elements:

$$H^n = \sum_{l=1}^{M_s} \left[\frac{(\bar{p}_{s,l}^n)^2}{2m_s} + \frac{k_{s,l}^*(\bar{u}_{s,l}^n)^2}{2} \right] + \sum_{l=1}^{M_p} \left[\frac{(\bar{p}_{p,l}^n)^2}{2m_p} + \frac{k_{p,l}^*(\bar{u}_{p,l}^n)^2}{2} \right] + \frac{(\bar{p}_b^n)^2}{2m_b} + \frac{k_L}{2} (u_1^2 + u_2^2) + V_1(u_1^n) + V_2(u_2^n). \quad (49)$$

Given that all the individual terms in (49) are non-negative, it follows that $H^n \geq 0$. In vector-matrix form, H^n can be written

$$H^n = (\bar{\mathbf{q}}^n)^T \Xi^{-1} \bar{\mathbf{q}}^n + (\bar{\mathbf{u}}^n)^T \Xi^{-1} \mathbf{A} \bar{\mathbf{u}}^n + \frac{k_L}{2} \|\mathbf{u}^n\|_2 + \|\mathbf{V}(\mathbf{u}^n)\|_1. \quad (50)$$

A numerical power balance that holds under time-invariant parameters is obtained by first pre-multiplying (31) with Ξ^{-1} and, subsequently, the left-hand side with $(\mu\mathbf{q})^T$ and the right-hand side by $(\delta\mathbf{u})^T$, which yields

$$\frac{\delta H}{\Delta_t} = \frac{H^{n+1} - H^n}{\Delta_t} = P^{n+\frac{1}{2}} - Q^{n+\frac{1}{2}}, \quad (51)$$

where $P^{n+\frac{1}{2}} = \frac{1}{2}\Delta_t^{-1}\mathbf{G}_e^T\delta\bar{\mathbf{u}}\mathbf{F}_e^{n+\frac{1}{2}}$ is the input power and

$$Q^{n+\frac{1}{2}} = \frac{(\delta\mathbf{u})^T \Xi^{-1} \mathbf{B} \delta\bar{\mathbf{u}}}{\Delta_t} + r_b \frac{(\delta\mathbf{u})^T \delta\mathbf{u}}{\Delta_t^2} + r_d \frac{(\delta u_d)^2}{\Delta_t^2} \geq 0 \quad (52)$$

is the dissipated power. Hence in the absence of input power, the system energy cannot grow. The fact that all terms in (50) are non-negative and either quadratic or monotone functions of one of the variables implies bounds on u^n and q^n , confirming numerical stability with no further conditions on the temporal step Δ_t . The above power balance does not hold under time variation of the system parameters. Hence under continuous parameter update, energy growth can potentially occur if the power injected by such variation outstrips the dissipation; our observations in off-line experiments are that this occurs only when parameters are varied at rates much higher than required for parametric control. A more thorny issue potentially arises when, during real-time operation, the iteration count spikes due to fast parameter sweeps. This may lead to (47) not being solved to high accuracy for a number of time instants, which in turn can cause violations of the power balance. The associated instability risk is currently managed in ad-hoc fashion by empirically constraining the rate at which parameters are varied (employing low-pass filtering at control rate of all the parameter signals).

4. PARAMETER SPACE EXPLORATION

4.1. Control Parameters

Exposing the user to the full set of parameters featuring in (1-3) makes it unnecessarily difficult to learn navigating the parameter space because of parameter redundancy. Without loss of generality, the parameter set is therefore reduced here by constraining the length parameters to $L_s = L_x L_y = 1$ and fixing the string mass per unity length at $\rho_s A_s = 0.001\text{kg/m}$. The following parameters

Table 1: Tunable parameters, the constraints imposed upon them for real-time operation, and example values.

		Fig. 4	Fig. 5	Fig. 6
STRING				
fundam. freq. [Hz]	$0 < \tilde{f}_s < \frac{\pi}{\Delta_t}$	100	47.3	80
inharmonicity coeff.	$0 \leq \mathcal{B}_s$	10^{-5}	10^{-5}	10^{-5}
damping [s^{-1}]	$0 \leq \sigma_{s,0}$	1	2	0.5
damping [m/s]	$0 \leq \sigma_{s,1}$	10^{-3}	$4 \cdot 10^{-4}$	10^{-2}
damping [m^3/s]	$0 \leq \sigma_{s,3}$	10^{-5}	$4 \cdot 10^{-6}$	10^{-4}
damper [s^{-1}]	$0 \leq \sigma_d$	500	0	0
connection position	$0 < z'_c < 1$	0.87	0.93	0.98
damper position	$0 < z'_d < 1$	0.99	0.99	0.99
excitation position	$0 < z'_x < 1$	0.07	0.5	0.5
BRIDGE				
modal mass ratio	$10^{-4} < R_{bs}$	6 0.6	10^{-4}	1
damping [s^{-1}]	$0 \leq \zeta_b$	1	2	10^{-2}
stiffness	$0 \leq k_b \leq 10^6$	10^5	10^5	10^6
nonlinearity	$0 \leq \eta \leq 1$	0	$0 1$	1
nonlinear exponent	$1 \leq \alpha \leq 3$	1	3	1.1
gravity [m/s ²]	$ g_b \leq 10$	0	0	-0.5
spring push level	$0 \leq G_1^+ \leq 1$	1	1	1
spring pull level	$0 \leq G_1^- \leq 1$	1	1	0
spring push level	$0 \leq G_2^+ \leq 1$	1	1	1
spring pull level	$0 \leq G_2^- \leq 1$	1	1	0
PLATE				
fundam. freq. [Hz]	$0 < \tilde{f}_p < \frac{\pi}{\Delta_t}$	17.7	50	30
dimensional ratio	$0 < R_{xy}$	0.89	0.98	0.77
modal mass ratio	$0 < R_{ps}$	10	1	10
damping [s^{-1}]	$0 \leq \sigma_{p,0}$	20	2	4
damping [m/s]	$0 \leq \sigma_{p,1}$	10^{-4}	$4 \cdot 10^{-4}$	10^{-2}
damping [m^3/s]	$0 \leq \sigma_{p,3}$	10^{-6}	$4 \cdot 10^{-6}$	10^{-4}
connection position	$0 < x'_c < 1$	0.61	0.17	0.61
connection position	$0 < y'_c < 1$	0.50	0.11	0.43
pickup position	$0 < x'_{a,k} < 1$	0.13	0.13	0.13
pickup position	$0 < y'_{a,k} < 1$	0.93	0.93	0.93

are then introduced with the aim of enabling intuitive control of the string, bridge, and plate characteristics:

$$\tilde{f}_s = f_{s,1} = \frac{1}{2} \sqrt{\left(\frac{E_s I_s}{\rho_s A_s} \right) \pi^2 + \left(\frac{T_s}{\rho_s A_s} \right)}, \quad (53)$$

$$\mathcal{B}_s = \pi^2 \frac{E_s I_s}{T_s}, \quad \zeta_b = \frac{r_b}{2m_s}, \quad z'_\kappa = z_\kappa \ (\kappa = x, c, d), \quad (54)$$

$$R_{bs} = \frac{m_b}{m_s}, \quad \zeta_b = \frac{r_b}{2m_b}, \quad R_{ps} = \frac{m_p}{m_s}, \quad R_{xy} = \frac{L_x}{L_y}, \quad (55)$$

$$\tilde{f}_p = f_{p,1,1} = \frac{1}{2} \sqrt{\frac{G_p \pi^2}{\rho_p h_p} (L_x^{-2} + L_y^{-2})}, \quad (56)$$

$$x'_c = \frac{x_c}{L_x}, \quad y'_c = \frac{y_c}{L_y}, \quad x'_{a,k} = \frac{x_{a,k}}{L_x}, \quad y'_{a,k} = \frac{y_{a,k}}{L_y}. \quad (57)$$

Furthermore, the connection spring stiffness constants in (4) are parameterised as follows:

$$k_L = (1 - \eta)k_b, \quad k_\ell^\pm = \eta k_b G_\ell^\pm \cdot 10^{4(\alpha-1)}, \quad (58)$$

where k_b is an overall bridge stiffness parameter, $0 \leq G_\ell^\pm \leq 1$ set the relative push and pulling levels of the spring and $0 \leq \eta \leq 1$ gives control over the level of nonlinear behaviour. The formulation in (58) helps ensuring, in combination with the constraints imposed on the parameters, that the overall stiffness does not exceed

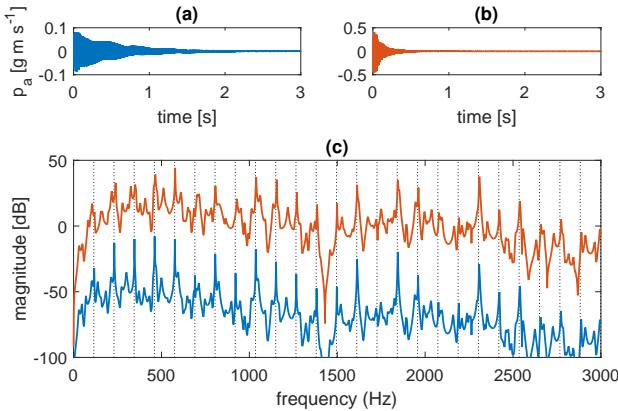


Figure 4: Example of linear coupling between the string, bridge and plate (see Table I for the model parameters). (a): large bridge mass ($R_{bs} = 6$). (b): small bridge mass ($R_{bs} = 0.6$). (c): corresponding magnitude spectra. The upper curve ($R_{bs} = 0.6$) is offset by 50dB for clarity. The vertical dotted lines indicate the positions of the string mode frequencies.

certain levels which would cause an excessive amount of iterations to be completed by the iterative solver. Finally, the external bridge force is recast here as a gravitational force $F_b(t) = g_b m_b$, where g_b is a gravitational acceleration value that can be varied at control rate by the user. A full list of tunable parameters is shown in Table 1, including the applied value constraints.

4.2. Numerical Experiments

The simplest configuration of the bridge is to use strictly linear spring connections (i.e. $\eta = 0$). The plots in Fig. 4 show signals and magnitude spectra of such a case when driving the string with a short smooth pulse. With a relative large bridge mass (setting $R_{bs} = 6$), strong standing waves develop in the string, and the transfer of energy is largely uni-directional, i.e. from the string to the plate, much in the way conventional string instruments operate. As can be seen in the corresponding spectrum (lower curve in Fig. 4(c)), the output also exhibits plate modes, which are more damped than the string resonances. Setting the bridge mass to a smaller value ($R_{bs} = 0.6$) leads to increased coupling between the plate and the string, and the system energy is then dissipated more quickly (see Fig. 4(b)), yielding an output in which the string modes are less dominant over the plate modes (see upper curve of Fig. 4(c)).

The notion of effecting strong string-plate coupling can be taken to its extreme by setting R_{bs} to a negligibly small value and R_{ps} to unity, while using similar damping values for the plate and string. The coupling is now very much bi-directional, and any standing waves on the string no longer correspond to the string eigenmodes. Although different than that of the modelled plate, the distribution of system modes across the frequency axis is nevertheless similar to that of a plate, so the perceived sound output is plate-like. Hence in this configuration, the string is effectively merely a mechanical interface to a plate-like instrument. An interesting feature to explore with this type of connection is to introduce nonlinear behaviour by varying the η parameter. Fig. 5 shows the spectrograms resulting for $\eta = 0$ (linear springs) and $\eta = 1$.

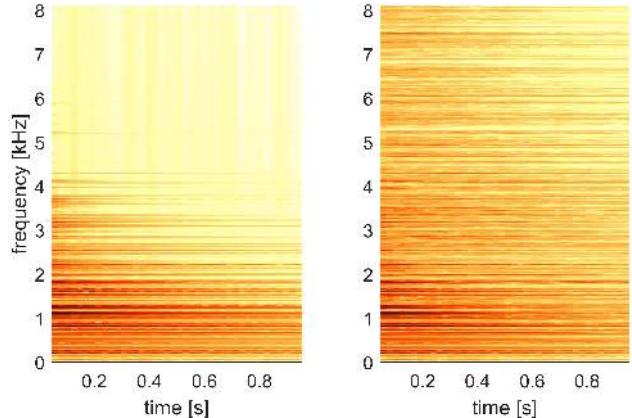


Figure 5: Example of plate-like sounds via strong string-plate coupling (see Table I for the model parameters). Left: plate momentum spectrogram for $\eta = 0$. Right: plate momentum spectrogram for $\eta = 1$.

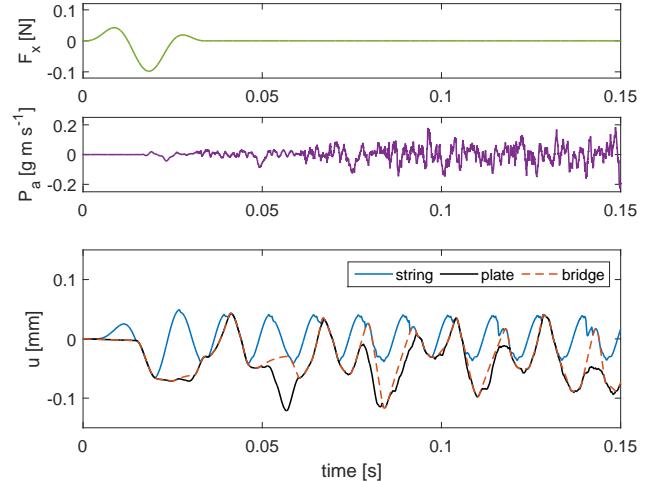


Figure 6: Example of a rattling bridge (see Table I for the model parameters). Top: string excitation force signal. Middle: plate momentum monitored at the pick-up position. Bottom: displacement at the connection point of the string, plate, and bridge mass.

(fully nonlinear springs), with $\alpha = 3$. As can be seen, the second case exhibits quick growth of densely spaced high-frequency partials, indicating strong nonlinear inter-mode coupling, and yielding a somewhat cymbal-like sound output.

An altogether different type of nonlinear behaviour is explored when imparting asymmetries in either or both of the spring connections, which introduces rattling effects. For example, setting $G_1^- = G_2^- = 0$ and $G_1^+ = G_2^+ = 1$ amounts to a bridge that is in contact with the plate and the string at equilibrium but that - in the absence of forces that pull it back to either object when it moves away from them - is free to rattle during vibration. Special effects are obtained when letting the string vibrate at a low frequency, as such enabling the generation of slowly evolving rattling patterns which partly play out at sub-sonic rates. An example case is demonstrated in Fig. 6, with the lower plot showing the complex motion of the bridge mass and its impactive interaction

```
// initialise output value with zero
pa = 0;
// loop over plate modes, index starts at Ms+1
for (int i = 0; i < Mp; i++) {
    pa += *(wa + i) * *(q_modes + Ms + 1 + i);
}
```

Figure 7: C++ code for implementing (39) for a single output.

```
// initialise __mm256 block with zeros
block = _mm256_set_pd(0, 0, 0, 0);
// loop over plate modes, sum blocks of four
for (int i = 0; i < Mp; i += 4) {
    block = _mm256_fnmadd_pd(_mm256_load_pd(&wa[i]),
        _mm256_load_pd(&q_modes[Ms+1+i]), block);
}
// store final block results and sum over them
temp = _mm256_store_pd(block);
pa = temp[0] + temp[1] + temp[2] + temp[3];
```

Figure 8: AVX code for implementing (39) for a single output.

with the string and plate, in response to a short windowed 40 Hz sine wave. For the chosen parameters (including gravity), the resulting sound is somewhat reminiscent of a snare drum roll. More generally, this type of configuration produces rattling and buzzing patterns of a semi-chaotic nature. Sound examples of all three of the above cases are available on the companion website¹ alongside various further explorative sounds and supporting material.

4.3. Real-Time Implementation

For real-time rendering, the system was built in Audio Unit plugin architecture, coding in C++ within the JUCE framework [16]. JUCE yields executable code within a standard plug-in API provided by Core Audio, and supports a host of plugin formats. In order to increase the number of modes that can be run in real-time, a second version of the code uses Advanced Vector Extensions (AVX) for the parts of the code that loop over M_s string modes or M_p plate modes. This allows performing arithmetic operations over multiple variables simultaneously; for double precision on standard processors, most of which currently use 256 bit registers, this means the code can operate on four modes at a time. To exemplify the coding difference, Figs. 7 and 8 show the instructions used in C++ and in AVX, respectively, for implementing the matrix operation in (39) when running the system with a single audio output. The iterative solver, which does not permit similar parallelisation, takes up to about 5% of the computation when the exit condition chosen as $\|\mathbf{s}_{j+1} - \mathbf{s}_j\| < 10^{-15}$, where j indicates the iteration.

The run-time computations also include re-calculation of system coefficients, such as the non-zero elements of \mathbf{A} , \mathbf{C} , \mathbf{h}_c , and \mathbf{h}_x , according to parameter changes made by the user at regular intervals. The shorter the audio buffer size, the faster the parameters can be varied by the user without significant artefacts, but the more this adds to the overall load. Table 2 lists the results of testing the number of modes that each code version can render in real-time without underflow when fixing the ratio between the number of string and plate modes as $M_p \approx 4M_s$, for different buffer sizes. Both codes were compiled with optimisation level -O3 and using AVX2. The first row shows the maximum number of modes when not performing any parameter updates during runtime. Similar to

Table 2: Maximum number of system modes ($M = M_p + M_s + 1$) for the plugin running at 44.1 kHz when choosing $M_p \approx 4M_s$.

audio buffer length (samples)	C++	C++ with AVX
no updates	3207	7037
512	2930	6505
256	2637	5925
128	2476	5253
64	2171	4571

the findings by Webb and Bilbao [8] for a finite difference based system, the AVX instructions accelerate the computations by about a factor two for double-precision floating point. The viability of single-precision AVX acceleration (which can be expected to provide a 4 times speed up, but may introduce round-off noise) is currently being investigated. Note that for both the C++ and the C++ with AVX code, the parameter updates were written entirely in C++. Further investigation could establish whether this part of the code can also be optimised by replacing C++ code with AVX instructions.

A relatively large number of string modes is chosen here because this enables interesting coupling phenomena. Experiments have indicated that a total of 5000 system modes is sufficient to generate a musically appropriate range of plate and string sizes (indeed, each of the sound examples available on the companion webpage were generated with this number of system modes). For a 44.1 kHz sample rate, this means that the mode series of larger plates is truncated before Nyquist, but - as reported in [11] - the resulting loss of high-frequency detail is not necessarily perceptually significant.

4.4. Control Interface

To facilitate ergonomic real-time control, each of the parameters is knob-controlled using a Knobbee board [17], which supports OSC messages at 10 bit resolution. Besides navigating the parameter space in search of interesting sounds, the user can employ the board to make gestures via parameter changes, many of which are not possible or difficult to achieve with a real-world counterpart of the model. The string can be excited either using a pre-stored signal or more interactively, using a silent string controller object based on the design proposed in [18]. Elsewhere [19] we report preliminary findings regarding the artistic use of these control interfaces with an earlier, simpler, bridge-less version of the model.

5. CONCLUSIONS AND PERSPECTIVES

The string-bridge-plate model does not enable faithful emulation of any existing musical instrument; instead, it offers synthesis of a range of sounds of an inherently mechano-acoustic nature. The novelty resides mainly in the design of the bridge, which incorporates parametrically detachable nonlinear spring connections on either side of the bridge mass; this is modelled numerically in a way that allows real-time simulation of the complex, semi-chaotic rattling and buzzing that entails when reconfiguring these connections on the fly. The realisation of this design extends on what existing real-time physical model implementations and environments currently offer in terms of contact dynamics. Further versatility derives from the ability to generate sounds with harmonic (string-

¹<http://www.socasites.qub.ac.uk/mvanwalstijn/dafx17a/>

like) as well as inharmonic (plate- or beam-like) spectra and from the control over the level of inter-object coupling through adjusting the modal mass values.

The tunability of the parameters indirectly relies on the modal expansion approach, which - as explained in § 3.1 - enables eliminating the dispersion and attenuation errors inherent to the unconditionally stable numerical scheme. The modal form has further beneficial features, in that (a) it facilitates direct control of frequency-dependent damping, (b) the algorithmic efficiency is greatly helped by the sparsity of the matrices \mathbf{G}_κ and \mathbf{H}_κ and the diagonality of matrices \mathbf{A} and \mathbf{C} , which is systematically exploited in our implementations of the system, and (c) the number of degrees of freedom (in the present model, the number of modes) can be reduced without affecting the audio bandwidth or introducing numerical dispersion, making the model's computational load easily scalable to the available processing power (though high-frequency detail is lost for larger strings and plates).

There are, however, also limitations to consider. Firstly, a relative large number of system coefficients needs to be regularly updated (compared to, e.g., finite difference schemes). Secondly, the parameter update remains simple and efficient only for objects for which the modal expansion is available in closed-form (e.g. a cantilever beam or a circular membrane), hence extending to more complex and varied boundary conditions is not straightforward. Thirdly, inputs, outputs, and connections are relatively expensive within a modal framework, due to the dot products required to translate between modal and spatial coordinates. The latter is of particular relevance if the generalised form of eqs. (30, 31) were to be taken as the basis for a modular environment.

It is also worthwhile briefly reflecting on the use of a Newton solver in simulating a system featuring multiple nonlinearities, which can present significant challenges, especially when non-smooth forces are involved. As discussed in § 3.4, the form of the nonlinear equation arising in the proposed model has the benefit of global convergence to a unique solution, while placing constraints on the system parameters provides an empirical handle on the iteration count; a useful addition would be to establish a theoretical iteration bound. Altogether this approach goes a long way towards ensuring computational robustness, which is paramount in a real-time synthesis context. Even better robustness would require either an explicit scheme, which - as far as the authors are aware - does not exist in provably stable form for the problem at hand, or an analytically solvable implicit form. The latter does not seem forthcoming either, but does exist for certain simplifications of (4). Further variations, possibly in combination with different discretisation choices, may offer different trade-offs than the ones made in the present study (e.g. sacrificing full tunability for higher efficiency), and as such are very much worth exploring.

The properties of the string-bridge-plate model are well aligned with design and control tasks, both of which can be performed through navigation of a continuous parameter space in real-time. As such, it offers new opportunities for creative application and control, with potential use in music performance and live improvisation. Hence a natural way forward with research on this topic is to investigate possible extensions, improvements, and variations of the model and its implementation in tandem with developing (and experimenting with) dedicated control interfaces, conducted in close collaboration with performers. The first live performance featuring the string-bridge-plate instrument took place at NIME 2017 [20].

6. REFERENCES

- [1] J. O. Smith, "Virtual acoustic musical instruments: Review and update," *Journal of New Music Research*, vol. 33, no. 3, pp. 283–304, 2004.
- [2] C. Cadoz, A. Luciani, and J. L. Florens, "CORDIS-ANIMA: A Modeling and Simulation System for Sound and Image Synthesis: The General Formalism," *Computer Music Journal*, vol. 17, no. 1, pp. 19–29, 1993.
- [3] J. Derek Morrison and J. M. Adrien, "MOSAIC: A Framework for Modal Synthesis," *Computer Music Journal*, vol. 17, no. 1, pp. 45–56, 1993.
- [4] M. Pearson, "TAO: a physical modelling system and related issues," *Organised Sound*, vol. 1, no. 1, pp. 43–50, 1996.
- [5] S. Bilbao, "A Modular Percussion Synthesis Environment," in *Proc. of the 12th Int. Conf. on Digital Audio Effects (DAFX-09)*, 2009.
- [6] A. S. Allen, *Ruratae: A Physics-Based Audio Engine*, Ph.D. thesis, University of California, San Diego, 2014.
- [7] J. Leonard and C. Cadoz, "Physical Modelling Concepts for a Collection of Multisensory Virtual Musical Instruments," in *New Interfaces for Musical Expression 2015*, 2015, pp. 150–155.
- [8] C. Webb and S. Bilbao, "On the Limits of Real-Time Physical Modelling Synthesis with a Modular Environment," in *Proc. of the 18th Int. Conf. on Digital Audio Effects (DAFX-15)*, 2015.
- [9] S. Bilbao and J. Ffitch, "Prepared Piano Sound Synthesis," in *Proc. of the 9th Int. Conf. on Digital Audio Effects (DAFX-06)*, 2006.
- [10] M. Schäfer, P. Frenstátský, and R. Rabenstein, "A Physical String Model with Adjustable Boundary Conditions," in *Proc. of the 19th Int. Conf. on Digital Audio Effects (DAFX-16)*, 2016, pp. 159–166.
- [11] S. Orr and M. van Walstijn, "Modal Representation of the Resonant Body within a Finite Difference Framework for Simulation of String Instruments," in *Proc. of the 17th Int. Conf. on Digital Audio Effects (DAFX-09)*, 2009.
- [12] M. van Walstijn, J. Bridges, and S. Mehes, "A Real-Time Synthesis Oriented Tanpura Model," in *Proc. Int. Conf. Digital Audio Effects (DAFx-16)*, 2016, pp. 175–182.
- [13] A. Falaize and T. Hélie, "Guaranteed-Passive Simulation of an Electro-Mechanical Piano: A Port-Hamiltonian Approach," in *Proc. of the 18th Int. Conf. on Digital Audio Effects (DAFX-15)*, 2015.
- [14] M. van Walstijn and J. Bridges, "Simulation of Distributed Contact in String Instruments: a Modal Expansion Approach," in *Proc. Europ. Sig. Proc Conf (EUSIPCO2016)*, 2016, pp. 1023–1027.
- [15] P. Deuflhard, *Newton methods for nonlinear problems: affine invariance and adaptive algorithms*, Springer, 2004.
- [16] M. Robinson, *Getting Started with JUCE*, Packt Publishing Ltd., Birmingham, UK, 1st edition, 2013.
- [17] C. Popp and R. Soria-Luz, "Developing Mixer-style Controllers Based On Arduino/Teensy Microcontrollers," in *The 12th Sound and Music Computer Conference*, Maynooth, 2015, pp. 37–41.
- [18] S. Mehes, M. van Walstijn, and P. Stapleton, "Towards a Virtual-Acoustic String Instrument," in *13th Sound and Music Computing Conference*, Hamburg, 2016, pp. 286–292.
- [19] S. Mehes, M. van Walstijn, and P. Stapleton, "Virtual-Acoustic Instrument Design: Exploring the Parameter Space of a String-Plate Model," in *New Interfaces for Musical Expression*, Copenhagen, 2017, pp. 399–403.
- [20] P. Stapleton and A. Pultz Melbye, "VASPI Performance," in *New Interfaces for Musical Expression*, Copenhagen, 2017.

MODAL BASED TANPURA SIMULATION: COMBINING TENSION MODULATION AND DISTRIBUTED BRIDGE INTERACTION

Jamie Bridges, Maarten Van Walstijn

Sonic Arts Research Centre

School of Electronics, Electrical Engineering, and Computer Science

Queen's University Belfast, UK

{jbridges05,m.vanwalstijn}@qub.ac.uk

ABSTRACT

Techniques for the simulation of the tanpura have advanced significantly in recent years allowing numerically stable inclusion of bridge contact. In this paper tension modulation is added to a tanpura model containing a stiff lossy string, distributed bridge contact and the thread. The model is proven to be unconditionally stable and the numerical solver used has a unique solution as a result of choices made in the discretisation process. Effects due to the distribution of the bridge contact forces by comparison to a single point bridge and of introducing the tension modulation are studied in simulations. This model is intended for use in furthering the understanding of the physics of the tanpura and for informing the development of algorithms for sound synthesis of the tanpura and similar stringed instruments.

1. INTRODUCTION

The tanpura is an Indian stringed instrument known for its distinctive droning sound which features a time-dependent formant of frequencies referred to as the "jvari". The generation of the jvari depends heavily on the interaction between the strings and the bridge over which they pass. The jvari also relies strongly on the position of a thread which is placed between each string and the bridge. This changes how much of the string can interact with the bridge and only when placed within a certain range of positions will the tanpura produce the jvari [1]. The sound of the jvari can be altered by moving the thread within the range.

The interaction between barriers and strings is not unique to this instrument; models of the guitar [2], violin [3] and many others have been developed which include this phenomenon. The non-analytic nature of contact forces requires some care to be taken to ensure that any model including them will be stable. Energy methods proved to be an effective way to maintain stability [4, 5, 6] by ensuring that the numerical energy or some analogous quantity is conserved for all time steps when there are no energy losses or gains from external factors. Within the umbrella of energy methods there are differing approaches to creating models; differentiation in time can be carried out with different approximations giving a temporal second order form [7] or a temporal first order form [5]. Models of stringed instruments have made use of finite difference methods [8, 5] as well as modal based methods [9]. The latter allow completely eliminating numerical dispersion by exactly fixing the modal frequencies and damping which removes mode detuning [9]. This is particularly important in the case of the tanpura as the jvari is sensitive to any numerical warping of the partial structure. In addition to unconditional stability it is possible to ensure that the numerical solver has a unique solution [9] which in conjunc-

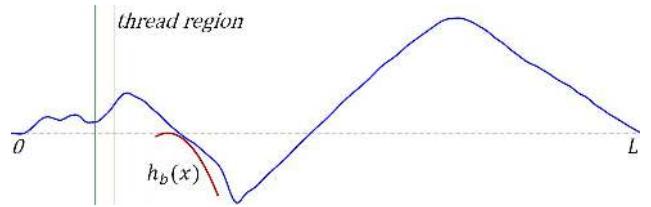


Figure 1: A simplified sketch of the tanpura model with the thread and bridge zoomed in on significantly.

tion with the stability allows the model to be run at standard audio sampling frequencies and maintain global trends.

In this paper tension modulation and a distributed bridge are included in a model of the tanpura. A sketch of the model is shown in Figure 1. Both of these effects have been modelled in musical acoustics before but they have not yet been brought together in a tanpura model. Tension modulation has been recently considered in the modelling of Portuguese twelve string guitars [10], the guitar [2] and more in depth for the general case [11]. However, none of these treatments have all of the desired attributes of having a modal basis, uniqueness of the iterative solver solution and provable stability. As this is the case a new formulation is developed here which brings all of these aspects together. The effect of the tension modulation in a string becomes particularly relevant when strings have a low tension (as they tend to on a tanpura) and/or a high Young's modulus and cross sectional radius. The manner in which tension modulation was included will be more discussed in more detail in Section 2.5.

The tanpura's bridge has traditionally been described as a two point bridge, meaning that the string is assumed to terminate at the thread and interacts at only one point on the bridge [12]. A recent paper written by Issanchou et al. [13] demonstrates an impressive agreement between experiments and a time-domain model for a two point bridge system, which is an important step in the validation of models which include non linear contact. This result was found for a high tension of 180.5N and using a guitar string whereas a more common tension for a tanpura string is between 30N and 40N. Hence questions remain regarding as to whether there is a difference between distributed bridge and point bridge interaction at lower tensions and what difference it would make to the jvari. An approach similar to that in [14] here is developed which involves not treating the thread as an end point but as a damper. Following techniques used there the thread and the plucking signal are modelled as single point forces but with a small spatial distribution associated with them. The incorporation of the distributed bridge is outlined in Section 2.7.

2. METHOD

2.1. Overview

The Newtonian form of the equation which describes the desired system in terms of the string's vertical displacement $y(x, t)$ (written as y for brevity), horizontal position x and time t can be written

$$\rho A \frac{\partial^2 y}{\partial t^2} = T_0 \frac{\partial^2 y}{\partial x^2} - EI \frac{\partial^4 y}{\partial x^4} - \gamma \frac{\partial y}{\partial t} + \mathcal{F}_{\text{tm}} + \mathcal{F}_{\text{c}} + \mathcal{F}_{\text{b}} + \mathcal{F}_{\text{e}}, \quad (1)$$

where ρ represents the mass density, A the cross sectional area, E the Young's modulus, I the moment of inertia and T_0 the resting tension of the string. γ gives the effect of frequency dependent damping and will be defined later. The \mathcal{F} terms on the right-hand side of the equation represent additional force densities which act on the string. Respectively these are due to tension modulation (\mathcal{F}_{tm}), the cotton thread (\mathcal{F}_{c}), the bridge (\mathcal{F}_{b}) and the excitation (\mathcal{F}_{e}). All of these are functions of both x and t and are defined over the length of the string (L), although some of them only have an effect over a small range of x . The string is assumed to be simply supported for the boundary conditions.

2.2. Modal version

In order to eliminate numerical dispersion a modal framework is used. This involves using the equation

$$y(x, t) = \sum_{i=1}^M v_i(x) \bar{y}_i(t), \quad (2)$$

where $\bar{y}_i(t)$ are the modal displacements, $v_i(x) = \sin(\beta_i x)$ are the modal shapes and $\beta_i = \frac{i\pi}{L}$. For this to be a perfect representation of the string the number of modes (M) would be infinity.

After substituting equation (3) into (1) and integrating spatially over the string the following equation is found which describes the dynamics of a single mode

$$m \frac{\partial^2 \bar{y}_i(t)}{\partial t^2} = -k_i \bar{y}_i(t) - r_i \frac{\partial \bar{y}_i(t)}{\partial t} + \sum_{\epsilon} \bar{F}_{\epsilon,i}(t), \quad (3)$$

where $m = \frac{\rho A L}{2}$, $k_i = \frac{L}{2}(T_0 \beta_i^2 + EI \beta_i^4)$, $r_i = \frac{L}{2}\gamma(\beta_i)$ and $\epsilon = \{\text{tm}, \text{c}, \text{b}, \text{e}\}$. $\gamma(\beta_i)$ is now expressed as

$$\gamma(\beta_i) = 2\rho A [\sigma_0 + (\sigma_1 + \sigma_3 \beta_i^2)|\beta_i|]. \quad (4)$$

σ_0 , σ_1 and σ_3 shape the frequency dependence to match to a real string as closely as possible [14]. $\bar{F}_{\epsilon,i}$, the modal driving forces, are derived in different ways according to how the physical effect being considered is modelled. Each one will be explained in its relevant section.

Equation (1) can be rewritten in first-order form, which is exactly equivalent, to read

$$\frac{\partial \bar{y}_i(t)}{\partial t} = \frac{\bar{p}_i(t)}{m} \quad (5)$$

$$\frac{\partial \bar{p}_i(t)}{\partial t} = -k_i \bar{y}_i(t) - r_i \frac{\partial \bar{y}_i(t)}{\partial t} + \sum_{\epsilon} \bar{F}_{\epsilon,i}(t) \quad (6)$$

Where $\bar{p}_i(t)$ is the momentum equivalent of the i^{th} mode.

2.3. Single forces of fixed spatial distribution

Some of the force densities will be represented as a single-variable force term which is assumed to be spatially applied to the string according to a fixed distribution function. This is affected by expressing the force densities as the force ($F_z(t)$) multiplied by a spatial distribution function in the following fashion

$$\mathcal{F}_z(x, t) = \psi_z(x) F_z(t), \quad (7)$$

where the spatial distribution $\psi_z(x)$ is defined as

$$\psi_z(x) = \frac{\pi}{2w_z} \cos \left[\frac{\pi}{w_z} (x - x_z) \right], \quad (8)$$

inside the domain $[x_z - \frac{1}{2}w_z, x_z + \frac{1}{2}w_z]$ and 0 otherwise [14]. w_z is the width of the spatial distribution and x_z the central point. The forces concerned will depend on the averaged string displacement at the point x_z , which can be written as

$$y_z(t) = \int_{x_z - w_z/2}^{x_z + w_z/2} \psi_z(x) y(x, t) dx \quad (9)$$

Evaluating equation (9) and converting to the modal form, the following equations can be found

$$\bar{F}_{z,i}(t) = g_{z,i} F_z(t), \quad (10)$$

and

$$y_z(t) = \sum_{i=1}^M g_{z,i} \bar{y}_i(t), \quad (11)$$

where

$$g_{z,i} = \frac{\pi^2 \sin(\beta_i x_z) \cos(\frac{\beta_i w_z}{2})}{\pi^2 - \beta_i^2 w_z^2}. \quad (12)$$

2.4. Discretisation in time

To solve equations (5) and (6) the following discretisation operators are used to find the system dynamics

$$\delta \phi(t) = \phi^{n+1} - \phi^n \approx \Delta t \frac{\partial \phi(t)}{\partial t} \Big|_{t=(n+\frac{1}{2})\Delta t} \quad (13)$$

$$\mu \phi(t) = \phi^{n+1} + \phi^n \approx 2\phi \Big|_{t=(n+\frac{1}{2})\Delta t}, \quad (14)$$

using $\Delta t = 1/f_s$. When discretising equations (5) and (6) it is more compact to use the substitution $\bar{q}_i^n = \frac{\Delta t}{2m} \bar{p}_i^n$ to give the equations

$$\delta \bar{y}_i^{n+\frac{1}{2}} = \mu \bar{q}_i^{n+\frac{1}{2}} \quad (15)$$

$$\delta \bar{q}_i^{n+\frac{1}{2}} = -a_i \mu \bar{y}_i^{n+\frac{1}{2}} - b_i \delta \bar{y}_i^{n+\frac{1}{2}} + \xi \sum_{\epsilon} \bar{F}_{\epsilon,i}^{n+\frac{1}{2}} \quad (16)$$

at the temporal mid point. The constants are defined as $\xi = \Delta t^2/(2m)$, $a_i = k_i \Delta t^2/(4m)$ and $b_i = r_i \Delta t/(2m)$. The individual cases of $\bar{F}_{\epsilon,i}^{n+\frac{1}{2}}$ will be explained in their relevant section. To correct for numerical dispersion a_i and b_i are replaced by a_i^* and b_i^* ; this enforces exact modal frequencies and damping for each mode [14]. The corrected parameters read

$$a_i^* = \frac{1 - 2R_i \Omega_i + R_i^2}{1 + 2R_i \Omega_i + R_i^2} \quad \text{and} \quad b_i^* = \frac{2(1 - R_i^2)}{1 + 2R_i \Omega_i + R_i^2}, \quad (17)$$

where $R_i = \exp(-\alpha_i \Delta t)$ and $\Omega_i = \cos(\omega_i \Delta t)$.

To solve equations (15) and (16) computationally they are rewritten as vectors in the following form:

$$\delta\bar{\mathbf{y}}^{n+\frac{1}{2}} = \mu\bar{\mathbf{q}}^{n+\frac{1}{2}} \quad (18)$$

$$\delta\bar{\mathbf{q}}^{n+\frac{1}{2}} = -\mathbf{A}\mu\bar{\mathbf{y}}^{n+\frac{1}{2}} - \mathbf{B}\delta\bar{\mathbf{y}}^{n+\frac{1}{2}} + \xi \sum_{\epsilon} \bar{\mathbf{F}}_{\epsilon,i}^{n+\frac{1}{2}} \quad (19)$$

In these equations $A_{ii} = a_i^*$ and $B_{ii} = b_i^*$ with all off-diagonal elements in both matrices being zero. Using the same procedure taken in previous papers [14] of setting $\bar{s} = \delta\bar{\mathbf{y}}^{n+\frac{1}{2}} = \mu\bar{\mathbf{q}}^{n+\frac{1}{2}}$ and then using the relations:

$$\bar{\mathbf{y}}^{n+1} = \bar{s} + \bar{\mathbf{y}}^n \quad \text{and} \quad \bar{\mathbf{q}}^{n+1} = \bar{s} - \bar{\mathbf{q}}^n, \quad (20)$$

equation (19) can be re-written as:

$$\mathbf{G} = (\mathbf{I} + \mathbf{A} + \mathbf{B})\bar{s} - 2(\bar{\mathbf{q}}^n - \mathbf{A}\bar{\mathbf{y}}^n) - \xi \sum_{\epsilon} \bar{\mathbf{F}}_{\epsilon,i}^{n+\frac{1}{2}} = \mathbf{0}, \quad (21)$$

Once the tension modulation, bridge, thread and input modal forces are discretised properly and put into this equation it can be solved iteratively using Newton's method. The Jacobian required for this is

$$\mathbf{J} = \mathbf{J}_s - \xi \sum_{\epsilon} \mathbf{J}_{\epsilon}, \quad (22)$$

where

$$\mathbf{J}_s = \mathbf{I} + \mathbf{A} + \mathbf{B}, \quad (23)$$

and \mathbf{J}_{ϵ} are the parts of the Jacobian due to the other forces. All of the parts of the Jacobian will be derived below.

To prevent spectral fold over in the model the number of modes (M) must be limited, setting the sizes of $\bar{\mathbf{y}}$ and $\bar{\mathbf{q}}$, so that the frequency of the highest partial is below the Nyquist frequency set by the sampling frequency.

2.5. Tension modulation

2.5.1. Continuous domain

The force density present in the string due to tension modulation is represented in the following equation

$$\mathcal{F}_{tm}(x, t) = \frac{EA}{2L} \int_0^L \left(\frac{\partial y}{\partial x} \right)^2 dx \frac{\partial^2 y}{\partial x^2}. \quad (24)$$

Following the procedure for converting to the modal domain this can be rewritten as

$$\mathcal{F}_{tm}(x, t) = -\frac{EA}{4} \left[\sum_{i=1}^M \bar{y}_i(t) v_i(x) \beta_i^2 \right] \left[\sum_{j=1}^M (\beta_j \bar{y}_j(t))^2 \right]. \quad (25)$$

This is found using the orthogonal nature of the sinusoidal basis set within the integral. From this the desired modal force can be derived by multiplying with the modal shape and integrating over the string giving

$$\bar{F}_{tm,i}(t) = -\Gamma \bar{y}_i(t) \beta_i^2 \left[\sum_{j=1}^M (\beta_j \bar{y}_j(t))^2 \right], \quad (26)$$

where $\Gamma = \frac{EAL}{8}$. This equation shows how the modes are mixed, with tension modulation having the effect of diffusing energy between modes. This is then rewritten in vector form to give:

$$\bar{\mathbf{F}}_{tm} = -\Gamma(\beta_2 \bar{\mathbf{y}}) \beta_2 \bar{\mathbf{y}}. \quad (27)$$

Where β_2 is a diagonal $M \times M$ matrix with entries along the diagonal of β_i^2 .

2.5.2. Discretisation of tension modulation equation in time

Equation (27) can be discretised in three ways and equation (28) shows the option that is used. This choice was made as it preserves the uniqueness of the solution to the iterative solver, this will be proven in Section 3.1.

$$\bar{\mathbf{F}}_{tm}^{n+\frac{1}{2}} = -\frac{\Gamma}{4}(\mu(\bar{\mathbf{y}}^T \beta_2 \bar{\mathbf{y}})) \beta_2 \mu \bar{\mathbf{y}}. \quad (28)$$

After the discretisation operator is applied and using equation (20) the following equation is obtained

$$\bar{\mathbf{F}}_{tm}(\bar{s}) = -\frac{\Gamma}{4}((\bar{\mathbf{y}}^n + \bar{s})^T \beta_2 (\bar{\mathbf{y}}^n + \bar{s}) + (\bar{\mathbf{y}}^n)^T \beta_2 \bar{\mathbf{y}}^n) \beta_2 (2\bar{\mathbf{y}}^n + \bar{s}). \quad (29)$$

This can then be inserted into equation (21). For ease of differentiation (29) is rewritten as

$$\bar{\mathbf{F}}_{tm}(\bar{s}) = -\frac{\Gamma}{4}(\alpha_2(\bar{s}) \mathbf{U}(\bar{s}) + \alpha_1 \mathbf{U}(\bar{s})), \quad (30)$$

where

$$\mathbf{U}(\bar{s}) = \beta_2(2\bar{\mathbf{y}}^n + \bar{s}) \quad (31)$$

$$\alpha_2(\bar{s}) = (\bar{\mathbf{y}}^n + \bar{s})^T \beta_2 (\bar{\mathbf{y}}^n + \bar{s}) \quad (32)$$

$$\alpha_1 = (\bar{\mathbf{y}}^n)^T \beta_2 \bar{\mathbf{y}}^n \quad (33)$$

To differentiate $\alpha_2(\bar{s})$ with respect to \bar{s} the vector calculus rule:

$$\frac{\partial g(x)^T C h(x)}{\partial x} = g(x)^T C \frac{\partial h(x)}{\partial x} + h(x)^T C^T \frac{\partial g(x)}{\partial x}, \quad (34)$$

is used. In the equation being considered β_2 which corresponds to \mathbf{C} is diagonal so it is equal to its own transpose. This gives:

$$\frac{\partial \alpha_2(\bar{s})}{\partial \bar{s}} = 2(\bar{s} + \bar{\mathbf{y}}^n)^T \beta_2. \quad (35)$$

Differentiating $\mathbf{U}(\bar{s})$ with respect to \bar{s} gives

$$\frac{\partial \mathbf{U}(\bar{s})}{\partial \bar{s}} = \beta_2. \quad (36)$$

Then using the product rule the Jacobian of the tension modulation part of the \mathbf{G} function can be written as

$$\mathbf{J}_{tm}(\bar{s}) = -\frac{\Gamma}{4}(\alpha_2 \beta_2 + \mathbf{U}(\bar{s})(2(\bar{s} + 2\bar{\mathbf{y}}^n)^T \beta_2) + \alpha_1 \beta_2). \quad (37)$$

As \mathbf{J}_{tm} is a full matrix the addition of the tension modulation significantly increases computational time due to the linear system solving required in each iteration.

It has been shown by Bilbao [8] that, using a similar modal formulation, significant pitch glide in the wrong direction can arise due to tension modulation if the number of modes used creates partials up to the Nyquist frequency. However it was found that this spurious effect could not be replicated in the model even for extreme amplitudes.

2.6. Thread

2.6.1. Continuous domain

In the model the thread is treated as described in Section 2.3. The equation for the force is

$$F_c(t) = K_c(h_c - y_c(t)) - R_c \frac{\partial y_c(t)}{\partial t}, \quad (38)$$

where K_c is the spring stiffness of the thread, y_c is the vertical displacement of the string at the thread point, t is the time, h_c is the thread equilibrium point and R_c is the thread loss parameter. h_c was set to zero for all modelling purposes and as such will be omitted in the following equations. When re-expressed in the modal domain this equation reads

$$\bar{F}_{c,i}(t) = -g_{c,i} \sum_{i=1}^M g_{c,i} \left(K_c \bar{y}_i(t) + R_c \frac{\partial \bar{y}_i(t)}{\partial t} \right). \quad (39)$$

2.6.2. Discretisation of thread equation in time

Using the discretisation operators on the force function and the equation $\bar{y}_i^{n+1} = \bar{s}_i + \bar{y}_i^n$ the equation for the modal force at time step $n + \frac{1}{2}$ can be written

$$\bar{F}_c^{n+\frac{1}{2}} = -\mathbf{g}_c \mathbf{g}_c^\top \left(\frac{K_c}{2} (\bar{s} + 2\bar{y}^n) + \frac{R_c}{\Delta t} \bar{s} \right), \quad (40)$$

in vector form. To solve for \bar{s} the Jacobian of this vector is required. This is:

$$\mathbf{J}_c = -\left(\frac{K_c}{2} + \frac{R_c}{\Delta t} \right) \mathbf{g}_c \mathbf{g}_c^\top. \quad (41)$$

2.7. Distributed Barrier

2.7.1. Continuous domain and spatial discretisation

To model the bridge a contact law is used to simulate the compression of the string and barrier. The contact law used is

$$\mathcal{F}_b(x, t) = k_b |h_b(x) - y(x, t)|^\alpha, \quad (42)$$

where k_b is the barrier stiffness per unit length, $h_b(x)$ is the height of the bridge at position x and α is the exponent that governs the bridge interaction. Following [14] the exponent is set according to Hertzian theory as $\alpha = 1$ and the formulation is carried out with this in place. Other choices of $\alpha \geq 1$ are possible and all of the terms required for solving related to the bridge can be derived in this general case. The force density and the string displacement in this equation are defined over $[x_b - \frac{1}{2}\omega_b, x_b + \frac{1}{2}\omega_b]$ which is the spatial domain of the bridge; the force density will be zero outside this region. The potential energy density for the bridge will also be required

$$\mathcal{V}_b(x, t) = \frac{k_b}{2} |h_b(x) - y(x_b, t)|^2, \quad (43)$$

as will another definition for the force

$$\mathcal{F}_b(x, t) = -\frac{\partial \mathcal{V}_b}{\partial y}. \quad (44)$$

For use in equation (21) the modal force vector $\bar{\mathbf{F}}_b$ must be found. This would usually be done by integrating over the length of the string as follows

$$\bar{F}_{b,i}(t) = \int_0^L v_i(x) \mathcal{F}_b(x, t) dx, \quad (45)$$

but due to the non-analytic nature of $\mathcal{F}_b(x, t)$ it is approximated as a Riemann sum [9]

$$\bar{F}_{b,i}(x, t) \approx \sum_{k=1}^K v_{i,k} \mathcal{F}_{b,k}(x_k, t) \Delta x. \quad (46)$$

The bridge is defined to have K points along its length, $v_{i,k}$ gives each modal amplitude at the spatial position k in the domain of the barrier and $\mathcal{F}_{b,k}(x_k, t)$ is the contact force density at point k and time t .

2.7.2. Discretisation of bridge equation in time

Writing the potential density at time step n gives

$$\mathcal{V}_{b,k}^n(t) = \frac{k_b}{2} |h_{b,k} - y_k^n|^2, \quad (47)$$

here y_k^n is at the corresponding bridge point to $h_{b,k}$ in each case. Equation (44) is discretised in time to give

$$\mathcal{F}_{b,k}^{n+\frac{1}{2}} = -\frac{\mathcal{V}_{b,k}^{n+1} - \mathcal{V}_{b,k}^n}{y_k^{n+1} - y_k^n}. \quad (48)$$

These equations are combined to give

$$\mathcal{F}_{b,k}^{n+\frac{1}{2}} = -\frac{k_b}{2} \frac{|h_{b,k} - (s_k + y_k^n)|^2 - |h_{b,k} - y_k^n|^2}{s_k}. \quad (49)$$

Where again s_k is at the corresponding barrier point to $h_{b,k}$. Following from equation (49) and vectorising gives the equation

$$\bar{\mathbf{F}}_b^{n+\frac{1}{2}} = \Delta x \mathbf{V}^\top \mathcal{F}_b^{n+\frac{1}{2}}, \quad (50)$$

where \mathbf{V} is a $K \times M$ matrix which holds all of the modal shapes at each point along the bridge. $\bar{\mathbf{F}}_b^{n+\frac{1}{2}}$ must be differentiated with respect to \bar{s} for use in Newton's method so this is done by using the chain rule, giving the Jacobian for the barrier as

$$\mathbf{J}_b = \Delta x \mathbf{V}^\top \frac{\partial \mathcal{F}_b^{n+\frac{1}{2}}}{\partial s} \frac{\partial s}{\partial \bar{s}}, \quad (51)$$

where

$$\mathbf{y} = \mathbf{V} \bar{\mathbf{y}} \quad \text{and} \quad \mathbf{s} = \mathbf{V} \bar{\mathbf{s}}. \quad (52)$$

Obviously $\frac{\partial \mathbf{s}}{\partial \bar{\mathbf{s}}} = \mathbf{V}$. Explicitly \mathbf{J}_b is written as

$$\mathbf{J}_b = \frac{k_b \Delta x}{2} \mathbf{V}^\top \zeta \mathbf{V}, \quad (53)$$

where ζ is a diagonal matrix with the dimensions $K \times K$. The diagonal entries in the matrix range from $\zeta_{1,1}$ which is the closest bridge point to the thread to $\zeta_{K,K}$ which is the furthest bridge point from the thread are:

$$\zeta_{k,k} = \frac{|h_{b,k} - (s_k + y_k^n)|^2 - |h_{b,k} - y_k^n|^2}{s_k^2} - \frac{2|h_{b,k} - (s_k + y_k^n)|}{s_k}. \quad (54)$$

2.7.3. Single point bridge

For comparison the distributed bridge was made replaceable by a single point bridge in the model. The single force bridge has a different equation for the force density which is

$$\mathcal{F}_b(x, t) = \delta(x - x_b) k_b |h_b(x) - y(x, t)|, \quad (55)$$

x_b here is the spatial position of the bridge point. When this is put into equation (45) the integral becomes exactly solvable and gives a modal force for mode i of

$$\bar{F}_{b,i}(t) = v_i(x_b) k_b |h_b(x_b) - y(x_b, t)|. \quad (56)$$

2.8. Plucking signal

The plucking signal used is a single point force spatially distributed following the procedure outlined in Section 2.3. The input force used [14] is

$$F_e(t) = A_e \sin^2 \left(\frac{\pi}{\tau_e} h_e(t - t_e) \right), \quad (57)$$

where

$$h_e(t) = \frac{1}{2} \cos \left(t + \frac{\tau_e \sinh(\Omega t)/\tau_e}{\sinh(\Omega t)} \right). \quad (58)$$

$F_e(t)$ is zero outside of the time domain $t_e : t_e + \tau_e$ where t_e is the excitation time and τ_e is the length of the excitation. This gives a malleable curve based on \sin^2 whose central point and gradients on either side can be altered according to the choice of Ω . This is to a certain extent an arbitrary choice and was chosen to simulate the plucking typically used in the playing of tanpura.

As F_e is known at all time steps it can simply be inserted into equation (21) by using the modal weight g_e and discretisation operator μ

$$\bar{F}_e^{n+\frac{1}{2}} = \frac{1}{2} g_e \mu F_e. \quad (59)$$

3. MODEL PROPERTIES

3.1. Uniqueness

For the scheme to have a unique existent solution the Jacobian must be positive definite. In this scheme the Jacobian is equation (22). For a $N \times N$ symmetric matrix (\mathbf{M}) to be positive definite it is required that

$$\mathbf{z}^\top \mathbf{M} \mathbf{z} > 0, \quad (60)$$

for all real column vectors \mathbf{z} of size N which are non-zero. It is useful in this instance to note that the addition of a positive semi-definite matrix to a positive definite matrix results in a matrix which is still positive definite. The definition for a positive semi-definite matrix being

$$\mathbf{z}^\top \mathbf{M} \mathbf{z} \geq 0, \quad (61)$$

\mathbf{J}_s is positive definite as every matrix in it has only real, positive entries and they are all diagonal. For the thread term $-\mathbf{J}_c$ can be proven to be positive semi definite simply by considering that any negatives in either \mathbf{z} or \mathbf{g}_c are squared in equation (62), proving that

$$\mathbf{z}^\top \mathbf{g}_c \mathbf{g}_c^\top \mathbf{z} \geq 0, \quad (62)$$

The term $-\zeta$ is positive semi-definite as $-\zeta_{k,k}$ is a convex function [5] of \bar{s} which means that $-\mathbf{J}_b$ is also positive semi-definite due to the following relation

$$-\mathbf{z}^\top \mathbf{J}_b \mathbf{z} = -\frac{\Delta x k_b}{2} (\mathbf{V}^\top \mathbf{z})^\top \zeta^n (\mathbf{V}^\top \mathbf{z}) \geq 0. \quad (63)$$

To prove that $-\mathbf{J}_{tm}$ is positive semi-definite it is easiest to consider it in its three parts as seen in equation (37). α_2 , α_1 and Γ are all positive and β_2 is a diagonal matrix with only positive entries. Following from this the following must be true for the first and third matrices on the right hand side of equation (37)

$$\mathbf{z}^\top \alpha_2 \Gamma \beta_2 \mathbf{z} \geq 0 \quad \text{and} \quad \mathbf{z}^\top \alpha_1 \Gamma \beta_2 \mathbf{z} \geq 0. \quad (64)$$

Expanding the remaining matrix and using $\beta_2^\top = \beta_2$ where appropriate and writing $\lambda = \bar{s} + 2\bar{y}^n$

$$\mathbf{U}(\bar{s})(2\lambda^\top \beta_2) = 2\beta_2 \lambda \lambda^\top \beta_2^\top. \quad (65)$$

After rearranging it can be seen that the following must be true.

$$2\Gamma \mathbf{z}^\top \beta_2 \lambda \lambda^\top \beta_2^\top \mathbf{z} \geq 0 \quad (66)$$

Therefore $-\mathbf{J}_{tm}$ is positive semi-definite. As all of the component matrices of \mathbf{J} are positive definite or positive semi-definite the uniqueness of the solution is proven.

3.2. Stability

The total numerical energy of the system at time-step n can be expressed as a summation of the total energies of the modes of the stiff string with the potentials due to the tension modulation, thread and barrier added on. The potential due to the barrier at time step n can be easily found using equation (47)

$$V_b^n = \Delta x (\mathbf{1}_K)^\top \mathcal{V}_b^n. \quad (67)$$

Where $\mathbf{1}_K$ is a vector of size K with 1 at every entry. The potential due to the tension modulation is

$$V_{tm}^n = \frac{\Gamma}{4} ((\bar{y}^n)^\top \beta_2 \bar{y}^n)^2, \quad (68)$$

and the potential due to the thread is

$$V_c^n = \frac{K_c}{2} (\mathbf{g}_c^\top \bar{y}^n)^2, \quad (69)$$

which in conjunction with the energy

$$H_s^n = \xi^{-1} ((\bar{y}^n)^\top \mathbf{A} \bar{y}^n + (\bar{q}^n)^\top \bar{q}^n), \quad (70)$$

give the total numerical energy at time-step n

$$H^n = V_b^n + V_{tm}^n + V_c^n + H_s^n. \quad (71)$$

To get a power balance the right hand side of equation (19) is multiplied by $(\delta \bar{y}^n)^\top$ and the left hand side by $(\mu \bar{q}^n)^\top$. Following various re-arrangements the equation for the energy balance is found to be

$$\Delta t^{-1} \delta H^{n+\frac{1}{2}} = P^{n+\frac{1}{2}} - Q^{n+\frac{1}{2}}. \quad (72)$$

The input power $P^{n+\frac{1}{2}}$ is

$$P^{n+\frac{1}{2}} = \frac{1}{2\Delta t} \mathbf{g}_e^\top \delta \bar{y}^{n+\frac{1}{2}} \mu F_e^{n+\frac{1}{2}}, \quad (73)$$

and the power losses $Q^{n+\frac{1}{2}}$ are

$$Q^{n+\frac{1}{2}} = \frac{(\delta \bar{y}^{n+\frac{1}{2}})^\top \mathbf{B} \delta \bar{y}^{n+\frac{1}{2}}}{\xi \Delta t} + \frac{R_c}{\Delta t^2} (\delta y_c^{n+\frac{1}{2}})^2. \quad (74)$$

As both \mathbf{A} and \mathbf{B} are positive definite $H^{n+\frac{1}{2}}$ and $Q^{n+\frac{1}{2}}$ can only be greater than or equal to zero so the change in the numerical energy of the system can only be conserved or decline outwith periods where energy is being directed into the system via the controlled input force.

String Parameters			
	C ₃	C ₂	unit
L	1.0	1.0	m
ρ	7850	8000	kg/m ³
A	6.16×10^{-8}	2.83×10^{-7}	m ²
E	2.0×10^{11}	1.0×10^{11}	N/m ²
I	3.02×10^{-16}	6.36×10^{-15}	kg/m ²
T ₀	33.1	38.7	N
σ_0	0.6	0.8	s ⁻¹
σ_1	6.5×10^{-3}	6.5×10^{-3}	m/s
σ_3	5×10^{-6}	5×10^{-6}	m ³ /s

Table 1: Physical parameter values used for C₃ and C₂ string.

4. RESULTS

4.1. String and contact parameters

The string parameters used in the model are detailed in Table 1. Parameters chosen to represent a steel C₃ string were the same as used in previous papers studying the tanpura [14]. For modelling a bronze C₂ string the values for σ_1 and σ_3 were set to be the same as for the steel string for convenience whilst the value for σ_0 was altered to get an empirically more realistic response.

The bridge profile was taken to be

$$h_b(x) = 3(x - x_b)^2, \quad (75)$$

which is based upon measurements taken by Guettler [15]. x_b is the central point of the barrier. For the spring stiffness of the barrier a value of $k_b = 1 \times 10^{11}$ N/m² was used. The width was chosen to be wide enough such that the end points of the barrier were not touched during vibration. This limits the string's interaction with the bridge to be within the space in which the bridge is defined. The position of the bridge maximum was set to be 10 mm while the bridge width required was found to be 1 mm for the C₃ and 1.5 mm for the C₂. The number of points was altered from being very high, around 60, down to 11 for C₃ and 16 for C₂. The smaller numbers gave almost exactly the same results as were found when 60 bridge points were used. K_c was chosen to be 1.2×10^5 N/m and R_c was taken to be 1.2 kg/s [14]. The thread position was set to 5 mm when modelling the C₃ string and 4 mm when modelling the C₂ string.

The plucking parameters used to generate results were $x_e = 0.37 \times L$ m, $w_e = 1.5$ mm, $A_e = -0.5$ N, $\Omega = 30$, $\tau_e = 0.01$ s and $t_e = 0$ s.

4.2. Convergence

The top row of graphs in Figure 2 shows the nut force over time with different sampling frequencies. It can be seen that the envelope shapes are very similar between the two cases with only small variations in the detail. The close match between the envelopes shows that the model does a good job of replicating global trends at lower sampling frequencies.

The second row of graphs shows the nut force over two different time frames, one very soon after the string is plucked and the other roughly 0.15 s after. The higher sampling rate plots which are the red and dashed black lines match up well over both plots. The lower sampling frequency causes the plot to be compressed

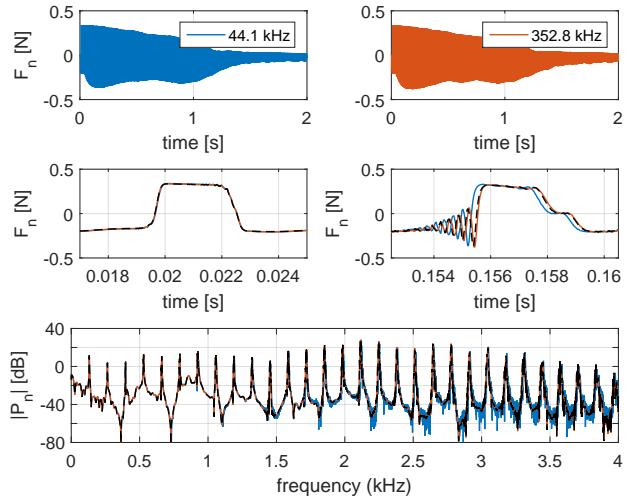


Figure 2: Plots showing how the sampling frequency affects the output from the model. The top two graphs show the nut force over time for two sampling frequencies. The middle plots display portions of the nut force signal over a short time frame. These can be compared to the black dashed line which represents $F_s = 352.8$ kHz but the number of modes is truncated to be the same as the number of modes in the $F_s = 44.1$ kHz case. The final plot shows the radiated pressure due to the nut force over a range of frequencies with the same colour scheme as the other graphs.

along the time axis which is what is shown in the later time snapshot. This is due to the numerical model with lower sampling rates effecting a reduced contact stiffness leading to increased contact durations. The shape of the signal is preserved however, meaning that general phenomena within the signal will be captured. This does however mean that for any comparison to experiment high sampling rates are required to get the signals to match up well over time. This was noted by Issanchou et al [13] when performing their comparison between experiment and simulation. While the black dashed line and the red line are signals generated at the same sampling frequency the number of modes used when generating the black dashed line were restricted to be the same as the number used in generating the blue signal. It is clear from these plots that including modes above the audio band has little effect on the area of interest. Hence these modes could, in the future, be ignored speeding up computation times significantly.

The final graph in Figure 2 shows the spectral envelope of the radiated nut pressure signal over a range of frequencies. Again it can be seen that there is a strong agreement between all of the lines. The differences are at such a low pressure level that they are perceptually insignificant, as are the small discrepancies in the partial frequencies. Indeed, informal listening test indicated that the three signals are aurally barely distinguishable.

4.3. Tanpura simulation

When analysing the information generated by the model the most important data is that which pertains to how the signal generated would sound. To investigate this the nut force is computed and is filtered by a body response measured with an impact hammer by a microphone 80 cm away [14]. This allows a physically meaningful

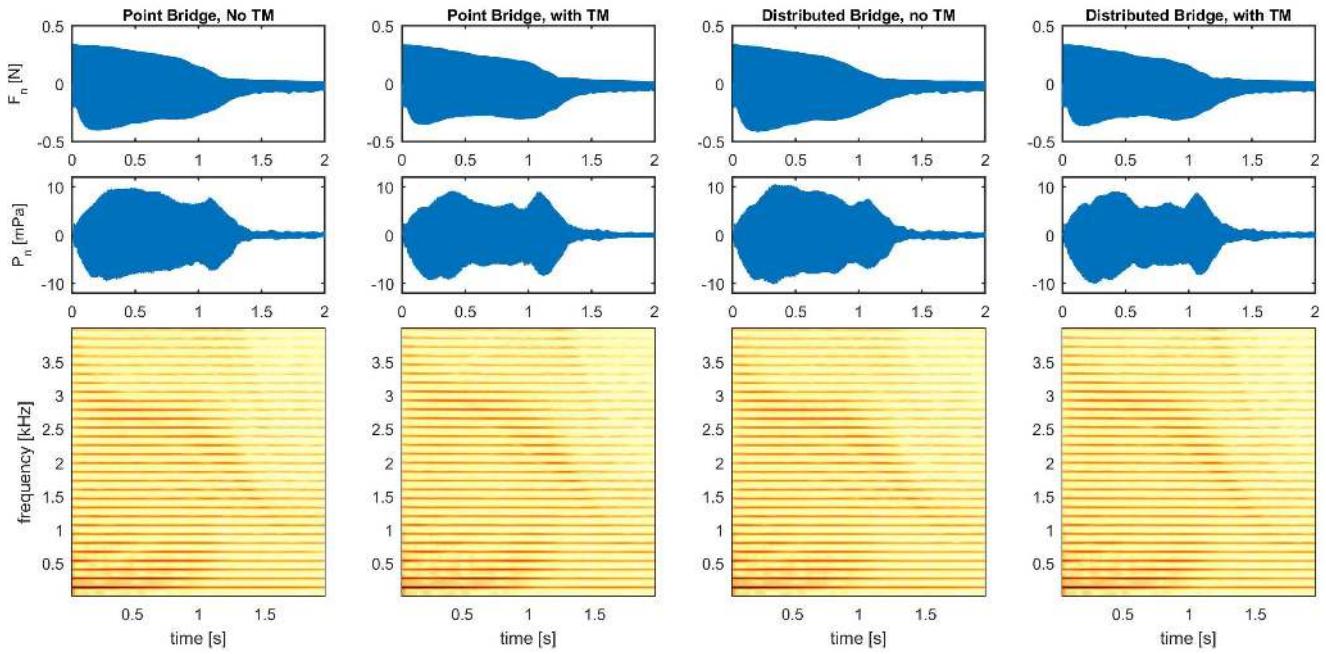


Figure 3: Plots showing two seconds of simulation after the start of the pluck. The top row shows the nut force envelope over time for the four cases indicated by the note on top of the graph. The second row shows the radiated pressure from the nut envelope over time. The third row shows spectrograms of the nut force.

signal to be generated from the model which when analysed will give information on how the changes being investigated will affect the perceived sound from the modelled instrument.

Figure 3 shows the nut force, radiation pressure from the nut force and spectrograms for the four different configurations of including tension modulation and having a point or distributed bridge. These were all set up with the same parameter values (barring the changes made to investigate) and identical plucking conditions. The simplest case presented here with no tension modulation and a point bridge is used as a point of comparison when studying the effects of adding tension modulation and a distributed barrier. Within the spectrograms the jvari is the time-dependent formant of pronounced frequencies which have the drop in spectral centroid followed by a plateau followed again by a drop. As can be seen from Figure 3 all of the spectrograms show this behaviour, however there are differences in the characteristics of the jvari shapes.

In the radiated pressure time-domain graph the tension modulation can be seen to significantly alter the envelope in the first half second of the simulation and the loss of energy at the end of the signal also follows a different shape. From the spectrograms it can be observed that the addition of tension modulation causes the spectral centroid of the jvari to start at a higher frequency and the plateau is also at a higher frequency relative to the situation without the tension modulation. The effect on the tail of the jvari is hard to see in the spectrograms. The tension modulation also has the expected effect of causing pitch glide in the partials. This can be seen by zooming in on the partials above 3 kHz in the second and fourth spectrograms.

Distribution of the bridge forces has the effect of shortening the plateau of the jvari when compared to the point barrier and it also rounds off the top right corner of the plateau. This effect can be seen in both the spectrogram and the radiated pressure plot. The

start of the tail can be seen to occur at roughly 1 second when the barrier is distributed and roughly 0.1 s later with the point barrier. Although they start to decay at different times they end up with a very similar level of energy left in the system.

When the tension modulation and distributed bridge are combined the spectrogram is distinct from all three of the others, indicating that both additions make a difference to the output of the model. This spectrogram shows the higher starting point frequency and plateau frequency for the jvari from the tension modulation and the more rounded, shorter plateau from the distributed barrier. The extra effect of having both together is difficult to see in plots, but when listening to sound examples¹ the combination appears to give a "livelier" sound than any of the signals generated by the other model options.

The effect due to the tension modulation can be seen more clearly in Figure 4. The plots show spectrograms of the nut force when the model is run with C₂ string parameters and a greater excitation amplitude of A_e = -0.8 N. It can be observed by comparing (a) and (b) to (c) and (d) that the tension modulation is noticeable when the barrier is absent but has a larger effect when the barrier is present. With these parameters the change in the shape of the jvari is more significant than in the C₃ case and is heard more clearly in sound examples.

5. CONCLUSIONS

This paper shows how tension modulation, a distributed barrier and the thread can be included in a model of the tanpura. The model is proven to be unconditionally stable and to have a unique solution to the non-linear vector equation that has to be solved at

¹<http://www.socasites.qub.ac.uk/mvanwalstijn/dafx17c/>

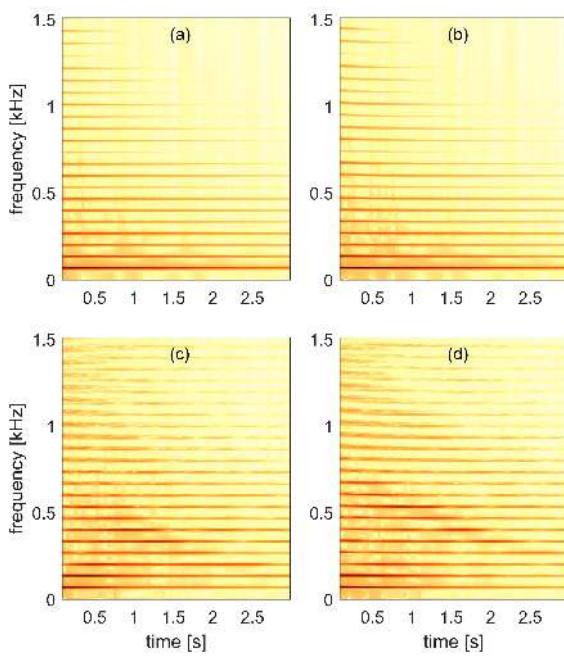


Figure 4: Spectrograms of the nut force for four configurations comparing the effects of tension modulation with and without tension modulation. a has no bridge or tension modulation, b has no bridge but with tension modulation, c has a bridge but with no tension modulation and d has both the bridge and tension modulation.

each time step. The results discussed show that both tension modulation and a distributed bridge have small effects on the shape of the jvari. A webpage² has been created with sound clips corresponding to each of the signals presented in Figures 3 and 4. These sound clips provide a way to interpret the results in a manner more natural to the study of the sound produced by a tanpura. It is worth noting that this model is not suitable for use in real time applications without dedicated hardware as 1 s of simulation at 44.1 kHz takes 81 s without tension modulation and 131 s with tension modulation.

The model presented here could be expanded further by including a transverse plane of oscillation in a similar way to other models detailed in the past [13, 3]. Coupling at termination points between transverse planes of oscillation and including friction as the string moves across the barrier has been found to make a difference to the jvari in simulations [16] and experimental investigation into the physical extent of these effects is another avenue that could be explored. Tension modulation in the two plane case will be important as it provides coupling when there is a phase difference between the oscillations in the two planes. A more quantitative method of comparing the spectrograms of signals would be useful in making comparisons between cases easier.

6. REFERENCES

- [1] C. V. Raman, “On some Indian stringed instruments,” *Proc. Indian Assoc. Cultiv. Sci*, vol. 7, pp. 29–33, 1921.
- [2] S. Bilbao and A. Torin, “Numerical Simulation of String/Barrier Collisions: The Fretboard.,” in *DAFx*, 2014, pp. 137–144.
- [3] C. Desvages and S. Bilbao, “Two-Polarisation Physical Model of Bowed Strings with Nonlinear Contact and Friction Forces, and Application to Gesture-Based Sound Synthesis,” *Applied Sciences*, vol. 6, no. 5, pp. 135, May 2016.
- [4] V. Chatzioannou and M. Van Walstijn, “An energy conserving finite difference scheme for simulation of collisions,” in *Proc. Stockholm Musical Acoust. Conf*, 2013.
- [5] V. Chatzioannou and M. van Walstijn, “Energy conserving schemes for the simulation of musical instrument contact dynamics,” *Journal of Sound and Vibration*, vol. 339, pp. 262–279, Mar. 2015.
- [6] Stefan Bilbao, Alberto Torin, and Vasileios Chatzioannou, “Numerical modeling of collisions in musical instruments,” *Acta Acustica united with Acustica*, vol. 101, no. 1, pp. 155–173, 2015.
- [7] S. Bilbao, “Numerical Modeling of String/Barrier Collisions,” in *Proc. ISMA*, 2014, pp. 267–272.
- [8] S. Bilbao, *Numerical Sound Synthesis*, Wiley & Sons, Chichester, UK, 2009.
- [9] M. van Walstijn and J. Bridges, “Simulation of distributed contact in string instruments: A modal expansion approach,” Aug. 2016, pp. 1023–1027, IEEE.
- [10] V. Debut, J. Antunes, M. Marques, and M. Carvalho, “Physics-based modeling techniques of a twelve-string Portuguese guitar: A non-linear time-domain computational approach for the multiple-strings/bridge/soundboard coupled dynamics,” *Applied Acoustics*, vol. 108, pp. 3–18, July 2016.
- [11] M Ducceschi, S. Bilbao, and C. Desvages, “Modelling collisions of nonlinear strings against rigid barriers: Conservative finite difference schemes with application to sound synthesis,” in *Proc. International Congress on Acoustics*, Sept. 2016.
- [12] C. Valette, C. Cuesta, M. Castellengo, and C. Besnainou, “The tampura bridge as a precursive wave generator,” *Acta Acustica united with Acustica*, vol. 74, no. 3, pp. 201–208, 1991.
- [13] C. Issanchou, S. Bilbao, JL. Le Carrou, C. Touze, and O. Doare, “A modal-based approach to the nonlinear vibration of strings against a unilateral obstacle: Simulations and experiments in the pointwise case,” *Journal of Sound and Vibration*, vol. 393, pp. 229–251, Apr. 2017.
- [14] M. van Walstijn, J. Bridges, and S. Mehes, “A real-time synthesis oriented tanpura model,” in *Proc. Int. Conf. Digital Audio Effects (DAFx-16)*, 2016.
- [15] K. Guettler, “On the string-bridge interaction in instruments with a one-sided (bridge) constraint,” available from knut-sacoustics. com, 2006.
- [16] J. Bridges and M. Van Walstijn, “Investigation of tanpura string vibrations using a two-dimensional time-domain model incorporating coupling and bridge friction,” *Proc. of the third Vienna Talk on Music Acoustics*, 2015.

²<http://www.socasites.qub.ac.uk/mvanwalstijn/dafx17c/>

PHYSICALLY DERIVED SYNTHESIS MODEL OF A CAVITY TONE

Rod Selfridge

Media and Arts Technology Doctoral College,
School of EECS
Queen Mary University of London
E1 4NS, U.K
r.selfridge@qmul.ac.uk

Joshua D Reiss

Centre for Digital Music
School of EECS
Queen Mary University of London
E1 4NS, U.K
joshua.reiss@qmul.ac.uk

Eldad J Avital

Centre for Simulation and Applied Mechanics
School of EMS
Queen Mary University of London
E1 4NS, U.K
e.avital@qmul.ac.uk

ABSTRACT

The cavity tone is the sound generated when air flows over the open surface of a cavity and a number of physical conditions are met. Equations obtained from fluid dynamics and aerodynamics research are utilised to produce authentic cavity tones without the need to solve complex computations. Synthesis is performed with a physical model where the geometry of the cavity is used in the sound synthesis calculations. The model operates in real-time making it ideal for integration within a game or virtual reality environment. Evaluation is carried out by comparing the output of our model to previously published experimental, theoretical and computational results. Results show an accurate implementation of theoretical acoustic intensity and sound propagation equations as well as very good frequency predictions.

NOMENCLATURE

c = speed of sound (m/s)
 f = frequency (Hz)
 ω = angular frequency = $2\pi f$ (rads/revolution)
 u = air flow speed (m/s)
 R_e = Reynolds number (dimensionless)
 S_t = Strouhal number (dimensionless)
 r = distance between listener and sound source (m)
 ϕ = elevation angle between listener and sound source
 φ = azimuth angle between listener and sound source
 ρ_{air} = mass density of air (kgm^{-3})
 μ_{air} = dynamic viscosity of air (Pa s)
 M = Mach number, $M = u/c$ (dimensionless)
 L = length of cavity (m)
 d = depth of cavity (m)
 b = width of cavity (m)
 κ = wave number, $\kappa = \omega/c$ (dimensionless)
 r = distance between source and listener (m)
 δ = shear layer thickness (m)
 δ^* = effective shear layer thickness (m)
 δ_0 = shear layer thickness at edge separation (m)
 θ_0 = shear layer momentum thickness at edge separation (m)
 C_2 = pressure coefficient (dimensionless)

1. INTRODUCTION

Aeroacoustic sounds comprise the class of sounds generated by the movement of air past objects or edges. Alternatively, the sounds can also be generated by objects moving through the air. Examples of aeroacoustic sounds are those created when a sword swings through the air, wind passes a doorway or a spinning propellor.

Research is undertaken to accurately determine the frequencies, gain and propagation patterns required to replicate the cavity tone. Key semi-empirical formulae are found within the aeroacoustic research field, allowing us to identify primary relationships and parameters. *Semi-empirical* equations are ones where an assumption or generalisation has been made to simplify the calculation or yield results in accordance with observations. Physical models have the major advantage of allowing users to continually change parameters and be confident that the underlying laws and principles are consistently obeyed, giving sounds produced an inherent authenticity.

The development of real-time sound synthesis models has great potential for use in nonlinear media such as virtual reality and games. A sound synthesis model that reacts in real-time to the variety of perspectives and interactions within these environments can offer an increased sense of realism that replaying sampled sounds may fail to capture. Linear media such as film and television may also benefit from the bespoke sound effects offered by our model. Our approach can be classified as *Procedural Audio*; sound is generated via a process dependent on evolving situations, i.e. speed of air/object motion, observer or camera position, etc. Parameters can be manipulated by a user or fully automated by a game engine producing instant changes in real-time.

The paper is organised as follows: in Section 2, we describe the state of the art for synthesising cavity sounds. The problem formulation is given in Section 3. Section 4 provides an overview of the fluid dynamics theory and equations used to predict the cavity tone. The implementation is given in detail in Section 5 with results presented in Section 6. Section 7 provides a discussion on these results including areas for future development. Finally, Section 8 concludes with a summary of our findings.

2. STATE OF THE ART

The cavity tone was synthesised in [1] in which Computational Fluid Dynamics (CFD) techniques were used to create several sound textures. The sound textures were used for real-time rendering of the audio, corresponding to air velocity and an object's motion specified by the user. The textures generated through CFD calculations can be more mathematically accurate than the equations used in this research but were not calculated in real-time due to computational complexity. Offline calculations had to be completed for each new cavity shape. This is not the case in our model.

The noise generated by jet aeroplanes was synthesised in [2]. This identifies tonal sounds generated by cavities left in the wheel wells as the landing gear is extended. An analysis approach was used to identify tonal components in aircraft noise then synthesised by using a sine wave with time varying amplitude and frequency. Perceptual tests revealed that tonal noise above 4000Hz

was extremely annoying; prediction of this along with simultaneous broadband aircraft noise is useful for aircraft engineers and designers.

Real-time sound effect synthesis using physically inspired models were presented in [3]. This includes a study into replicating the sounds generated from microscopic cavities in building walls and doorways by noise filtering. These were expressed in a “howls” model that generates filtered noise tones, triggered when the windspeed passes above a threshold. These were generalised for numerous cavities with no exact dimensions.

Often physical models of musical instruments include synthesising the coupling of vibrating air to cavities present in the instruments. Typical examples of these are [4] where a timpani model was created, and [5], which modelled stringed instruments including a Chinese Ruan. Both synthesis techniques used a finite difference methods to solve differential equations of the vibrating space; [4] using general purpose graphical processing units and [5] field programmable gate arrays to perform the computationally intensive calculations required. Our model includes the fundamental natural resonant mode of the cavity as well as tones created by a feedback mechanism. Our implementation uses equations derived from the Navier-Stokes equations but requiring much less computational power, (see Section 4.1).

Real-time synthesis of another fundamental aeroacoustic sound, the Aeolian tone was presented by the authors in [6]. The Aeolian tone is the sound generated as air flows around a solid object creating oscillating forces due to vortex shedding, for example, a sword sweeping through the air. A physically derived model was implemented enabling accurate sound synthesis results with real-time operation. Semi-empirical equations from fluid dynamics were used in [6] to calculate frequencies and other acoustic properties. The results of this paper were extended to include mechanical properties in [7]. A similar approach is undertaken in this paper.

This work is related to the fluid dynamics of flue instruments, like the pipe organ [8], where the pipe mode resonance is similar to that of a deep cavity, (see Section 4.1). Instead of being coupled to a cavity tone these instruments have an air jet striking a wedge which generates the edge tone [9].

3. PROBLEM FORMULATION

The goal is to create an accurate cavity tone sound effect that operates in real-time. The parameters that we wish a user to be able to adjust are cavity dimensions, airspeed and listener position. The important aspects of the tones produced by air flowing over a cavity are the frequency components, magnitude of propagation based on observer’s distance and angle, and the bandwidth of the tones. The purpose of the research is to create a sound synthesis model of a fundamental aeroacoustic sound source which can then be used as an integral component of practical models, for example a wheel well as part of an aircraft model.

Semi-empirical equations defined or derived from fundamental fluid dynamics principles have been created and used by aeroacoustic engineers to diminish complex computations yet provide accurate acoustic results. Relevant equations from this field have been identified, based on tangible parameters, allowing us to build a physical model sound effect. Should explicit equations not exist then approximate ones based on observations from previously published results are implemented.

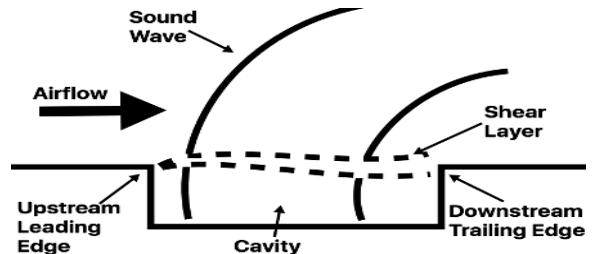


Figure 1: Basic components of cavity used in tone generation.

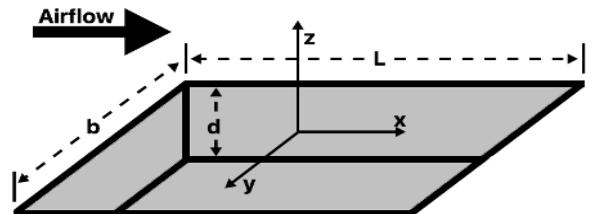


Figure 2: Diagram showing coordinates of a cavity.

4. BACKGROUND THEORY OF CAVITY TONE

4.1. Basic Principles

The generation of an acoustic tone due to air flowing over the open side of a cavity is described in [10–12]. Figure 1 illustrates air flowing over a cavity. As air passes over the leading edge vortices are generated due to the difference in airspeed between the free stream air and air in the cavity. The region of vortices is called the *shear layer*.

There are two operating modes exhibited by cavities: shear mode and wake mode. The shear mode is when the 3 dimensional shear layer collides with the trailing edge in a flapping motion. The wake mode is far less common and where a large 2 dimensional vortex is shed from the cavity itself. This report is focused on the shear mode.

As the vortices in the shear layer travel to the far side of the cavity they collide with the trailing edge generating an acoustic pressure wave. Outside the cavity the pressure wave propagates to the far field and is perceived as sound. Inside the cavity the pressure waves travel back towards the leading edge, interacting with the vortices being shed, re-enforcing the shedding process and creating a feedback system. The dimensions of the cavity used in this paper are shown in Fig. 2.

A shallow cavity is described in [13] as having the dimension ratio $L/d > 1$. Similarly a deep cavity has dimensions $L/d < 1$. Shallow cavities are dominated by a dipole sound source while deep cavities are dominated by the cavity resonance tone which is monopole in nature. [14] states the contribution from cavity resonance is important for deep cavities only and when the ratio $L/d > 5/2$ these resonances can dominate the acoustic radiation. The width b of the cavity only affects the frequency of the cavity resonance and has no effect on the feedback system.

Dipoles and monopoles are compact sound sources used to represent the sounds generated when the distance between observer and source is small compared to the sound wavelength. A monopole is a pulsating sphere radiating spherical waves while a dipole is a sphere oscillating from side to side.

Shallow cavities are either open or closed. Open cavities are

when the shear layer re-attaches around the trailing edge, as shown in Fig. 1. A closed cavity is a longer shallow cavity when the shear layer attaches to the bottom of the cavity prior to the trailing edge and does not produce the same dipole output. The shear layer should always reattach to the trailing edge in a deep cavity.

A condition that can indicate whether a cavity is closed or open is the ratio between the length and depth. [15] found that cavities become closed when the ratio $L/d > 8$, whilst quoting a separate study which found $L/d > 11$.

4.2. Frequency Calculations

In 1878 the Czech physicist Vincenc Strouhal defined one of the fundamental relationships of fluid dynamics which linked a physical dimension L , airspeed u and frequency f of the tone produced to a dimensionless variable known as the Strouhal number S_t . This is given in Eqn 1:

$$f = \frac{S_t u}{L} \quad (1)$$

where for the case of the cavity tone, L represents the cavity length. The first main research that predicted the cavity tone frequencies was done by Rossiter [10] defining the following formula to determine the Strouhal number:

$$S_{t\lambda} = \frac{\lambda - \alpha}{\frac{1}{K} + M} \quad (2)$$

where α is a constant representing the difference in phase between the acoustic wave arriving at the leading edge and the vortex being shed. It was fitted to the measured experimental data and a value of 0.25 was found. The constant K represents the ratio of convection velocity of vortices to the free stream flow speed. Rossiter gave K the value of 0.57, again to fit the observed data. λ is an integer number representing the mode.

Rossiter's equation, Eqn. 2, was extended by Heller et al. [16] to include the correction for the speed of sound within the cavity. This is shown in Eqn. 3

$$S_{t\lambda} = \frac{\lambda - \alpha}{\frac{1}{K} + \frac{M}{[1 + \frac{\gamma-1}{2} M^2]^{\frac{1}{2}}}} \quad (3)$$

where γ is the ratio of specific heats; [16] gives $\gamma = 1.4$. Rewriting Eqn. 1 to include the Strouhal number calculated by Eqn. 3 we obtain the frequency of each Rossiter mode f_λ :

$$f_\lambda = \frac{\lambda - \alpha}{\frac{1}{K} + \frac{M}{[1 + \frac{\gamma-1}{2} M^2]^{\frac{1}{2}}}} \frac{u}{L} \quad (4)$$

4.3. Acoustic Intensity Calculations

Pioneering research into aeroacoustics was carried out by Lighthill [17, 18] who took the fundamental fluid dynamics equations, Navier Stokes equations, and defined them for the wave equation to predict the acoustic sound. In [14] Howe uses a Green's Function to solve Lighthill's Acoustic Analogy in relation to the acoustic intensity of cavity tones. This solution provides a relationship defining the acoustic intensity based on the tone frequency, airspeed, cavity dimensions and propagation angle. The Green's function has components relating to a dipole field generated by the feedback system G_D and a monopole field created by the cavity resonance G_M .

Using the the Green's function Howe [14] derived the non-dimensional far field acoustic pressure frequency spectrum, $\Phi(\omega, x) \approx$

$$\frac{M^2 (\omega \delta^*/U)^{5/3}}{(1 + M \cos \phi)^2 \{(\omega \delta^*/U)^2 + \tau_p^2\}^{3/2}} \left| \frac{C_2 \sin(\kappa d)}{\cos\{\kappa(d + \zeta + i(\kappa A/2\pi))\}} + i(\cos \phi - M) \right|^2 \quad (5)$$

where x is the distance along the x axis of the cavity. The first and second terms in the magnitude brackets correspond to the monopole and dipole radiation respectively. $A = bL$ is the area of the cavity mouth, τ_p is a constant set to 0.12 and ζ is an *end correction* set to $\sqrt{\pi A/4}$. The end correction is the effective length to which the cavity must be extended to account for the inertia of fluid above the mouth of the cavity also set into reciprocal motion by the cavity resonance (see [14] for a more detailed explanation).

A fixed value for δ^* is set in [14], making the second Rossiter frequency dominant. The value of δ^* has a large influence over the dominant mode [19]. Dipole sources model Rossiter modes, with frequencies obtained from Eqn 4. The monopole source frequency, relating to the lowest order mode, is calculated when the complex frequency satisfies:

$$\kappa \left(d + \zeta + i \frac{\kappa A}{2\pi} \right) = \frac{\pi}{2} \quad (6)$$

4.4. Tone Bandwidth

To date no research has been found that describes the relationship between the peak frequency and the bandwidth of the tone. The Reynolds number is a dimensionless measure of the turbulence in a flow given by:

$$Re = \frac{\rho_{air} d u}{\mu_{air}} \quad (7)$$

It is known that the higher the Reynolds number, the smaller scale the vortices will be. This would imply that the higher the Reynolds number the wider the tone bandwidth.

5. IMPLEMENTATION

The equations in this section are discrete compared to the continuous formulas previously stated. The discrete time operator $[n]$ has been omitted until the final output (section 5.6), to reduce complexity. It should be noted that the airspeed u is sampled at 44.1KHz to give $u[n]$ and hence, $M[n], \omega[n], \kappa[n], St[n], f[n], Re[n], \Phi[n], \delta^*[n], Q[n]$ and $a[n]$.

Our synthesis model was implemented using the software Pure Data. This was chosen due to the open source nature of the code and ease of repeatability rather than high performance computations. Airspeed, cavity dimensions and observer position are adjustable in real-time. Properties like air density, speed of sound etc., are set as constants for design purposes but can be adjusted with minimal effort if a real-time evolving environment is required. All intermediate calculations are performed in real-time and no preprocessing is required.

Table 1: Values of the ratio L/θ_0 for different L/d values.

Reference	L/d	L/θ_0
[19]	2	52.8
	4	60.24
	4	86.06
	6	90.36
[20]	4	82

5.1. Conditions of Activation

The change from open to closed cavity occurs when the length to depth ratio passes a value around $L/d > 8 \rightarrow 11$ (Section 4.1). This is implemented by a sigmoid function, indicating whether the cavity is open or closed when L/d passes a value between 8 and 11. If the cavity is flagged as closed then Rossiter mode dipoles do not produce sound.

5.2. Frequency Calculations

A discrete implementation of Eqn. 4 is used with $\lambda = [1, 4]$ to give the first 4 Rossiter frequencies $f_{D\lambda}$. These relate to the dipole sources associated with the feedback loop.

To calculate the monopole resonant mode frequency a solution to the real part of Eqn. 6 is found; shown in Eqn. 8:

$$\kappa(d + \zeta) = \frac{\pi}{2} \quad (8)$$

with $\kappa = \omega/c$ and $\omega = 2\pi f_M$. Rearranging reveals the monopole frequency, f_M as:

$$f_M = \frac{c}{4(d + \zeta)} \quad (9)$$

5.3. Shear Layer Thickness

[19] and [20] state values for the ratio of L/θ_0 and d/θ_0 for a number of different ratios of L/d . These are shown in Table 1. A linear regression line is calculated for this data giving the relationship:

$$\frac{L}{\theta_0} = 9.39 \frac{L}{d} + 36.732 \quad (10)$$

Since the parameters L and d are set we can calculate a predicted value for θ_0 . The shear layer effective thickness at separation, δ_0^* and θ_0 are related by Eqn. 11 [21]:

$$H = \delta_0^*/\theta_0 \quad (11)$$

where H is a *shape factor* given in [20] as 1.29 for turbulent flow and 2.69 for laminar. Using Eqn. 10 and the relationship with the shape factor we are able to calculate δ_0^* .

The shear layer thickness over the cavity δ_c is stated in [21] and given as:

$$\delta_c = \left(\frac{xL}{R_{e_L}} \right)^{\frac{1}{2}} \quad (12)$$

for a laminar flow, where R_{e_L} is the Reynolds number with respect to the cavity length. For turbulent flow,

$$\delta_c = \left(\frac{x}{\sigma\sqrt{8}} \right) \quad (13)$$

where σ is the *Gortler Parameter* calculated from [19] to lie between 5 and 7. We chose $\sigma = 6$ for this model. x is the distance along the cavity length. We need to select a value of x to obtain the shear layer thickness for the calculations. Since the dipole is positioned near the trailing edge [19], we set $x = 0.75L$.

The relationship between δ^* and δ is given in [21] as:

$$\delta^* = \frac{1}{1+n} \delta \quad (14)$$

where $n = 7$. From Eqn. 14 we can calculate δ_c^* using δ_c as found from either Eqn. 12 or 13. The total shear layer effective thickness is obtained from Eqn. 15.

$$\delta^* = \delta_c^* + \delta_0^* \quad (15)$$

A critical value for R_{e_L} of 25000 is set [22], and implemented with a sigmoid function providing a transition from laminar to turbulent flow.

5.4. Acoustic Intensity Calculations

The acoustic intensity is calculated from a real-time discrete version of Eqn. 5 for the previously calculated monopole and dipole frequencies, Eqn. 9 and a discrete version of Eqn. 4. To achieve this the real and imaginary parts of the equation within the magnitude brackets are separated out. The denominator of the first component is:

$$\cos \left[\kappa \left(d + \zeta + i \frac{\kappa A}{2\pi} \right) \right] \quad (16)$$

multiplying κ into the brackets becomes:

$$\cos \left[\kappa(d + \zeta) + i \left(\frac{\kappa^2 A}{2\pi} \right) \right] \quad (17)$$

Using the identity:

$$\cos(a + ib) = \cos a \frac{e^b + e^{-b}}{2} - i \sin a \frac{e^b - e^{-b}}{2}$$

Let

$$X = \cos [\kappa(d + \zeta)] \left[\frac{e^{(\frac{\kappa^2 A}{2\pi})} + e^{-(\frac{\kappa^2 A}{2\pi})}}{2} \right]$$

$$Y = \sin [\kappa(d + \zeta)] \left[\frac{e^{(\frac{\kappa^2 A}{2\pi})} - e^{-(\frac{\kappa^2 A}{2\pi})}}{2} \right]$$

Expanding the magnitude brackets, the discrete implementation of Eqn. 5 becomes:

$$\Phi(\omega, x) \approx \frac{M^2 (\omega \delta^*/U)^{5/3}}{(1 + M \cos \phi)^2 \{(\omega \delta^*/U)^2 + \tau_p^2\}^{3/2}} \left[\left[\frac{X C_2 \sin(\kappa d)}{X^2 + Y^2} \right]^2 + \left[\frac{Y C_2 \sin(\kappa d)}{X^2 + Y^2} + (\cos \phi - M) \right]^2 \right] \quad (18)$$

Howe [14] gives a value of $C_2 = 1.02$ for a cavity with d/L ratio = 0.5 and τ_p is given as 0.12. To enable us to identify gains for the dipole sounds and monopole sound the second element within the large brackets is expanded giving:

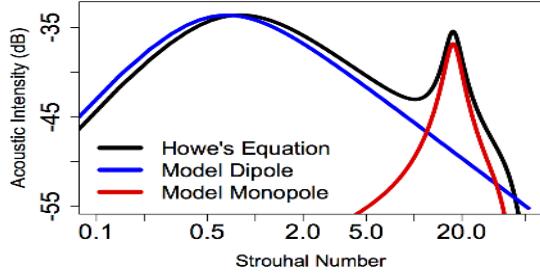


Figure 3: Far field spectrum gains calculated by Eqn. 20 & Eqn. 21 as compared to Howe's derivation from [14]. $L = b = 0.03m$, $d = 0.015m$ and $u = 3.43m/s$.

$$\left(\frac{YC_2 \sin(\kappa d)}{X^2 + Y^2}\right)^2 + 2\left(\frac{YC_2 \sin(\kappa d)}{X^2 + Y^2}(\cos \phi - M)\right) + (\cos \phi - M)^2 \quad (19)$$

It is seen that the middle term in Eqn. 19 contains elements from both monopole and dipole sources. Since the monopole is the more efficient compact source [23] this term is included in the monopole equation to minimise error. The monopole gain is shown in Eqn. 20:

$$G_M(\omega_M, r, \phi) \sim \frac{M^2(\omega_M \delta^*/U)^{5/3}}{r(1+M \cos \phi)^2 \{(\omega_M \delta^*/U)^2 + \tau_p^2\}^{3/2}} \left[\left[\frac{XC_2 \sin(\kappa_M d)}{X^2 + Y^2} \right]^2 + \left[\frac{YC_2 \sin(\kappa_M d)}{X^2 + Y^2} \right]^2 + 2 \frac{YC_2 \sin(\kappa d)}{X^2 + Y^2} (\cos \phi - M) \right] \quad (20)$$

where $G_M(\omega_M, r, \phi)$ is the discrete far field spectrum gain in relation to the monopole. For the dipole:

$$G_D(\omega_{D\lambda}, r, \phi) \sim \frac{M^2(\omega_{D\lambda} \delta^*/U)^{5/3}}{r(1+M \cos \phi)^2 \{(\omega_{D\lambda} \delta^*/U)^2 + \alpha_p^2\}^{3/2}} \left[[\cos \phi - M]^2 \right] \quad (21)$$

where $G_D(\omega_{D\lambda}, r, \phi)$ is the discrete far field spectrum gain in relation to the dipole representing one of the Rossiter modes (λ). In Eqns. 20 and Eqn. 21, the distance between the source and listener, r , represents how the sound amplitude diminishes with respect to distance in the far field. Figure 3 shows the output from Eqns. 5, 20 and 21, with $r = 1$.

5.5. Tone Bandwidth

As stated in Section 4.4, no exact relationship for the tone bandwidth has been found. It is known that the higher the Reynolds number the more the vortices reduce in size with increased complex interactions leading to a broader width around the peak frequency; laminar flow will have larger, simpler interactions and be closer to a pure tone.

Table 2: Q values with corresponding Reynolds numbers measured from plots given in publications.

Reference	Q	R_{e_L}
[13]	5	5.64×10^5
	4.5	4.52×10^5
	4.5	3.45×10^5
[24]	25	3.10×10^5
[25]	11.875	1.27×10^5
[26]	6	1.47×10^5
	14	3.54×10^6
	14	7.08×10^6
[27]	6.875	2.48×10^6
[28]	7.875	2.95×10^6
[29]	22.5	4.01×10^6
[30]	76	4.51×10^4
	15	6.46×10^5

In signal processing the relationship between the peak frequency and bandwidth is called the Q value, ($Q = f_{D\lambda}/\Delta f$, where Δf is the tone bandwidth at -3dB of the peak). To approximate the relationship between Q and R_{e_L} , the peak frequencies and bandwidths from previously published plots are measured [13, 24–30]. Results are shown in Table 2.

A linear regression line, fitted to the natural logarithm of the Reynolds number, was obtained from the data in Table 2. The equation is given in Eqn. 22:

$$Q = 87.715 - 5.296 \log(R_{e_L}) \quad (22)$$

To prevent Q reaching unrealistic values a limit has been set so that $2 \leq Q \leq 90$.

5.6. Total Output

The implementation described above determines the values used in the final output. The sound is generated from a noise source shaped by a number of bandpass filters. A flow diagram of the synthesis process for a dipole source is shown in figure 4. It can be seen from Fig. 4 how the parameters are interrelated through the different semi-empirical equations ensuring accuracy throughout the sound effect characteristics.

The monopole source frequency due to the depth resonant mode $f_M[n]$ is calculated from Eqn. 9. The noise source is filtered through a bandpass filter with the centre frequency set at $f_M[n]$, Q value set by Eqn. 22, giving the output from the filter, $B_M[n]$. The intensity is calculated from Eqn. 20 as $G_M(\omega_m[n], r, \theta)$, where $\omega_m[n] = 2\pi f_M[n]$, giving total monopole output, $M[n]$:

$$M[n] = G_M(\omega_m[n], r, \theta)B_M[n] \quad (23)$$

The dipole frequencies due to the Rossiter modes are set from a discrete implementation of Eqn. 3 with $\lambda = [1, 4]$; the corresponding frequency values, $f_{D\lambda}[n]$ are calculated from Eqn. 4. The noise source is filtered through a bandpass filter with the centre frequency set at $f_{D\lambda}[n]$, Q value set by Eqn. 22, giving the output from the filter, $B_{D\lambda}[n]$. The intensity is calculated from Eqn. 21 giving a single dipole output, $D_\lambda[n]$:

$$D_\lambda[n] = G_D(\omega_{D\lambda}[n], r, \theta)B_{D\lambda}[n] \quad (24)$$

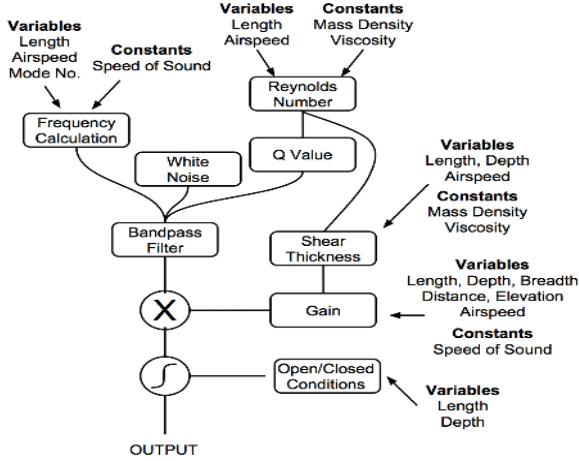


Figure 4: Flow diagram showing typical dipole synthesis process

where $\omega_m[n] = 2\pi f_D[n]$. The total output for the cavity tone $y[n]$ is given as the summation of the output of the resonant monopole and dipoles representing the first 4 Rossiter modes, shown in Eqn. 25;

$$y[n] = M[n] + D_1[n] + D_2[n] + D_3[n] + D_4[n] \quad (25)$$

6. RESULTS

Figure 3 illustrates the outputs from Eqns. 20 and 21 as compared to that given by [14]. The peak gain value occurs at a slightly lower Strouhal number in our model than in Howe’s equation 5. The difference is most likely due to the way the boundary layer thickness is calculated, which is fixed in [14] but varies due to changing conditions in our model.

The plot of the monopole output from our model matches well with that given in [14]. This indicates, for this airspeed, the additional component added in to Eqn 20 containing a dipole term does not diminish accuracy.

Comparison of the model’s frequency calculations to previously published results is shown in Table 3. No specific frequencies are published in [14] but our model shows good agreement with the dominant peak indicated in [14] and the second dominant mode from the model. Our model is designed around the theory presented in [14] and hence a close result was expected.

There are several results presented in [13]; the emphasis of that publication is to provide benchmark data to validate computational aeroacoustic codes. Our physical synthesis model obtains excellent results compared to [13]. The results from our model for the lowest airspeeds (89.2 m/s and 137.2 m/s) are closer to the theoretical predictions while the higher airspeeds are all closer to the published wind tunnel results. The difference between results pertaining to the lower speeds and the higher speed is most likely due to the use of Rossiter’s formula in [13]; our model uses Eqn. 3 while [13] uses Eqn. 2.

Results presented in [31] are in relation to a larger sized cavity. Although the published measured, computational and theoretical results are all close, they do highlight the difference in all three methods. It can be seen that results from our model lie within the range of all the published results. The monopole due to cavity

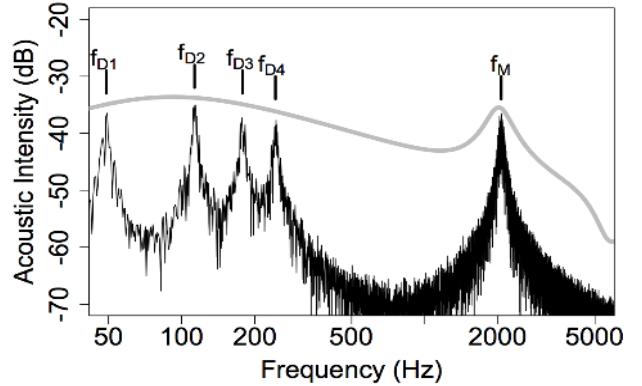


Figure 5: Output from the cavity synthesis model. The grey line indicates the value from Eqn. 5 for comparison. L and $b = 0.03\text{m}$, $d = 0.015\text{m}$, $u = 3.43\text{m/s}$, $\theta = 30^\circ$.

resonance is found to be dominant under these conditions in our model.

The difference between the three results; measured, theoretical and computational, is again highlighted in [32]. The frequencies presented here are higher than perceptible by the human ear but are useful to see how our model performs under these conditions. Again it can be seen that the published theoretical results and our model results are similar but the measured / computational results are slightly higher.

A deep cavity is tested in [24] at a subsonic airspeed; the Rossiter frequencies are not published. The cavity resonance is dominant in our physical model at a frequency of 213Hz, close to the published result of 225Hz. This indicates that our model is accurate when calculating the monopole frequency as well as the Rossiter frequencies, with similar discrepancies previously between measured and theoretical results.

Examination of low airspeed over a wide cavity is examined in [33] as a comparison to a measurement technique called Particle Image Velocimetry, (beyond the scope of this paper). Results indicate a more pronounced difference between the frequencies from this publication to those from our model. Our model correctly predicts the dominant modes but gives noticeably lower frequencies, especially for the 12 m/s case. It is known that theoretical models are less accurate at very low air speeds and at the same time the variation in measured frequencies is much wider [20].

The last example is from [34] when a cube shaped cavity is tested. The results are read from a graph but it can be seen that the values from our model are lower than the published values. For these dimensions the monopole due to cavity resonance is dominant, not the Rossiter frequencies. This is not altogether unexpected since it was stated that if $L/d > 5/2$ resonances can dominate [14], (section 4.1).

Comparing our model’s average frequency prediction to published results we found it was 0.3% lower than theoretical frequencies, 2.0% lower than computed frequencies and 6.4% lower than measured frequencies.

An example output of the cavity tone synthesis model is shown in Fig. 5 indicating f_{D1} , f_{D2} , f_{D3} and f_{D4} where f_{D2} is the dominant mode. The conditions are set to match the example given by Howe [14]. The monopole output is clearly visible at the same frequency as calculated by Howe’s equation.

Table 3: Comparison of published measured, computed and theoretical results and our synthesis model. **Bold** indicates dominant frequency. Ref. = Reference. (* - read from a graph, † - theoretical answer, ‡ - computational answer, ? - Unknown)

Ref.	Airspeed (m/s)	Dimensions (m)			Published Results (Hz)					Physical Model Results (Hz)						
		u	l	w	d	f_{D1}	f_{D2}	f_{D3}	f_{D4}	f_M	f_{D1}	f_{D2}	f_{D3}	f_{D4}	f_M	
[14]	3.43	0.03	0.03	0.015		114*†					49	113	178	243	2061	
[13]	89.2 137.2 181.8 230.5 274.4 308.7	0.0191	0.1016	0.0127		4530					1739	4057	6376	8695	1656	
						4063†										
						5854†					2506	5848	9190	12532		
						7339					3141	7330	11519	15707		
						8062†										
						8809					3773	8804	13835	18865		
						9401†										
						10027					4294	10019	15745	21470		
[31]	291.6	0.4572	0.1016	0.1016		10372†					4680	10919	17159	23398	293	
						10925										
						10941†										
[32]	514.5	0.0045	?	0.0015		195	419	693	947		188	438	689	938	293	
						181†	422†	663†	904†							
						193‡	460‡	667‡	940‡							
[24]	40	0.06	0.06	0.35		31899	71344	110789			28567	66657	104707	142837		
						28469†	66542†	104615†								
[33]	12 15	0.03	0.6	0.015		32242‡	66542‡	126567‡			225	267	623	980	213	
						454		908								
						454‡		908‡			168	391	615	838		
						496		992								
[34]	31	0.15	0.15	0.15	125*	245*	375*				209	487	766	1043	640	
											84	196	308	420	303	

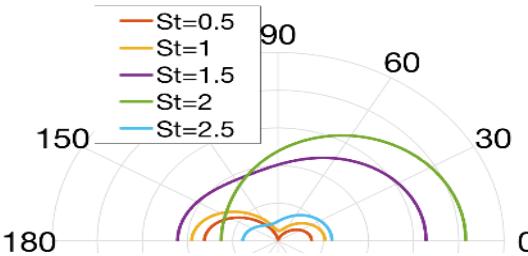


Figure 6: Directional output from synthesis model. L and $b = 0.03\text{m}$, $d = 0.015\text{m}$, airspeed = 34.3m/s. St = Strouhal number.

The directional output is shown in Fig. 6. For different Strouhal numbers there are different propagation angles. The propagation becomes more circular as the frequency increases, indicating the influence of the monopole. Comparing Fig. 6 with published results by Howe [14] indicates that for our model the circular monopole occurs at $St \approx 2$ whereas it occurs at $St \approx 2.5$ in [14].

7. DISCUSSION

The formula given in [16], implemented by our model, performs well for frequency prediction compared to the previously published results in Table 3 obtained through wind tunnel measurements, theoretical predictions and computational fluid dynamics.

The papers with more than one result for the same conditions highlight that it is difficult to predict an exact frequency value from theory, indicating that the theoretical and computational models may not capture all the underlying fluid dynamic processes ongoing when a cavity tone is produced. The wind tunnel recordings may also have been affected by noise and interference from equipment and the tunnel itself.

The values of C_2 and τ_p introduced by Howe [14] in Eqn. 5 may change depending on the cavity dimensions. If so, this would introduce discrepancies between our model and measured outputs.

The acoustic intensity calculated by the model matches well with the theory introduced by Howe [14]. The sound effect is given an independent gain to allow sound designers to achieve the loudness they desire while maintaining the relationships over the frequency range and if combining multiple compact sources.

The directional output from our model is like that produced by Howe [14] but is found to differ at higher strouhal numbers. The cause of this is unknown as there is no difference between the theoretical equation identified by Howe for directional propagation and the equation implemented in the physical model.

The equation from Howe [14], Eqn. 5, takes into account the elevation of the observer to the tone. It does not take into consideration the azimuth angle making the radiated output inherently 2 dimensional rather than 3D. The calculation of the acoustic resonance frequency of the cavity is still based on three dimensions.

This research focused on the sound generated from a cavity as the air passes over it parallel to the length dimension, (Fig. 2). The vast majority of research by the fluid dynamics community

has this condition due to the need to minimise noise generated by wheel wells or bomb bay doors in aircraft. This physical model can be used to calculate the sound produced by cavities under such circumstances. The physics change when there is an angle of attack on the incoming air which is beyond the scope of this paper.

Other sound effects that can be developed in the future using our model as a component part could range from a pipe organ to grooves in the cross section of a sword. The model may also be used to calculate the sound generated by a car travelling with a sunroof or window open or wind passing doorways and other exposed cavities.

A demo of our cavity tone synthesis model is available at <https://code.soundsoftware.ac.uk/hg/cavity-tone>.

8. CONCLUSIONS

We have developed a compact sound source representing the cavity tone based on empirical formulas from fundamental fluid dynamics equations. This has produced a sound synthesis model which closely represents what would be generated in nature. The model operates in real-time and unlike other models, our approach accounts for the angle and distance between source and receiver. Although the solution presented is less detailed than solutions obtained from CFD techniques, results show that the model predicts frequencies close to measured and computational results however unlike other methods our model operates in real-time.

Acoustic intensity and propagation angles derived from fundamental fluid dynamics principles are implemented which overall have good agreement with published results.

9. ACKNOWLEDGMENTS

Supported by EPSRC, grant EP/G03723X/1. Thanks to Will Wilkinson, Brecht De Man, Stewart Black and Dave Moffat for their comments.

10. REFERENCES

- [1] Y Dobashi, T Yamamoto, and T Nishita. Real-time rendering of aerodynamic sound using sound textures based on computational fluid dynamics. In *ACM Transactions on Graphics*, California, USA, 2003.
- [2] D Berckmans et al. Model-based synthesis of aircraft noise to quantify human perception of sound quality and annoyance. *Journal of Sound and Vibration*, Vol. 331, 2008.
- [3] A Farnell. *Designing sound*. MIT Press Cambridge, 2010.
- [4] S Bilbao and CJ Webb. Physical modeling of timpani drums in 3d on gpgpus. *Journal of the Audio Engineering Society*, Vol. 61, 2013.
- [5] F Pfeifle and R Bader. Real-time finite difference physical models of musical instruments on a field programmable gate array. In *Proc. of the 15th Int. Conference on Digital Audio Effects*, York, UK, 2012.
- [6] R Selfridge et al. Physically derived synthesis model of an Aeolian tone. In *Audio Engineering Society Convention 141*, Los Angeles, USA, Best Student Paper Award, 2016.
- [7] R Selfridge, DJ Moffat, and JD Reiss. Real-time physical model of an Aeolian harp. In *24th International Congress on Sound and Vision (accepted)*, London, UK, 2017.
- [8] JW Coltrane. Jet drive mechanisms in edge tones and organ pipes. *The Journal of the Acoustical Society of America*, Vol. 60, 1976.
- [9] MS Howe. Edge, cavity and aperture tones at very low mach numbers. *Journal of Fluid Mechanics*, Vol. 330, 1997.
- [10] JE Rossiter. Wind tunnel experiments on the flow over rectangular cavities at subsonic and transonic speeds. Technical report, Ministry of Aviation; Royal Aircraft Establishment; RAE Farnborough, 1964.
- [11] CKW Tam and PJW Block. On the tones and pressure oscillations induced by flow over rectangular cavities. *Journal of Fluid Mechanics*, Vol. 89, 1978.
- [12] T Colonius, AJ Basu, and CW Rowley. Numerical investigation of flow past a cavity. In *AIAA/CEAS Aeroacoustics Conference*, Seattle, USA, 1999.
- [13] KK Ahuja and J Mendoza. Effects of cavity dimensions, boundary layer, and temperature on cavity noise with emphasis on benchmark data to validate computational aeroacoustic codes. Technical report, NASA 4653, 1995.
- [14] MS Howe. Mechanism of sound generation by low mach number flow over a wall cavity. *Journal of sound and vibration*, Vol. 273, 2004.
- [15] V Sarohia. Experimental investigation of oscillations in flows over shallow cavities. *AIAA Journal*, Vol. 15, 1977.
- [16] HH Heller, DG Holmes, and EEE Covert. Flow-induced pressure oscillations in shallow cavities. *Journal of sound and Vibration*, Vol. 18, 1971.
- [17] MJ Lighthill. On sound generated aerodynamically. i. general theory. *Proc. of the Royal Society of London. Series A. Mathematical and Physical Sciences*, 1952.
- [18] MJ Lighthill. On sound generated aerodynamically. ii. turbulence as a source of sound. In *Proc. of the Royal Society of London A. Mathematical, Physical and Engineering Sciences*, 1954.
- [19] CW Rowley, T Colonius, and AJ Basu. On self-sustained oscillations in two-dimensional compressible flow over rectangular cavities. *Journal of Fluid Mechanics*, Vol. 455, 2002.
- [20] V Suponitsky, E Avital, and M Gaster. On three-dimensionality and control of incompressible cavity flow. *Physics of Fluids*, Vol. 17, 2005.
- [21] T Cebeci and P Bradshaw. *Physical and computational aspects of convective heat transfer*. Springer Science & Business Media, 2012.
- [22] H Schlichting et al. *Boundary-layer theory*. Springer, 1960.
- [23] D G Crighton et al. *Modern methods in analytical acoustics: lecture notes*. Springer Science & Business Media, 2012.
- [24] DD Erickson and WW Durgin. Tone generation by flow past deep wall cavities. In *25th AIAA Aerospace Sciences Meeting*, Reno, USA, 1987.
- [25] YH Yu. Measurements of sound radiation from cavities at subsonic speeds. *Journal of Aircraft*, Vol. 14, 1977.
- [26] HE Plumbe, JS Gibson, and LW Lassiter. A theoretical and experimental investigation of the acoustic response of cavities in an aerodynamic flow. Technical report, DTIC Document, 1962.
- [27] J Malone et al. Analysis of the spectral relationships of cavity tones in subsonic resonant cavity flows. *Physics of Fluids*, Vol. 21, 2009.
- [28] CW Rowley et al. Model-based control of cavity oscillations part ii: system identification and analysis. In *40th AIAA Aerospace Sciences Meeting & Exhibit*, Reno, USA, 2002.
- [29] K Krishnamurty. Acoustic radiation from two-dimensional rectangular cutouts in aerodynamic surfaces. *NACA Technical Note 3487*, 1955.
- [30] N Murray, E Sällström, and L Ukeiley. Properties of subsonic open cavity flow fields. *Physics of Fluids*, Vol. 21, 2009.
- [31] D Fuglsang and A Cain. Evaluation of shear layer cavity resonance mechanisms by numerical simulation. In *30th Aerospace Sciences Meeting and Exhibit*, Reno, USA, 1992.
- [32] X Zhang, A Rona, and JA Edwards. An observation of pressure waves around a shallow cavity. *Journal of sound and vibration*, Vol. 214, 1998.
- [33] V Koschatzky et al. High speed PIV applied to aerodynamic noise investigation. *Experiments in fluids*, Vol. 50, 2011.
- [34] L Chatellier, J Laumonier, and Y Gervais. Theoretical and experimental investigations of low mach number turbulent cavity flows. *Experiments in fluids*, Vol. 36, 2004.

A CONTINUOUS FREQUENCY DOMAIN DESCRIPTION OF ADJUSTABLE BOUNDARY CONDITIONS FOR MULTIDIMENSIONAL TRANSFER FUNCTION MODELS

Maximilian Schäfer and Rudolf Rabenstein

Multimedia Communications and Signal Processing,
Friedrich-Alexander Universität Erlangen-Nürnberg (FAU)
Cauerstr. 7, D-91058 Erlangen, Germany
Max.Schaefer@fau.de, Rudolf.Rabenstein@fau.de

ABSTRACT

Physical modeling of string vibrations strongly depends on the conditions at the system boundaries. The more complex the boundary conditions are the more complex is the process of physical modeling. Based on prior works, this contribution derives a general concept for the incorporation of complex boundary conditions into a transfer function model designed with simple boundary conditions. The concept is related to control theory and separates the treatment of the boundary conditions from the design of the string model.

1. INTRODUCTION

The physical modeling of vibrating strings, e.g. guitar or piano strings is a well studied topic in the area of sound synthesis. A description of the physical phenomena in terms of partial differential equations (PDE) allows the application of different modeling techniques. These methods transform the physical description into computational models by transformations into the respective frequency domain or by discretization in time and space [1, 2].

While a PDE describes the vibrations of a guitar string itself, a suitable set of boundary conditions has to be chosen to achieve a realistic scenario. For a guitar string different kinds of boundary conditions are discussed in [3]. Simple boundary conditions can be defined as conditions for deflection (fixed ends) or its first order space derivatives (free ends). Complex boundary conditions, e.g. impedance boundary conditions are described by a linear combination of physical quantities by a complex boundary impedance.

A suitable modelling technique is the Functional Transformation Method (FTM), based on an expansion into eigenfunctions [4]. These eigenfunctions are most easily determined for simple boundary conditions [5]. The incorporation of complex boundary conditions leads to non-linear equations for the eigenvalues [6].

An alternative approach for the incorporation of complex boundary conditions is based on a suitable state-space description. It is shown for the FTM in [7] for the case of frequency-independent impedance boundary conditions. These concepts are highly related to the basics of open-loop and closed loop systems in control theory as the synthesis algorithm is kept separate from the boundary model [8–10]. Thus the eigenvalues of the simple boundary value problem act as eigenvalues of the open loop system and the eigenvalues of the complex boundary value problem belong to the closed loop system. The closed loop eigenvalues do not need to be calculated explicitly, instead their effect is created by the feedback loop.

This contribution extends the concepts presented in [7, 11] in the continuous frequency domain which hold for a wide range of PDE's of physical interest. The synthesis models based on the

FTM are formulated in terms of a state-space description and a well defined input/output model for the system boundaries is derived. This model leads to a generalized boundary circuit, which shows how the simple boundary conditions are connected to more complex ones.

The presentation is not completely self-contained, since important steps have already been discussed in [7] and references therein. Therefore readers are occasionally referred to the corresponding sections of [7] for details not presented here.

The paper is structured as follows: Sec. 2 gives an overview on the Functional Transformation Method to obtain a transfer function model by modal expansion. Subsequently the synthesis of such systems is reformulated in terms of a state space description in Sec. 3. Sec. 4 presents an extensive input/output model for physical systems and clarifies the terms simple and complex boundary conditions. Especially the connection between both kinds of boundary conditions is highlighted in this section, which leads to a modified state space model in Sec. 5. Sec. 6 recapitulates the string model presented in [7], which is reformulated to fit into the input/output model. In Sec. 7 a bridge model for a guitar is proposed, which is connected to the string model. The simulation results are shown in Sec. 8. Finally Sec. 9 presents several ways in which the model can be extended and applied in further works.

2. MULTIDIMENSIONAL TRANSFER FUNCTION MODEL

For the synthesis of physical systems a multidimensional transfer function model can be obtained by suitable transformation methods, e.g. with the Functional Transformation Method. The method is based on a combination of the Laplace transformation for the time variable and an expansion into spatial eigenfunctions by the Sturm-Liouville transformation.

2.1. Physical Description

The PDEs describing a physical system along with their initial and boundary conditions can be formulated in vector form. The application of a Laplace transform removes the time derivatives and leaves a set of ordinary differential equations. Linear systems which are at rest for $t < 0$ can be formulated in a generic vector form on the spatial region $0 \leq x \leq \ell$

$$[s\mathbf{C} - L]\mathbf{Y}(x, s) = \mathbf{F}_e(x, s), \quad L = \mathbf{A} + \mathbf{I}\frac{\partial}{\partial x}, \quad (1)$$

with the spatial differential operator L and the complex frequency variable s . The vector $\mathbf{F}_e(x, s)$ is the continuous frequency domain equivalent of a vector of time and space dependent functions

$\mathbf{f}_e(x, t)$ exciting the system. The matrix \mathbf{C} contains weighting parameters for the time derivatives, and the matrix \mathbf{A} contains loss parameters and is included in the spatial differential operator L . The vector of variables $\mathbf{Y}(x, s)$ contains the unknown physical quantities. The matrix \mathbf{I} is the identity matrix.

The boundary conditions of the system prescribe the existence of a boundary value Φ for one variable or a linear combination of variables in the vector $\mathbf{Y}(x, s)$ at $x = 0, \ell$. The conditions are formulated in terms of the vector of boundary values and a matrix of boundary conditions \mathbf{F}_b^H acting on the vector of variables

$$\mathbf{F}_b^H(x, s)\mathbf{Y}(x, s) = \Phi(x, s), \quad x = 0, \ell, \quad (2)$$

where the superscript H denotes the hermitian matrix (conjugate transpose).

2.2. Sturm-Liouville Transformation

For application to the space dependent quantities in (1) a Sturm-Liouville Transformation (SLT) is defined in terms of an integral transformation [4]

$$\bar{\mathbf{Y}}(\mu, s) = \mathcal{T}\{\mathbf{Y}(x, s)\} = \int_0^\ell \tilde{\mathbf{K}}^H(x, \mu) \mathbf{C} \mathbf{Y}(x, s) dx, \quad (3)$$

where the vector $\tilde{\mathbf{K}}$ is the kernel of the transformation and $\bar{\mathbf{Y}}(\mu, s)$ is the representation of the vector $\mathbf{Y}(x, s)$ in the complex temporal and spatial transform domain. Defining a suitable kernel \mathbf{K} , the inverse transformation is expressed in terms of a generalized Fourier series expansion

$$\mathbf{Y}(x, s) = \mathcal{T}\{\bar{\mathbf{Y}}(\mu, s)\} = \sum_{\mu=-\infty}^{\infty} \frac{1}{N_\mu} \bar{\mathbf{Y}}(\mu, s) \mathbf{K}(x, \mu), \quad (4)$$

with the scaling factor N_μ . The kernel functions are unknown in general, they can be determined for each problem (1) as the solution of an arising eigenvalue problem [4, 12]. The integer index μ is the index of a discrete spatial frequency variable s_μ for which the PDE (1) has nontrivial solutions [4].

2.3. Properties of the Transformation

The SL-transform described above has to fulfill different properties to be applicable to a PDE (1), e.g. the discrete nature of the eigenvalues s_μ , the adjoint nature of the kernel functions \mathbf{K} and $\tilde{\mathbf{K}}$ and the bi-orthogonality of the kernel functions. The properties of the SLT and the FTM are described in detail in [4, 12]. The mathematical derivations regarding the SLT and the FTM are omitted here for brevity, as this contribution is focussed on the input and output behaviour at the system boundaries.

2.4. Transform Domain Representation

The application of the transformation (3) turns a PDE in the form of Eq. (1) into an algebraic equation of the form (see [7, Eq. (16)])

$$s\bar{\mathbf{Y}}(\mu, s) - s_\mu \bar{\mathbf{Y}}(\mu, s) = \bar{F}_e(\mu, s) + \bar{\Phi}(\mu, s), \quad (5)$$

with the transformed vector of boundary values and the transformed excitation function

$$\bar{\Phi}(\mu, s) = - \left[\tilde{\mathbf{K}}^H(x, \mu) \Phi(x, s) \right]_0^\ell, \quad (6)$$

$$\bar{F}_e(\mu, s) = \int_0^\ell \tilde{\mathbf{K}}^H(x, \mu) \mathbf{F}_e(x, s) dx. \quad (7)$$

Solving (5) for the transformed vector of variables leads to the representation

$$\bar{\mathbf{Y}}(\mu, s) = \bar{H}(\mu, s) [\bar{F}_e(\mu, s) + \bar{\Phi}(\mu, s)], \quad (8)$$

with the multidimensional transfer function

$$\bar{H}(\mu, s) = \frac{1}{s - s_\mu}, \quad \text{Re}\{s_\mu\} < 0. \quad (9)$$

3. STATE SPACE MODEL

The description by a multidimensional transfer function from Sec. 2 can be formulated in terms of a state-space description, basically shown in [7]. This formulation provides many computational benefits (e.g. to avoid delay free loops) and allows the incorporation of boundary conditions of different kinds.

3.1. State Equation

The transform domain representation (5) is reformulated to a form which resembles a state equation for each single value of μ

$$s\bar{\mathbf{Y}}(\mu, s) = s_\mu \bar{\mathbf{Y}}(\mu, s) + \bar{F}_e(\mu, s). \quad (10)$$

Combining all μ -dependent variables into matrices and vectors, expands the state equation to any number of eigenvalues s_μ

$$s\bar{\mathbf{Y}}(s) = \mathbf{A}\bar{\mathbf{Y}}(s) + \bar{\Phi}(s) + \bar{F}_e(s), \quad (11)$$

with the vectors

$$\bar{\mathbf{Y}}(s) = [\dots, \bar{\mathbf{Y}}(\mu, s), \dots]^T, \quad (12)$$

$$\bar{\Phi}(s) = [\dots, \bar{\Phi}(\mu, s), \dots]^T, \quad (13)$$

$$\bar{F}_e(s) = [\dots, \bar{F}_e(\mu, s), \dots]^T, \quad (14)$$

and the state matrix is given as

$$\mathbf{A} = \text{diag}(\dots, s_\mu, \dots). \quad (15)$$

3.2. Output Equation

The inverse transformation from Eq. (4) can be reformulated in vector form as an output equation for the state-space model

$$\mathbf{Y}(x, s) = \mathbf{C}(x)\bar{\mathbf{Y}}(s), \quad (16)$$

with the matrix

$$\mathbf{C}(x) = \left[\dots, \frac{1}{N_\mu} \mathbf{K}(x, \mu), \dots \right]. \quad (17)$$

3.3. Transformed Boundary Term

The transformed boundary term $\bar{\Phi}$ in Eq. (6), can be reformulated in terms of a vector notation to fit into the state equation (11)

$$\bar{\Phi}(s) = \mathbf{B}(0)\Phi(0, s) - \mathbf{B}(\ell)\Phi(\ell, s), \quad (18)$$

with the matrix

$$\mathbf{B}(x) = \begin{bmatrix} \vdots \\ \tilde{\mathbf{K}}^H(x, \mu) \\ \vdots \end{bmatrix}, \quad x = 0, \ell. \quad (19)$$

The state space description with matrices of possibly infinite size is only used to show the parallelism to other state-space descriptions. The matrices $\mathbf{A}, \mathbf{B}(x)$ and $\mathbf{C}(x)$ act as transformation operators rather than as matrices in the sense of linear algebra.

4. INPUT/OUTPUT BEHAVIOUR

When dealing with boundary conditions of physical systems it is important to select which quantities are defined as input and which are outputs of the system (see [10, p. 55]). The number of boundary conditions corresponds to the order of the spatial differential operator, respectively the number of variables in the vector $\mathbf{Y}(x, s)$. In many practical cases, boundary conditions are assigned to half the variables at $x = 0$ and to half of the variables at $x = \ell$. The other variables at either side depend on the system dynamics and can be observed, they are called boundary observations here.

With this distribution of boundary conditions and boundary observations, the vector of variables \mathbf{Y} can be sorted in a way, that the boundary conditions appear on the top of the vector $\mathbf{Y}(x, s)$ as $\mathbf{Y}_1(x, s)$ and those variables, which are assigned to boundary observations appear on the bottom as $\mathbf{Y}_2(x, s)$. To construct these vectors, the variable vector is multiplied by suitable permutation matrices to extract the relevant entries, which leads to a separated representation of the vector of variables

$$\mathbf{Y}(x, s) = \begin{bmatrix} \mathbf{Y}_1(x, s) \\ \mathbf{Y}_2(x, s) \end{bmatrix}. \quad (20)$$

This partitioning carries over to the whole state space description. The eigenfunctions \mathbf{K} and $\tilde{\mathbf{K}}$ from (3), (4) have the same size as the vector $\mathbf{Y}(x, s)$, they can also be divided by applying the same permutation

$$\mathbf{K}(x, \mu) = \begin{bmatrix} \mathbf{K}_1(x, \mu) \\ \mathbf{K}_2(x, \mu) \end{bmatrix}, \quad \tilde{\mathbf{K}}(x, \mu) = \begin{bmatrix} \tilde{\mathbf{K}}_1(x, \mu) \\ \tilde{\mathbf{K}}_2(x, \mu) \end{bmatrix}. \quad (21)$$

With this partitioning, the matrices in the output equation (16) and the transformed boundary vector (18) are also divided

$$\mathbf{C}(x) = \begin{bmatrix} \mathbf{C}_1(x) \\ \mathbf{C}_2(x) \end{bmatrix}, \quad \mathbf{B}(x) = [\mathbf{B}_1(x) \quad \mathbf{B}_2(x)]. \quad (22)$$

4.1. Boundary Conditions

After selecting the input and output variables, the following sections discuss different kinds of boundary conditions and their connection to each other. Sec. 6 presents an application of these general concepts to a physical model of a vibrating string.

4.1.1. Simple Boundary Conditions

This section defines simple boundary conditions at $x = 0, \ell$ of the physical system. Simple boundary conditions are conditions acting on individual entries of $\mathbf{Y}_1(x, s)$ at the outputs of the system. At first the boundary matrices and vectors according to (2) are specialized to

$$\mathbf{F}_b(x, s) = \mathbf{F}_s(x), \quad \Phi(x, s) = \Phi_s(x, s). \quad (23)$$

with

$$\mathbf{F}_s(x) = \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix}, \quad \Phi_s(x, s) = \begin{bmatrix} \Phi_{s1}(x, s) \\ \mathbf{0} \end{bmatrix}. \quad (24)$$

Additionally a matrix of boundary observations is defined

$$\mathbf{G}_s^H(x) = \mathbf{I} - \mathbf{F}_s^H(x) = \begin{bmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{I} \end{bmatrix}, \quad \mathbf{F}_s^H(x) + \mathbf{G}_s^H(x) = \mathbf{I}. \quad (25)$$

With these two matrices the input and output behaviour of the system is described completely. Applying the matrices of boundary conditions (23) and boundary observations (25) to the vector of variables (22) leads to (some arguments (x, s) omitted)

$$\mathbf{F}_s^H \mathbf{Y} = \begin{bmatrix} \Phi_{s1} \\ \mathbf{0} \end{bmatrix}, \quad \mathbf{G}_s^H \mathbf{Y} = \begin{bmatrix} \mathbf{0} \\ \mathbf{Y}_{so} \end{bmatrix}, \quad (26)$$

$$\mathbf{Y}_1(x, s) = \Phi_{s1}(x, s), \quad \mathbf{Y}_2(x, s) = \mathbf{Y}_{so}(x, s). \quad (27)$$

4.1.2. Complex Boundary Conditions

Complex boundary conditions are defined here as linear combinations of physical quantities at the system boundaries $x = 0, \ell$, including time derivatives or complex parameters and functions. Like simple boundary conditions, also complex boundary conditions can be defined in a more general form. Again the matrix of boundary conditions is specialized

$$\mathbf{F}_b(x, s) = \mathbf{F}_c(x, s), \quad \Phi(x, s) = \Phi_c(x, s). \quad (28)$$

with

$$\mathbf{F}_c(x, s) = \begin{bmatrix} \mathbf{F}_{c1}(x, s) & \mathbf{0} \\ \mathbf{F}_{c2}(x, s) & \mathbf{0} \end{bmatrix}, \quad \Phi_c = \begin{bmatrix} \Phi_{c1}(x, s) \\ \mathbf{0} \end{bmatrix}. \quad (29)$$

The matrix $\mathbf{F}_{c1}(x, s)$ has to be chosen as non-singular, except for isolated values of s . Additionally, as in the case of simple boundary conditions, a matrix of boundary observations is defined

$$\mathbf{G}_c(x, s) = \begin{bmatrix} \mathbf{0} & \mathbf{G}_{c1}(x, s) \\ \mathbf{0} & \mathbf{G}_{c2}(x, s) \end{bmatrix}, \quad (30)$$

which transforms the vector of variables into a vector of complex boundary observations

$$\mathbf{G}_c^H(x, s) \mathbf{Y}(x, s) = \begin{bmatrix} \mathbf{0} \\ \mathbf{Y}_{co}(x, s) \end{bmatrix}. \quad (31)$$

Also with the complex boundary conditions the input and output behaviour of the system can be described in a closed form, by applying both matrices (29), (30) to the vector of variables, which leads to

$$\mathbf{F}_{c1}^H(x, s) \mathbf{Y}_1(x, s) + \mathbf{F}_{c2}^H(x, s) \mathbf{Y}_2(x, s) = \Phi_{c1}(x, s), \quad (32)$$

$$\mathbf{G}_{c1}^H(x, s) \mathbf{Y}_1(x, s) + \mathbf{G}_{c2}^H(x, s) \mathbf{Y}_2(x, s) = \mathbf{Y}_{co}(x, s). \quad (33)$$

4.2. Connection between simple and complex Conditions

With the definitions of simple and complex boundary conditions from Sec. 4.1.1 – 4.1.2 their connections can be explored, to express complex boundary conditions in terms of the simple ones. So in general the physical system is described by a bi-orthogonal basis for simple boundary conditions. Then the external connections are formulated in terms of relations between the input and output variables of the complex boundary conditions.

4.2.1. Simple and Complex Boundary Conditions

Starting with the application of complex boundary conditions to the vector of variables \mathbf{Y} and exploiting Eq. (25) leads to

$$\mathbf{F}_c^H(x, s) (\mathbf{F}_s^H(x) + \mathbf{G}_s^H(x)) \mathbf{Y}(x, s) = \Phi_c(x, s). \quad (34)$$

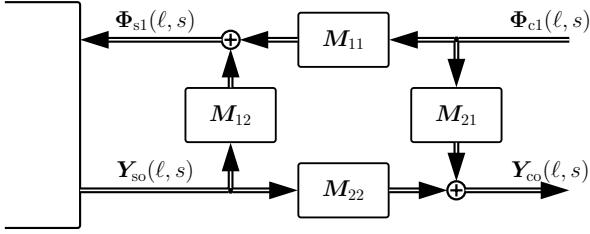


Figure 1: *Boundary circuit for the general connection between simple and complex boundary conditions according to Eq. (35), (38). $\Phi_{s1}(\ell, s)$, $\mathbf{Y}_{so}(\ell, s)$: Inputs and observations of the model for simple boundary conditions. $\Phi_{c1}(\ell, s)$, $\mathbf{Y}_{co}(\ell, s)$: Inputs and observations of the complex boundary circuit. M : Matrices defining the connection between simple and complex boundary conditions.*

Inserting the matrix of complex boundary conditions (29) and the definitions of matrices of the simple case (24), (25) and applying the block partitioning leads to an expression of the simple boundary conditions Φ_{s1} in terms of the complex boundary conditions Φ_{c1} and simple boundary observations \mathbf{Y}_{so}

$$\Phi_{s1}(x, s) = M_{11}(x, s)\Phi_{c1}(x, s) + M_{12}(x, s)\mathbf{Y}_{so}, \quad (35)$$

with the connection matrices

$$M_{11}(x, s) = \mathbf{F}_{c1}^{-H}(x, s), \quad (36)$$

$$M_{12}(x, s) = -\mathbf{F}_{c1}^{-H}(x, s)\mathbf{F}_{c2}^H(x, s). \quad (37)$$

4.2.2. Simple and Complex Boundary Observations

Similar to the boundary conditions, also the boundary observation of the simple and complex case can be connected. The goal is to express the complex boundary observations \mathbf{Y}_{co} in terms of the simple boundary values Φ_{s1} and the simple boundary observations \mathbf{Y}_{so} . Starting with Eq. (33) and exploiting the block structure of matrices and vectors, together with (27), leads to

$$\mathbf{Y}_{co}(x, s) = M_{21}(x, s)\Phi_{c1}(x, s) + M_{22}\mathbf{Y}_{so}(x, s), \quad (38)$$

with the connection matrices

$$M_{21}(x, s) = \mathbf{G}_{c1}^H(x, s)\mathbf{F}_{c1}^{-H}(x, s), \quad (39)$$

$$M_{22}(x, s) = \mathbf{G}_{c2}^H(x, s) - \mathbf{G}_{c1}^H(x, s)\mathbf{F}_{c1}^{-H}(x, s)\mathbf{F}_{c2}^H(x, s). \quad (40)$$

Eq. (35) and (38) describe the connection between simple and complex boundary conditions in terms of boundary values and boundary observations.

Fig. 1 shows this connection between simple and complex boundary conditions for a system boundary (open box on the left side) at $x = \ell$. The physical system on the left side is designed with simple boundary conditions, so that the behaviour at $x = \ell$ is defined by the partitioned vector \mathbf{Y} from (20), respectively by the simple boundary values Φ_{s1} and the simple boundary observations \mathbf{Y}_{so} according to Eq. (27). The figure shows that whenever a system with complex boundary conditions is desired, it is not necessary to redesign the whole physical model. Instead Eqs. (35) and (38) allow to impose the complex boundary values Φ_{c1} on the simple model and to express the complex boundary observations \mathbf{Y}_{co} in terms of the simple ones.

5. MODIFIED STATE SPACE DESCRIPTION

In this section the connection between simple and complex boundary conditions from Sec. 4.2 is incorporated into the state-space description from Sec. 3.

5.1. Transformed Boundary Term

The simple boundary observations \mathbf{Y}_{so} can be derived by the application of the partitioning (20), (22) to the output equation (16)

$$\mathbf{Y}_{so}(x, s) = \mathcal{C}_2(x)\bar{\mathbf{Y}}(s), \quad x = 0, \ell. \quad (41)$$

In the same way, the partitioning (20), (22) together with the simple boundary conditions from Eq. (24) is applied to the transformed boundary term from Eq. (18). The result is

$$\bar{\Phi}(s) = \mathcal{B}_1(0)\Phi_{s1}(0, s) - \mathcal{B}_1(\ell)\Phi_{s1}(\ell, s) = \hat{\mathcal{B}}\hat{\Phi}(s), \quad (42)$$

with the block matrices

$$\hat{\mathcal{B}} = \begin{bmatrix} \mathcal{B}_1(0) \\ -\mathcal{B}_1(\ell) \end{bmatrix}^T, \quad \hat{\Phi}_s(s) = \begin{bmatrix} \Phi_{s1}(0, s) \\ \Phi_{s1}(\ell, s) \end{bmatrix}. \quad (43)$$

Now the connection between simple and complex boundary conditions can be incorporated into the transformed boundary term by inserting Eq. (35) with the boundary observations from (41) into (42). Solving for $\bar{\Phi}(s)$ leads to a representation of the boundary term $\bar{\Phi}$ by the state variable $\bar{\mathbf{Y}}(s)$ and the complex boundary conditions

$$\bar{\Phi}(s) = -\hat{\mathcal{B}}\hat{\mathcal{K}}\bar{\mathbf{Y}}(s) + \mathcal{B}_c\hat{\Phi}_c(s), \quad (44)$$

with the block matrices

$$\hat{\mathcal{K}} = -\begin{bmatrix} M_{12}(0, s)\mathcal{C}_2(0) \\ M_{12}(\ell, s)\mathcal{C}_2(\ell) \end{bmatrix}, \quad \hat{\Phi}_c(s) = \begin{bmatrix} \Phi_{c1}(0, s) \\ \Phi_{c1}(\ell, s) \end{bmatrix}, \quad (45)$$

$$\mathcal{B}_c = [\mathcal{B}_1(0)M_{11}(0, s) \quad -\mathcal{B}_1(\ell)M_{11}(\ell, s)]. \quad (46)$$

5.2. State Equation and Output Equation

Including (44) into the state equation (11) leads to a modified state equation, where the complex boundary conditions are incorporated

$$s\bar{\mathbf{Y}}(s) = \mathcal{A}_c\bar{\mathbf{Y}}(s) + \mathcal{B}_c\hat{\Phi}_c(s) + \bar{\mathbf{F}}_e(s), \quad (47)$$

with the modified state feedback matrix

$$\mathcal{A}_c = \mathcal{A} - \hat{\mathcal{B}}\hat{\mathcal{K}}. \quad (48)$$

The structure of the modified state equation shows that the boundary circuit from Fig. 1 is equivalent to a state feedback structure. The state matrix \mathcal{A} of the state space description contains the poles of the physical system, Eq. (48) shows the influence of the boundary circuit which shifts the poles of the system with simple boundary conditions.

The output equation (16) of the state-space description is not directly affected by the incorporation of complex boundary conditions.

5.3. Relation to Control Theory

The modification of the open loop behaviour by feedback as introduced in Sec. 5.2 is well known in the literature on control theory [8, 10]. The fact that system interconnection constitutes a feedback control loop is highlighted e.g. in [10, Fig. 15]. The notation in (48) has been chosen to reflect the description of closed loop systems in e.g. [9, Eq. (19)].

6. STRING MODEL

This section presents a multidimensional transfer function model of a guitar string based on [7]. The derivations are omitted for brevity; instead the results and especially the boundary conditions from [7] are rewritten to fit the input/output model from Sec. 4.

6.1. Physical Description

A single vibrating string can be described by a PDE [1, Eq. (6)] in terms of the deflection $y = y(x, t)$ depending on time t and space x on the string

$$\rho A \ddot{y} + EI \ddot{y}''' - T_s y'' + d_1 \dot{y} - d_3 y'' = f_e, \quad (49)$$

where \dot{y} represents the time- and y' the space-derivative and with the cross-section area A , moment of inertia I and the length ℓ . The material is characterized by the density ρ and Young's modulus E . T_s describes the tension and d_1 and d_3 are frequency independent and frequency dependent damping [1]. The space and time dependent excitation function is defined as $f_e = f_e(x, t)$.

The PDE in (49) can be rearranged in the vector form (1) as shown in [7]. To fit the vector of variables into the input/output model (20), the vector is rearranged with suitable permutation matrices. Therefore the vector of variables defined in [7] can be partitioned

$$\mathbf{Y}_1(x, s) = \begin{bmatrix} sY(x, s) \\ Y''(x, s) \end{bmatrix}, \quad \mathbf{Y}_2(x, s) = \begin{bmatrix} Y'(x, s) \\ Y'''(x, s) \end{bmatrix}. \quad (50)$$

6.2. Simple Boundary Conditions

A set of simple boundary conditions for a string fixed at both ends $x = 0, \ell$ is formulated as a set of equations for the deflection and bending moment

$$sY(x, s) = \Phi_1(x, s), \quad Y''(x, s) = \Phi_2(x, s) \quad x = 0, \ell. \quad (51)$$

This set of linear equations can be formulated in terms of a matrix of boundary conditions and a vector of boundary excitations according to (2) as shown in [7]. From here on boundary conditions of the first kind are called simple boundary conditions and they are formulated as described in Sec. 4.1.1 in Eq. (24) with

$$\Phi_{s1}(x, s) = \begin{bmatrix} \Phi_1(x, s) \\ \Phi_2(x, s) \end{bmatrix}, \quad x = 0, \ell. \quad (52)$$

6.3. Complex Boundary Conditions

As a set of complex boundary conditions, impedance boundary conditions are used at $x = 0$ according to [7]. They are formulated as a set of linear equations, combining the string deflection Y with a force F by a frequency dependent admittance $Y_b(s)$

$$sY(0, s) - Y_b(s)F(0, s) = \Phi_{c1}, \quad Y''(0, s) = \Phi_{c2}. \quad (53)$$

The force can be expressed as a combination of two forces reformulated in terms of the derivatives of string deflection [13]

$$F(0, s) = T_s Y'(0, s) - EI Y'''(0, s). \quad (54)$$

The boundary conditions are rewritten as shown in Sec. 4.1.2 to fit into the input/output model, with the matrices of boundary conditions

$$\mathbf{F}_{c1}^H(0, s) = \mathbf{I}, \quad \mathbf{F}_{c2}^H(0, s) = Y_b(s) \begin{bmatrix} -T_s & EI \\ 0 & 0 \end{bmatrix}, \quad (55)$$

and the vector of boundary excitations

$$\Phi_{c1}(0, s) = \begin{bmatrix} \Phi_{c1}(0, s) \\ \Phi_{c2}(0, s) \end{bmatrix}. \quad (56)$$

For the position $x = \ell$ the simple boundary conditions from Sec. 6.2 are applied. It then follows for the matrix of boundary conditions at $x = \ell$

$$\mathbf{F}_{c1}^H(\ell, s) = \mathbf{I}, \quad \mathbf{F}_{c2}^H(\ell, s) = \mathbf{0}, \quad (57)$$

and for the boundary observations

$$\Phi_{c1}(\ell, s) = \Phi_{s1}(\ell, s). \quad (58)$$

6.4. Kernel Functions

To fit into the input/output model from Sec. 4 also the kernel functions from [7] have to be rearranged. For the primal kernel function follows according to (21) with suitable permutations

$$\mathbf{K}_1(x, \mu) = \begin{bmatrix} \frac{s_\mu}{\gamma_\mu} \sin(\gamma_\mu x) \\ -\gamma_\mu \sin(\gamma_\mu x) \end{bmatrix}, \quad \mathbf{K}_2(x, \mu) = \begin{bmatrix} \cos(\gamma_\mu x) \\ -\gamma_\mu^2 \cos(\gamma_\mu x) \end{bmatrix}, \quad (59)$$

where \mathbf{K}_1 is assigned to the boundary conditions and \mathbf{K}_2 to the boundary observations. The same partitioning is applied to the adjoint kernel function

$$\tilde{\mathbf{K}}_1(x, \mu) = \begin{bmatrix} q_1^* \cos(\gamma_\mu x) \\ -\gamma_\mu^2 \cos(\gamma_\mu x) \end{bmatrix}, \quad \tilde{\mathbf{K}}_2(x, \mu) = \begin{bmatrix} -\frac{s_\mu^* q_1^*}{\gamma_\mu} \sin(\gamma_\mu x) \\ \gamma_\mu \sin(\gamma_\mu x) \end{bmatrix}, \quad (60)$$

with the wave numbers γ_μ and the coefficient q_1 defined in [7].

6.5. Modified State Space Model

For the synthesis of the string model (49) normally Eq. (4) can be used as a superposition of first order systems. Alternatively a state-space model can be set up for the synthesis (see Sec. 3). Here the modified state-space model from Sec. 5 is applied for the incorporation of complex boundary conditions (see Sec. 6.3).

6.5.1. Output Equation

The output equation is designed according to Eq. (16) with the output matrix $\mathcal{C}(x)$ from Eq. (17), which is divided according to (22). For the design of the matrices \mathcal{C}_1 and \mathcal{C}_2 , the kernel functions (59), (60) and the scaling factor N_μ (see [7, Eq. (15)]) are necessary.

6.5.2. State Equation

The state equation for the synthesis of the string model is designed according to Eq. (47). The modified state feedback matrix (48) is constructed via the state matrix \mathbf{A} using the eigenfrequencies as the solution of the dispersion relation [7]

$$s_\mu^2 + \left(\frac{a_1}{c_1} - \lambda^2 \frac{c_2}{c_1} \right) s_\mu - \frac{\lambda^2}{c_1} (\lambda^2 + a_2) = 0. \quad (61)$$

The extension of the state matrix to incorporate complex boundary conditions follows the principle shown in Sec. 5 using the matrices in Eq. (45), (46).

These matrices are established with

- the feedback matrices M_{12} and M_{11} from (36)-(37) with the complex boundary conditions from (55), (57),
- the partitioned output matrices $\mathcal{C}(x)$ according to (22) and (17),
- the complex boundary excitations (56), (58),
- the transformation matrices of the transformed boundary term $\tilde{\Phi}$ according to (42) and (19).

With a state equation designed in this form, the complex boundary conditions are incorporated into a string model designed with simple boundary conditions as shown in Fig. 1.

7. BRIDGE MODEL

As a realistic boundary circuit for the impedance boundary conditions (53), a frequency dependent bridge model is chosen as discussed in [6]. A bridge model resembles the influence of a guitar bridge, where the string is attached to the bridge at the position $x = 0$.

For simplicity the bridge admittance is designed as the superposition of N second order bandpass filters

$$Y_b(s) = \sum_{n=1}^N Y_n \frac{s \frac{\omega_n}{Q_n}}{s^2 + \frac{\omega_n}{Q_n} s + \omega_0^2}, \quad (62)$$

with the quality factor Q_n and the resonance frequency $\omega_n = 2\pi f_n$ for each mode. An additional gain factor Y_n is added to each bandpass. In general the admittance functions for the bridge are position and direction dependent, so the admittance differs for different strings [6].

The physical parameters for the bridge resonances can be taken from measurements as e.g. shown in [14]. There the lowest six most significant eigenmodes of the bridge are realized with damped sinusoids.

Figure 2 shows the first six prominent modes of the bridge admittance $Y_b(s)$ in the frequency domain. The resonance frequencies ω_n and the quality factors Q_n are taken from [14]. The gain factors are related to the effective masses $Y_n = \frac{1}{m_n}$ as presented in [6], with the physical values for the effective masses from [14].

For simulations in the continuous frequency domain, the admittance function from Eq. (62) is directly used. The function is just inserted into the boundary conditions (55) and incorporated into the modified state space model described in Sec. 6.5.

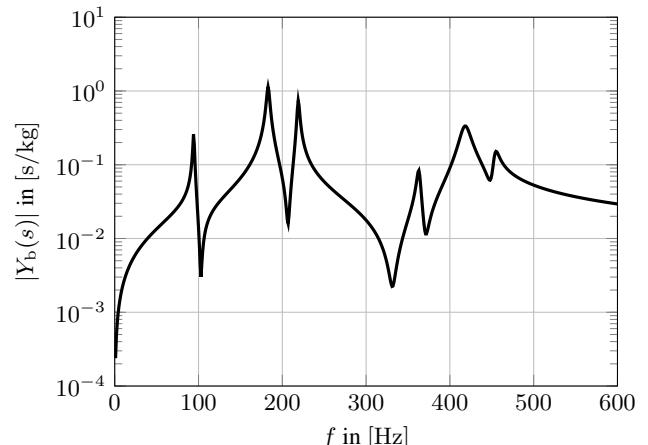


Figure 2: Absolute value of the bridge admittance $Y_b(s)$ according to Eq. (62). Shown are the $N = 6$ prominent modes [14].

8. SIMULATION RESULTS

The following section shows the simulation results for a guitar string including a boundary circuit. All simulation results are computed using the modified state-space model as shown in Sec. 6.5, with the state equation (47) and the output equation (16).

All calculations are performed in the continuous frequency domain and the results are presented in terms of the amplitude spectra of the bending force $F_{\text{bend}}(x, s)$ in y -direction in the lower frequency range. The bending force is calculated according to

$$F_{\text{bend}}(x, s) = EIY'''(x, s), \quad (63)$$

where several valid simplifications were applied [3, 13].

8.1. Boundary Conditions and Excitation Function

The complex boundary conditions are chosen for the frequency domain simulations of the string according to Sec. 6.3. The boundary values Φ_{c1}, Φ_{c2} of the complex boundary conditions are set to zero for brevity. The admittance $Y_b(s)$ is interpreted as the admittance of the guitar bridge, where the string is placed on. The admittance is varied for the following simulations.

The function $f_e(x, t)$ exciting the string is a single impulse at the position $x_e = 0.3$ m on the string, as shown in [7]. In the frequency domain this excitation leads to a uniform excitation of all frequency components. The simulation of output variables in response to the excitation function can be seen as an impedance-analysis of the string.

For all simulations a nylon guitar low E-String is used. The values for the physical parameters in Eq.(49) are taken from [6, 15, 16].

8.2. Frequency independent Bridge Impedance

In a first step the simulation is performed for a frequency independent constant bridge admittance $Y_b(s) = \text{const.}$ Figure 3 shows the normalized amplitude spectra of the bending force $|F_{\text{bend}}(0, s)|$ at the bridge position $x = 0$ for the lower frequency range. The bridge admittance is varied $Y_b(s) = 0 \text{ s/kg}, 2 \text{ s/kg}, 5 \text{ s/kg}$.

For zero admittance (solid curve in Fig. 3) the modes of the guitar string can be seen clearly at the fundamental frequency and the higher modes. The behaviour complies with that of a string model with simple boundary conditions as the feedback path in the modified state matrix (48) is zero. The results for a string with simple boundary conditions are confirmed in [5].

For increasing bridge admittance (dotted, dashed curve in Fig. 3) all modes of the guitar string are damped equally. These results are not realistic as a real bridge impedance is strongly frequency dependent. The results for a frequency independent admittance show the validity of the presented concepts for a feedback structure for a generic example and the results of [7] are verified.

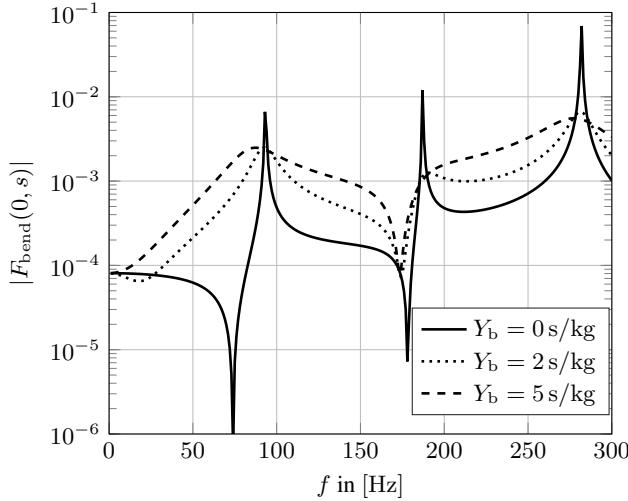


Figure 3: Absolute value of the spectrum of bending force $F_{\text{bend}}(0, s)$ at the bridge position $x = 0$, for three different frequency independent bridge admittances $Y_b = \text{const}$.

8.3. Frequency dependent Bridge Impedance

Now a frequency dependent bridge admittance $Y_b(s)$ is used for the simulations according to Sec. 7 including the first six ($N = 1, \dots, 6$) modes of a measured bridge impedance (see Fig. 2). The physical values for the mode resonance frequencies f_n , the Q -values and the effective masses are taken from [14, Table 1].

The results of the string simulation for a frequency dependent bridge model are pictured in Fig. 4. The figure shows the amplitude spectrum of bending force $F_{\text{bend}}(0, s)$ at $x = 0$ for a zero bridge admittance $Y_b = 0$ (solid curve) and a frequency dependent bridge admittance $Y_b(s)$ according to Eq. (62) (dotted curve).

To have a comprehensive representation of the string behaviour the bridge admittance (as shown in Fig. 2) is plotted into Fig. 4 to indicate the mode positions of the bridge model (dashed curve). The frequency range is limited to the interesting range, where the bridge modes influence the string vibration [14, Table 1].

In Fig. 4 the influence of the bridge admittance on the string vibration can be seen clearly. Depending on the resonance frequencies f_n and the corresponding amplitudes of the bridge admittance, the modes of the vibrating string are shaped. Some of the peaks are damped completely (around 180 Hz in Fig. 4) other peaks are shifted in frequency according to the profile of $Y_b(s)$.

Fig. 5 shows the influence of the bridge admittance $Y_b(s)$ on the bending force $F_{\text{bend}}(x_e, s)$ according to Eq. (63) at the excitation position $x = x_e$. It can be seen that in the frequency independent case (dotted line) all string resonances are damped in the same way. For a frequency dependent bridge admittance (dashed line) the modes are influenced according to the shape of the bridge admittance.

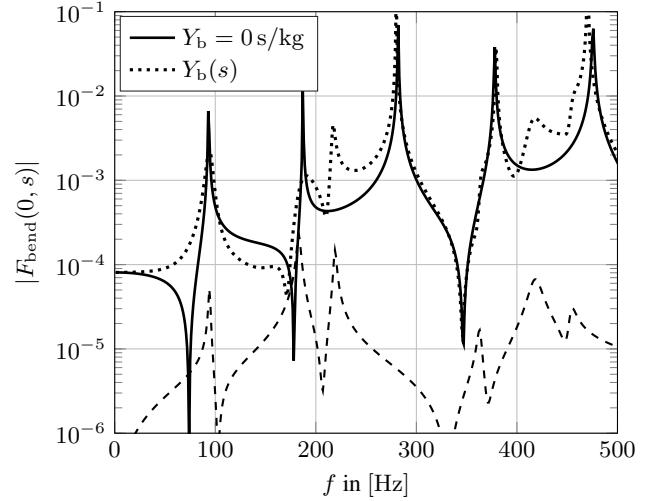


Figure 4: Absolute value of the spectrum of bending force $F_{\text{bend}}(0, s)$ at the bridge position $x = 0$ for zero bridge admittance (solid line) and for a frequency dependent bridge admittance according to Eq. (62) in Sec. 7 (dotted line). The dashed black line is an amplitude shifted version of the bridge admittance $Y_b(s)$, plotted for illustration.

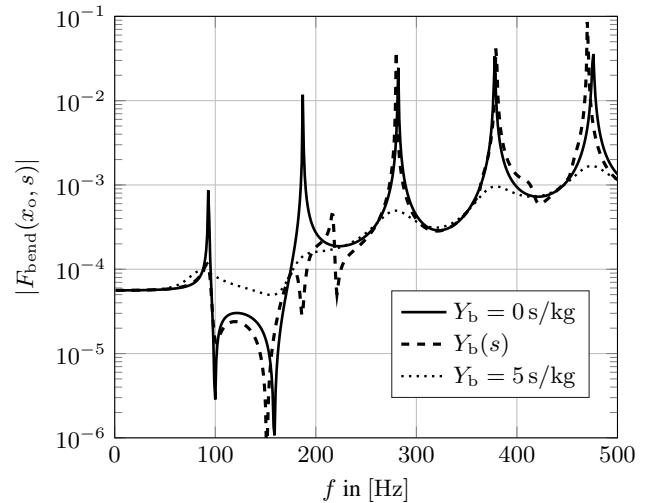


Figure 5: Absolute value of the spectrum of bending force $F_{\text{bend}}(x_e, s)$ at the excitation position $x_e = 0.3$ m for zero bridge admittance (solid line), for a frequency dependent bridge admittance according to Eq. (62) in Sec. 7 (dashed line) and for a constant frequency independent bridge admittance $Y_b = \text{const}$ (dotted line).

9. FURTHER WORKS

This contribution presents a general approach for the incorporation of complex boundary conditions into systems designed with simple boundary conditions based on concepts from control theory [8–10]. This method can be used in many applications and can be further improved. Therefore the following further works are envisaged: The bridge admittance (62) was chosen exemplary as a superposition of bandpass filters including realistic resonance frequencies and quality factors based on [14]. For further works the model can be extended to include more realistic damping effects. The concepts shown here can be used to link a body model to the bridge in a block-based style.

The presented concepts are based on a modified state-space description (see Sec. 5) including a feedback matrix in the continuous frequency domain. These concepts can be transformed into the discrete-time domain for real-time simulations, which requires a re-design of the state-space description from Sec. 3 and Sec. 5. Suitable state space techniques have been presented in [7, Secs. 3–5].

Interactions of the model at any position on the string also can be realized with the presented concepts. According to Sec. 4 an input/output model including the excitation function $f_e(x, t)$ can be derived, e.g. for string-fretboard or string-finger interaction.

The concept for the incorporation of complex boundary conditions can be extended to more than one spatial dimension. E.g. the eigenfunctions for the 2D plate equations cannot be derived analytically except for the most simple boundary conditions [17]. The presented concept appears to be a promising approach for the simulation of the plate equation with complex boundary conditions.

10. CONCLUSIONS

This paper presented a concept for the incorporation of complex boundary conditions into systems designed with simple boundary conditions. Building on prior work this contribution develops an input/output description for systems based on transfer function models. Subsequently it has been shown that complex boundary conditions (e.g. impedance boundary conditions) can be included into a model by the design of a feedback loop related to control theory. The concepts allow to change the boundary behaviour of a system without changing the interior model of the system.

The validity of the presented concepts is verified by an extensive example for modelling of guitar strings. A bridge model is added to an existing string model based on a multidimensional transfer function. The results are presented in terms of several spectra of the string resonances.

Acknowledgment: The authors wish to thank Joachim Deutscher for valuable discussions on issues related to control theory and the anonymous reviewers for numerous suggestions.

11. REFERENCES

- [1] J. Bensa, S. Bilbao, R. Kronland-Martinet, and J. O. Smith, “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. [Online]. Available: <http://scitation.aip.org/content/asa/journal/jasa/114/2/10.1121/1.1587146>
- [2] A. Chaigne and A. Askenfelt, “Numerical simulations of piano strings. Part 1: A physical model for a struck string using finite difference methods,” Tech. Rep., Apr. 1993.
- [3] N. H. Fletcher and T. D. Rossing, *The Physics of Musical Instruments*, 2nd ed. New York, USA: Springer-Verlag, 1998.
- [4] R. V. Churchill, *Operational Mathematics*. Boston, Massachusetts: Mc Graw Hill, 1972.
- [5] R. Rabenstein and L. Trautmann, “Digital sound synthesis of string instruments with the functional transformation method,” *Signal Processing*, vol. 83, pp. 1673–1688, 2003.
- [6] L. Trautmann, S. Petrausch, and M. Bauer, “Simulations of string vibrations with boundary conditions of third kind using the functional transformation method,” *The Journal of the Acoustical Society of America (JASA)*, vol. 118, no. 3, pp. 1763–1775, September 2005.
- [7] M. Schäfer, P. Frenštátský, and R. Rabenstein, “A physical string model with adjustable boundary conditions,” in *19th International Conference on Digital Audio Effects (DAFx-16)*, Brno, Czech Republic, September 2016, pp. 159 – 166.
- [8] J. Deutscher, *Zustandsregelung verteilt-parametrischer Systeme*. Heidelberg: Springer, 2012.
- [9] A. Mohr and J. Deutscher, “Parametric state feedback design for linear infinite-dimensional systems,” in *2013 European Control Conference (ECC)*, July 2013, pp. 2086–2091.
- [10] J. C. Willems, “The behavioral approach to open and interconnected systems,” *IEEE Control Systems Magazine*, vol. 27, no. 6, pp. 46–99, Dec 2007.
- [11] R. Rabenstein and S. Petrausch, “Adjustable boundary conditions for multidimensional transfer function models,” in *10th International Conference on Digital Audio Effects (DAFx-07)*, Bordeaux, France, September 2007, pp. 305–310.
- [12] S. Petrausch and R. Rabenstein, “A simplified design of multidimensional transfer function models,” in *International Workshop on Spectral Methods and Multirate Signal Processing (SMMS2004)*, Vienna, Austria, September 2004, pp. 35–40.
- [13] L. Trautmann and R. Rabenstein, *Digital Sound Synthesis by Physical Modeling using the Functional Transformation Method*. New York, USA: Kluwer Academic Publishers, 2003.
- [14] T. J. W. Hill, B. E. Richardson, and S. J. Richardson, “Acoustic parameters for the characterisation of the classical guitar,” *Acta Acustica united with Acustica*, vol. 90, no. 2, pp. 335–348, 2004.
- [15] J. D’Addario & Company, Inc. (2017) A complete technical reference for fretted instrument string tensions. Last access: June 5, 2017. [Online]. Available: http://www.daddario.com/upload/tension_chart_13934.pdf
- [16] O. Christensen, “An oscillator model for analysis of guitar sound pressure response,” *Acta Acustica united with Acustica*, vol. 54, no. 5, 1984.
- [17] S. Bilbao, *Numerical Sound Synthesis*. Chichester, UK: John Wiley and Sons, 2009.

EVERTIMS: OPEN SOURCE FRAMEWORK FOR REAL-TIME AURALIZATION IN ARCHITECTURAL ACOUSTICS AND VIRTUAL REALITY

David Poirier-Quinot

Acoustic and Cognitive Spaces Group
IRCAM, CNRS, Sorbonne University, UPMC
Paris 6, UMR 9912, STMS
Paris, France
david.poirier-quinot@ircam.fr

Brian F.G. Katz

Institut Jean Le Rond d'Alembert
Sorbonne Universités
UPMC Univ Paris 06, CNRS
Paris, France
brian.katz@upmc.fr

Markus Noisternig

Acoustic and Cognitive Spaces Group
IRCAM, CNRS, Sorbonne University, UPMC
Paris 6, UMR 9912, STMS
Paris, France
markus.noisternig@ircam.fr

ABSTRACT

This paper presents recent developments of the EVERTims project, an auralization framework for virtual acoustics and real-time room acoustic simulation. The EVERTims framework relies on three independent components: a scene graph editor, a room acoustic modeler, and a spatial audio renderer for auralization. The framework was first published and detailed in [1, 2]. Recent developments presented here concern the complete re-design of the scene graph editor unit, and the C++ implementation of a new spatial renderer based on the JUCE framework. EVERTims now functions as a Blender add-on to support real-time auralization of any 3D room model, both for its creation in Blender and its exploration in the Blender Game Engine. The EVERTims framework is published as open source software: <http://evertims.ircam.fr>.

1. INTRODUCTION

Auralization is “*the process of rendering audible, by physical or mathematical modeling, the sound field of a source in space [...] at a given position in the modeled space*” [3]. Room acoustic auralization generally relies on Room Impulse Responses (RIR), either measured or synthesized. An RIR represents the spatio-temporal distribution of reflections in a given room upon propagation from an emitter (source) to a receiver (listener). Fig. 1 details several conceptual components of an RIR.

Auralization has some history of being used in architectural design, e.g. for the subjective evaluation of an acoustic space during the early stage of its conception [4]. It also serves for perceptually evaluating the impact of potential renovation designs on existing spaces. Auralization is often used as a supplement to objective parameter metrics [5, 6, 7]. While previously limited to industry and laboratory settings, auralization is now receiving particular attention due to the emergence of Virtual Reality (VR) technologies [8, 9]. Lately, most research has been driven by the need for (1) real-time optimization and (2) accuracy and realism of the simulated acoustics [10, 11, 12], where these two goals drive developments in opposing directions. The phenomenon is similar to the race for real-time lighting engines and the advent of programmable shaders [13].

An overview of the many available techniques for simulating the acoustics of a given geometry has been presented in [4, 10, 14], detailing both ray-based [15, 16] and wave-based techniques [17, 18]. Ray-based techniques are based on a high frequency approximation of acoustic propagation as geometric rays, ignoring diffraction and other wave effects. They usually outperform wave-based techniques regarding calculation speed.

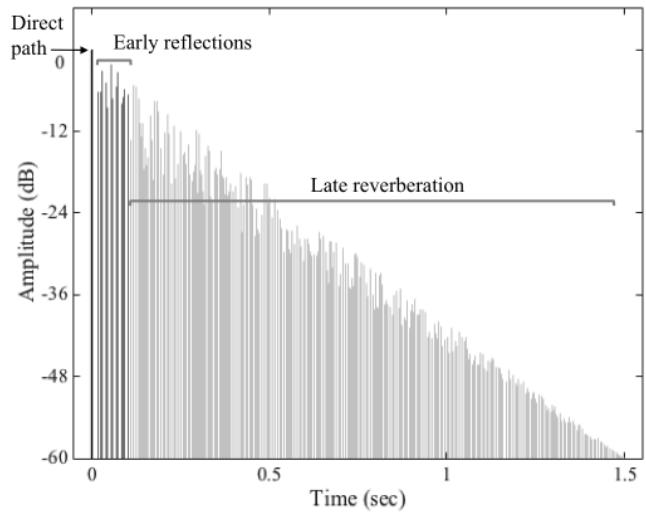


Figure 1: Theoretical plot (not generated in EVERTims) illustrating the conceptual components of a RIR: direct path, early reflections, and late reverberation. EVERTims relies on an image source model to simulate the early reflections, and an FDN to generate the late reverberation.

A further distinction is made here between ray-based techniques, referring to the family of methods based on the geometrical acoustics hypothesis, and ray-tracing techniques [19], one of its subfamily. As detailed in [2], the EVERTims room acoustic modeler relies on an *image source* technique [16] (ray-based). The implementation is based on a beam tracing algorithm [2], focused on real-time estimation of specular early reflections.

A number of geometrical acoustic simulation programs have been developed having auralization capabilities, either tailored for architectural acoustics [20, 21, 22] or acoustic game design [23, 24], e.g.: 3DCeption Spatial Workstation¹, DearVR², and VisiSonics RealSpace 3D³ (as most of these programs are closed-source, the authors cannot guarantee that they are strictly based on geometrical acoustic models). The level of realistic detail achieved is typically inversely proportional to the real-time capabilities. The ambition in developing EVERTims has been to combine the accuracy of the former and the performance of the latter, much like the approach of the RAVEN [25] based plugin presented in [22].

¹Two Big Ears website: www.twobigears.com/spatworks

²DearVR website: www.dearvr.com

³VisiSonics website: www.visisonics.com

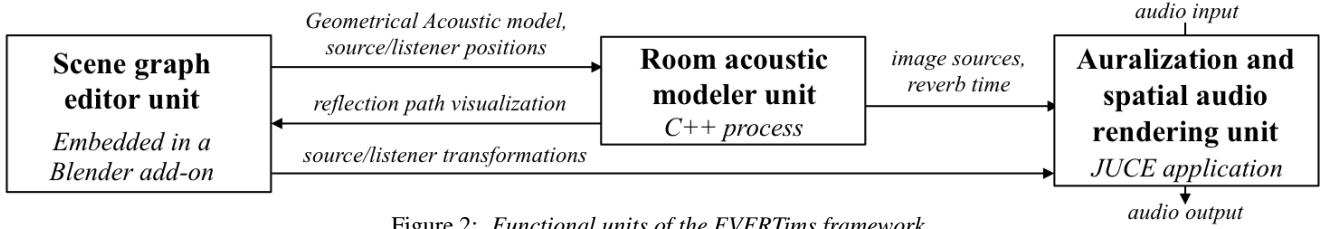


Figure 2: Functional units of the EVERTims framework.

It does not claim to outperform any of the above-mentioned softwares, but rather to provide a reliable, simple, and extensible tool for acousticians, researchers, and sound designers.

The main contribution of this paper is to present the evolution of the EVERTims project since its first publication in [2]. Recent developments include the implementation of a standalone auralization rendering unit and a Blender add-on for controlling the auralization process. Sec. 3 introduces the Blender add-on, which replaces the previously implemented VirChor scene graph editor. Only minor revisions have been made to the room acoustic modeler that is briefly described in Sec. 4. The implementation of the new auralization and spatial audio rendering unit is described in Sec. 5. Sec. 6 summarizes the current state of the framework and outlines future developments.

2. FRAMEWORK OVERVIEW

Fig. 2 shows the functional units of the EVERTims framework. The scene graph editor handles the room geometry and the acoustic properties of the wall surface materials, referred to as the room’s Geometrical Acoustic (GA) model, as well as the user interaction. This editor, implemented as a Blender add-on, provides a Graphical User Interface (GUI) from which the room acoustic modeler, the auralization unit, and the real-time auralization process (i.e. initiating the data exchange between the functional units) can be started. The room acoustic modeler runs as a background subprocess, while the auralization unit runs as a standalone application.

In order to support the interchangeability of algorithms the main functional units are fully encapsulated and self-contained. The communication and data exchange is based on the Open Sound Control (OSC) protocol [26]. This allows the distribution of sub-tasks to different computers, or to run the entire environment on a single computer. During the auralization, the Blender add-on streams GA model information, along with listener and source positions, to the acoustic modeler. Based on these data, the acoustic modeler builds a list of image sources, corresponding to a minimum reflection order defined at start-up (see Sec. 4). These image sources, along with an estimated reverberation time for the current GA model, are then sent to the auralization unit. The Blender add-on also streams source and listener transformation matrices to the auralization unit. Based on the data issued from both the add-on and the modeler, the auralization unit generates and spatialises image sources as early reflections, as well as constructs the late reverberation. An additional callback can be initiated by the add-on to display the results of the acoustic modeler simulation as reflection paths drawn in the 3D scene for monitoring.

EVERTims uses an iterative refinement procedure and computes higher reflection orders progressively. Whenever there is a change in the geometry or a sound source position, an approximate beam-tree up to the minimum reflection order is computed.

The visible paths are then sent to the auralization unit. When there is no change, it continues to compute the solution up to the next reflection order until the chosen maximum order is reached. When a listener moves, visibility tests are computed for all the paths in the beam-trees for that listener and changes are sent to the auralization unit. Changes in source/listener orientation do not affect the beam-tree and visibility of paths; there is no need to perform any recalculation. A change to the geometry or the source position is computationally expensive, as it requires a new computation of the underlying beam tree, reconstructing the beam-tree up to the minimum order, and beginning the iterative order refinement of the solution once again.

For architects and acousticians, the core feature of the framework is to provide both a reactive auralization during room design and exploration, providing an interactive rendering based on the actual geometry of the space, and an accurate auralization for final room acoustics assessment. The same framework allows for integration into game engines for real-time applications. At any time, intermediate rendering results can be exported as an RIR in different audio formats (e.g., binaural, Ambisonic), suitable for use in any convolution-based audio renderer.

3. SCENE GRAPH EDITOR AND 3D VISUALIZATION

The EVERTims scene graph editor and 3D visualizations are handled in Blender. The add-on is written in Python, integrated in Blender 3D View Toolbox. The approach is similar to the one proposed in [27]. The add-on itself handles EVERTims specific data (GA, assets import, OSC setup, etc.) and processes. To start the auralization, the add-on requires four specific EVERTims elements: *room*, *source*, *listener*, and *logic*. These elements can either be defined from existing objects in the scene, or imported from an assets library packaged with the add-on.

The *source* and *listener* can be attached to any object in the 3D scene. The *room* can be defined from any mesh geometry; the add-on does not apply any checks on the room geometry (e.g., convexity, closed form), allowing EVERTims to work for open or sparse scenes as well as closed spaces. An acoustic material must be assigned to each room mesh face. The wall surface materials are defined in the EVERTims materials library, imported from the assets pack, and shared with the room acoustic modeler unit (see Sec. 4). Each acoustic material is defined by a set of absorption coefficients (10 octave bands: 32 Hz–16 kHz). The materials library consists of a human-readable text file which is easy to modify and extend.

The *logic* object handles the parameters of the Blender add-on when EVERTims is running in *game engine mode* for real-time auralization. This mode uses the Blender Game Engine for building virtual walkthroughs. Its counterpart, the EVERTims *edit mode*, auralizes the room model in real-time during design and creation

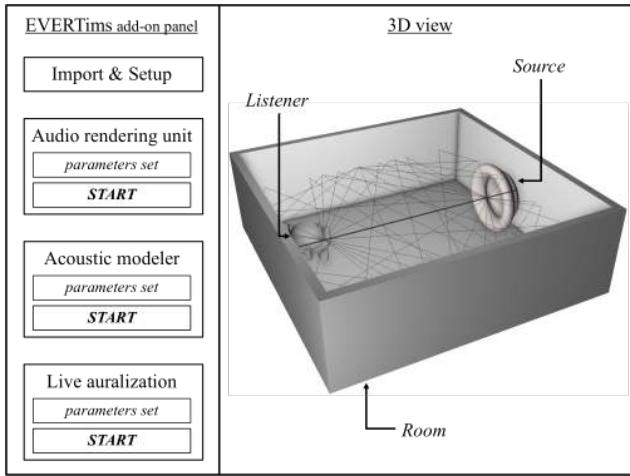


Figure 3: Simplified Blender interface during an EVERTims auralization session. The EVERTims add-on is displayed on the left panel. The “reflection path” debug mode has been activated to monitor the output of the acoustic modeler (reflection paths) for the current geometry.

phases. Both modes rely on the same `evertims` Python module, called either from add-on GUI callbacks or logic bricks attached to the EVERTims logic object. Fig. 3 illustrates the Blender interface during an EVERTims auralization session in edit mode.

During an auralization session, callbacks of the `evertims` Python module stream GA related information along with listener and source positions to the acoustic modeler. In edit mode, any room geometry modifications and changes of the wall surface materials are forwarded to the acoustic modeler for real-time updates of the auralization unit. Spatial and temporal thresholds can be defined to throttle the update mechanism, e.g., the minimum amount of movement required for either source or listener to send an update status in order to control communication traffic in the case of head-tracker jitter. For monitoring and debugging, simulation results can be visualized as rays in the 3D scene. This is depicted in Fig. 3. In addition, a low-priority thread can be started to output local or acoustic modeler subprocess logs to the Blender console.

4. ROOM ACOUSTIC MODELER

This section gives a brief description of the room acoustic modeler unit (see [2] for details). The modeler unit is a console application, implemented in C++. Upon reception of the room geometry and listener & source positions, the modeler unit constructs a beam tree for the current scene geometry. The maximum height of this tree is defined by the highest simulated reflection order passed as a parameter to the modeler. From this tree, a list of image sources is generated and sent to the auralization unit. For visualization, a list of acoustic paths is sent back to the Blender add-on (see Fig. 3). This iterative order refinement of the solution ensures a reactive auralization for dynamic scenes and high accuracy for scenes with fixed source(s) position(s) and room geometry.

Each specular reflection path is characterized by its direction of arrival relative to the listener, propagation delay, absorption due to frequency-dependent material properties, and air absorption (currently not used by the auralization unit, see Sec. 6). Results of

the room acoustic model consisting of visible reflection paths and their accumulated material attenuation are sent to the audio renderer. The reflection path message also contains the “first reflection point” for implementing the source directivity (see Sec. 5). The generic form of an image source message is:

$$\text{pathID order } r1_x \ r1_y \ r1_z \ rN_x \ rN_y \ rN_z \ dist \ abs0 \dots abs9 \quad (1)$$

where $[r1_x, r1_y, r1_z]$ and $[rN_x, rN_y, rN_z]$ are the position vectors of the first and last reflections respectively, $dist$ is the length of the whole reflection path, and abs_M is the absorption coefficient for the M^{th} octave band.

As the reflection paths can only be computed up to a limited order in real-time, a statistical model, currently based on Sabine’s formula [28], approximates the late reverberation time of the current room. The reverberation time is estimated per octave band and sent to the auralization unit’s feedback delay network processor.

The room acoustic modeler runs on multiple threads: one for input processing, one for updating visibility changes, and one for calculating new solutions. The modeler supports multi-source & multi-listener scenarios. It can achieve interactive update rates for a moving listener while calculating up to 7th-order reflections (for frequency-dependent surface absorption characteristics), with a model containing less 1000 polygons. A complete characterization of the EVERTims modeler unit, including performance assessment and an evaluation of its simulation fidelity as compared to other auralization engines, is presented in [2].

5. SPATIAL AUDIO RENDERING FOR AURALIZATION

The general architecture of the auralization unit is detailed in Fig. 4. The unit is implemented in C++ using the JUCE framework⁴ and packaged as a standalone application. At present, the auralization unit is designed for binaural playback over headphones and processes either an audio file or a microphone input signal.

To each image source sent by the acoustic modeler is associated a delayed tap of the current audio input (e.g., audio file buffer). To this tap is applied an attenuation proportional to the length of the whole reflection path of the image source. A frequency specific attenuation is then applied, based on the abs_M coefficients in Msg. (1). The number of bands of the filter-bank used for signal frequency decomposition can be defined from 3–10. The 3-band option is designed for simulations with a large number of image sources and/or for architectures with reduced CPU resources. To further reduce CPU consumption, the filter-bank implementation is based on successively applied low-pass filters (e.g., see implementation in [29]) rather than a series of parallel bandpass filters.

The $[r1_x, r1_y, r1_z]$ position vector of Msg. (1) along with the source orientation are used to compute each image source’s specific Direction of Departure (DoD). Based on this DoD and a directivity diagram loaded from a *GeneralTF* SOFA file (Spatially Oriented Format for Acoustics⁵, cf. [30, 31]), an octave-band specific directivity attenuation is applied to the image source audio buffer. The current implementation proposes a basic set of predefined directivity diagrams (omnidirectional, cardioid, etc.).

The resulting audio buffers of each image source are encoded in 3rd-order Ambisonics and summed to create a single Ambisonics stream sound-field. The sound-field is then decoded to binaural, based on the virtual speaker approach [32]. For line-of-sight

⁴JUCE website: www.juce.com

⁵SOFA website: www.sofaconventions.org

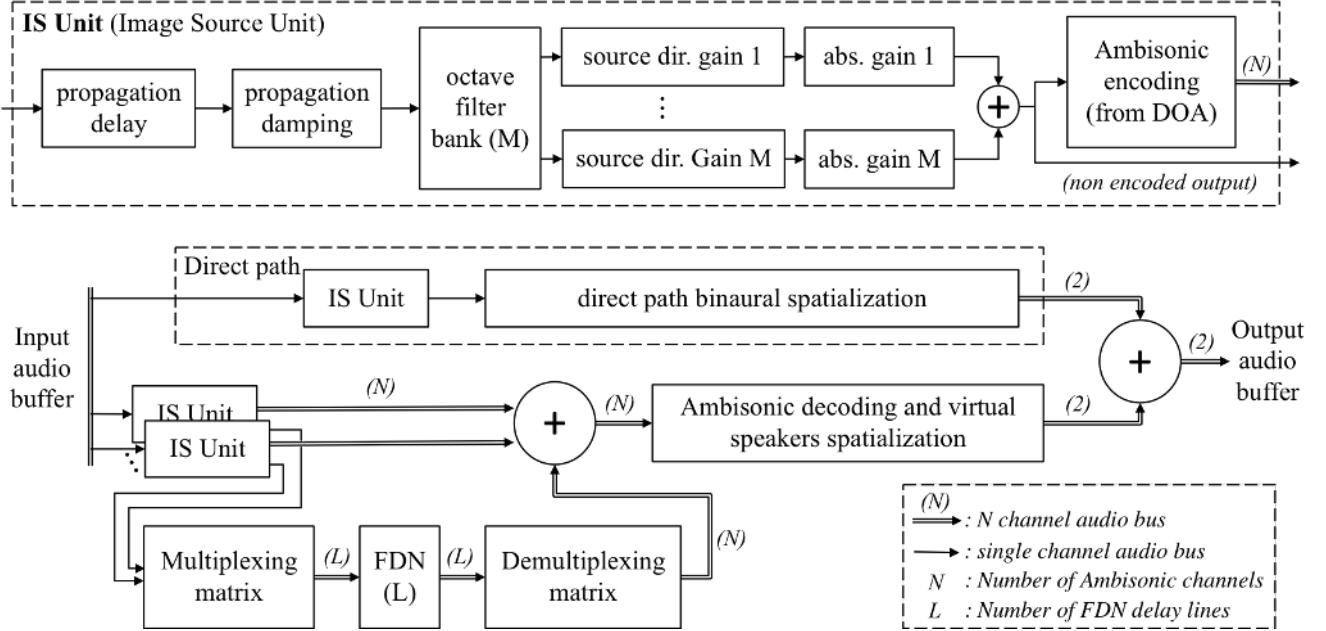


Figure 4: General architecture of the auralization engine.

scenarios, the audio tap of the direct path is handled by a dedicated binaural encoder rather than to the Ambisonic processing unit. The objective of this direct encoding, already applied in the previous version of the auralization unit [2], is to further increase the accuracy of the perceived sound source location. Both decoding schemes are based on filters dynamically loaded from a *Simple-FreeFieldHRIR* SOFA file. The interpolation of incomplete HRIR measurement grids is not yet supported by the auralization unit, being handled via Matlab routines at the moment.

While the acoustic modeler efficiently computes high reflection orders for static geometries, it uses a “low reflection order mode” for dynamic geometries. The auralization unit then simulates the late reverberation using a Feedback Delay Network (FDN) [33]. The current FDN implementation uses 16 feedback channels with mutually prime delay times, and thus assures for a high modal density in all frequency bands to avoid any “ringing tone” effect. The FDN input, output, and feedback matrices are tailored to limit inter-channel correlation and optimize network density [34]. The delays and gains are defined following the approach taken in [34], to match the frequency-specific room response time estimated by the room acoustic modeler. For spatial audio playback, the FDN outputs are encoded into 1st-order Ambisonics. The 16 decorrelated output signals are mapped to the four 1st-order Ambisonics channels (e.g., the outputs 0, 4, 8, and 12 are summed to Ambisonic channel 0) to simulate diffuse late reverberation. Higher-order Ambisonics encoding for high-resolution binaural playback is planned for future updates.

The simulated RIRs can be exported as an audio file in binaural or Ambisonic formats. In addition, all auralization parameters and results (source, listener, and image sources positions, image sources absorption coefficients, etc.) can be exported as a text file for use in other programs.

6. CONCLUSION

This paper presented the latest developments of the EVERTim framework. The aim of the underlying project is to design an accurate auralization tool to support research and room acoustic design. Besides accuracy, the framework is focused on real-time rendering for the auralization of dynamic environments. Its integration as a Blender add-on allows for real-time assessment of room acoustics during its creation.

Each unit of the EVERTim framework is available as an open source project. These units can be used separately for integration in any other framework. Sources, tutorials, recordings and exchange protocol information are available on the EVERTim website: <http://evertim.ircam.fr>.

Latest developments mainly concerned the re-design of the EVERTim framework interface and the integration of its different components. The novel Blender add-on simplifies the control of the overall auralization simulation, providing end-users with a state-of-the-art mesh editing tool in the process. The implementation of the auralization and spatial audio rendering unit as a JUCE C++ graphical application, rather than a PureData patch, should simplify future maintenance and cross-platform compatibility. The new auralization unit now supports the SOFA format to import either HRIR or source directivity patterns.

Foreseen short-term developments include multi-source and multi-listener integration along with cross-platform support. The acoustic modeler unit already supports multi-source and multi-listener, the feature only lacks its counterpart in the auralization unit. The auralization engine is available as a cross-platform application, thanks to the versatility of the JUCE framework. Only minor modifications to the Blender add-on will be required to support both features. Once the multi source/listener support is implemented, a performance comparison of the framework against exist-

ing auralization solutions will be conducted. This comparison will concern both auralization accuracy and efficiency, following that presented in [2]. The assessment will be conducted on the ESPRO model (available from the EVERTims website), a 3D reproduction of the IRCAM projection space [35].

The Sabine based statistical model used for estimation of the room response decay time will be replaced by an Eyring based model [36], or other more modern methods taking into account some simple geometrical room properties [37]. The estimation will focus on early decay times, rather than on the current RT60 room reverberation time, as these are more relevant from a perceptual point of view. Spatial Impulse Response Rendering [38] or DirAC [39] methods will be considered for the encoding of the diffuse field generated from FDN outputs.

A graphical editor to handle acoustic material creation and modification will be added to the add-on, along with an editor for source directivity.

The impact of room temperature, humidity, etc. on frequency-specific air absorption is not implemented at the moment (propagation damping in Fig. 4 is based on the inverse propagation law only). This feature will soon be added to the auralization unit, applied to both image sources and FDN absorption gains.

The impact of surface scattering on image sources propagation will be integrated into both modeler and auralization units, based on an hybrid approach similar to that suggested in [40]. The Binary Space Partitioning used in the room acoustic modeler unit to handle the room geometry [2] will be replaced by a Spatial Hashing implementation [41], optimized for dynamic geometry updates (see [42] and Fig. 3 of [43]).

7. REFERENCES

- [1] Markus Noisternig, Lauri Savioja, and Brian FG Katz, “Real-time auralization system based on beam-tracing and mixed-order ambisonics,” *J Acous Soc Am*, vol. 123, no. 5, pp. 3935–3935, 2008.
- [2] Markus Noisternig, Brian FG Katz, Samuel Siltanen, and Lauri Savioja, “Framework for real-time auralization in architectural acoustics,” *Acta Acustica United with Acustica*, vol. 94, no. 6, pp. 1000–1015, 2008.
- [3] Mendel Kleiner, Bengt-Inge Dalenbäck, and Peter Svensson, “Auralization—an overview,” *J Audio Eng Soc*, vol. 41, no. 11, pp. 861–875, 1993.
- [4] Peter Svensson and Ulf R Kristiansen, “Computational modelling and simulation of acoustic spaces,” in *Audio Eng Soc Conf 22*, 2002, pp. 11–30.
- [5] John S Bradley, “Review of objective room acoustics measures and future needs,” *Applied Acoustics*, vol. 72, no. 10, pp. 713–720, 2011.
- [6] Brian FG Katz and Eckhard Kahle, “Design of the new opera house of the Suzhou Science & Arts Cultural Center,” in *Western Pacific Acoustics Conf*, 2006, pp. 1–8.
- [7] Brian FG Katz and Eckhard Kahle, “Auditorium of the Morgan library, computer aided design and post-construction results,” in *Intl Conf on Auditorium Acoustics*, 2008, vol. 30, pp. 123–130.
- [8] Barteld NJ Postma, David Poirier-Quinot, Julie Meyer, and Brian FG Katz, “Virtual reality performance auralization in a calibrated model of Notre-Dame Cathedral,” in *Euroregio*, 2016, pp. 6:1–10.
- [9] Brian FG Katz, David Poirier-Quinot, and Jean-Marc Lyzwa, “Interactive production of the “virtual concert in Notre-Dame”,” in *19th Forum Intl du Son Multicanal*, 2016.
- [10] Michael Vorländer, *Auralization, Fundamentals of Acoustics, Modelling, Simulation, Algorithms and Acoustic Virtual Reality*. Springer, 2008.
- [11] Lauri Savioja, “Real-time 3D finite-difference time-domain simulation of low-and mid-frequency room acoustics,” in *Intl Conf on Digital Audio Effects*, 2010, vol. 1, pp. 77–84.
- [12] Dirk Schröder, *Physically based real-time auralization of interactive virtual environments*, vol. 11, Logos Verlag Berlin GmbH, 2011.
- [13] Tomas Akenine-Möller, Eric Haines, and Naty Hoffman, *Real-time rendering*, CRC Press, 2008.
- [14] Lauri Savioja and U Peter Svensson, “Overview of geometrical room acoustic modeling techniques,” *J Acous Soc Am*, vol. 138, no. 2, pp. 708–730, 2015.
- [15] Asbjørn Krokstad, Staffan Strom, and Svein Sørdsdal, “Calculating the acoustical room response by the use of a ray tracing technique,” *J Sound and Vib*, vol. 8, no. 1, pp. 118–125, 1968.
- [16] Jont B Allen and David A Berkley, “Image method for efficiently simulating small-room acoustics,” *J Acous Soc Am*, vol. 65, no. 4, pp. 943–950, 1979.
- [17] Brian Hamilton, *Finite Difference and Finite Volume Methods for Wave-based Modelling of Room Acoustics*, Ph.D. thesis, Univ of Edinburgh, 2016.
- [18] Stefan Bilbao and Brian Hamilton, “Wave-based room acoustics simulation: Explicit/implicit finite volume modeling of viscothermal losses and frequency-dependent boundaries,” *J Audio Eng Soc*, vol. 65, no. 1/2, pp. 78–89, 2017.
- [19] K Heinrich Kuttruff, “Auralization of impulse responses modeled on the basis of ray-tracing results,” *J Audio Eng Soc*, vol. 41, no. 11, pp. 876–880, 1993.
- [20] Bengt-Inge L Dalenbäck, “Room acoustic prediction based on a unified treatment of diffuse and specular reflection,” *J Acous Soc Am*, vol. 100, no. 2, pp. 899–909, 1996.
- [21] Graham M Naylor, “Odeon, another hybrid room acoustical model,” *Applied Acoustics*, vol. 38, no. 2–4, pp. 131–143, 1993.
- [22] Lukas Aspöck, Sönke Pelzer, Frank Wefers, and Michael Vorländer, “A real-time auralization plugin for architectural design and education,” in *Proc of the EAA Joint Symp on Auralization and Ambisonics*, 2014, pp. 156–161.
- [23] Regis Faria and Joao Zuffo, “An auralization engine adapting a 3D image source acoustic model to an Ambisonics coder for immersive virtual reality,” in *Audio Eng Soc Conf: Intl Conf 28*, 2006, pp. 1–4.
- [24] Wolfgang Ahnert and Rainer Feistel, “Ears auralization software,” in *Audio Eng Soc Conf 12*, 1992, pp. 894–904.
- [25] Dirk Schröder and Michael Vorländer, “RAVEN: A real-time framework for the auralization of interactive virtual environments,” in *Forum Acusticum*, 2011, pp. 1541–1546.

- [26] Matthew Wright, Adrian Freed, et al., “Open SoundControl: A new protocol for communicating with sound synthesizers,” in *Intl Computer Music Conf*, 1997, pp. 1–4.
- [27] Jelle van Mourik and Damian Murphy, “Geometric and wave-based acoustic modelling using blender,” in *Audio Eng Soc Conf: Intl Conf 49*, 2013, pp. 2–9.
- [28] Wallace Clement Sabine and M David Egan, “Collected papers on acoustics,” *J Acous Soc Am*, vol. 95, no. 6, pp. 3679–3680, 1994.
- [29] Yukio Iwaya, Kanji Watanabe, Piotr Majdak, Markus Noisternig, Yoiti Suzuki, Shuichi Sakamoto, and Shouichi Takane, “Spatially oriented format for acoustics (SOFA) for exchange data of head-related transfer functions,” in *Proc Inst of Elec, Info, and Comm Eng of Japan*, 2014, pp. 19–23.
- [30] Piotr Majdak, Yukio Iwaya, Thibaut Carpentier, Rozenn Nicol, Matthieu Parmentier, Agnieszka Roginska, Yōiti Suzuki, Kanji Watanabe, Hagen Wierstorf, Harald Ziegelwanger, and Markus Noisternig, “Spatially Oriented Format for Acoustics: A Data Exchange Format Representing Head-Related Transfer Functions,” in *Audio Eng Soc Conv 134*, 2013, pp. 1–11.
- [31] Piotr Majdak and Markus Noisternig, “AES69-2015: AES standard for file exchange - Spatial acoustic data file format,” *Audio Eng Soc*, 2015.
- [32] Markus Noisternig, Thomas Musil, Alois Sontacchi, and Robert Holdrich, “3D binaural sound reproduction using a virtual Ambisonic approach,” in *Intl Symp on Virtual Environments, Human-Computer Interfaces and Measurement Systems*. IEEE, 2003, pp. 174–178.
- [33] Manfred R Schroeder, “Natural sounding artificial reverberation,” *J Audio Eng Soc*, vol. 10, no. 3, pp. 219–223, 1962.
- [34] Jean-Marc Jot and Antoine Chaigne, “Digital delay networks for designing artificial reverberators,” in *Audio Eng Soc Conv 90*, 1991, vol. 39, pp. 383–399.
- [35] Markus Noisternig, Thibaut Carpentier, and Olivier Warusfel, “ESPRO 2.0—Implementation of a surrounding 350-loudspeaker array for 3D sound field reproduction,” in *Audio Eng Soc Conf 25*, 2012, pp. 1–6.
- [36] Carl F Eyring, “Reverberation time in “dead” rooms,” *J Acous Soc Am*, vol. 1, no. 2A, pp. 217–241, 1930.
- [37] J Kang and RO Neubauer, “Predicting reverberation time: Comparison between analytic formulae and computer simulation,” in *Proc of the 17th Intl Conf on Acoustics*, 2001, vol. 7, pp. 1–2.
- [38] Juha Merimaa and Ville Pulkki, “Spatial impulse response rendering i: Analysis and synthesis,” *J of the Audio Eng Soc*, vol. 53, no. 12, pp. 1115–1127, 2005.
- [39] Ville Pulkki, “Spatial sound reproduction with directional audio coding,” *J of the Audio Eng Soc*, vol. 55, no. 6, pp. 503–516, 2007.
- [40] Jens Holger Rindel and Claus Lynge Christensen, “Room acoustic simulation and auralization—how close can we get to the real room,” in *Proc. 8th Western Pacific Acoustics Conf*, 2003, pp. 1–8.
- [41] Matthias Teschner, Bruno Heidelberger, Matthias Müller, Danat Pomerantes, and Markus H Gross, “Optimized spatial hashing for collision detection of deformable objects,” in *Vision Modeling and Visualization*, 2003, vol. 3, pp. 47–54.
- [42] Dirk Schröder, Alexander Ryba, and Michael Vorländer, “Spatial data structures for dynamic acoustic virtual reality,” in *Proc of the 20th Intl Conf on Acoustics*, 2010, pp. 1–6.
- [43] Sönke Pelzer, Lukas Aspöck, Dirk Schröder, and Michael Vorländer, “Interactive real-time simulation and auralization for modifiable rooms,” *Building Acoustics*, vol. 21, no. 1, pp. 65–73, 2014.

A COMPARISON OF PLAYER PERFORMANCE IN A GAMIFIED LOCALISATION TASK BETWEEN SPATIAL LOUDSPEAKER SYSTEMS

Joe Rees-Jones

Audio Lab,
Department of Electronic Engineering,
University of York
York, UK
jrq504@york.ac.uk

Damian Murphy

Audio Lab,
Department of Electronic Engineering,
University of York
York, UK
damian.murphy@york.ac.uk

ABSTRACT

This paper presents an experiment comparing player performance in a gamified localisation task between three loudspeaker configurations: stereo, 7.1 surround-sound and an equidistantly spaced octagonal array. The test was designed as a step towards determining whether spatialised game audio can improve player performance in a video game, thus influencing their overall experience. The game required players to find as many sound sources as possible, by using only sonic cues, in a 3D virtual game environment. Results suggest that the task was significantly easier when listening over a 7.1 surround-sound system, based on feedback from 24 participants. 7.1 was also the most preferred of the three listening conditions. The result was not entirely expected in that the octagonal array did not outperform 7.1. It is thought that, for the given stimuli, this may be a repercussion due to the octagonal array sacrificing an optimal front stereo pair, for more consistent imaging all around the listening space.

1. INTRODUCTION

As an entertainment medium, interactive video games are well suited to the benefits of spatial audio. Spatialised sound cues can be used to fully envelop the player in audio, creating immersive and dynamic soundscapes that contribute to more engaging gameplay experiences. In addition to this, an international survey carried out by Goodwin [1] in 2009 suggests that video game players consider spatialised sound to be an important factor of a game experience. At the time of writing, the majority of video game content is able to output multichannel audio conforming to home theater loudspeaker configurations, such as 5.1 and 7.1 surround-sound. More recently, Dolby Atmos has been employed in a handful of titles such as *Star Wars Battlefront* (2015) [2] and Blizzard's *Overwatch* (2016) [3].

A potential benefit arises when considering the influence a multichannel listening system may have on a player's performance in the game world. If a player is able to better determine the location of an in-game event, such as the position of a narrative-progressing item or a potential threat, will this inform their gameplay decisions? From this concept, a novel idea for a comparative listening test was derived, in an attempt to assess the influence a spatial rendering system may have on a player's ability to localise a sound source in an interactive game environment. The test presented in this paper was based on the gamification of a simple localisation task, designed according to three core principles:

1. The player's objective is to locate as many sound sources as possible in the given time limit.

2. The player does not receive any visual feedback regarding the position of a sound source.
3. The player receives a final score determined by how many sound sources are found.

Three loudspeaker configurations commonly used for spatial audio rendering were compared in this study: stereo, 7.1 surround-sound and an octagonal array with a center channel placed at 0° relative to a front-facing listening position. Stereo and 7.1 were chosen as they represent popular, commercially available, rendering solutions used by video game players. The octagonal array was chosen for its relatively stable sound source imaging all around the listener, although it is not a standard configuration currently used for game audio. Player performance was quantified based on the number of correct localisations in each condition, represented by their final score.

The paper is structured as follows: In Section 2, the phantom image stability of loudspeaker arrays is discussed with an emphasis on surround-sound standards. Game design and the implementation of the three listening conditions is described in Section 3. Results, analysis and a discussion of the results are presented in Section 4. The paper conclusion is given in Section 5.

2. BACKGROUND

A listener's ability to infer the location of a sound source, presented using a physical loudspeaker array, is reliant on stable phantom imaging between two adjacent loudspeakers. By manipulating the relative amplitude of two loudspeakers (g_1 and g_2 in Figure 1), the illusion of a phantom (or virtual) sound source emanating at some point between them can be achieved [4].

The ratio of gain values (g_1 and g_2) between the two loudspeakers shown in Figure 1 is given according to Bauer's sine panning law [6]:

$$\frac{\sin\theta}{\sin\theta_0} = \frac{g_1 - g_2}{g_1 + g_2} \quad (1)$$

where θ is the perceived angle of the virtual source and θ_0 is the angle of the loudspeakers relative to a listener facing forward at 0°. If the listener's head is to be more formally considered then it is suggested that replacing the \sin term with \tan in (1) will provide more consistent imaging [5]. The actual gain values with a constant loudness can then be derived using:

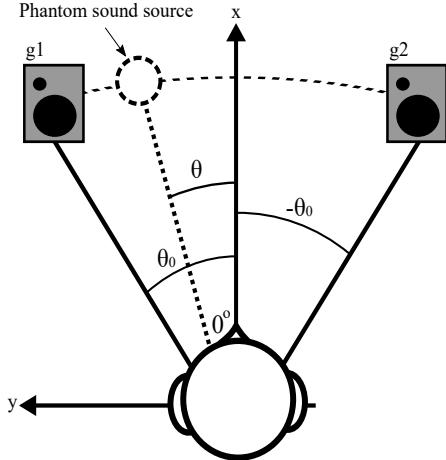


Figure 1: The relationship between the desired angle of a phantom sound source (θ) and two loudspeaker angles ($\pm\theta_0$) [5].

$$\sqrt{\sum_{n=1}^N g_n^2} = 1 \quad (2)$$

where N is the number of loudspeakers. The wider the angle between the two loudspeakers, the more unstable the phantom imaging becomes, in that sounds are perceived to ‘pull’ towards the closest speaker relative to the intended position of the sound source. It is widely accepted that the angle between two loudspeakers should not exceed 60° if stable imaging is to be preserved, where $\pm 30^\circ$ relative to a forward facing listener is optimal [7]. Imaging can be further improved by placing a centre loudspeaker, in-between the pair, at 0° .

For spatialised game audio, it is often the case that the mix will be rendered according to cinema listening standards, such as 5.1 and 7.1 surround-sound. Figure 2 shows the positioning of loudspeakers for 5.1 surround sound listening as suggested in ITU-R BS: 775 [8]. When considering the left and right surround channels (LS and RS in Figure 2), the angles between them exceed the recommended 60° for stable imaging. Such standards are appropriate for film/television viewing in that left, right and center loudspeaker channels are reserved for dialogue, music and diagetic effects whereas the surround channels are more generally used to emphasise the ambiance of a scene. Although mixing strategies vary between different video games, it is not uncommon for game audio to be rendered to every channel of a surround-sound system [9]. One aim is to give the player the impression of a reactive soundscape that dynamically responds to their physical input (i.e. through using a control pad). Consistent and stable phantom imaging around the listening space would therefore be desirable.

A review of the literature suggests that phantom imaging becomes unstable due to the wide angle between loudspeakers as is standard in surround-sound systems, especially when considering lateral sound sources. Cabot [10] assessed the localisation of both rectangular and diamond quadrophonic (4 loudspeaker) systems, finding phantom imaging to be most stable in the front quadrants but very unstable to the sides and rear of the listener. This result is further emphasised by Martin et. al [11] who considered image stability in a 5.1 surround sound system. Their results suggest

phantom imaging is both reliable and predictable using the front three loudspeakers but is highly unstable when a sound at a position greater than 90° relative to a front facing listener is desired. An improved panning law for lateral virtual sound sources in 5.1 surround-sound was derived by Kim et. al and gave promising results, although the authors admit their relatively small subject base (5 participants) is insufficient to draw general conclusions [12]. Theile and Plenge [13] suggest that for more stable lateral imaging, sound sources intended to be perceived at $\pm 90^\circ$ relative to the listening position should be represented by a real sound source i.e. a loudspeaker. They propose an equally spaced arrangement of six loudspeakers to get a suitable ‘all-around’ effect. This configuration was extended by Martin et. al [14] to an equally spaced octagonal array with a front center speaker placed at 0° relative to the listener. The array was found to give relatively stable imaging around the listening space for amplitude-based panning algorithms.

The conclusions drawn from these studies provide evidence that a listener’s ability to successfully localise a sound will be influenced by the phantom image stability of the loudspeaker array used. However, none of these studies asked participants to directly interact with audio stimuli by playing a game. Therefore it is of interest to investigate whether similar comparisons can be made between different loudspeaker arrays, with varying degrees of phantom sound stability, in the context of an interactive game task. In order to retain consistency with these studies, amplitude panning of interactive game audio was compared over stereo, surround-sound and octagonal loudspeaker arrays. Stereo is standard in all modern video game content and, according to Goodwin [1], has the highest user base amongst gamers. The configuration gives strong frontal phantom imaging due to the placement of the left and right loudspeaker at $\pm 30^\circ$ relative to the central listening position. However, imaging to the sides and rear is not possible due to the lack of loudspeakers at these positions. It would therefore be expected that a listener would find it difficult to locate a sound anywhere but within the $\pm 30^\circ$ of the stereo pair.

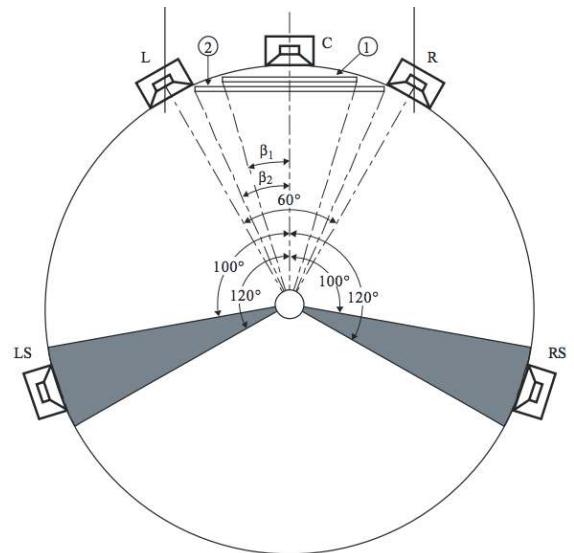


Figure 2: The loudspeaker configuration recommended by ITU-R BS:775 for 5.1 surround-sound listening [8]. The placement of LS and RS is not optimal for stable imaging.

7.1 surround-sound represents the current state-of-the-art in spatial game audio playback, hence its inclusion in the study. The format is an extension of 5.1 surround-sound through the addition of two loudspeakers either behind or to the sides of the listener [15], giving an improvement in spatialisation capability compared to stereo. The $\pm 30^\circ$ stereo arrangement is retained, with an additional center channel placed in-between. For this study the subwoofer ('.1' channel) was not included, as it is intended for further defining low frequency effects which were not used.

As stated previously, an equidistant array of 8 loudspeakers arranged as an octagon gives stable phantom imaging all around the listener. This is an improvement on both 5.1 and 7.1 surround-sound where the inconsistent placement of loudspeakers leads to instability at the sides and rear. However, unlike stereo and 7.1 surround, octagonal arrays are not used to render audio in consumer gaming. Also, the loudspeakers in-front of the listener need to be spaced at a wider angle than those in stereo and 7.1 configurations if equidistant placement is to be achieved, with two loudspeakers placed at $\pm 90^\circ$ for reliable lateral imaging. Therefore the trade-off in ease of localisation between more consistent imaging all around the listener, and the potential for higher resolution frontal imaging in 7.1, is of interest.

3. METHODS

This section outlines the localisation task participants were asked to complete and how it was implemented using a game-like virtual environment. The methods used to render game audio to the three listening conditions - stereo, 7.1 surround-sound and an octagonal array - are then covered. It was decided early in the design process that a custom-made game environment would be used. Previous experiments by the authors have used program material taken from commercially available video game content for current-generation gaming consoles. However, it is not possible to access the source code of such content making it difficult to determine the exact audio rendering methods used, beyond the way in which loudspeakers should be placed. The repeatability between participants is also questionable, along with potential learning effects that may occur due to multiple play-throughs of the same piece of game content. Creating a custom video game gave more control over the underlying mechanics/systems and the effectiveness of an octagonal loudspeaker array could be more easily explored.

For clarity, some game specific terms used in reference to the game design are defined here:

Game engine: A basic framework of code and scripts for creating video games, usually bundled as a complete software package. Handles the game's visual rendering, physics systems and underlying rules/mechanics.

Game object: Conceptually, objects refer to the game's building blocks. They act as containers for all the systems and code required to construct anything needed to make the game operate as intended, such as walls, characters, weapons or on-screen text [16].

Game world: The virtual environment/space in which the game is played. In the present study this refers to a virtual 3-dimensional space.

Player avatar: The player's virtual representation within the game world. The avatar's actions are directly controlled by the player, allowing the player to interact with and navigate through the game world.

Sound source: A game object placed at some position in the game world, from which sound is emitted.

3.1. Game Design

The virtual environment and underlying systems for the localisation task were designed and implemented using the Unity game engine [17]. Sound spatialisation and rendering was done separately in Max/MSP [18]. A single sound source was used in the game, the position of which changed as soon as it was successfully located by the player. The sound source was represented by a spherical Unity game object with a radius of 0.5 metres and its visual renderer turned off, ensuring that the source would be invisible to participants. The position of the sound source was always determined randomly within the boundaries of the game world, represented by a 20x20 metre square room. Random positioning was implemented so that players would not learn sound source positions after playing the game multiple times. The virtual room comprised of four grey coloured walls, a floor and a ceiling to serve a visual reference regarding the player's position within the game world. According to Zielinski et al. [19], visuals can distract significantly from an audio-based task, therefore visuals were deliberately simplified.

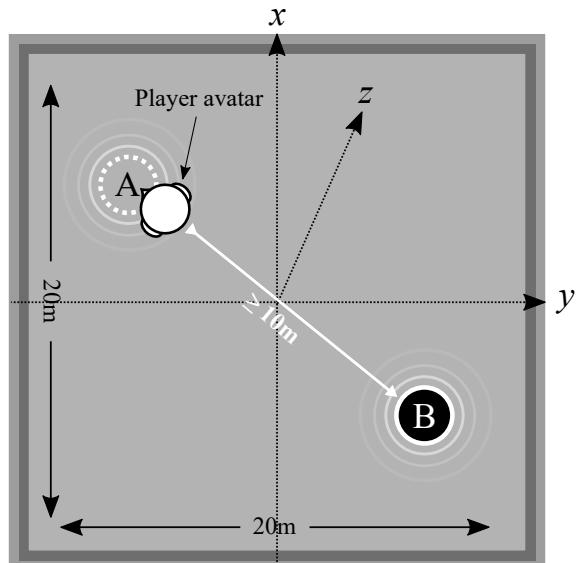


Figure 3: A conceptual illustration of a player correctly locating the sound source in its current position (A) by entering its radius and pressing 'x' on the gamepad. The sound source then moves to a new random position (B) at least 10m away from the player's current position.

Players were able to navigate the game world through the eyes of a virtual avatar, using a control system similar to those found in the majority of first-person point of view games. The position and rotation of the avatar, within the boundaries of the game world, could be controlled by the player using the left and right joysticks of a standard Playstation 4 gamepad. This allowed for full 360° movement in all directions on a horizontal plane. The gamepad's 'x' button was used to trigger a simple if statement within the game's code to determine whether the player had successfully found the sound source. If, upon pressing the 'x' button,

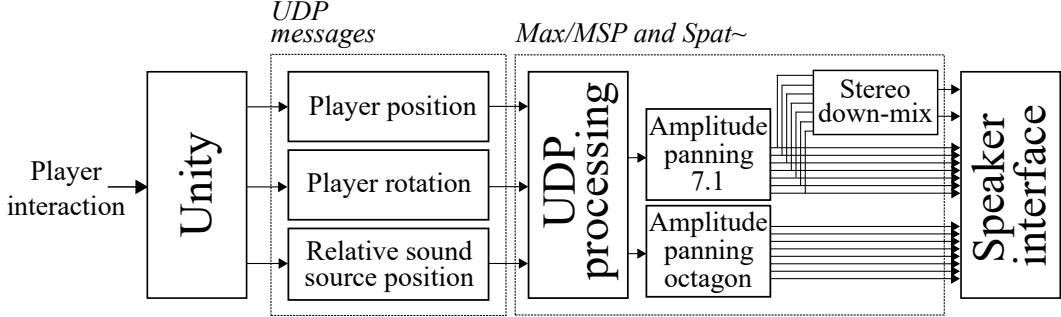


Figure 4: Outline of data flow from Unity to Max/MSP to the loudspeaker interface. Coordinates from Unity were sent to the Max/MSP patch via UDP. UDP messages were then processed to pan a sound source at different positions for the three listening conditions.

the player avatar was within the radius of the sphere representing the sound source's current location, the sphere would move to a random new location at least 10 metres away from the player, within the room's boundaries. Upon triggering this event, an on-screen value depicting the player's score increased by one. A top-down interpretation is illustrated in Figure 3, where position A represents the current position of the sound-source and position B is the new position. If the 'x' button was pressed and the player was not within the radius of the sound source then the current position was maintained with no increase in score. A count-down timer set to 2 minutes 30 seconds was also implemented. The timer was not displayed to players and once it reached 0, "Game Over" was displayed to the player, along with their final, score to signify the end of the game session. The game was played three times by each participant.

3.2. Game Audio Rendering

Game audio was rendered separately to the main game using the *Spatialisateur* (Spat~) object library for Max/MSP provided by IRCAM [20]. Communication between Unity and Max/MSP was achieved using the User Datagram Protocol (UDP). The player avatar's x, y, and z coordinates in the game world were packed and transmitted over UDP on every frame update of the game. This ensured the Max/MSP patch would be synced to the game systems and visuals. The x, y and z coordinates of the sound source relative to the player were sent to Max/MSP in the same way. A diagram of the data flow from Unity to Max/MSP is given in Figure 4. Players were asked to locate a sine tone at a frequency of 440Hz repeating every half a second with an attack time of 5 milliseconds to give a hard onset. A short delay was also applied to the tone, giving a sonar-like effect. An ascending sequence of tones were played to the player upon every correct localisation to give some auditory feedback as to their success, in-line with the increase in score. If the player was incorrect, a descending sequence was played. It was decided other effects commonly found in video games, like music, ambiance and footsteps, would not be included for this test, so as to not confuse the listener.

3.2.1. Sound Spatialisation

The amplitude panning law given in (1) can be extended for more than two loudspeakers using pairwise amplitude panning algorithms [7]. In this work, pairwise panning was implemented for the 7.1 and octagonal loudspeaker configurations using a 'spat.pan~'

Max/MSP object. The 'spat.pan~' object takes a sound source (in this case the 440Hz repeating sine tone) as its input, and pans it according to x, y and z coordinates around the pre-defined loudspeaker layout. The x, y and z coordinates used for panning corresponded to the relative position of the sound source to the player, as transmitted from Unity via UDP. The number of loudspeakers and their placement around the listening area were defined for the panners as follows:

7.1 surround-sound: $0^\circ, 30^\circ, 90^\circ, 135^\circ, -135^\circ, -90^\circ, -30^\circ$

Octagon: $0^\circ, 45^\circ, 90^\circ, 135^\circ, 180^\circ, -135^\circ, -90^\circ, -45^\circ$

Angles used in the 7.1 condition were defined such that they conformed to the ITU-R BS: 775 for surround-sound listening [8]. The octagonal array was arranged in the same configuration used by Martin et al. [14], inclusive of a front centre loudspeaker at 0° relative to a forward-facing listener. Adjacent loudspeakers were defined equidistantly with an angle of 45° between them. The angles used for both conditions are reflected in Figure 5 by the 7.1 surround-sound and Octagon labelled loudspeakers.

3.2.2. Distance Attenuation

Since the player was able to move around the game world freely, it was necessary to include distance attenuation in the audio rendering. This made the sound appear louder as the player moved towards its source and quieter as they moved away. This was achieved by taking the inverse square of the relative distance (in metres) between the sound source and the player. This can be expressed in decibels (dB) using:

$$10 \log_{10} \left(\frac{1}{d^2} \right) \quad (3)$$

where d is the distance between the sound source and the listener. The same distance attenuation was used across the three conditions in order to keep changes in amplitude consistent. The amplitude of the sound remained constant as the player stayed within the radius of the sound source. This was implemented after informal testing, as it was found that otherwise, the sound would only ever reach maximum amplitude if the player was stood directly in the centre of the sound source position.

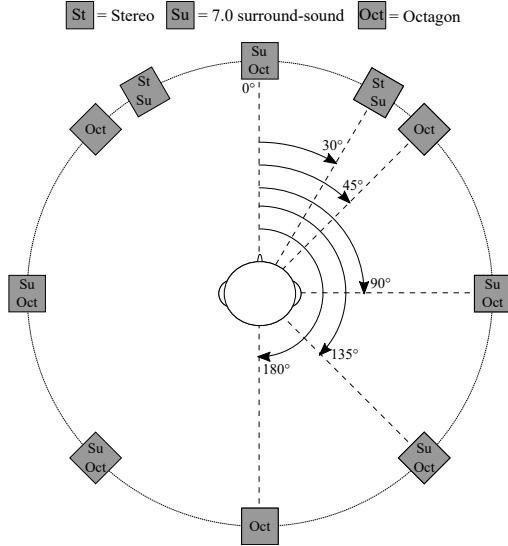


Figure 5: Loudspeaker angles used for all three listening conditions. Angles are symmetrical to the left and right of a front-facing listener.

3.2.3. Stereo Down-mix

It is not uncommon for a stereo down-mix to be generated from a game's 7.1 audio output. Audio content intended for surround-sound listening can be presented over any regular two-channel stereo system by down-mixing non-stereo channels. Additional channels are combined with the front left and right channels to ensure sound cues mixed for the centre and surround channels are not lost. As illustrated in Figure 4, the down-mix used for this experiment was done using the same 7 audio channels (not including the low-frequency effects) used in the 7.1 surround-sound listening condition, according to ITU-R BS.775-3 [14]. The left/right surround (Ls/Rs) channels and left/right back surround (Lbs/Rbs) channels were attenuated by 3dB and sent to the respective left and right front channels. The centre channel was also attenuated by 3dB and sent to the front left and right channels. This can be expressed as:

$$\begin{aligned} L_D &= L + 0.707C + 0.707Ls + 0.707Lbs \\ R_D &= R + 0.707C + 0.707Rs + 0.707Rbs \end{aligned} \quad (4)$$

where L_D and R_D are the down-mixed left and right channels, respectively.

4. EXPERIMENTAL PROCEDURE

A total of 24 subjects participated in the experiment, 16 of which were male, 6 female, and 2 non-binary. All subjects were aged between 18 and 35. Before participating, all potential subjects were asked if they were familiar with using a gamepad to control a game. If not, they were asked not to participate in order to reduce the amount of time needed to learn the game's control system. All provided a signature to confirm their consent.

Participants played the game in all three of the listening conditions - stereo, 7.1 surround-sound and octagon - but were not

Table 1: Counterbalanced participant groupings for the three listening conditions.

Allocation	Condition		
	Stereo	7.1	Octagon
Group 1	Stereo	7.1	Octagon
Group 2	Stereo	Octagon	7.1
Group 3	7.1	Stereo	Octagon
Group 4	7.1	Octagon	Stereo
Group 5	Octagon	Stereo	7.1
Group 6	Octagon	7.1	Stereo

made aware of any of the conditions prior to, or during, the test. Repeated-measures test designs such as this are susceptible to learning effects, in that participant results may be influenced through being exposed to the same program material multiple times. To reduce this risk, the order of listening conditions was counterbalanced as suggested in [21]. This design with 24 participants and 3 listening conditions gave 4 participants in each counterbalanced group. Furthermore, a training session was provided, as described below.

Each of the three game sessions lasted 2 minutes 30 seconds. A 'Game Over' message and the player's score (i.e how many times the sounds source was correctly located) were displayed on-screen at the end of each session. The number of correct localisations was output to a separate text file after each game session, giving each subject a final score for each of the three listening conditions. Once a subject had been exposed to all of the listening conditions, they were asked to state which of the three conditions they preferred, and provide any comments regarding the experiment.

Before the formal test began, participants were asked to complete a training session based on a simplified version of the game, allowing them to become familiar with the control scheme. The training version of the game took place in the same virtual room, with the addition of 5 coloured rings placed in its center and each corner. During the training, the sound source would only ever appear at one of these pre-defined locations. Participants were asked to move the in-game avatar to each of these locations and press the gamepad's 'x' button if they believed that to be the origin of the sound source. Once each of the pre-defined sound sources had been found once, the coloured rings were removed, and participants were asked to find the sound sources again, without a visual cue. Training was done in mono to eliminate the possible learning effect due to playing the game in an experimental condition more than once. The distance attenuation was preserved, allowing participants to familiarise themselves with amplitude changes as they moved closer to and further away from the sound source. The training session was not timed and only finished once a subject had found each of the 5 sound sources twice.

4.1. Apparatus

10 Genelec 8040a loudspeakers were arranged as shown in Figure 5, 1.5 metres from the central listening position. Those intended for 7.1 surround-sound conformed to ITU-R BS: 755 [8]. The Unity game and Max/MSP patch were run from the same Windows PC. Participants interacted with the game using a standard Playstation 4 gamepad connected to the PC via USB. Loudspeakers were driven by a MOTU PCI-424 soundcard. Visuals were

Table 2: Wilcoxon signed-rank output for the mean-adjusted player scores. T is the signed-rank and p is the significance value. The z value is used to determine the significance value (p) and the effect size (r). A value of 1 in the h column signifies a rejection of the null hypothesis.

Conditions compared		Median		T	$p(<.05)$	z	r	h
Stereo	7.1 Surround-sound	-1.167	1.167	17.5	<0.001	-3.559	0.514	1
Stereo	Octagon	-1.167	-0.667	87.5	0.202	-1.277	0.184	0
7.1 Surround Sound	Octagon	1.167	-0.667	173.5	0.043	2.023	0.292	1

presented using an Optoma HD200X projector, projecting onto an acoustically transparent screen. Loudspeakers positioned at 0° and $\pm 30^\circ$ were located behind the screen.

5. RESULTS

This section presents the results from statistical analysis of the player scores obtained during the experiment. Player scores (i.e. the number of correct localisations) were compared between pairs of the three listening conditions. Relationships between participants' success at the game and their preference for a listening condition are also given. All statistical analysis was performed using the statistics and machine learning toolbox in MATLAB.

5.1. Player scores

Player scores for the three listening conditions were first checked for normal distribution using a Kolmogorov-Smirnov goodness-of-fit test. Scores were found to be non-normally distributed (non-parametric), therefore Wilcoxon signed-rank tests were used to check for significances between pairs of conditions, as suggested by [21]. Scores were standardised before analysis due to the overall differences in scores between participants. This was done by subtracting a subject's mean score from their three individual condition scores. This ensured the relative distances between a player's own scores would be preserved. The null hypothesis for analysis was: *There is no statistically significant difference in the number of correct localisations between pairs of listening conditions.* The output from the Wilcoxon signed-rank tests are presented in Table 2. A value of 1 in the column labeled h of Table 2 signifies a rejection of the null hypothesis at the $p < 0.05$ significance level.

Analysis suggests there was a statistically significant difference in scores between stereo and 7.1 surround-sound as well as between 7.1 surround-sound and octagon conditions. Upon viewing the boxplot given in Figure 6 it can be seen that participants achieved higher localisation scores in the 7.1 surround-sound condition compared to both stereo and the octagonal array. This result implies that players were better at the game when listening to the audio using 7.1 surround-sound, in that they were able to more successfully localise sound sources. When considering stereo and 7.1 surround-sound, the effect size ($r = 0.514$) also signifies that listening condition had a large effect on player scores. This is higher than the moderate effect size observed between the 7.1 and octagon conditions ($r = 0.292$). This suggests scores achieved in the 7.1 condition were consistently higher in comparison to stereo than when compared to the octagonal array.

The null hypothesis could not be rejected for the comparison between the stereo and octagon conditions, suggesting there was no statistically significant difference in player scores between the two at the $p < 0.05$ significance level. This is reflected by the boxplot in Figure 6, where it can be seen a similar range in values

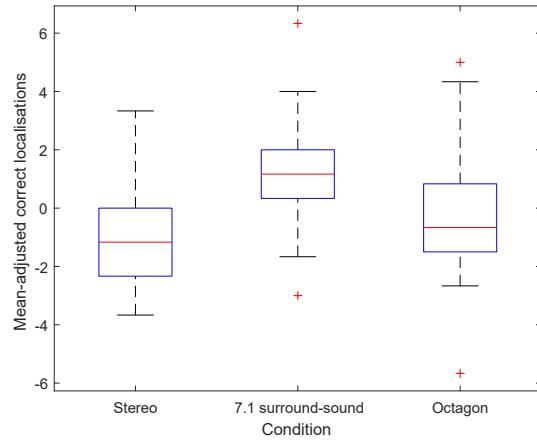


Figure 6: Mean-adjusted distribution of player scores for the three listening conditions: stereo, 7.1 surround-sound and octagonal array. Analysis suggests the highest scores were achieved during the 7.1 surround-sound condition.

is spanned by the stereo and octagon plots. The result implies that participant performance neither improved nor worsened between the two conditions, in that the number of correct localisations was similar.

5.2. Player preference

Once participants had played the game using all three listening conditions, they were asked to state which of the three they preferred, and were also encouraged to provide comments regarding their decision. In general, 7.1 surround-sound was the most preferred of the three conditions, as chosen by 70.8% of participants. Both stereo and the octagonal array were preferred by significantly fewer subjects. The preference scores for each condition are illustrated in Figure 7.

A number of participants commented that their preference was influenced by the condition in which their highest score was achieved. Table 3 shows the percentage of highest player scores attained in each condition, alongside the corresponding percentage of overall preference. 60.4% of the highest scores were obtained in the 7.1 surround-sound condition, which was also most preferred by 70.8% of players. This implies that there was a preference for the condition in which the game was found to be easiest, which in the majority of cases was 7.1 surround-sound. Although the stereo condition contributed to 12.5% of the highest scores, the majority of subjects who achieved those scores stated that, perceptually, sounds were easier to localise in 7.1 surround-sound, hence it was

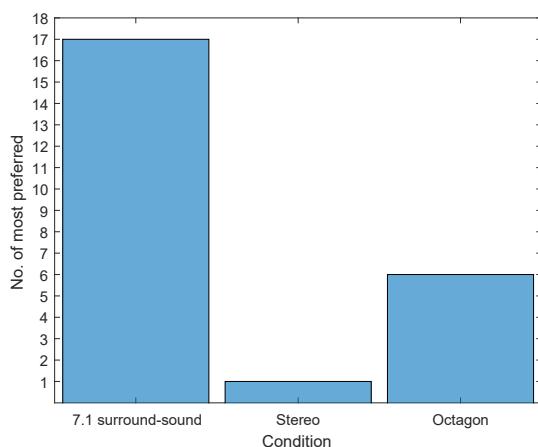


Figure 7: Preference ratings given for each listening condition. 7.1 surround-sound was preferred by the majority of participants.

Table 3: The percentage of highest scores, alongside the percentage of preference ratings, for each condition.

Condition	% highest score	% most preferred
Stereo	12.5%	4.2%
7.1 surround-sound	60.4%	70.8%
Octagon	27.1%	25.0%

more preferred. This may explain the minor discrepancy between the score and preference percentages for the stereo and surround-sound conditions.

5.3. Discussion

In general, players had greater success in the game when listening to audio over a 7.1 surround-sound loudspeaker array. The number of correct localisations was consistently higher for the majority of participants than when compared to stereo and octagonal loudspeaker systems. 7.1 was also the most preferred of the three listening conditions, with participant comments suggesting this was due to it being the condition in which the highest scores were achieved. It was expected that 7.1 would outperform stereo due to the increased number of channels available in the system, and from these results it can be said that players did benefit from using a more spatial listening array. However, the same could not be said in regard to the octagonal array.

As stated in section 2, it would be expected that the localisation of sound sources, especially those positioned laterally and to the rear of the listener, would be easiest when listening over an octagonal array of loudspeakers. However, the more consistent and stable phantom imaging that can be achieved using such a system seems to have had little impact on the results obtained in this experiment. Visuals were presented to the player using a stationary screen, therefore players were only ever required to look forwards. For this reason it may have been that those loudspeakers located directly in front of the listener's forward facing position were of most use in the localisation task. The front left and right loudspeakers of the octagonal array were spaced wider than the $\pm 30^\circ$ used in the

7.1 arrangement. Although both conditions made use of a centrally spaced loudspeaker at 0° , the increased resolution generated by the narrower angles between loudspeakers had in 7.1 surround-sound may have been more helpful than consistent imaging from all directions. Also, the directionality that can be achieved with a 7.1 array, although not perfect, would at least allow a listener to gain a vague sense of a sound source's general direction. It may therefore be the case that once a player had positioned their in-game avatar such that the sound was perceived to emanate at some point straight ahead, triangulating its specific location was then easiest using a more optimally spaced stereo pair. This observation is reflected in the comments given by participants, where it was stated on multiple occasions that it was easiest to triangulate/focus on the sound source in the 7.1 surround-sound condition.

It is important to note, however, that in comparison to commercial games, the game used in this experiment was a relatively simple example. Generally, modern games include more in-depth sound design and visual effects that work together in forming the entire game experience. It would therefore be of interest to determine whether the results obtained from this experiment could be replicated using a more complex game task, inclusive of more 'true-to-life' game systems. This would provide clarity as to whether the results from this study were dependent on the stimulus used, and is the proposed next step for this work. In comparison to previous work by authors, the use of a custom game environment was found to allow for far more control over experimental variables, and is therefore recommended for such studies.

6. CONCLUSION

This paper presented an experiment designed to determine whether enhanced spatial audio feedback has an influence on how well a player performs at a video game. Player performance was quantified by how many correct localisations of a randomly positioned sound source were achieved within a time limit of 2.30 minutes. This was compared between three listening conditions: stereo, 7.1 surround-sound and an octagonal array of loudspeakers. Results suggest that by using a more spatial listening system, player performance was improved, in that significantly higher localisation scores were achieved when using a 7.1 system in comparison to stereo. 7.1 was also consistently the most preferred of the three conditions by participants. However, the same result was not observed for the octagonal array. A possible explanation is that the angles between the front three loudspeakers used in the octagonal array ($-45^\circ, 0^\circ, +45^\circ$) were wider than those in the 7.1 surround-sound system ($-30^\circ, 0^\circ, +30^\circ$). Therefore frontal sound source imaging may not have been as well defined in comparison to the 7.1 condition.

7. REFERENCES

- [1] Simon N Goodwin, "How players listen," in *Audio Engineering Society Conference: 35th International Conference: Audio for Games*. Audio Engineering Society, 2009.
- [2] "Star Wars™ Battlefront™ in Dolby Atmos for PC, note=Available at <https://www.dolby.com/us/en/categories/games/star-wars-battlefront-in-dolby-atmos.html>, year=accessed March 28, 2017," .

- [3] “Dolby Atmos Unleashes the Power of Blizzard’s Overwatch®,” Available at <https://www.dolby.com/us/en/categories/games/overwatch.html>, accessed March 28, 2017.
- [4] Francis Rumsey and Tim McCormick, *Sound and Recording: An Introduction*, Focal Press, Oxford, UK, fifth edition, 2006.
- [5] Ville Pulkki and Matti Karjalainen, “Localization of amplitude-panned virtual sources 1: stereophonic panning,” *Journal of the Audio Engineering Society*, vol. 49, no. 9, pp. 739–752, 2001.
- [6] Benjamin B Bauer, “Phasor analysis of some stereophonic phenomena,” *The Journal of the Acoustical Society of America*, vol. 33, no. 11, pp. 1536–1539, 1961.
- [7] Francis Rumsey, *Spatial Audio*, Focal Press, Oxford, UK, 2001.
- [8] ITU-R, “Recommendation BS. 775-3: Multichannel stereophonic sound system with and without accompanying picture,” *International Telecommunications Union*, 2012.
- [9] Mark Kerins, *The Oxford Handbook of New Audiovisual Aesthetics*, chapter Multichannel Gaming and the Aesthetics of Interactive Surround, pp. 585–604, Oxford University Press, Oxford, UK, 2013.
- [10] Richard C Cabot, “Sound localization in 2 and 4 channel systems: A comparison of phantom image prediction equations and experimental data,” in *Audio Engineering Society Convention 58*. Audio Engineering Society, 1977.
- [11] Geoff Martin, Wieslaw Woszczyk, Jason Corey, and René Quesnel, “Sound source localization in a five-channel surround sound reproduction system,” in *Audio Engineering Society Convention 107*. Audio Engineering Society, 1999.
- [12] Sungyoung Kim, Masahiro Ikeda, and Akio Takahashi, “An optimized pair-wise constant power panning algorithm for stable lateral sound imagery in the 5.1 reproduction system,” in *Audio Engineering Society Convention 125*. Audio Engineering Society, 2008.
- [13] Gunther Theile and Georg Plenge, “Localization of lateral phantom sources,” *Journal of the Audio Engineering Society*, vol. 25, no. 4, pp. 196–200, 1977.
- [14] Geoff Martin, Wieslaw Woszczyk, Jason Corey, and René Quesnel, “Controlling phantom image focus in a multichannel reproduction system,” in *Audio Engineering Society Convention 107*. Audio Engineering Society, 1999.
- [15] Tomlinson Holman, *Surround Sound: up and running*, CRC Press, FL, US, 2014.
- [16] “GameObjects,” Available at <https://docs.unity3d.com/Manual/GameObjects.html>, accessed March 28, 2017.
- [17] “Download Unity Personal,” Available at <https://store.unity.com/download?ref=personal>, accessed April 7, 2017.
- [18] “Cycling 74: Tools for sound, graphics, and interactivity,” Available at <https://cycling74.com/products/max/>, accessed April 7, 2017.
- [19] Slawomir K Zielinski, Francis Rumsey, Søren Bech, Bart De Bruyn, and Rafael Kassier, “Computer games and multichannel audio quality—the effect of division of attention between auditory and visual modalities,” in *Audio Engineering Society Conference: 24th International Conference: Multi-channel Audio, The New Reality*. Audio Engineering Society, 2003.
- [20] Jean-Marc Jot and Olivier Warusfel, “Spat: a spatial processor for musicians and sound engineers,” in *CIARM: International Conference on Acoustics and Musical Research*, 1995.
- [21] Andy Field and Graham Hole, *How to Design and Report Experiments*, Sage, London, UK, 2003.

ACCURATE REVERBERATION TIME CONTROL IN FEEDBACK DELAY NETWORKS

Sebastian J. Schlecht and Emanuël A.P. Habets

International Audio Laboratories Erlangen *

Erlangen, Germany

sebastian.schlecht@audiolabs-erlangen.de

ABSTRACT

The reverberation time is one of the most prominent acoustical qualities of a physical room. Therefore, it is crucial that artificial reverberation algorithms match a specified target reverberation time accurately. In feedback delay networks, a popular framework for modeling room acoustics, the reverberation time is determined by combining delay and attenuation filters such that the frequency-dependent attenuation response is proportional to the delay length and by this complying to a global attenuation-per-second. However, only few details are available on the attenuation filter design as the approximation errors of the filter design are often regarded negligible. In this work, we demonstrate that the error of the filter approximation propagates in a non-linear fashion to the resulting reverberation time possibly causing large deviation from the specified target. For the special case of a proportional graphic equalizer, we propose a non-linear least squares solution and demonstrate the improved accuracy with a Monte Carlo simulation.

1. INTRODUCTION

Reverberation time is one of the most prominent acoustical qualities of a physical room dominating the perceptual quality of a space depending on the intended purpose. The reverberation time, denoted by T_{60} , is the most common decay rate measure and is defined as the time needed for the energy decay curve of an impulse response to drop by 60 dB [1]. The frequency-dependent reverberation time $T_{60}(\omega)$ can be similarly derived from the energy decay relief [2]. The accuracy of perceiving the reverberation time has been studied from various application standpoints [3, 4], however, specific just-noticeable-differences may vary depending on the stimulus signal, early to late reverberation ratio and other properties. For a rough orientation, JNDs of 4% have been reported by participants listening to bands of noise [3], 5-12% for impulse responses and 3-9% for speech signals [5].

Consequently, generative algorithms for artificial reverberation strive to recreate the reverberation time of the desired virtual space accurately. Among the algorithms for artificial reverberation, the feedback delay networks (FDNs) enjoy popularity for its versatility and its efficient implementation [6, 7, 8]. The reverberation time of FDNs is commonly determined in two steps: Firstly, a lossless FDN is designed, i.e., the energy entering the FDN cycles unattenuated through the feedback loop such that there is no decay of energy. Secondly, the attenuation filters are introduced in the feedback loop to control the decay rate of the energy and by this the resulting reverberation time of the system. Fig. 1 shows a single feedback comb filter with delay line z^{-m} and attenuation filter $A(z)$.

* The International Audio Laboratories Erlangen are a joint institution of the Friedrich-Alexander-Universität Erlangen-Nürnberg (FAU) and Fraunhofer Institut für Integrierte Schaltungen IIS.

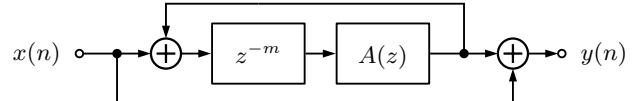
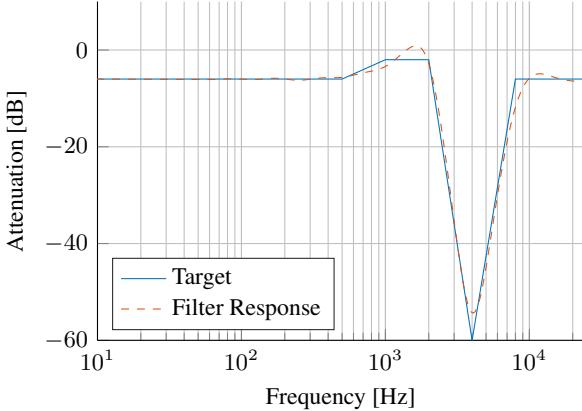


Figure 1: Feedback comb filter with delay length of m samples and attenuation filter $A(z)$ determining the resulting reverberation time.

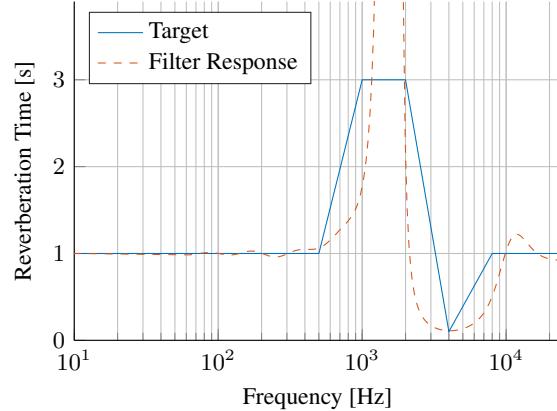
There are commonly two paradigms for designing the attenuation filters: The first being motivated by physical room acoustics, where the reverberation time results from different absorptive boundary materials and the free path in-between [9, 10, 11]. The resulting reverberation time can be predicted by a modified version of Sabine's [12] and Eyring's [13] formulas. Whereas this attenuation filter design is useful for recreating real room acoustics, it is not optimal for achieving precise control of the reverberation time. The second paradigm is motivated from a system theoretic perspective, where attenuation filters influence directly the magnitude of the system poles of the FDN which in turn have a close relation to the resulting reverberation time. The idea is to choose the strength of the attenuation filter proportional to the corresponding delay lengths, i.e., the longer the delay, the stronger is the attenuation [6]. By this, it is possible to define a global *attenuation-per-second* no matter how the signal travels through the delay network.

Various filter types have been proposed to realize the attenuation filter $A(z)$ depending on the control flexibility, computational complexity and required accuracy. In the early days, the most cheaply available filter was a one-pole lowpass filter [6, 14, 15, 16, 17, 18]. Biquadratic filters allow control of the decay time in three independent frequency bands, with adjustable crossover frequencies [19, 20]. More advanced studies try to emulate the frequency response of reflection coefficients in octave bands by applying high-order filter IIR filters [9, 10, 21]. Recently, Jot proposed a proportional graphic equalizer being a simple yet effective method to control an arbitrary number of logarithmic bands [22]. Because of its beneficial design, this graphic equalizer is central to the discussion in this work.

Despite of the large number of proposals on the FDN structure and the attenuation filter, only few details are given on the design of the attenuation filter $A(z)$ assuming that the approximation error made is negligible. However, the example given in Fig. 2 demonstrates that an attenuation filter with a maximum approximation error of a few dB may yield a poor approximation of the resulting reverberation time, in fact it has an infinite T_{60} at 2 kHz. The reason for this inconsistency can be found in the reciprocal relation between the attenuation filter response and the



(a) Target magnitude response of the attenuation filter and an approximated filter.



(b) Corresponding target T_{60} response and the resulting T_{60} of the approximated filter.

Figure 2: Target reverberation time and corresponding target attenuation filter magnitude response on the dB scale for a delay m of 100 ms. Although the approximated filter exhibits relatively small attenuation errors, it exhibits large errors for the reverberation time.

reverberation time. The following paper is dedicated to improving the design of the attenuation filter to achieve a more accurate reverberation time. Section 2 introduces the background on FDNs and the error propagation between the magnitude response and the reverberation time. Further it introduces the proportional graphic equalizer used for the implementation of the attenuation filter. Section 3 introduces different techniques to determine the parameters of the graphic equalizers and demonstrates their impact on a case example. Further, a Monte Carlo simulation with randomized frequency-dependent target reverberation time evaluates the quality of the different parameter estimation methods.

2. REVERBERATION TIME OF AN FDN

2.1. Combination of Comb Filters

An FDN consists of multiple delay lines interconnected by a feedback matrix, which is commonly chosen to be a unilossless matrix¹. The constituting lossless FDN is adaptable to produce a specified reverberation time by extending every delay line with an attenuation filter [6]. Because of the lossless prototype, each attenuation filter can be considered independently and only dependent on the corresponding delay line. Consequently, the remaining paper only considers single delay line FDNs, which are commonly referred to as absorptive feedback comb filters (see Fig. 1). The transfer function of the absorptive feedback comb filter in Fig. 1 is

$$H(z) = \frac{1}{1 - A(z)z^{-m}}, \quad (1)$$

where $A(z)$ is the transfer function of the attenuation filter and m is the delay length in samples. Every time the signal traverses the feedback loop, it is affected by the attenuation filter such that this recursive effect can be described as an *attenuation-per-time interval*. Particular care should be taken for the design of the attenuation filter $A(z)$ to ensure the stability of (1). The global tar-

¹In contrast to lossless feedback matrices used for example in [7], unilossless feedback matrices are lossless for all possible delays. For more information, the reader is referred to [23].

get *attenuation-per-sample* $\delta(\omega)$ on dB scale can be related to the frequency-dependent reverberation time $T_{60}(\omega)$ by

$$\delta(\omega) = -60 \frac{1}{f_s T_{60}(\omega)}, \quad (2)$$

where f_s is the sampling frequency. On a linear scale, the attenuation-per-sample is given by

$$D(\omega) = 10^{\delta(\omega)/20}. \quad (3)$$

The attenuation filter $A(z)$ is commonly idealized in having zero-phase such that all system poles of (1) lie on the line specified by $|A(\omega)|^{1/m}$ which in turn determines the decay rate of the FDN [6]. Although this cannot be satisfied strictly, in many designs the attenuation filter delay is small compared to the delay m and can be neglected. To achieve the target reverberation time, the attenuation filter is designed such that [6]

$$\begin{aligned} |A(\omega)| &\approx D^m(\omega) \\ \text{or} \\ \alpha(\omega) &\approx m\delta(\omega), \end{aligned} \quad (4)$$

where

$$\alpha(\omega) = 20 \log_{10} |A(\omega)|. \quad (5)$$

In the following section, we explain the error propagation of the filter response as seen in the example given in Fig. 2.

2.2. Error Propagation of Attenuation Filter Approximation

The design of the attenuation filter may be performed by approximating the target magnitude response either on the dB or linear scale in (4) minimizing an appropriate error norm [24]. First, we focus on a least-squares design approach performed on the dB scale as the main objective such that (4) is expressed as²

$$\|\alpha - m\delta\|_2^2. \quad (6)$$

²The Euclidean norm $\|\cdot\|_2$ of a continuous frequency response F is given by $\|F\|_2 = \left(\int_0^{2\pi} |F(\omega)|^2 d\omega \right)^{1/2}$ whereas for a $n \times 1$ vector v the Euclidean norm is $\|v\|_2 = \left(\sum_{i=1}^n |v_i|^2 \right)^{1/2}$.

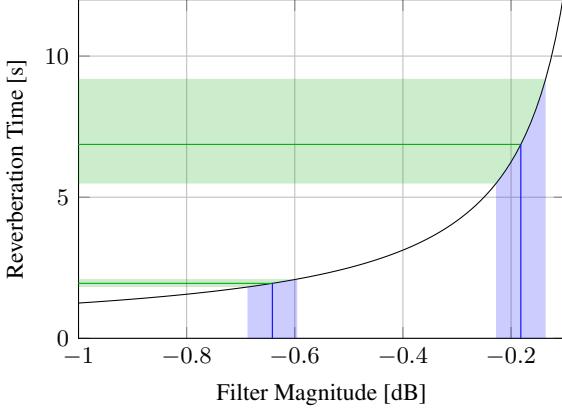


Figure 3: Error propagation from the filter magnitude response to the resulting reverberation time for a delay m of 100 ms. The black line depicts the relation between filter magnitude and reverberation time as given in (2) and (4). The blue shaded intervals indicate error intervals of ± 0.045 dB around two target values (solid blue lines) and the green shaded intervals depict the corresponding error of the reverberation time around the two target values (solid green lines).

We refer to this approach as the *magnitude least-squares (MLS)* approach.

Because of (2), the approximation error however propagates in a non-linear fashion to the resulting reverberation time. Fig. 3 shows different target magnitudes of an attenuation filter on the dB scale with three equidistant error intervals and corresponding reverberation times. Although the approximation error of the attenuation magnitude is homogenous, the resulting reverberation time can deviate arbitrarily depending on the target value. For strong attenuation of -1 dB, an approximation error of 0.2 dB has negligible influence on the resulting reverberation time. Whereas for weak attenuation of -0.18 dB the same error can lead to instability as depicted in Fig. 2.

To overcome this problem, we propose to minimize the error directly on the resulting reverberation time, i.e.,

$$\left\| \frac{1}{\alpha} - \frac{1}{m\delta} \right\|_2^2. \quad (7)$$

We refer to this approach as the *T₆₀ least-squares (TLS)* approach. In the following, we discuss the squared errors (6) and (7) for a recently proposed attenuation filter implementation for FDN based on a proportional graphic equalizer [22].

2.3. Proportional Graphic EQ

In the following, we give details on a multiband graphic equalizer allowing logarithmic band adjustment of the reverberation time [22]. The technical details are similar to the ones given in [25, 26]. The graphic multiband equalizer consist of a number of cascaded 2nd order IIR filters, each of which controls a certain band. The lowest and highest band filters are shelving filters [25]:

$$H_{LS,g}(z) = g^{1/2} \frac{p_0 + p_1 z^{-1} + p_2 z^{-2}}{q_0 + q_1 z^{-1} + q_2 z^{-2}} \quad (8)$$

$$H_{HS,g}(z) = g / H_{LS,g}(z) \quad (9)$$

with

$$p_0 = g^{1/2} \Omega^2 + \sqrt{2}\Omega g^{1/4} + 1 \quad (10)$$

$$p_1 = 2g^{1/2} \Omega^2 - 2 \quad (11)$$

$$p_2 = g^{1/2} \Omega^2 - \sqrt{2}\Omega g^{1/4} + 1 \quad (12)$$

$$q_0 = g^{1/2} + \sqrt{2}\Omega g^{1/4} + \Omega^2 \quad (13)$$

$$q_1 = 2g^{1/2} \Omega^2 - 2 \quad (14)$$

$$q_2 = g^{1/2} - \sqrt{2}\Omega g^{1/4} + \Omega^2, \quad (15)$$

where g is the gain at DC ($\omega = 0$) for $H_{LS,g}$ and at the Nyquist frequency ($\omega = f_s/2$) for $H_{HS,g}$; and $\Omega = \tan(\omega_c/2)$, where the gain is $g^{1/2}$ at the cutoff frequency ω_c in radians. Whereas the remaining filters are peak-notch filters being defined by the transfer function [25]:

$$H_{PN,g}(z) = \frac{p_0 + p_1 z^{-1} + p_2 z^{-2}}{q_0 + q_1 z^{-1} + q_2 z^{-2}} \quad (16)$$

$$p_0 = g^{1/2} + g \tan(B/2) \quad (17)$$

$$p_1 = -2g^{1/2} \cos(\omega_c) \quad (18)$$

$$p_2 = g^{1/2} - g \tan(B/2) \quad (19)$$

$$q_0 = g^{1/2} + \tan(B/2) \quad (20)$$

$$q_1 = -2g^{1/2} \cos(\omega_c) \quad (21)$$

$$q_2 = g^{1/2} + \tan(B/2), \quad (22)$$

where B is the bandwidth which can be alternatively determined by the quality factor Q using $B = \frac{\omega_c}{Q}$. The cutoff and center frequencies ω_c of the respective band filters are spaced logarithmically over the full frequency range. The so-called *command gain* g of the band filter indicates the maximum boost or attenuation of the magnitude response, respectively.

This parametrization allows that the magnitude responses of the shelving and peak/notch filters at different command gain settings (but with fixed center frequency and bandwidth) are self-similar on the dB scale [27]:

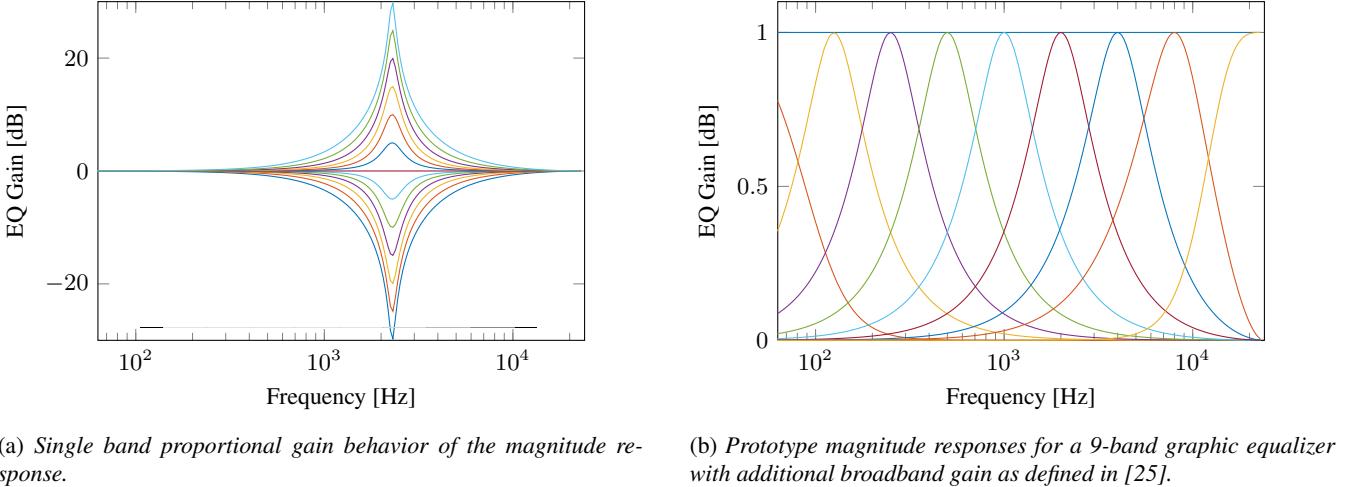
$$g \log_{10} |H_{X,1}| \approx \log_{10} |H_{X,g}|, \quad (23)$$

where $X \in \{\text{LS, HS, PN}\}$ for $H_{X,g}$. Fig. 4a shows the magnitude response of a peak/notch filter for command gains between -30 and 30 dB. It can be observed that the self-similarity property is well-approximated for absolute gains up to 10 dB, but deteriorates for higher absolute gain values. Fig. 4b shows the magnitude response of the individual biquadratic filters of a 9-band graphic equalizer for a prototype gain of 1 dB. There are many refinements for the parametrization of the graphic equalizer, which can be incorporated to further improve the filter design results, e.g. additional opposite filters to reduce the interference between the bands [28], high-precision iterative optimization [29], or improved interpolation of the target response between the center frequencies [30].

The transfer function of the complete graphic equalizer with L -bands is then given by:

$$A_{GE}(z) = g_0 \prod_{l=1}^L H_{l,g_l}(z), \quad (24)$$

where g_0 is an overall broadband gain and H_{l,g_l} is the l^{th} -band with gain g_l , first and last band being low- and high shelving fil-



(a) Single band proportional gain behavior of the magnitude response.

(b) Prototype magnitude responses for a 9-band graphic equalizer with additional broadband gain as defined in [25].

Figure 4: Proportional graphic equalizer as proposed in [22].

ters and the remaining bands being peak filters. The vector of command gains is then

$$\begin{aligned} \mathbf{g} &= [g_0, g_1, \dots, g_L]^\top \\ \gamma &= 20 \log_{10}(\mathbf{g}), \end{aligned} \quad (25)$$

where the \log_{10} is applied element-wise, and $(\cdot)^\top$ denotes the transpose operation.

In the next section, we explain the impact of the different error norms on the filter design method for the graphic equalizer and the resulting reverberation time.

3. MULTIBAND GAIN ESTIMATION

3.1. Linear Least-Squares Solution

Because of the self-similarity in (23), the magnitude responses of the band filters can be used as basis functions to approximate the magnitude response of the overall graphic equalizer. The filters are sampled at a $K \times 1$ vector of control frequencies ω_p spaced logarithmically over the complete frequency range, where $K \geq L$. The resulting interaction matrix is

$$\begin{aligned} \mathbf{B} &= 20 \log_{10} |[\mathbf{g}', H_{1,g'}(\omega_p), \dots, H_{L,g'}(\omega_p)]| \\ &= \begin{matrix} & \text{[Color Bar]} \\ \text{[Color Bar]} & \text{[Color Bar]} \end{matrix} \end{aligned} \quad (26)$$

of size $K \times (L+1)$, where g' is a prototype gain and $\mathbf{g}' = g' \mathbf{1}_{K \times 1}$. The white color indicates 0 dB gain and dark color indicates gains up to 1 dB in Fig. 4b. The interaction matrix \mathbf{B} represents how much the response of each band filter leaks to other bands. Similarly, the target vector is

$$\boldsymbol{\tau} = m\delta(\omega_p), \quad (27)$$

where δ is applied element-wise. The command gains γ for the individual filters yield the resulting magnitude response on the dB

scale of the entire attenuation filter:

$$\alpha_{GE}(\omega_p) = \mathbf{B}\gamma. \quad (28)$$

The magnitude least squares problem in (6) can then be restated as a linear least-squares problem:

$$\begin{aligned} \gamma_{MLS} &= \arg \min_{\gamma} \|\mathbf{B}\gamma - \boldsymbol{\tau}\|_2^2 \\ &= (\mathbf{B}^\top \mathbf{B})^{-1} \mathbf{B}^\top \boldsymbol{\tau}, \end{aligned} \quad (29)$$

where $(\mathbf{B}^\top \mathbf{B})^{-1} \mathbf{B}^\top$ is often called the Moore-Penrose pseudoinverse of \mathbf{B} . The condition number of $\mathbf{B}^\top \mathbf{B}$ is between 10^5 - $2 \cdot 10^5$, which indicates the inverse operation to be numerically unstable. The pseudoinverse matrix can be computed in advance using the QR-decomposition approach which is numerically more stable than the direct approach, and can then be stored.

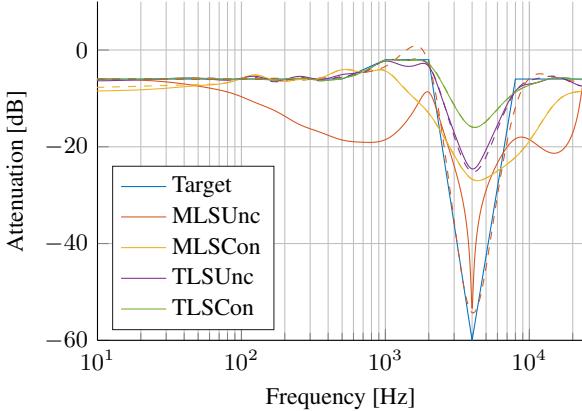
3.2. Non-Linear Least-Squares Solution

The T_{60} least-squares problem given in (7) yields then the following nonlinear least-squares problem:

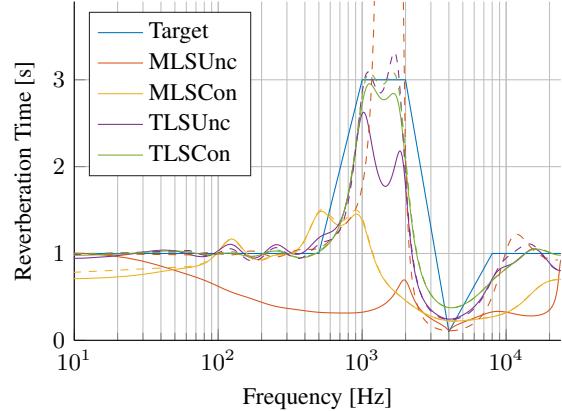
$$\gamma_{TLS} = \arg \min_{\gamma} \left\| \frac{1}{\mathbf{B}\gamma} - \frac{1}{\boldsymbol{\tau}} \right\|_2^2 = \sum_{k=1}^K \left(\frac{1}{(\mathbf{B}\gamma)_k} - \frac{1}{\tau_k} \right)^2, \quad (30)$$

where $(\mathbf{B}\gamma)_k = \sum_{l=0}^L B_{kl} \gamma_l$. Unfortunately, there is no explicit solution for this non-linear problem such that it has to be solved by a numerical optimization algorithm, e.g., gradient descent algorithm [31]. For gradient based approaches it is efficient and more stable to provide the first and second derivatives analytically. The gradient of (30) is given by

$$\frac{\partial}{\partial \gamma_i} \left\| \frac{1}{\mathbf{B}\gamma} - \frac{1}{\boldsymbol{\tau}} \right\|_2^2 = 2 \sum_{k=1}^K \left(\frac{1}{(\mathbf{B}\gamma)_k} - \frac{1}{\tau_k} \right) \left(\frac{-B_{ki}}{(\mathbf{B}\gamma)_k^2} \right)$$



(a) Magnitude response with different filter design methods.



(b) Reverberation time with different filter design methods.

Figure 5: Case study for target reverberation time given in Section 4.1. The dashed lines indicate the approximation response resulting from (28), whereas the solid lines indicate the actual filter response which may differ because of violations of the self-similarity property (23). Both unconstrained methods MLSUnc and TLSUnc has large deviations between the approximation and actual filter response. The TLSCon results in the least error in the resulting reverberation time.

and the Hessian matrix is given by

$$\frac{\partial^2}{\partial \gamma_i \partial \gamma_j} \left\| \frac{1}{B\gamma} - \frac{1}{\tau} \right\|_2^2 = 2 \sum_{k=1}^K \frac{B_{ki}B_{kj}}{(B\gamma)_k^4} + \left(\frac{1}{(B\gamma)_k} - \frac{1}{\tau_k} \right) \left(\frac{2B_{ki}B_{kj}}{(B\gamma)_k^3} \right),$$

where $0 \leq i, j \leq L$.

3.3. Constrained LLS and NLLS

Additionally to the least squares problem, we introduce linear constraints on the command gains γ to account for the deteriorating self-similarity for large gains in (23):

$$-10 \leq \gamma_l \leq 10 \text{ for } 1 \leq l \leq L, \quad (31)$$

where the value of 10 dB was found empirically. Please note that the first gain γ_0 , which is a broadband gain is not limited by a constraint.

In the following, we employ the gradient descent implementation *fmincon* and *fminunc* in MATLAB with and without gain constraints, respectively, to approximate the optimal command gains γ . For non-linear problems, it is inherent to find a local minimum instead of the globally optimal solution and therefore the choice of the initial value γ^{init} can impact the quality of the solution considerably. Different initial values were tested informally, and for simplicity, we settled for the broadband average gain:

$$\gamma^{init} = [\bar{\tau}, 0, \dots, 0], \quad (32)$$

where $\bar{\tau}$ is the arithmetic mean of τ .

4. EVALUATION

Four filter design methods are evaluated in this section:

- MLSUnc: linear unconstrained LS - (29)

- MLSCon: linear constrained LS - (29) & (31)
- TLSUnc: nonlinear unconstrained LS - (30)
- TLSCon: nonlinear constrained LS - (30) & (31)

4.1. Case study

To illustrate some of the shortcomings of a magnitude least squares method for the graphic equalizer, we choose an example target reverberation time and the corresponding target magnitude response values for a delay length of 100 ms. The centre frequencies, the reverberation time definitions and the corresponding target magnitude attenuation are given in Table 1. The large fluctuation in this example was chosen to demonstrate more clearly the potential issues and may be exaggerated in the context of many practical use cases. Table 1 also gives the solutions to the four filter design methods MLSUnc, MLSCon, TLSUnc and TLSCon for a 9-band graphic equalizer and an additional broadband gain. The Fig. 5 shows the resulting attenuation and reverberation time responses for the four methods:

- MLSUnc: The approximation response on the magnitude domain is as expected the best fit to the target magnitude. The corresponding reverberation time deviates strongly, in fact being infinite at 2 kHz. The actual filter response deviates in turn from the approximation in both domains. This is because of a violation of the self-similarity property (23).
- MLSCon: The additional constraints on the command gains γ result in a close match between the approximation and the actual filter response. However, the overall filter design quality is poor.
- TLSUnc: The non-linear filter design yields a optimal fit to the target reverberation time, but with a similar deviation from the actual filter response as observed with MLSUnc.
- TLSCon: The additional constraints on the command gains leads to well corresponding approximation and actual filter response and results in a good overall approximation of the target reverberation time response.

Table 1: Case study corresponding to Fig. 5. The MSE value of the best performing method is indicated in bold.

Center Frequency ω_c [Hz]	MSE	63	125	250	500	1k	2k	4k	8k	16k
Reverberation Time [s]	T_{60}	1	1	1	1	3	3	0.1	1	1
Magnitude Attenuation [dB]	τ	6	-6	-6	-6	-2	-2	-60	-6	-6
Magnitude errors [dB] at ω_c										
Linear unconstrained LS	MLSUnc	12.08	-1.98	-4.86	-9.31	-12.59	-16.52	-6.72	6.68	-12.35
Linear constrained LS	MLSCon	10.28	-1.15	0.85	-0.11	1.92	-2.35	-11.07	33.32	-16.16
Nonlinear unconstrained LS	TLSUnc	9.72	-0.07	0.55	0.55	0.92	-0.30	-1.38	35.44	-2.48
Nonlinear constrained LS	TLSCon	12.17	0.08	0.02	0.16	0.11	-0.20	-0.94	44.08	-2.75
T_{60} errors [s] at ω_c										
Linear unconstrained LS	MLSUnc	0.90	-0.25	-0.45	-0.61	-0.68	-2.68	-2.31	0.01	-0.67
Linear constrained LS	MLSCon	0.77	-0.16	0.17	-0.02	0.47	-1.62	-2.54	0.12	-0.73
Nonlinear unconstrained LS	TLSUnc	0.42	-0.01	0.10	0.10	0.18	-0.39	-1.22	0.14	-0.29
Nonlinear constrained LS	TLSCon	0.32	0.01	0.00	0.03	0.02	-0.28	-0.96	0.28	-0.31
Command gain solutions [dB]										
		γ_0	γ_1	γ_2	γ_3	γ_4	γ_5	γ_6	γ_7	γ_8
Linear unconstrained LS	MLSUnc	-0.74	-5.14	-2.84	-3.27	-2.80	-3.48	22.30	-65.06	9.74
Linear constrained LS	MLSCon	-18.55	10.00	8.41	5.54	8.13	10.00	5.09	-10.00	-2.36
Nonlinear unconstrained LS	TLSUnc	-19.77	13.32	7.66	7.89	6.30	9.66	17.64	-18.64	14.05
Nonlinear constrained LS	TLSCon	-12.15	5.98	3.35	3.76	1.86	6.20	10.00	-10.00	5.02
										γ_9

To summarize, there are two reasons for poor filter designs: Firstly, the command gains are too large, causing the self-similarity property to deteriorate. The solutions listed in Table 1 also demonstrate that the command gains cannot be simply constraint after the approximation. Secondly, good approximation in the attenuation domain may still result in a poor approximation of the reverberation time. Both shortcomings are overcome by the TLSCon solution resulting in a considerable improvement. In the next section, we expand this result from this specific case study by a Monte Carlo simulation.

4.2. Monte Carlo Simulations

To evaluate the accuracy of the four filter design methods for a large number of target functions, we perform a Monte Carlo simulation. For this, we randomize the target reverberation time T_{60} at nine octave band points with uniform distribution between 0.1 and 5 s. We consider three delay lengths of 10, 100 and 1000 ms. For each condition, we computed 1000 randomized T_{60} responses. The approximation error is quantified by the *mean squared error (MSE)* between the target reverberation time and the approximated reverberation time. Solutions which created an unstable feedback loop were counted separately and are referred to as the *probability of instability*.

Fig. 6 depicts the probability distribution of the MSE for the different filter design methods. The quality of approximation can be observed consistently to the case study in increasing order: MLSUnc, MLSCon, TLSUnc and as the best filter design method TLSCon. The probability of instability is roughly in reversed order with some exceptions. Regarding the three different delay lengths, we can observe:

- $m = 10$ ms: The MSE is relatively small for all four filter design methods. Constrained and unconstrained meth-

ods behave largely similar. The MLS methods have a 3-5% chance that the MSE is larger than 2 s, whereas this probability is zero for the TLS methods. The probability of instability is 20% for the MLS methods, whereas it is 0% for TLS methods.

- $m = 100$ ms: Overall, the MSE is slightly larger than for $m = 10$ ms. However, the probability of instability for MLS decreases by up to 7%.
- $m = 1000$ ms: All methods but TLSCon perform considerably worse with the MSE being distributed almost equally up to 2 s. The probability to have an MSE over 2 s is between 15-22% for these methods. The probability of instability for the unconstraint methods is between 19-21%, whereas the constraint methods have a 0% chance to become unstable.

Regarding the impact of the delay length on the MSE, we can observe the following two effects. Firstly, for short delays the target magnitude response is relatively small, e.g., a reverberation time of 5 s corresponds to an attenuation of -0.12 dB per 10 ms, and 0.1 s corresponds to an attenuation of -6 dB per 10 ms. Because of the small target attenuation there is almost no effect from the approximation constraint on the command gains. Further, for MLS methods the target attenuation is close to 0 dB such that the error margin before instability is very narrow causing a high probability of instability. Secondly, long delays deteriorate the approximation quality as the attenuation filter has to attenuate more in one go: a reverberation time of 0.1 s corresponds to an attenuation of -600 dB per 1000 ms, -60 dB per 100 ms and -6 dB per 10 ms. The large attenuation of -600 dB is naturally more difficult to achieve in a controlled fashion than smaller attenuations. The unconstraint methods adapt large command gains causing uncontrolled ripples in the frequency response and therefore unstable feedback. On

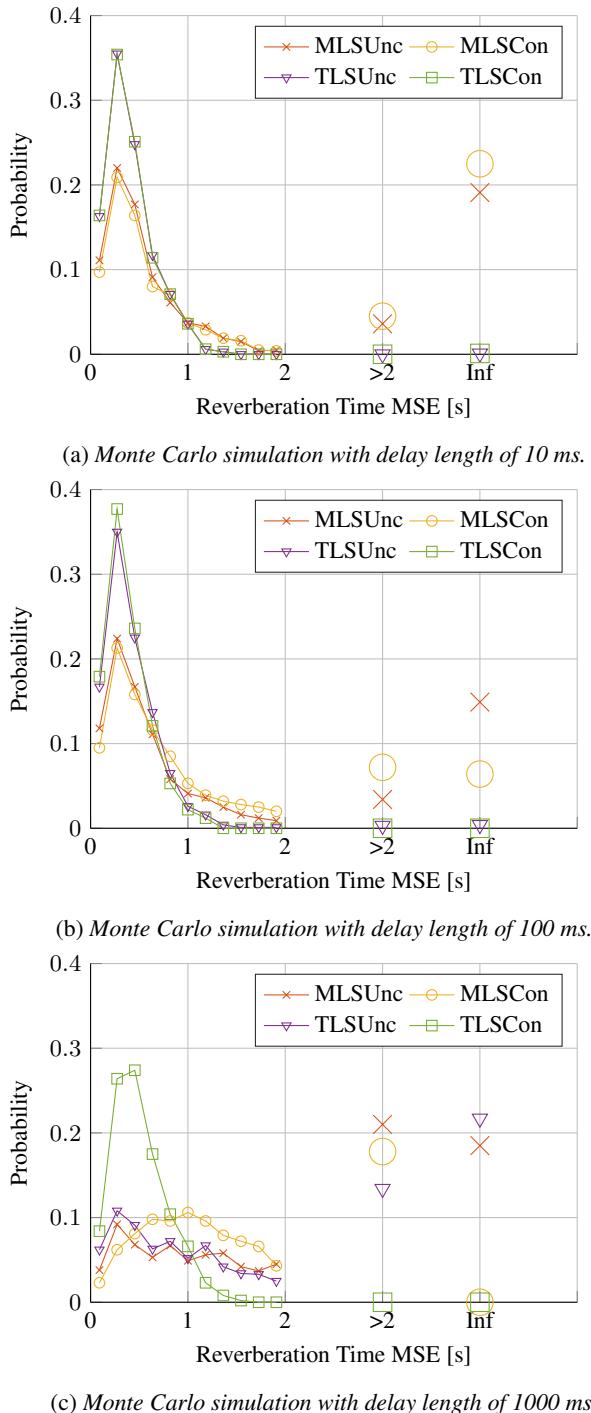


Figure 6: Numerical evaluation of the MSE distribution for different filter design methods.

the contrary, MLSCon chooses a large broadband command gain, and relatively small band command gains such that the filter overattenuates, which guarantees stable feedback, but causes also large deviation from the target reverberation time. Overall, it can be noted that the proposed filter design method TLSCon outperforms

the other methods considerably, especially as it guarantees a stable FDN.

5. CONCLUSION

The impact of different filter design methods of the attenuation filter in an FDN on the resulting reverberation time was investigated. The attenuation filter was implemented by a proportional graphic equalizer. Four methods to compute the parameters were discussed: a linear least-squares problem with and without linear constraint on the command gains approximating the target attenuation magnitude response, and the nonlinear least-squares problem with and without linear constraint approximating the target reverberation time response directly. In a Monte Carlo simulation, we demonstrated that the nonlinear LS solution with linear constraints outperforms the other filter design methods considerably, especially as it guarantees a stable FDN.

This study focus for practicality on a particular filter implementation, however we suggest that it is possible to extend the presented results to other filter types. The proposed approach is difficult to perform in interactive real-time environments such that further investigation in more efficient solutions is required.

References

- [1] Manfred R Schroeder, “New Method of Measuring Reverberation Time,” *J. Acoust. Soc. Amer.*, vol. 37, no. 3, pp. 409–412, 1965.
- [2] Jean Marc Jot, “An analysis/synthesis approach to real-time artificial reverberation,” in *Proc. Int. Conf. Acoustics, Speech and Signal Processing*, San Francisco, California, USA, 1992, pp. 221–224.
- [3] Hans-Peter Seraphim, “Untersuchungen über die Unterschiedsschwelle exponentiellen Abklingens von Rauschbandimpulsen,” *Acta Acustica united with Acustica*, vol. 8, no. 4, pp. 280–284, 1958.
- [4] Theodoros I Niaounakis and William J Davies, “Perception of Reverberation Time in Small Listening Rooms,” *J. Audio Eng. Soc.*, vol. 50, no. 5, pp. 343–350, 2002.
- [5] Matti Karjalainen and Hanna Jarvelainen, “More About This Reverberation Science: Perceptually Good Late Reverberation,” in *Proc. Audio Eng. Soc. Conv.*, New York, NY, USA, Nov. 2001, pp. 1–8.
- [6] Jean Marc Jot and Antoine Chaigne, “Digital delay networks for designing artificial reverberators,” in *Proc. Audio Eng. Soc. Conv.*, Paris, France, Feb. 1991, pp. 1–12.
- [7] Davide Rocchesso and Julius O Smith III, “Circulant and elliptic feedback delay networks for artificial reverberation,” *IEEE Trans. Speech, Audio Process.*, vol. 5, no. 1, pp. 51–63, 1997.
- [8] Vesa Välimäki, Julian D Parker, Lauri Savioja, Julius O Smith III, and Jonathan S Abel, “Fifty Years of Artificial Reverberation,” *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 20, no. 5, pp. 1421–1448, July 2012.

- [9] Huseyin Hacihibiboglu, Enzo De Sena, and Zoran Cvetkovic, “Frequency-Domain Scattering Delay Networks for Simulating Room Acoustics in Virtual Environments,” in *Proc. Int. Conf. Signal-Image Technology Internet-Based Systems*, Dijon, France, 2011, pp. 180–187.
- [10] Enzo De Sena, Huseyin Hacihibiboglu, and Zoran Cvetkovic, “Scattering Delay Network: An Interactive Reverberator for Computer Games,” in *Proc. Audio Eng. Soc. Conf.*, London, UK, Feb. 2011, pp. 1–11.
- [11] Enzo De Sena, Huseyin Hacihibiboglu, Zoran Cvetkovic, and Julius O Smith III, “Efficient Synthesis of Room Acoustics via Scattering Delay Networks,” *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 23, no. 9, pp. 1478–1492, 2015.
- [12] Wallace Clement Sabine, *Collected papers on acoustics*, Reverberation. Harvard University Press, Cambridge, 1922.
- [13] Carl F Eyring, “Reverberation Time in “Dead” Rooms,” *J. Acoust. Soc. Amer.*, vol. 1, no. 2A, pp. 168–168, Jan. 1930.
- [14] James Anderson Moorer, “About This Reverberation Business,” *Comput. Music J.*, vol. 3, no. 2, pp. 13–17, June 1979.
- [15] William G Gardner, “A real-time multichannel room simulator,” *J. Acoust. Soc. Amer.*, vol. 92, no. 4, pp. 2395, 1992.
- [16] Davide Rocchesso, Stefano Baldan, and Stefano Delle Monache, “Reverberation Still In Business: Thickening And Propagating Micro-Textures In Physics-Based Sound Modeling,” in *Proc. Int. Conf. Digital Audio Effects*, Trondheim, Norway, Nov. 2015, pp. 1–7.
- [17] Riitta Väänänen, Vesa Välimäki, Jyri Huopaniemi, and Matti Karjalainen, “Efficient and Parametric Reverberator for Room Acoustics Modeling,” in *Proc. Int. Comput. Music Conf.*, Thessaloniki, Greece, 1997, pp. 200–203.
- [18] Jon Dattorro, “Effect Design, Part 1: Reverberator and Other Filters,” *J. Audio Eng. Soc.*, vol. 45, no. 9, pp. 660–684, 1997.
- [19] Jean Marc Jot, “Efficient models for reverberation and distance rendering in computer music and virtual audio reality,” in *Proc. Int. Comput. Music Conf.*, Thessaloniki, Greece, 1997, pp. 1–8.
- [20] Thibaut Carpentier, Markus Noisternig, and Olivier Warusfel, “Hybrid Reverberation Processor with Perceptual Control.,” in *Proc. Int. Conf. Digital Audio Effects*, Erlangen, Germany, 2014, pp. 93–100.
- [21] Torben Wendt, Steven van de Par, and Stephan D Ewert, “A Computationally-Efficient and Perceptually-Plausible Algorithm for Binaural Room Impulse Response Simulation,” *J. Audio Eng. Soc.*, vol. 62, no. 11, pp. 748–766, 2014.
- [22] Jean Marc Jot, “Proportional Parametric Equalizers - Application to Digital Reverberation and Environmental Audio Processing,” in *Proc. Audio Eng. Soc. Conv.*, New York, NY, USA, Oct. 2015, pp. 1–8.
- [23] Sebastian J Schlecht and Emanuel A P Habets, “On Lossless Feedback Delay Networks,” *IEEE Trans. Signal Process.*, vol. 65, no. 6, pp. 1554–1564, Mar. 2017.
- [24] John G Proakis and Dimitris G Manolakis, *Digital Signal Processing*, Upper Saddle River, New Jersey, 4th edition, 2007.
- [25] Vesa Välimäki and Joshua Reiss, “All About Audio Equalization: Solutions and Frontiers,” *Applied Sciences*, vol. 6, no. 5, pp. 129, May 2016.
- [26] Robert Bristow-Johnson, “The Equivalence of Various Methods of Computing Biquad Coefficients for Audio Parametric Equalizers,” in *Proc. Audio Eng. Soc. Conv.*, San Francisco, CA, USA, Nov. 1994, pp. 1–10.
- [27] Jonathan S Abel and David P Berners, “Filter Design Using Second-Order Peaking and Shelving Sections.,” in *Proc. Int. Comput. Music Conf.*, Miami, USA, 2004, pp. 1–4.
- [28] Seyed-Ali Azizi, “A New Concept of Interference Compensation for Parametric and Graphic Equalizer Banks,” in *Proc. Audio Eng. Soc. Conv.*, Munich, Germany, Apr. 2002, pp. 1–8.
- [29] Jussi Rämö, Vesa Välimäki, and Balázs Bank, “High-Precision Parallel Graphic Equalizer,” *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 22, no. 12, pp. 1894–1904, Dec. 2014.
- [30] Jose A Belloch and Vesa Välimäki, “Efficient target-response interpolation for a graphic equalizer,” in *Proc. Int. Conf. Acoustics, Speech and Signal Processing*, Shanghai, China, 2016, pp. 564–568.
- [31] Jorge Nocedal and Stephen J Wright, *Numerical Optimization*, Springer Series in Operations Research and Financial Engineering. Springer Science & Business Media, New York, Jan. 1999.

BINAURALIZATION OF OMNIDIRECTIONAL ROOM IMPULSE RESPONSES - ALGORITHM AND TECHNICAL EVALUATION

Christoph Pörschmann *, Philipp Stade * †, Johannes M. Arend * †

*TH Köln, Institute of Communication Engineering, Cologne, Germany

†TU Berlin, Audio Communication Group, Berlin, Germany

christoph.poerschmann@th-koeln.de

ABSTRACT

The auralization of acoustic environments over headphones is often realized with data-based dynamic binaural synthesis. The required binaural room impulse responses (BRIRs) for the convolution process can be acquired by performing measurements with an artificial head for different head orientations and positions. This procedure is rather costly and therefore not always feasible in practice. Because a plausible representation is sufficient for many practical applications, a simpler approach is of interest.

In this paper we present the *BinRIR* (Binauralization of omnidirectional room impulse responses) algorithm, which synthesizes BRIR datasets for dynamic auralization based on a single measured omnidirectional room impulse response (RIR). Direct sound, early reflections, and diffuse reverberation are extracted from the omnidirectional RIR and are separately spatialized. Spatial information is added according to assumptions about the room geometry and on typical properties of diffuse reverberation. The early part of the RIR is described by a parametric model and can easily be modified and adapted. Thus the approach can even be enhanced by considering modifications of the listener position. The late reverberation part is synthesized using binaural noise, which is adapted to the energy decay curve of the measured RIR.

In order to examine differences between measured and synthesized BRIRs, we performed a technical evaluation for two rooms. Measured BRIRs are compared to synthesized BRIRs and thus we analyzed the inaccuracies of the proposed algorithm.

1. INTRODUCTION

Binaural synthesis is a powerful tool for headphone-based presentation of virtual acoustic environments (VAEs). It can be applied for auralization purposes in various areas like audio engineering, telecommunication, or architectural acoustics. For many of these applications, a plausible presentation is sufficient; an authenticity reproduction of the sound field is often not pursued. In this context plausibility refers to the illusion that the scenario being depicted is actually occurring [1] while authentic refers to a perception that the scenario cannot be distinguished from a real reference.

Binaural room impulse responses (BRIRs) can be applied, which are either simulated or measured with an artificial head for different orientations (and positions). Finally the BRIRs are convolved with anechoic signals in a binaural renderer. By considering the listener's head movements in the auralization, localization accuracy increases [2], front-back confusion can be decreased [2] and externalization of virtual sound sources improves [3][4]. Several commercial or scientific rendering engines are available, which adapt the sound field presented through headphones according to the orientation of the listener in real time (e.g. [5][6][7][8]). Depending on the head movements, which shall be considered in the

auralization, these measurements need to be done for circular orientations in the horizontal plane or even for spherical head orientations considering horizontal and vertical rotations. However, measuring such BRIR datasets requires a large amount of time and the use of complex devices (e.g. a rotatable artificial head). Furthermore, for each listening position, another set of BRIRs needs to be captured. Thus, for many applications in the field of spatial audio and virtual environments, the effort is so high that circular sets of BRIRs are not used. To approach this issue, we developed the *BinRIR* (Binauralization of omnidirectional room impulse responses) algorithm, which aims for an auralization based on a simple measurement procedure. Only one single measured omnidirectional room impulse response (RIR) is required to obtain a plausible auralization when using dynamic binaural synthesis. The algorithm even allows to shift the listener position. Thus, one single measured RIR is sufficient to synthesize a BRIR dataset for a freely chosen head orientation and position in the room.

In literature, several approaches to obtain BRIRs from measured RIRs have been described. In [9][10] a synthesis of BRIRs from B-format measurements has been proposed. The spatial impulse response rendering (SIRR) method applies a decomposition of the sound field into direct and diffuse parts. While the diffuse part is decorrelated, vector-based amplitude panning is used to distribute the direct sound on different loudspeakers. In [11] a directional audio coding (DirAC) method is proposed which can capture, code, and resynthesize spatial sound fields. DirAC analyzes the audio signal in short time frames and determines the spectrum together with direction and diffuseness in the frequency bands of human hearing. As this method does not work impulse response-based it is quite different to the one presented in this paper. Another simple approach to synthesize BRIRs has been presented by Menzer. In [12][13] RIRs measured in the B-format are used to synthesize BRIRs. Direct sound, spectral shape and temporal structure are extracted from the RIR. Additionally, the incidence direction of the direct sound is estimated from the measured data. No specific treatment of the early reflections is proposed. All reflections and the diffuse reverberation are synthesized by performing an adequate reconstruction of the interaural coherence.

In this paper, we present research results on the binauralization of omnidirectionally measured RIRs. Parts of the studies including the basic idea and a basic description of the approach as well as results of a perceptual evaluation have already been published [14][15][16]. This paper is organized as follows: In section 2 we introduce and explain the *BinRIR* algorithm performing the spatialization of omnidirectional RIRs in detail. In section 3 we describe the results of a technical evaluation. We compare measured BRIRs of two different rooms to the synthesized counterparts and elaborate differences caused by the simplifications of the algorithm. Finally, section 4 concludes the paper and provides an outlook.

2. ALGORITHM DESIGN

2.1. General Structure

The basic idea of the *BinRIR* algorithm is to use only one measured omnidirectional RIR for the synthesis of BRIR datasets which can be used for dynamic auralization. The algorithm was implemented in Matlab and applies predictable information from sound propagation in enclosed spaces as well as knowledge regarding the human perception of diffuse sound fields. For processing, the RIR is split into different parts. The early part contains the direct sound and strong early reflections. For this part, the directions of incidence are modeled reaching the listener from arbitrarily chosen directions. The late part of the RIR is considered being diffuse and is synthesized by convolving binaural noise with small sections of the omnidirectional RIR. By this, the properties of diffuse reverberation are approximated. The algorithm includes an additional enhancement: The synthesized BRIRs can be adapted to shifts of the listener and thus freely chosen positions in the virtual room can be auralized.

The *BinRIR* algorithm incorporates several inaccuracies and deviates significantly from a measured BRIR. The directions of incidence of the synthesized early reflections are not in line with the real ones. Hence, differences in the perception of spatial properties (e.g. envelopment) between the original room and the synthesized room may occur. Furthermore, a point source is assumed for all synthetic BRIR datasets. Thus, it is not possible to rebuild source width and other properties of the source correctly. Finally, the diffusely reflected part of the early reflections cannot be precisely reconstructed.

The basic structure of the *BinRIR* algorithm is shown in Figure 1. As input data the algorithm only requires the omnidirectional RIR and the position of the sound source. Furthermore, the algorithm accesses an appropriate set of HRIRs and a preprocessed sequence of binaural noise. Both were obtained from measurements with a Neumann KU100 artificial head [17].

The algorithm is only applied to frequencies above 200 Hz. For lower frequencies the interaural coherence of a typical BRIR is nearly one and the omnidirectional RIR can be maintained. 7th order Chebyshev Type II filters are used to separate the low frequency part from the rest of the signal.

2.2. Direct sound and early reflections

Onset detection is used to identify the direct sound in the omnidirectional RIR. The direct sound frame starts with the onset and ends after 10 ms (5 ms followed by 5 ms raised cosine offset). The following time section is assigned to the early reflections and the transition towards the diffuse reverberation. For small and non-reverberant rooms ($V < 200 \text{ m}^3$ and $RT_{60} < 0.5 \text{ s}$) a section length of 50 ms is chosen, otherwise the section is extended to 150 ms. In order to determine sections with strong early reflections in the omnidirectional RIR, the energy is calculated in a sliding window of 8 ms length and time sections which contain high energy are marked. Peaks which are 6 dB above the RMS level of the sliding window are determined and assigned to geometric reflections. A windowed section (raised cosine, 5 ms ramp) around each of the peaks is considered as one reflection. If several very dense reflections occur in adjacent sections, these sections are merged. Following this procedure, small windowed sections of the omnidirectional RIR are extracted describing the early reflections. The incidence directions of the synthesized reflections base on a spatial re-

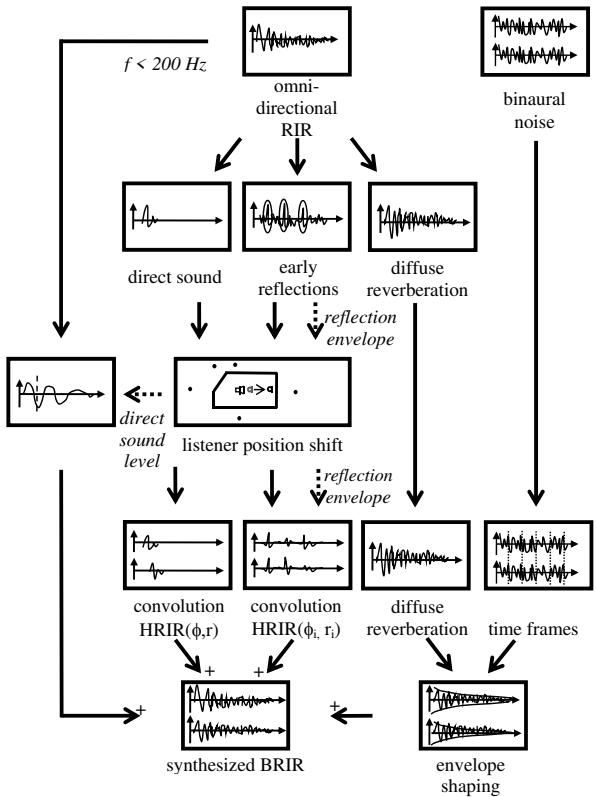


Figure 1: Block diagram of the *BinRIR* algorithm for synthesizing a BRIR based on one single omnidirectional RIR.

flection pattern adapted from a shoebox room with non-symmetric positioned source and receiver. Thus a fixed lookup-table containing the incidence directions is used. By this a simple parametric model of the direct sound and the early reflections is created. Amplitude, incidence direction, delay and the envelope of each of the reflections are stored. The design of the algorithm identifying the reflections and its parameterization (e.g. window length, peak detection) was done based on empiric tests. Informal listening experiments during the development have shown that the exact way in which the reflections are determined is not substantial.

By convolving each windowed section of the RIR with the $HRIR(\phi)$ of each of the directions, a binaural representation of the early geometric reflective part is obtained. To synthesize interim directions between the given HRIRs, interpolation in the spherical domain is performed [18].

2.3. Diffuse Reverberation

The diffuse reverberation is considered reaching the listener temporally and spatially equally distributed. Several studies have shown that after a so-called perceptual mixing time, no specific differences to completely diffuse reverberation can be perceived and thus, an exact reconstruction of the temporal and spatial structure is not required (e.g. [19]). It was found that the perceptual mixing time is room dependent and can be chosen according to predictors which are calculated based on geometric room prop-

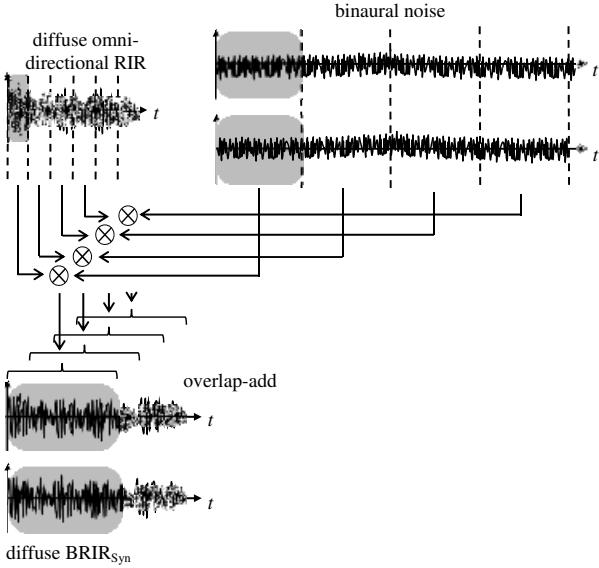


Figure 2: Synthesis of the binaural diffuse reverberation: Sections of the diffuse omnidirectional RIR (0.67 ms; 32 taps at 48 kHz sampling rate) and the binaural noise (2.67 ms; 128 taps at 48 kHz sampling rate) are convolved. Both sections are raised-cosine windowed. The diffuse BRIR is synthesized by summing up the results of the convolutions applying overlap-add.

erties. However, in [20] it has been shown that small perceptual differences still remain. Thus recent studies (e.g. [21][22]) proposed models applying an overlap between early reflections and the diffuse part instead of using a fixed mixing time. By this some diffuse energy is embedded in the early part of the BRIR. A similar approach is used in the *BinRIR* algorithm: All parts of the RIR excluding the sections of the direct sound and the detected early reflections are assigned to the diffuse part.

To synthesize the binaural diffuse reverberation we developed two different methods. In [14][15] the RIR was split up into 1/6 octave bands by applying a near perfect reconstruction filter bank [23] and the binaural diffuse part was synthesized for each frequency band. In [16] we proposed another method for the synthesis of the diffuse part which is used in this publication. The diffuse reverberation is synthesized by convolving small time sections (0.67 ms) of the omnidirectional RIR with sections of 2.67 ms binaural noise (both sections raised-cosine windowed). The results of the convolutions of all time sections are summed up with overlap-and-add. Figure 2 explains the synthesis of the diffuse reverberation in greater detail. By this, both the binaural features (e.g. interaural coherence) of the binaural noise and the frequency-dependent envelope of the omnidirectional RIR are maintained. The lengths of the time sections were determined by informal listening tests during the development of the algorithm. This method requires less computational power than the one proposed in [14][15]. Informal listening tests showed that both methods are perceptually comparable.

2.4. Listener position shifts

The algorithm includes a further enhancement: The synthesized BRIR can be adapted to listener position shifts (LPS) and thus freely chosen positions of the listener in the virtual room can be

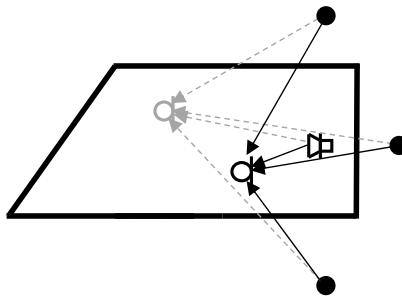


Figure 3: Basic principle of the listener position shifts (LPS): A mirror-image model is applied to modify the amplitude and the temporal structure of the direct sound and the early reflections. The receiver is moved from an initial position (grey) to a modified position (black). By this the paths of the direct sound and the reflections are changed.

auralized. For this, a simple geometric model based on mirror-image sound sources is used. The distance between the listener and each of the mirror-image sound sources is determined from the delay of the corresponding reflection peak to the direct sound peak. In a next step, a shifted position of the listener is considered and amplitudes (based on the $1/r$ law), distances, and directions of incidence are recalculated for each reflection (Fig. 3). Optimizing an earlier version of the *BinRIR* algorithm (e.g [16]) we modified the low-frequency component below 200Hz when applying LPS. If, for example, the listener approaches the sound source, the amplitude of the direct sound increases and the low-frequency energy for the direct sound needs to be adapted accordingly. For this the low-frequency part of the direct sound (first 10ms followed by 10 ms raised cosine set) is adjusted according to the $1/r$ law.

2.5. Synthesis of Circular BRIR sets

The synthesis of the BRIRs is repeated for constant shifts in the azimuthal angle (e.g. 1°) for the direct and the reflected sound. Thus, a circular set of BRIRs is obtained, which can be applied by different rendering engines for dynamic binaural synthesis. The synthesized sets of BRIRs can be stored in various formats, e.g. the miro-Format [24], a multi-channel-wave format to be used by the SoundScape Renderer [7] and can be converted to the SOFA-format [25].

3. TECHNICAL EVALUATION

To analyze the performance of the algorithm, a series of listening experiments has already been conducted [14][16]. These experiments mainly aimed at a quantification of the perceptual differences between measured and synthesized BRIRs. In this paper we focus on a technical evaluation of the algorithm and compare different properties of the synthesized BRIRs to the properties of measured BRIRs. Therefore we analyzed the detected reflections regarding their direction, their time of incidence, and their amplitude to measured data. Furthermore, we compared the reverberation tails and the reverberation times of the synthesized BRIRs to the ones of measured BRIRs. We investigated to what extent the clarity (C_{50}) of the synthesized room matches the measured room's

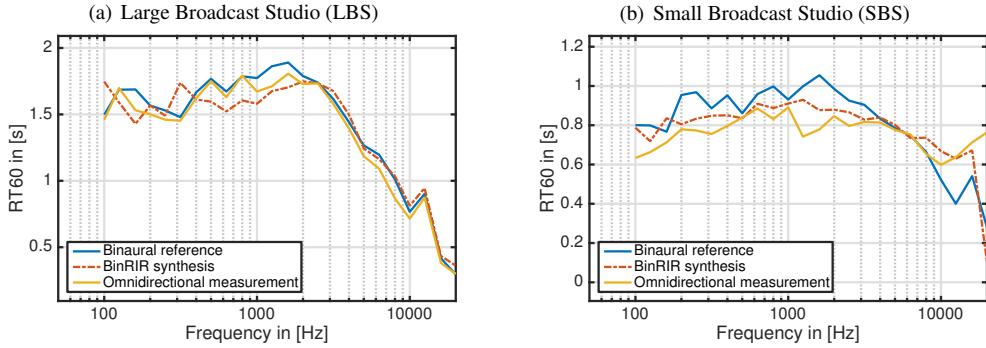


Figure 4: Reverberation Time (RT₆₀) of the Large Broadcast Studio (a) and the Small Broadcast Studio (b). In each plot the RT₆₀ for the binaurally measured reference, for the synthesis with the *BinRIR* algorithm and for the omnidirectional measurement are shown. The RT₆₀ was calculated in 1/3 octave bands in the time domain

clarity. Finally we looked briefly in which way the use of the LPS influences the early part of the BRIR.

3.1. Measured rooms

The performance of the algorithm was analyzed for two different rooms. Both rooms are located at the WDR radiobroadcast studio in Cologne and are used for various recordings of concerts and performances. The "KVB-Saal" (Large Broadcast Studio - LBS) has a volume of 6100 m³, a base area of 579 m² and can seat up to 637 persons. We measured the impulse responses in the 6th row (Distance_{SrcRec} = 13.0 m). The "kleiner Sendesaal" (Small Broadcast Studio - SBS) has a volume of 1247 m³, a base area of 220 m² and 160 seats. The Distance_{SrcRec} in this room was 7.0 m. In order to evaluate the algorithm, measured impulse responses from these rooms were used. In addition to the omnidirectional RIRs, which are required to feed the *BinRIR* algorithm, we measured circular BRIR datasets at the same position as a reference. This dataset was measured in steps of 1° on the horizontal plane with a Neumann KU100 artificial head. Finally we used data from spherical microphone array measurements, conducted with the VariSphere measurement system [26]. For this, we applied a rigid sphere array configuration with 1202 sample points on a Lebedev grid at a diameter of 17.5 cm. The omnidirectional RIRs and the array data were measured with an Earthworks M30 microphone. As sound source, a PA stack involving an AD Systems Stium Mid/High unit combined with 3 AD Systems Flex 15 subwoofers was used. The complete series of measurements is described in detail in [27]. Based on the microphone array measurements, we identified reflections in the room using sound field analysis techniques [28]. For this the array impulse responses were temporally segmented (time resolution = 0.5 ms) and transformed into the frequency domain. Applying a spatial Fourier transformation, the impulse responses were transformed into the spherical wave spectrum domain [29]. Then the sound field was decomposed into multiple plane waves using the respective spatial Fourier coefficients. Data was extracted for a decomposition order of N = 5, a spherical composite grid with 3074 Lebedev points at a frequency f = 4500 Hz. For this frequency quite robust results for the detection of the reflections were found. By this, a spatio-temporal intensity matrix of the sound field at the listener position was calculated. Each time slice of the matrix was analyzed with a specific algorithm in or-

der to classify reflections which are represented as local maxima in the matrix [30]. The detected reflections were stored with their attributes "time", "direction" and "level" in a reflection list and can be used for further comparison with the *BinRIR* algorithm.

3.2. Direct Sound and early reflections

In a first step we looked at the early part of the synthetic BRIRs and compared the direct sound and the early reflections determined by the *BinRIR* algorithm to the reflections which are identified using sound field analysis techniques based on the microphone array measurements. To describe the directions of the reflections we used a spherical coordinate system. The azimuthal angle denotes the orientation on the horizontal plane with $\varphi=0^\circ$ corresponding to the front direction. The elevation angle is $\delta=0^\circ$ for frontal orientation and $\delta=90^\circ$ for sound incidence from above.

Table 1 and 2 show the reflections detected by *BinRIR* as well as the 14 strongest reflections which were identified based on the array data. For each reflection, the time of arrival relative to the direct sound, the incidence direction, and the level are shown. The temporal structure and the energy of many of the reflections show similarities. For example in the LBS for reflection # 7, # 10 and # 13 and in the SBS for reflection # 13 the level and the time of arrival match quite well. Furthermore, some of the reflections were detected in the array measurements as impinging from different directions nearly simultaneously. These reflections are typically merged to one reflection with an increased level by the *BinRIR* algorithm (e.g. LBS: # 4 and # 5; SBS: # 6 and # 7; # 9 - # 11).

As the incidence directions of the synthetic reflections are chosen by *BinRIR* based on a lookup-table, it is not surprising that there are significant differences compared to the directions of the measured reflections. However, if the proportions as well as source and listener position of the synthesized room to some extent match the modelled shoebox room some of the incidence directions are appropriate. Thus, at least for the first reflection and as well for some other reflections detected by *BinRIR*, an acceptable congruence exists both for the LBS and the SBS. The azimuthal deviation of the first reflection is less than 40° and in elevation no relevant deviation exists.

As already explained in 2.2 the *BinRIR* algorithm starts detecting reflections 10 ms after the direct sound. Thus no reflections reaching the listener within the first 10 ms can be found. The time frame

#	Array measurement				BinRIR algorithm			
	Delay [ms]	Azimuth [°]	Elevation [°]	Level [dB]	Delay [ms]	Azimuth [°]	Elevation [°]	Level [dB]
0	0.0	0	3	0.0	0.0	0	0	0.0
1	5.0	8	-58	-27.6				
2	7.5	142	-18	-27.5				
3	17.5	0	-8	-27.3				
4	24.0	309	0	-25.6	24.9	319	-2	-21.0
5	26.5	58	5	-27.9				
6	28.0	278	0	-28.1				
7	31.0	0	58	-20.3	31.1	10	-2	-19.4
8	38.0	253	-2	-28.0	38.4	85	-3	-22.9
9	50.5	180	20	-27.2				
10	79.5	178	17	-12.7	79.6	343	4	-14.9
11	92.5	353	73	-26.6	92.5	319	-54	-40.5
12					108.2	0	67	-34.4
13	117.5	183	35	-26.3	117.6	85	-50	-31.4
14	174.0	356	8	-25.0				
15	202.0	3	3	-24.3				

Table 1: Properties of the direct sound and the early reflections for the Large Broadcast Studio (LBS). Left side: Reflections determined from the analysis of the array data. Right side: Reflections detected by the *BinRIR* algorithm

#	Array measurement				BinRIR algorithm			
	Delay [ms]	Azimuth [°]	Elevation [°]	Level [dB]	Delay [ms]	Azimuth [°]	Elevation [°]	Level [dB]
0	0.0	5	5	0.0	0.0	0	0	0.0
1	1.5	7	-22	-6.9				
2	3.5	16	-76	-21.9				
3	7.5	210	-22	-27.5				
4	12.0	204	-17	-27.7				
5	15.5	27	66	-23.9	22.2	319	-2	-19.2
6	20.0	284	0	-21.1				
7	23.0	58	75	-24.5	35.4	10	-2	-21.6
8	35.0	203	2	-27.7				
9	50.0	152	1	-21.5	51.3	85	-3	-18.5
10	50.5	151	-2	-21.5				
11	51.5	147	29	-25.5				
12	57.0	129	0	-21.4				
13	59.0	173	7	-19.7	59.6	343	4	-15.9
14	99.5	331	-2	-19.5	100.3	319	-54	-29.3

Table 2: Properties of the direct sound and the early reflections for the Small Broadcast Studio (SBS). Left side: Reflections determined from the analysis of the array data. Right side: Reflections detected by the *BinRIR* algorithm

determining the geometric reflections ends after 150 ms and no reflections are detected by *BinRIR* after this period.

Furthermore, several reflections which can be extracted from the array data measurements are not detected by *BinRIR* (e.g. # 3 and # 9 in the LBS and # 5 and # 12 in the SBS). However, in total more than 2/3 of the reflections in the section from 10 ms to 150 ms determined from the array data correspond to a reflection determined by *BinRIR*.

3.3. Energy Decay and Reverberation Time

In a next step we compared the energy decay curves of the synthesis and of the binaurally measured reference BRIRs. As already explained, the *BinRIR* algorithm synthesizes diffuse reverberation by applying a frame-based convolution of the omnidirectional RIR with binaural noise. Thus in addition to the synthesized and the measured BRIR (reference) we analyzed the energy decay of the measured omnidirectional RIR as well. The following analysis is based on the impulse responses for the frontal viewing direction ($\varphi=0^\circ$, $\delta=0^\circ$). Analyzing the reverberation time RT₆₀ (Figure 4) we observed that the general structure of the curves is similar, but variations between the three curves exist. The average un-

signed deviation between the synthesis and the reference is 0.10 s for the LBS and 0.09 s for the SBS, the maxima are 0.26 s (LBS) and 0.23 s (SBS).

3.4. Interaural Coherence

Next, we compared the interaural coherence (IC) of the synthesized BRIRs and of the reference BRIRs (Figure 5). We calculated the IC according to [31] applying hamming-windowed blocks with a length of 256 taps (5.33 ms) and an overlap of 128 taps (2.67 ms). In each plot the IC calculated with three different starting points is shown. For reference and synthesis in both rooms the IC is significantly different when direct sound is included in the calculation ($t > 0$ ms). For the medium condition (LBS: $t > 150$ ms; SBS: $t > 50$ ms) significant differences between synthesis and reference can be observed. This is not surprising as no shaping of the IC is performed in the *BinRIR* algorithm. However, this difference is smaller for the SBS, because the impulse response is probably nearly diffuse at 50 ms. For the condition with the maximal starting point (LBS: $t > 300$ ms; SBS: $t > 150$ ms) which mainly comprises the diffuse reverberation the IC of the synthesized BRIR matches the reference quite well.

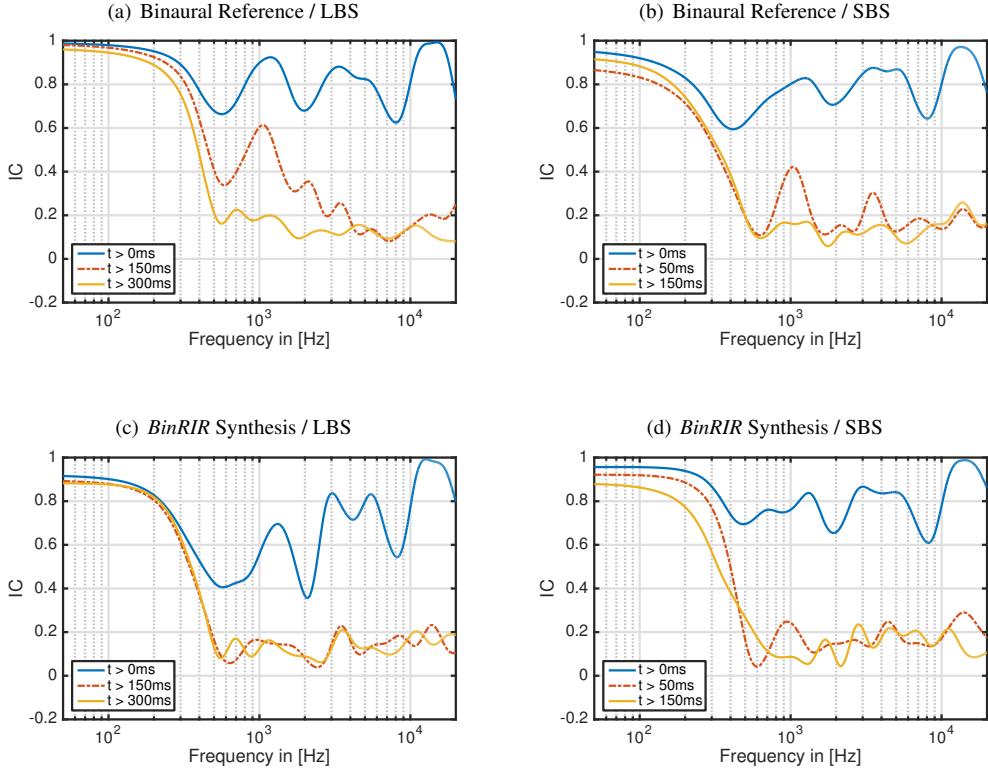


Figure 5: Interaural coherence (IC) of the Large Broadcast Studio (LBS) and the the Small Broadcast Studio (SBS). In plot (a) and (b) the data for the binaural reference is shown, in (c) and (d) the data for the *BinRIR* synthesis. For the LBS the IC is plotted for $t > 0\text{ ms}$, $t > 150\text{ ms}$ and $t > 300\text{ ms}$, for the SBS for $t > 0\text{ ms}$, $t > 50\text{ ms}$ and $t > 150\text{ ms}$.

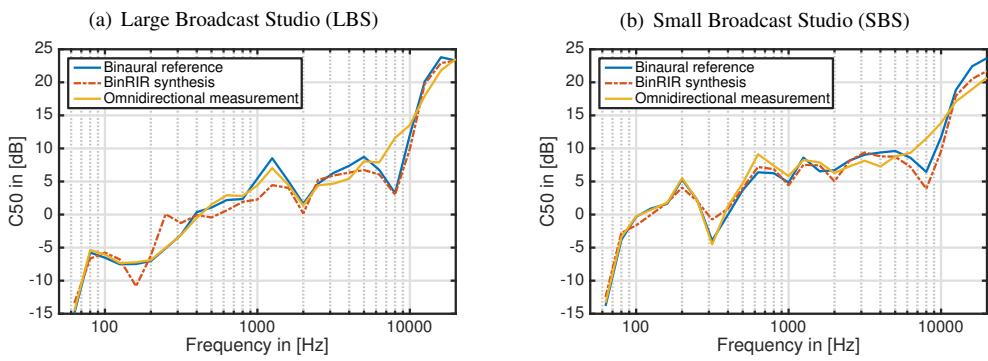


Figure 6: Clarity (C_{50}) of the Large Broadcast Studio (LBS) and the the Small Broadcast Studio (SBS). In each plot the C_{50} for the binaurally measured reference, for the synthesis with the *BinRIR* algorithm and for the omnidirectional measurement are shown.

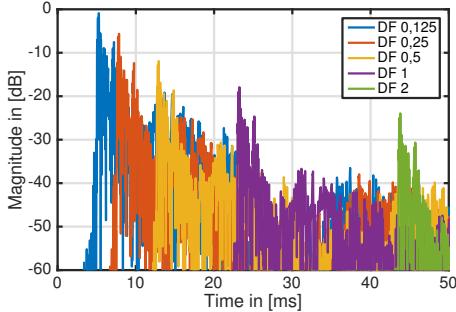


Figure 7: Influence of the Listener Position Shift (LPS) on the early part of the time response for the Small Broadcast Studio (SBS). The time responses for Distance Factors (DFs) from 0.125 - 2 are shown in different colors

3.5. Clarity

Next, we examined the clarity C_{50} over frequency for each of the conditions (Figure 6). The differences between the omnidirectional RIR, the synthesis and the reference are minor. The average unsigned deviation between the synthesis and the reference is 1.4 dB for the LBS and 1.1 dB for the SBS. The maxima are 5.2 dB (LBS) and 3.2 dB (SBS). Thus the ratio of the energy of the early part of the BRIR and the late diffuse part of the BRIR can be regarded as appropriate.

3.6. Listener position shifts

Finally we analyzed for the SBS in which way the early part of the synthesized BRIR is changed when listener position shifts (LPS) are performed. The results are shown in Figure 7 for synthesized Distances_{SrcRec} between 0.875 m and 14 m (distance factor 1/8 - 2 of the original distance). It can be observed that the level and the time of arrival of the direct sound are changed significantly (according to the 1/r distance law) when performing LPS. The influence of the LPS on reflections is hard to observe from the plot, but changes in amplitude and time of arrival according to the geometric room model can be found here as well. The diffuse part and thus the complete late reverberation remain unchanged when performing LPS (not shown in Figure 7).

4. CONCLUSION

In this paper, the *BinRIR* algorithm was presented, which aims for a plausible dynamic binaural synthesis based on one measured omnidirectional RIR. In two different rooms, RIRs were measured and binauralized applying the presented *BinRIR* algorithm, so that synthetic BRIR datasets were generated. The presented method separately treats direct sound, reflections and diffuse reverberation. The early parts of the impulse responses are convolved with HRIRs of arbitrary chosen directions while the reverberation tail is rebuilt from an appropriately shaped binaural noise sequence. In an extension, the algorithm allows to modify the sound source distance of a measured RIR by changing several parameters of an underlying simple room acoustic model.

The synthetic BRIRs were compared to reference BRIRs measured with an artificial head. Due to missing information on spatial as-

pects, a perfect reconstruction of the sound field is generally not possible. An analysis of the early reflections showed that neither all reflections are detected by the *BinRIR* algorithm nor their directions match to the physical ones of the room. However, the reflections which were identified by the *BinRIR* algorithm correlate with the times of incidence and partly with the direction of incidence of the physical reflections in the room quite well. For the diffuse part, small differences in the reverberation time and the interaural coherence were observed. However, in general, the synthesis can be regarded as appropriate. An evaluation of the reverberation time RT_{60} and of the clarity C_{50} only showed minor differences between reference and synthesis. Analyzing the perceptual influences of the determined differences is not covered in the study presented here. Please refer to [14][16] for an analysis of these topics.

The approach presented in this paper can be combined with other modifications of measured RIRs. In [32][33], we discussed a predictive auralization of room modifications by an appropriate adaptation of the BRIRs. Thus the measurement of one single RIR is sufficient to obtain a plausible representation of the modified room. Furthermore, the opportunity to shift the listener position freely in the room can be employed when perceptual aspects of listener movements shall be investigated as e.g. proposed in [34].

5. ACKNOWLEDGEMENTS

The research presented here has been carried out in the Research Project MoNRA which is funded by the Federal Ministry of Education and Research in Germany. Support Code: 03FH005I3-MoNRA. We appreciate the support.

The code of the Matlab-based implementation and a GUI-based version of the *BinRIR* algorithm which is available under the GNU GPL License 4.1 can be accessed via the following webpage: <http://www.audiogroup.web.th-koeln.de/DAFX2017.html>

6. REFERENCES

- [1] Mel Slater, “Place illusion and plausibility can lead to realistic behaviour in immersive virtual environments,” *Philosophical Transactions of the Royal Society B: Biological Sciences*, vol. 364, no. 1535, pp. 3549–3557, 2009.
- [2] Jens Blauert, *Spatial Hearing - Revised Edition: The Psychoacoustics of Human Sound Source Localisation*, MIT Press, Cambridge, MA, 1997.
- [3] Etienne Hendrickx, Peter Stitt, Jean-Christophe Messonnier, Jean-Marc Lyzwa, Brian FG Katz, and Catherine de Boishéraud, “Influence of head tracking on the externalization of speech stimuli for non-individualized binaural synthesis,” *The Journal of the Acoustical Society of America*, vol. 141, no. 3, pp. 2011–2023, 2017.
- [4] W. Owen Brimijoin, Alan W. Boyd, and Michael A. Akeroyd, “The contribution of head movement to the externalization and internalization of sounds,” *PLoS ONE*, vol. 8, no. 12, pp. 1–12, 2013.
- [5] Ulrich Horbach, Attila Karamustafaoglu, Renato S. Pellegrini, and Philip Mackensen, “Design and Applications of a Data-based Auralization System for Surround Sound,” in *Proceedings of 106th AES Convention, Convention Paper 4976*, 1999.

- [6] Jens Blauert, Hilmar Lehnert, Jörg Sahrhage, and Holger Strauss, “An Interactive Virtual-Environment Generator for Psychoacoustic Research I: Architecture and Implementation,” *Acta Acustica united with Acustica*, pp. 94–102, 2000.
- [7] Matthias Geier, Jens Ahrens, and Sascha Spors, “The soundscape renderer: A unified spatial audio reproduction framework for arbitrary rendering methods,” in *Proceedings of 124th Audio Engineering Society Convention 2008*, 2008, pp. 179–184.
- [8] Dirk Schröder and Michael Vorländer, “RAVEN: A Real-Time Framework for the Auralization of Interactive Virtual Environments,” *Forum Acusticum*, pp. 1541–1546, 2011.
- [9] Juha Merimaa and Ville Pulkki, “Spatial impulse response rendering I: Analysis and synthesis,” *Journal of the Audio Engineering Society*, vol. 53, no. 12, pp. 1115–1127, 2005.
- [10] Ville Pulkki and Juha Merimaa, “Spatial impulse response rendering II: Reproduction of diffuse sound and listening tests,” *Journal of the Audio Engineering Society*, vol. 54, no. 1-2, pp. 3–20, 2006.
- [11] Ville Pulkki, Mikko-Ville Laitinen, Juha Vilkamo, Jukka Ahonen, Tapio Lokki, and Tapani Pihlajamäki, “Directional audio coding-perception-based reproduction of spatial sound,” *International Workshop On The Principles And Applications of Spatial Hearing (IWPASH 2009)*, 2009.
- [12] Fritz Menzer, *Binaural Audio Signal Processing Using Interaural Coherence Matching*, Dissertation, École polytechnique fédérale de Lausanne, 2010.
- [13] Fritz Menzer, Christof Faller, and Hervé Lissek, “Obtaining binaural room impulse responses from b-format impulse responses using frequency-dependent coherence matching,” *IEEE Transactions on Audio, Speech and Language Processing*, vol. 19, no. 2, pp. 396–405, 2011.
- [14] Christoph Pörschmann and Stephan Wiefling, “Perceptual Aspects of Dynamic Binaural Synthesis based on Measured Omnidirectional Room Impulse Responses,” in *International Conference on Spatial Audio*, 2015.
- [15] Christoph Pörschmann and Stephan Wiefling, “Dynamische Binauralsynthese auf Basis gemessener einkanaliger Raumimpulsantworten,” in *Proceedings of the DAGA 2015*, 2015, pp. 1595–1598.
- [16] Christoph Pörschmann and Philipp Stade, “Auralizing Listener Position Shifts of Measured Room Impulse Responses,” *Proceedings of the DAGA 2016*, pp. 1308–1311, 2016.
- [17] Benjamin Bernschütz, “A Spherical Far Field HRIR / HRTF Compilation of the Neumann KU 100,” *Proceedings of the DAGA 2013*, pp. 592–595, 2013.
- [18] Benjamin Bernschütz, *Microphone Arrays and Sound Field Decomposition for Dynamic Binaural Recording*, Dissertation, TU Berlin, 2016.
- [19] Alexander Lindau, Linda Kosanke, and Stefan Weinzierl, “Perceptual evaluation of physical predictors of the mixing time in binaural room impulse responses,” *Journal of the Audio Engineering Society*, vol. 60, no. 11, pp. 887–898, 2012.
- [20] Philipp Stade, “Perzeptive Untersuchung zur Mixing Time und deren Einfluss auf die Auralisation,” *Proceedings of the DAGA 2015*, pp. 1103–1106, 2015.
- [21] Philipp Stade and Johannes M. Arend, “Perceptual Evaluation of Synthetic Late Binaural Reverberation Based on a Parametric Model,” in *AES Conference on Headphone Technology*, 2016.
- [22] Philip Coleman, Andreas Franck, Philip J.B. Jackson, Luca Remaggi, and Frank Melchior, “Object-Based Reverberation for Spatial Audio,” *Journal of the Audio Engineering Society*, vol. 65, no. 1/2, pp. 66–76, 2017.
- [23] Wessel Lubberhuizen, “Near perfect reconstruction polyphase filterbank, Matlab Central, www.mathworks.com/matlabcentral/fileexchange/15813 assessed 04/04/2017,” 2007.
- [24] Benjamin Bernschütz, “MIRO - measured impulse response object: data type description,” 2013.
- [25] Piotr Majdak, Yukio Iwaya, Thibaut Carpentier, Rozenn Nicol, Matthieu Parmentier, Agnieszka Roginska, Yôiti Suzuki, Kanji Watanabe, Hagen Wierstorf, Harald Ziegelwanger, and Markus Noisternig, “Spatially oriented format for acoustics: A data exchange format representing head-related transfer functions,” in *Proceedings of the 134th Audio Engineering Society Convention 2013*, 2013, number May, pp. 262–272.
- [26] Benjamin Bernschütz, Christoph Pörschmann, Sascha Spors, and Stefan Weinzierl, “SOFiA - Sound Field Analysis Toolbox,” in *Proceedings of the International Conference on Spatial Audio - ICSA*, 2011.
- [27] Philipp Stade, Benjamin Bernschütz, and Maximilian Rühl, “A Spatial Audio Impulse Response Compilation Captured at the WDR Broadcast Studios,” in *27th Tonmeistertagung - VDT International Convention*, 2012, pp. 551–567.
- [28] Benjamin Bernschütz, Philipp Stade, and Maximilian Rühl, “Sound Field Analysis in Room Acoustics,” *27th Tonmeistertagung - VDT International Convention*, pp. 568–589, 2012.
- [29] Earl G. Williams, *Fourier Acoustics - Sound Radiation and Nearfield Acoustical Holography*, Academic Press, London, UK, 1999.
- [30] Philipp Stade, Johannes M. Arend, and Christoph Pörschmann, “Perceptual Evaluation of Synthetic Early Binaural Room Impulse Responses Based on a Parametric Model,” in *Proceedings of 142nd AES Convention, Convention Paper 9688*, Berlin, Germany, 2017, pp. 1–10.
- [31] Fritz Menzer and Christof Faller, “Stereo-to-Binaural Conversion Using Interaural Coherence Matching,” in *Proceedings of the 128th AES Convention, London UK*, 2010.
- [32] Christoph Pörschmann, Sebastian Schmitter, and Aline Jaritz, “Predictive Auralization of Room Modifications,” *Proceedings of the DAGA 2013*, pp. 1653–1656, 2013.
- [33] Christoph Pörschmann, Philipp Stade, and Johannes M. Arend, “Binaural auralization of proposed room modifications based on measured omnidirectional room impulse responses,” in *Proceedings of the 173rd Meeting of the Acoustical Society of America*, 2017.
- [34] Annika Neidhardt and Niklas Knoop, “Binaural walk-through scenarios with actual self-walking using an HTC Vive Real-time rendering of binaural audio Interactive binaural audio scenes Creating scenes allowing self-translation,” in *Proceedings of the DAGA 2017*, 2017, pp. 283–286.

PINNA MORPHOLOGICAL PARAMETERS INFLUENCING HRTF SETS

Slim Ghorbal

3D Sound Labs

Paris, France

s.ghorbal@3dsoundlabs.com

Théo Auclair

3D Sound Labs

Paris, France

t.auclair@3dsoundlabs.com

Catherine Soladie

FAST,

CentraleSupélec

Rennes, France

catherine.soladier@centralesupelec.fr

Renaud Séguier

FAST,

CentraleSupélec

Rennes, France

renaud.seguier@centralesupelec.fr

ABSTRACT

Head-Related Transfer Functions (HRTFs) are one of the main aspects of binaural rendering. By definition, these functions express the deep linkage that exists between hearing and morphology - especially of the torso, head and ears. Although the perceptive effects of HRTFs is undeniable, the exact influence of the human morphology is still unclear. Its reduction into few anthropometric measurements have led to numerous studies aiming at establishing a ranking of these parameters. However, no consensus has yet been set. In this paper, we study the influence of the anthropometric measurements of the ear, as defined by the CIPIC database, on the HRTFs. This is done through the computation of HRTFs by *Fast Multipole Boundary Element Method* (FM-BEM) from a parametric model of torso, head and ears. Their variations are measured with 4 different spectral metrics over 4 frequency bands spanning from 0 to 16kHz. Our contribution is the establishment of a ranking of the selected parameters and a comparison to what has already been obtained by the community. Additionally, a discussion over the relevance of each approach is conducted, especially when it relies on the CIPIC data, as well as a discussion over the CIPIC database limitations.

1. INTRODUCTION

The HRTFs of a listener are intimately related to his morphology. Thus, a good knowledge of his shape should be a sufficient condition for inferring his HRTFs. Following that idea, many efforts have been done to personalise HRTF sets using anthropometric data. Therefore, the literature is rich of articles exploring this path.

Inoue et al. [1] measured the HRTFs and nine physical features of the head and ears of 86 Japanese subjects. Then, they studied their relationship through multiple regression analysis and used it as an estimation method.

For their part, Zotkin et al. [2] proposed an HRTF personalisation algorithm based on digital images of the ear taken by a video camera. They perform 7 measurements on it and compute out of them a distance between subjects of a given database. The closest match is selected and his HRTFs are used as raw material for the individualisation experiment.

Bilinski et al. [3] propose a method for the synthesis of the magnitude of HRTFs using a sparse representation of anthropometric features. They use a super-set of the features defined by

the CIPIC database and learn the sparse representation of subjects of a training database. Then a l_1 -minimisation problem is solved for finding the best sparse representation of a new subject. This representation is then used for the synthesis of his HRTF set.

However, in these studies, all the parameters are not necessarily independent nor even decisive. Hence, several researchers proposed new means for refining their selection.

Among them, Hu et al. introduced a correlation analysis in two steps [4, 5] prior to the personalisation process. They used the CIPIC database and highlighted significant correlations leading to the selection of only 8 parameters out of the available 27. 5 of them were related to the ear.

Xu et al. [6, 7] retained ten measurements after performing a correlation analysis between the CIPIC HRTFs and anthropometric parameters. It is worth noting that the analysis was restrained to 7 directions and 4 frequencies and that only two of the retained measurements were related to the ear.

Hugeng et al. [8] also realised a correlation analysis over the CIPIC data but ended up in retaining 8 measurements. 4 of them were ear measurements.

Grijalva et al. [9] applied a customisation process of HRTFs using Isomap and Artificial Neural Networks on the CIPIC database. Prior to this, they used the results of [8] as the appropriate morphological parameters to focus on.

While the previous studies added the measurements selection into a wider personalisation process, others exclusively focused on the relative influence of each parameter. This is what did Zhang et al. [10], who concluded after a correlation analysis that 7 ear measurements were among the 8 most significant ones, and Fels et al. [11] who use a parametric model of head, torso and ear for generating new HRTF sets by *Boundary Element Method* (BEM).

The latter study compared separately the influence of 6 parameters describing the head and 6 others describing the pinna based on the modifications introduced in the HRTFs. The evaluation took into account the spectral distance, the *interaural time difference* (ITD) and the *interaural level difference* (ILD) variations. One parameter at a time was modified, ranged in limits derived from anthropometric statistics of adults and children.

Although it provided insights on the relative weights of the parameters in each group, a clear ranking was not established. Moreover, the simulations were limited to frequencies below 8 kHz. This is a major limitation as the human hearing ranges up to 20

kHz and that localisation information due to the pinna is classically comprised between 3-4 kHz and 14-15 kHz or more [12, 13].

This multiplicity of works and conclusions reveals an absence of clear consensus about the way each anthropometric parameter modifies - or not - the HRTFs. In the present paper, we take an approach similar as Fels et al. but extended to frequencies up to 16 kHz and establish a categorisation of the pinna parameters.

More in detail, section 2 goes through the process of selection of parameters, the generation of meshes, the computation of HRTFs and the choice and definition of the metrics. Section 3 presents the results themselves, discusses the impact of the chosen metric and establishes a ranking between the retained pinna parameters based on their relative influence over the DTFs. In section 4, we effectively compare our results to the conclusions proposed up to now by the community and lead a discussion over the convergences and points of disagreement. Finally, section 5 sums up the results and conclusions of the present paper and gather the opened questions and perspective of future works.

2. PROTOCOL

2.1. Parameters

2.1.1. CIPIC database

As it is widely used in the community, we have chosen to consider the morphological parameters defined by CIPIC [14]. This database consists in sets of HRTF and morphological parameters measured on 45 subjects. These parameters - 27 in total - are intended to describe the torso, head and ears of the human body with a focus on what could likely impact the HRTF. In particular, 12 parameters describe the position and shape of the ear, the remaining ones describe the head and the body.

It is worth noting that only 35 subjects have been fully measured, meaning that each parameter comes with a set of measures comprised between 35 and 45 values. The database also comes with their mean values μ and standard deviations σ .

2.1.2. Selection and values

Based on the sets of ear parameters selected in [11, 1, 2, 8, 10], we retain the set of parameter $\mathcal{P} = \{d_1, d_2, d_3, d_4, d_5, d_6, \theta_1, \theta_2\}$ - defined in figure 2 -, namely the cavum concha height, the cyma concha height, the cavum concha width, the fossa height, the pinna height, the pinna width, the pinna rotation angle and the pinna flare angle (See table 1). This choice is a result of the number of occurrences of each parameter in these studies, their selection or not as major parameters and our ability to set them with precision in our 3D model (as a reminder, the CIPIC parameters are defined in 2D).

To stick to plausible deformations, for each $p \in \mathcal{P}$, we target the following set of values:

$$\mathcal{V}_p = \{\mu_p + k * \sigma_p, k \in [-2, 2]\} \quad (1)$$

Moreover, as we intend to study the influence of each p independently, only one at a time can be different from its mean value. When possible - or when it makes sense - , the other CIPIC parameters are set to their mean value. In what follows, we will denote p^{ko} the simulation where parameter p is set to $\mu_p + k * \sigma_p$.

Table 1: Anthropometric statistics for parameters in \mathcal{P} . Distances are in cm and angles in degrees.

Var	Measurement	μ	σ
d_1	cavum concha height	1.91	0.18
d_2	cyma concha height	0.68	0.12
d_3	cavum concha width	1.58	0.28
d_4	fossa height	1.51	0.33
d_5	pinna height	6.41	0.51
d_6	pinna width	2.92	0.27
θ_1	pinna rotation angle	24.01	6.59
θ_2	pinna flare angle	28.53	6.70

2.2. Morphological model

The model used in this paper is a parametric one, result of a merge between a schematic representation of head and torso - to which we will refer as *snowman*, although it does not strictly match the one introduced by Algazi & Duda [15] - and an ear model realised thanks to Blender and deformable through multiple blend shapes. The snowman is used in order to add more realism to the generated HRTFs and get closer to what could actually be measured on a real person. The ear model is the true source of interest. It is designed to be as close as possible to real ears, there again for realism. Figure 1 represents the mean ear before and after merge on the snowman - referred as the mean shape.

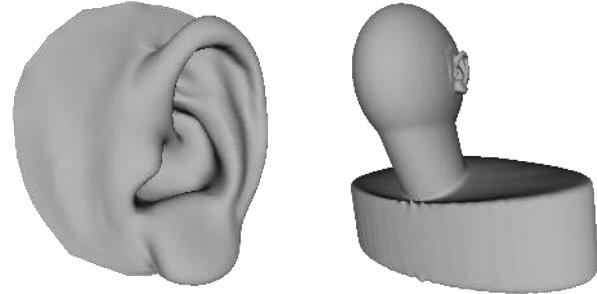


Figure 1: Mean ear alone (left) and after merge (right)

Although the CIPIC parameters definition may seem simple at first sight on a 2D drawing - see fig 2 -, for them to be fully usable in our 3D environment we need whether to carry out projections of the model on well-chosen plans or to extend them to 3D. Although both options come with drawbacks, the latter seemed more appropriated.

2.3. HRTF generation

Each mesh obtained from the model is then used to feed the FM-BEM computation software *mesh2hrtf* [16, 17]. A virtual source is placed inside the ear canal and virtual microphones are distributed on a sphere of radius 1.2 m whose centre coincides with the centre of the interaural axis. It is worth noting that this sphere is not strictly uniform but slightly denser on the pole than on the equator. Moreover, the directions of elevation inferior to -60° have been excluded from the computations.

The output is a set of 2141 HRTFs computed for every frequency between 100 Hz and 16 kHz by steps of 100 Hz.

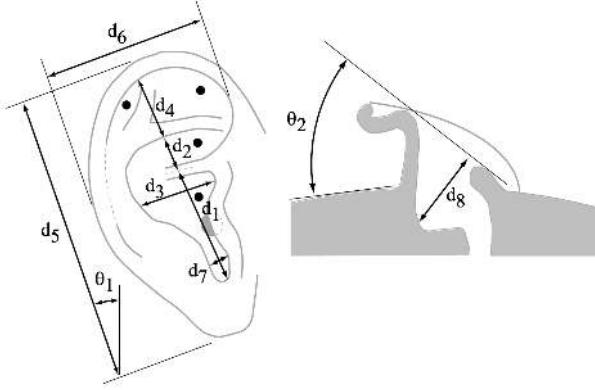


Figure 2: Pinna measurements definition

2.4. Metrics

In order to compare the variations introduced by the ear deformations in the DTF sets, the following metrics are used:

- The widely-used [1, 8, 9] *Spectral Distortion* (SD) - also sometimes referred as *Log Spectral Distortion* -, defined as:

$$SD = \sqrt{\frac{1}{N} \sum_{k=0}^{N-1} \left(20 * \log_{10} \left(\frac{H_1(f_k)}{H_2(f_k)} \right) \right)^2} [dB] \quad (2)$$

where the frequencies f_k are regularly spaced on a linear scale.

- The *Spectral Distance Measure* (SDM), introduced by Huopaniemi [18], corresponding to the SD distance where the frequencies f_k are regularly spaced on a logarithmic scale.
- The *Inter-Subject Spectral Distortion* (ISSD) introduced by Middlebrooks [19] and defined as the mean over the directions of the variance of the difference between the DTFs to compare.
- The *log-ISSG* introduced by Rugeles [20] and corresponding to the ISSD distance where the frequencies f_k are regularly spaced on a logarithmic scale.

Additionally, the frequency band [0, 16] kHz is split into 4 subbands of 4 kHz width each.

3. EXPERIMENTAL RESULTS

3.1. General observations

As an introductory example, the ipsilateral DTFs of the simulations $d_3^{k\sigma}$, $k \in \{-2, -1, 1, 2\}$ are compared to the ipsilateral DTFs of the mean ear simulation in figure 3. The corresponding ears are gathered in figure 4. As expected, only the high frequencies are affected by the change of the shape (frequencies above 6kHz in the present case).

For coherence between the outputs, the simulations have been gathered into 4 different groups corresponding to the deviation applied to the parameter under study. In practice, it means that for each $k \in \{-2, -1, 1, 2\}$, the simulations $d_1^{k\sigma}, d_2^{k\sigma}, \dots, \theta_2^{k\sigma}$ are

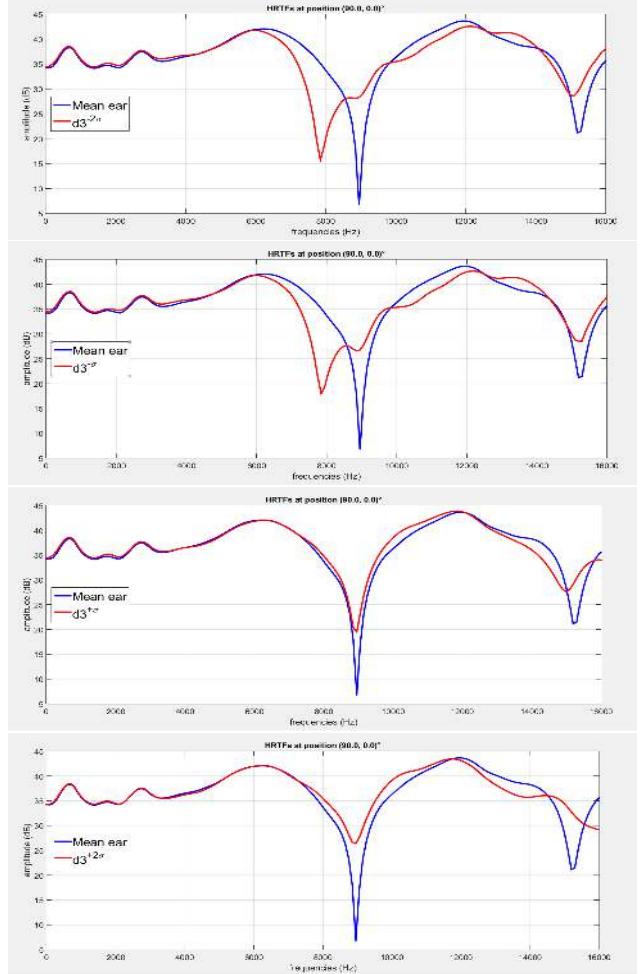


Figure 3: DTFs in the ipsilateral direction for the mean ear (blue) and for the deformed ear (red).

studied together. Figure 5 presents the impact of a deviation of -2σ on each parameter, frequency band by frequency band and for each retained metric.

3.2. Metrics choice

An immediate observation is that the log-ISSD (resp. SDM) yields almost the same results as the ISSD (resp. SD). In fact, their average absolute difference varies between 0.05 and 0.13 (resp. 0.05 and 0.11) for each frequency band, while their average values are around 2.2 (resp 2.3).

Another straightforward observation is that all the curves are monotonically increasing, with a low initial value. This is coherent with what has been seen in the introductory example (see fig 3). The ear has almost no effect on the low part of the spectrum, the real gap occurring in the [4 – 8] kHz band. Moreover, as expected, we find that the higher the frequencies, the greater the sensitiveness to pinna deformations. Finally, the ranking obtained through the ISSD or through the SD appear to be very similar. Focusing on the -2σ group, we observe indeed in each case a major influence of parameters d_3, d_4 and θ_2 while d_1, d_2 and particularly d_6 have

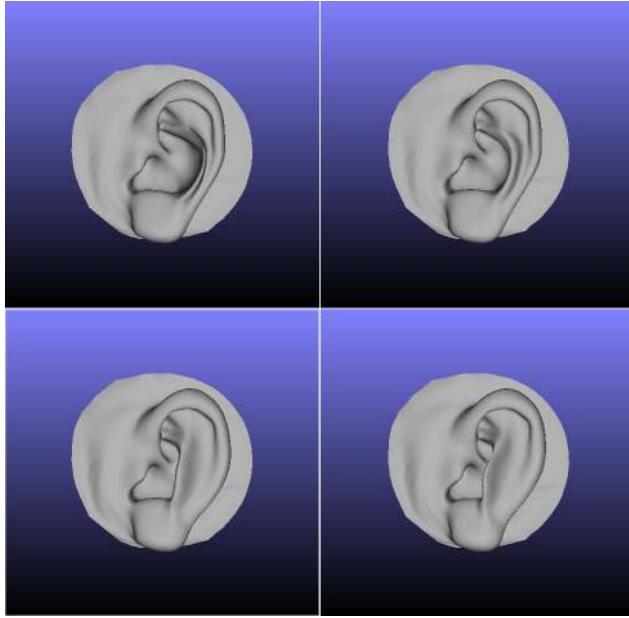


Figure 4: The 4 ears generated for the d_3 simulations.

little impact in comparison.

3.3. Deformation impact

The figure 6 shows the parameters influence for each applied deviation with respect to the ISSD metric.

As expected, the greater the deviation, the greater the influence of the parameters. In fact, for each parameter, and excepted the band [0 - 4]kHz where the impact on the HRTFs is not significant enough, the norms computed for a deviation of -2σ (resp. $+2\sigma$) are greater than the ones computed for a deviation of $-\sigma$ (resp. $+\sigma$).

However, it is worth noticing that these changes are not linear with respect to the deviation. In other words, doubling the deviation will not necessarily imply doubling the metric. In particular, the simulations $d_5^{+2\sigma}$ and $\theta_2^{-2\sigma}$ appear to strongly change the HRTFs while the other simulations for these 2 parameters show a moderate or weak influence.

Nevertheless, some regularities exist. This is the case for parameters d_3 and d_4 , which systematically rank among the most influential ones and for d_1 , d_2 and d_6 which almost systematically rank among the least influential ones.

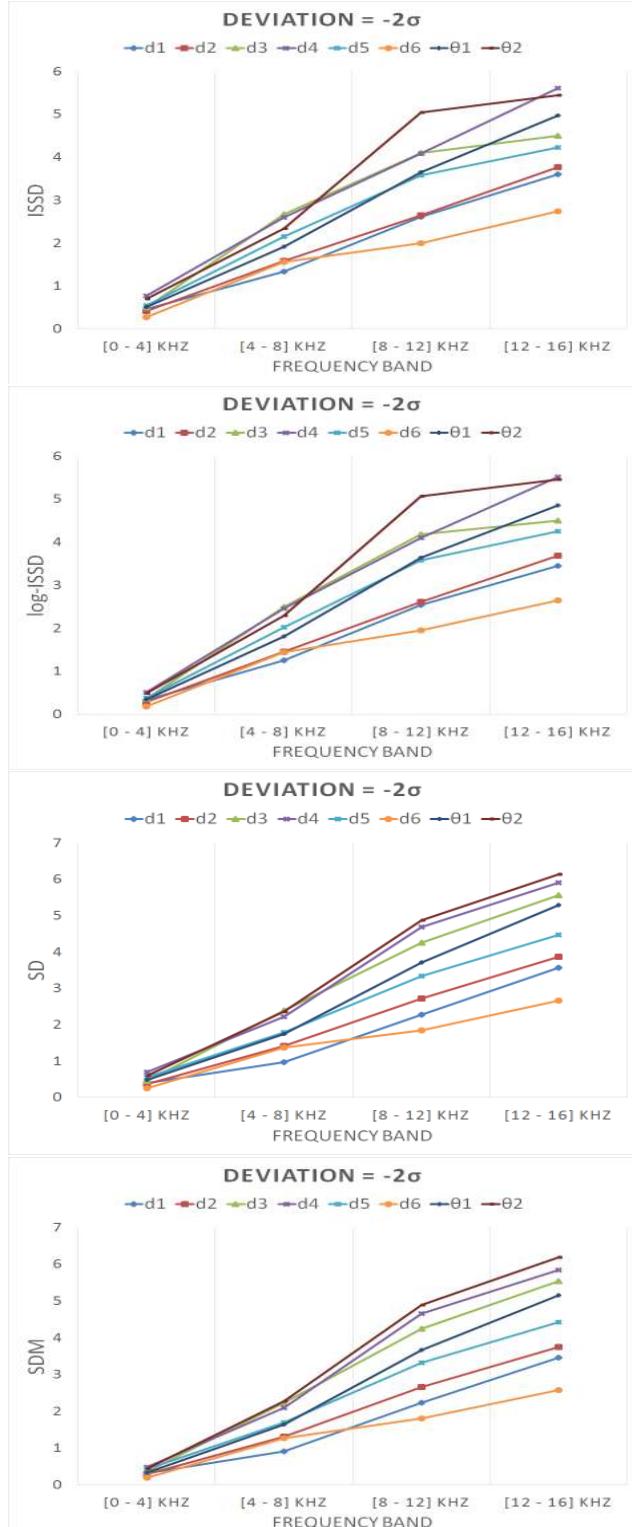


Figure 5: Parameters' influence - deviation = -2σ . From top to bottom, metrics ISSD, log-ISSD, SD and SDM.

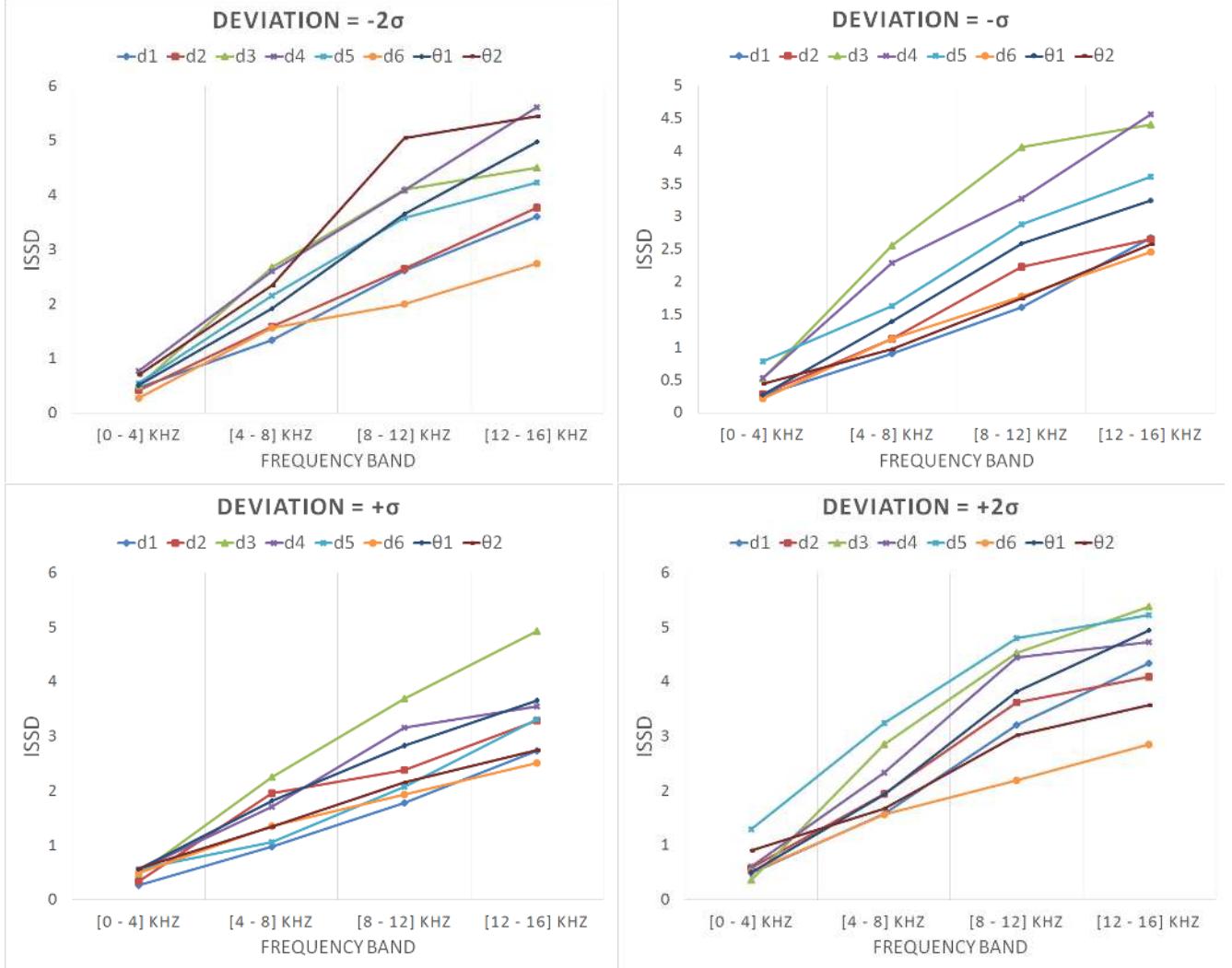


Figure 6: Parameters influence for deviations -2σ (top left), $-\sigma$ (top right), $+\sigma$ (bottom left) and $+2\sigma$ (bottom right).

4. DISCUSSION

The first conclusions we can draw out of these results have to do with the choice of metric. The use of a logarithmic scale does not bring any additional information to what could be extracted from the ISSD and the SD metrics and cannot be seen as a wiser option. On the contrary, its computation forces to interpolate the data, adding computational complexity and inducing a loss of precision. Moreover, the SD and the ISSD reveal similar trends in the data and can be considered here as equivalent. Nevertheless, it must not be forgotten that the 4 retained metrics derive all from HRTF amplitudes differences and are part of the same family of metrics, making the similarity of their behaviour less surprising.

Regarding the ranking previously established, a comparison to the other studies from the community is presented in table 2:

Its analysis leads to the following facts: Xu et al. [6, 7] only retained θ_1 and θ_2 as significant measures. Our results suggest that θ_s can effectively be of a certain importance without being major terms. Such a divergence can partly be explained by the ambi-

tion behind Xu et al.'s studies. More in details, they performed a correlation analysis over the whole set of 27 CIPIC parameters using the CIPIC database, which only contains 45 subjects - among which 10 do not come with complete sets of measures - while the reliability of such a technique strongly depends on the size of the underlying database. Additionally, they only used a very small subset of the available HRTFs: 7 directions and 4 frequencies.

Hu et al. [4, 5] retained d_1, d_3, d_4, d_5 and d_6 as main factors. d_3 is indeed one of our main factor but d_1 and d_6 are not. It is worth noticing that their first regression analysis, performed to select the parameters with large correlations with the DTFs did not retain d_2, θ_1 and θ_2 . This is at least mind confusing if compared to the previous conclusions. It must then be recalled that Xu et al. only used 7 directions and 4 frequencies. However, they also used the CIPIC database and a statistical analysis. Hence, the same remark as the previous one can be done here.

Hugeng et al. [8] retained d_1, d_3, d_5 and d_6 as main factors. As in the previous case, d_3 is indeed one of our main factors but d_1 and d_6 are not. However, their framework being very close to

Table 2: Comparison of our results to the ones of the community.

Authors	Method	Data	Major parameters	Minor parameters
Xu et al.	Correlation analysis	CIPIC database (statistics and HRTFs)	θ_1, θ_2	all others ¹
Hu et al.	Multiple Regressions analysis	CIPIC database (statistics and HRTFs)	d_1, d_3, d_4, d_5, d_6	d_2, θ_1, θ_2
Hugeng et al.	Correlation analysis	CIPIC database (statistics and HRTFs)	d_1, d_3, d_5, d_6	all others ¹
Zhang et al.	Correlation analysis	CIPIC database (statistics and HRTFs)	$d_3, d_4, d_5, d_6, \theta_1, \theta_2$	all others ¹
Fels et al.	Numerical simulations	Own statistics and Numerical HRTFs	d_3, d_8	d_5
this work	Numerical simulations	CIPIC statistics and Numerical HRTFs	d_3, d_4	d_1, d_2, d_6

¹ No intermediate category available in these cases. Parameters could only be significant or not.

the one presented by Xu et al., they are also subject to the same remarks.

Zhang et al. [10] have exhibited 8 parameters, among which 6 describe the pinna shape. Namely, they are $d_3, d_4, d_5, d_6, \theta_1$ and θ_2 . The first 2 are the ones we have seen as the most important here and d_1 and d_2 are not in their selected set. Nevertheless, d_6 is here again presented as a prominent parameter while it completely fails to present this characteristic in our simulations.

Last, Fels et al. [11] retained d_3 as the most important factor as well as d_8 (out of the scope of this study) and rejected d_5 , while it was retained by the 2 previous studies. Here, d_5 appears to be a good example of non-linearities, as $d_5^{+2\sigma}$ proves to have a strong effect whereas $d_5^{+\sigma}$ does not.

As it can be observed, the only clear consensus that can be reached is for d_3 . As it represents the cavum concha width, this conclusion is also coherent with the prior intuition one could have about it.

That being said, another point worthy of interest is the case of parameter d_6 , twice retained as an important parameter, in total disagreement with our observations. In order to investigate it, the original ear meshes are presented in figure 7 hereafter.

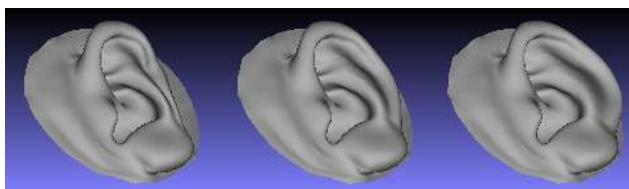


Figure 7: From left to right, $d_6^{-2\sigma}$, the mean ear and $d_6^{+2\sigma}$.

As we can see, the introduced distortions are not visible from the ear pit, where lies the virtual sound source. Hence, the concha operates as a mask, considerably reducing the potential effect of d_6 . An immediate consequence is that another value of θ_2 could have yielded a totally different outcome. This fact reveals in particular that not only the parameters' values are important but so are their combinations, making any statistical analysis more challenging.

5. CONCLUSIONS

In the present work, we have built a database of meshes and computed HRTFs fully dedicated to the study of pinna influence. The anthropometric data were carefully selected to be as relevant as possible. This starts with the use of the CIPIC parameters definition and statistics, known as a reference in the community. Moreover, the choice of the parameters themselves emerges from previous works published in the literature. Furthermore, the anthropometric values as well as the HRTFs generation parameters were set so as to correspond as much as possible to real life problematics and data. In particular, we have covered the whole bandwidth usually said to contain spectral cues. Finally, 4 different metrics have been used to perform the study and to compare the results to previous studies.

Regarding the metrics, it has been shown that they all led to the same conclusions. Thus, one can simply pick and choose the metric that best fits its use case. In our case, retaining the ISSD, parameters d_3 and d_4 showed a stronger effect on the HRTFs than the other ones while d_1 , d_2 and d_6 had, comparatively, much less importance. These conclusions have been confronted to results issued from the community, unveiling a consensus about d_3 .

In addition, it has been observed that non-linearities exist between the CIPIC parameters and the HRTFs. The specific study of d_6 has underscored the need for numerous different ear shapes, i.e. for a bigger database, especially when performing statistical analyses. It also raises the question of the relevance of the parameters choices as introduced by CIPIC, perhaps not perfectly suited for HRTFs analyses, and their definitions, not easily adaptable to 3D data.

Finally, the lack of reachable consensus between the studies aiming at defining a clear set of major parameters also question in general the validity of the studies that present a selection step prior to other treatments (as HRTF individualisation).

However, the current set of data has not delivered all of its information yet. More specifically, future works will investigate the directionality of the impact of each pinna parameter over the HRTFs.

6. REFERENCES

- [1] Naoya Inoue, Toshiyuki Kimura, Takanori Nishino, Katsunobu Itou, and Kazuya Takeda, “Evaluation of hrtfs estimated using physical features,” *Acoustical science and technology*, vol. 26, no. 5, pp. 453–455, 2005.
- [2] DYN Zotkin, Jane Hwang, R Duraiswaini, and Larry S Davis, “Hrtf personalization using anthropometric measurements,” in *Applications of Signal Processing to Audio and Acoustics, 2003 IEEE Workshop on*. Ieee, 2003, pp. 157–160.
- [3] Piotr Bilinski, Jens Ahrens, Mark RP Thomas, Ivan J Tashnev, and John C Platt, “Hrtf magnitude synthesis via sparse representation of anthropometric features,” in *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2014, pp. 4468–4472.
- [4] Hongmei Hu, Lin Zhou, Jie Zhang, Hao Ma, and Zhenyang Wu, “Head related transfer function personalization based on multiple regression analysis,” in *2006 International Conference on Computational Intelligence and Security*. IEEE, 2006, vol. 2, pp. 1829–1832.
- [5] Hongmei Hu, Lin Zhou, Hao Ma, and Zhenyang Wu, “Hrtf personalization based on artificial neural network in individual virtual auditory space,” *Applied Acoustics*, vol. 69, no. 2, pp. 163–172, 2008.
- [6] S Xu, ZZ Li, L Zeng, and G Salvendy, “A study of morphological influence on head-related transfer functions,” in *2007 IEEE International Conference on Industrial Engineering and Engineering Management*. IEEE, 2007, pp. 472–476.
- [7] Song Xu, Zhizhong Li, and Gavriel Salvendy, “Improved method to individualize head-related transfer function using anthropometric measurements,” *Acoustical science and technology*, vol. 29, no. 6, pp. 388–390, 2008.
- [8] W Wahab Hugeng and Dadang Gunawan, “Improved method for individualization of head-related transfer functions on horizontal plane using reduced number of anthropometric measurements,” *arXiv preprint arXiv:1005.5137*, 2010.
- [9] Felipe Grijalva, Luiz Martini, Siome Goldenstein, and Dinei Florencio, “Anthropometric-based customization of head-related transfer functions using isomap in the horizontal plane,” in *Acoustics, Speech and Signal Processing (ICASSP), 2014 IEEE International Conference on*. IEEE, 2014, pp. 4473–4477.
- [10] M Zhang, RA Kennedy, TD Abhayapala, and Wen Zhang, “Statistical method to identify key anthropometric parameters in hrtf individualization,” in *Hands-free Speech Communication and Microphone Arrays (HSCMA), 2011 Joint Workshop on*. IEEE, 2011, pp. 213–218.
- [11] Janina Fels and Michael Vorländer, “Anthropometric parameters influencing head-related transfer functions,” *Acta Acustica united with Acustica*, vol. 95, no. 2, pp. 331–342, 2009.
- [12] Simone Spagnol, Michele Geronazzo, and Federico Avanzini, “Fitting pinna-related transfer functions to anthropometry for binaural sound rendering,” in *Multimedia Signal Processing (MMSP), 2010 IEEE International Workshop on*. IEEE, 2010, pp. 194–199.
- [13] Robert B King and Simon R Oldfield, “The impact of signal bandwidth on auditory localization: Implications for the design of three-dimensional audio displays,” *Human Factors: The Journal of the Human Factors and Ergonomics Society*, vol. 39, no. 2, pp. 287–295, 1997.
- [14] V Ralph Algazi, Richard O Duda, Dennis M Thompson, and Carlos Avendano, “The cipic hrtf database,” in *Applications of Signal Processing to Audio and Acoustics, 2001 IEEE Workshop on the*. IEEE, 2001, pp. 99–102.
- [15] V Ralph Algazi, Richard O Duda, Ramani Duraiswami, Nail A Gumerov, and Zhihui Tang, “Approximating the head-related transfer function using simple geometric models of the head and torso,” *The Journal of the Acoustical Society of America*, vol. 112, no. 5, pp. 2053–2064, 2002.
- [16] Harald Ziegelwanger, Wolfgang Kreuzer, and Piotr Majdak, “Mesh2hrtf: Open-source software package for the numerical calculation of head-related transfer functions,” in *22st International Congress on Sound and Vibration*, 2015.
- [17] Harald Ziegelwanger, Piotr Majdak, and Wolfgang Kreuzer, “Numerical calculation of listener-specific head-related transfer functions and sound localization: Microphone model and mesh discretization,” *The Journal of the Acoustical Society of America*, vol. 138, no. 1, pp. 208–222, 2015.
- [18] Jyri Huopaniemi and Matti Karjalainen, “Review of digital filter design and implementation methods for 3-d sound,” in *Audio Engineering Society Convention 102*. Audio Engineering Society, 1997.
- [19] John C Middlebrooks, “Individual differences in external-ear transfer functions reduced by scaling in frequency,” *The Journal of the Acoustical Society of America*, vol. 106, no. 3, pp. 1480–1492, 1999.
- [20] F Rugeles, M Emerit, and B Katz, “Évaluation objective et subjective de différentes méthodes de lissage des hrtf,” in *Cong Français d’Acoustique (CFA)*. CFA, 2014, pp. 2213–2219.

2D SPATIAL AUDIO IN A MOLECULAR NAVIGATOR/EDITOR FOR BLIND AND VISUALLY IMPAIRED USERS

Ian Rodrigues, Ricardo Teixeira, Sofia Cavaco*	Vasco D. B. Bonifácio	Daniela Peixoto, Yuri Binev, Florbela Pereira, Ana M. Lobo, João Aires-de-Sousa*
NOVA LINCS Department of Computer Science Faculty of Science and Technology Universidade Nova de Lisboa 2829-516 Caparica, Portugal in.rodrigues@campus.fct.unl.pt, scavaco@fct.unl.pt	Centro de Química-Física Molecular and Institute of Nanoscience and Nanotechnology Instituto Superior Técnico Universidade de Lisboa Av. Rovisco Pais 1049-001 Lisboa, Portugal vasco.bonifacio@tecnico.ulisboa.pt	LAQV-REQUIMTE Department of Chemistry Faculty of Science and Technology Universidade Nova de Lisboa 2829-516 Caparica, Portugal {d.peixoto,y.binev, mf.pereira, aml, jas}@fct.unl.pt

ABSTRACT

In order to contribute to the access of blind and visually impaired (BVI) people to the study of chemistry, we are developing Navmol, an application that helps BVI chemistry students to interpret molecular structures. This application uses sound to transmit the information about the structure of the molecules. Navmol uses voice synthesis and describes the molecules using the clock polar type coordinates. In order to help the users to mentally conceptualize the molecular structure representations more easily, we propose to use 2D spatial audio. This way, the audio signal generated by the application gives the user the perception of sound originating from the directions of the bonds between the atoms in the molecules. The sound spatialization is obtained with head related transfer functions. The results of a usability study show that the combination of spatial audio with the description of the molecules using the clock reference system helps BVI users to understand the molecules' structure.

1. INTRODUCTION

BVI people have difficulty in achieving a higher education degree due to the lack of material adapted to their special needs. Moreover, the few students who reach higher education, are influenced to pursue a degree in humanities or music and to avoid science and engineering degrees. According to the 2011 census, there were 16 500 BVI people in Portugal, of which 42 000 were school-age people (5-29 years old) [1]. However, only 10 000 are referenced as students. In 2001 the percentage of BVI people in Portugal who completed a higher education degree was 0.9% while in the general population this number increases to 10.8% [2, 3]. In the European Union in 2002 the percentage of people without disabilities with 25-64 years of age who completed a university degree was 22%, but this number decreases to 10% when we consider people with disabilities [4]. The same study in Portugal concluded that 17% of the people without disabilities completed a university degree while only 1% of the people with disabilities completed a university degree.

The representation of molecular structures is a major challenge for the accessibility of blind students and professionals to chemistry and other science degrees that include chemistry courses. This is particularly true in organic chemistry that critically requires the interpretation and transmission of molecular structures - and these are commonly processed visually, by means of chemical drawings.

Only a few applications that tackle this issue have been proposed in the literature. The AsteriX-BVI web server resorts to tactile information, such as one using 3D printing techniques to construct tactile representations of molecules with Braille annotations [5], or another that functions as a molecular editor that uses 2D sketches and static Braille reports [6]. While this solution may help blind users to construct a mental representation of the molecules, it may not be accessible to everyone.

In order to contribute to the access of BVI people to higher education, we are developing tools adapted to their special needs that can be used to study chemistry courses. These consist of chemoinformatics strategies to teach chemistry to BVI users [7]. In particular, here we focus on the structure of the molecules. These strategies rely on computer representations of molecular structures not only as drawings, but as graphs, specifying the atoms in a molecule, their bond connections, and their 3D configuration [8].

Here we propose a solution based on spatial audio implemented with head related transfer functions (HRTFs) to transmit information about the structure of molecules to BVI students and chemists. This solution has the advantage of being able to reach more BVI users than previously proposed applications.

With the intention to facilitate the integration of blind students in the study of chemistry, we are developing Navmol, a navigator and editor of molecular structures designed for BVI students and chemists that uses voice synthesis and other accessibility tools familiar to BVI users (text-to-Braille refreshable displays and the keyboard) [9]. Furthermore, it contains a graphical interface that replicates the same information that is heard so that sighted users can better communicate with BVI users, helping with their integration for example in a classroom (figure 1). Along with these features, we propose to use spatial audio to help BVI users to more easily understand the structure of the molecules.

The main novelty of this work is the use of spatial audio to transmit information about the structure of the molecules to BVI users. This work can have a real impact in the lives of people in a special needs group that is typically under-served, contributing to their access to higher education in science and engineering degrees.

Below we discuss the sound spatialization technique used in Navmol, and the results from a usability test performed with a group of chemists, a group of non-chemistry university students and a group of BVI volunteers. The results from all groups show that the users can successfully reconstruct the molecular structure of the molecules presented to them when Navmol describes the



Figure 1: Two BVI students using Navmol.

structure of the molecules with spatial audio. Moreover, the comments from BVI users also indicate that the sound spatialization improves Navmol, that is, with spatial audio it is easier and faster to grasp the structure of the molecules.

2. NAVMOL

Navmol¹ is a software application that provides BVI people with the possibility of navigating, interpreting and editing the representation of molecular structures in a plane (figure 2). For the communication of molecular structures to BVI users, Navmol uses the clock reference system, which is a polar coordinate system based on the representation of an analogue clock making an analogy between the hours of the clock and their corresponding directions. The clock reference system is widely used by BVI people in many situations, including during meals as a way to know where the food is located in the plate [10].

After selecting a certain atom in a molecule, the user gets information about the clock directions of all the neighbor atoms (that are bonded to the selected atom, except hydrogens which are omitted). For instance, the sentence “atom C 2 at 9 o’clock through a double bond” means that, in the loaded molecular representation (figure 2), there is a C 2 carbon atom to the left of the carbon atom C 1 that the user is currently selecting. Navmol uses a speech synthesizer to transmit this information.

In order to give the users more cues that help them to mentally conceptualize representations of the molecular structures, Navmol modifies the speech synthesizer signal with HRTFs so that the speech is spatialized. Provided the user wears headphones, the speech can be heard as if coming from different analogue clock directions.

Spatial audio has been used before in computer games to give a sense of immersion to the users. Other applications that use sound spatialization with HRTFs have been successfully used by BVI users. *Demor* [11] and *AudioSpaceStation* [12] are two such examples. These are games that are solely based on spatial audio sound to give BVI users the feeling of immersion in a virtual 3D

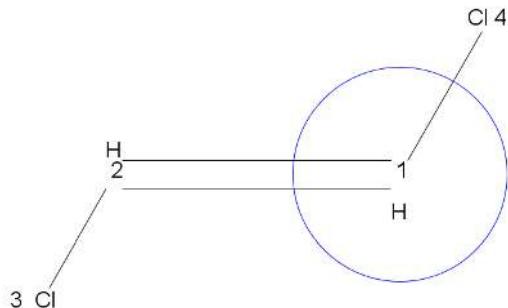


Figure 2: Navmol’s graphic representation of *trans*-1,2-dichloroethene. The circle marks the selected carbon atom C 1.

world. The novelty of this work is to use spatial audio in an application whose purpose is to contribute to the integration of BVI users in chemistry courses.

Navmol is implemented in Java and uses the Chemistry Development Kit (CDK) library and the FreeTTS text-to-voice synthesizer. For the spatial audio it uses the JOAL library [13].

3. SPATIALIZED SOUND

When a sound is emitted, it suffers many modifications before it reaches our ears. Some of these modifications are caused by our own body, namely by the head, torso and pinnae. In particular, when sound travels through the pinnae it is reflected in different ways depending on the orientation of the head relative to the sound source. In other words, a series of reflected waves are generated due to the shape of the pinnae. The phase differences between the direct and reflected waves depend on the sound source’s location [14]. As a result, the intensity of the sound’s frequency components change, that is, the pinnae have a directional filtering effect. These spectral differences as well as the delays of the reflected waves are some of the cues used by the brain to perform sound localization.

Synthesized spatial audio can be generated by changing the right and left channel signal to simulate the pinnae filtering effect. This can be performed with HRTFs, which are frequency fil-

¹Navmol is available at <https://sourceforge.net/projects/navmol/files/Navmol-SoundSpatialization-beta.zip/download>.

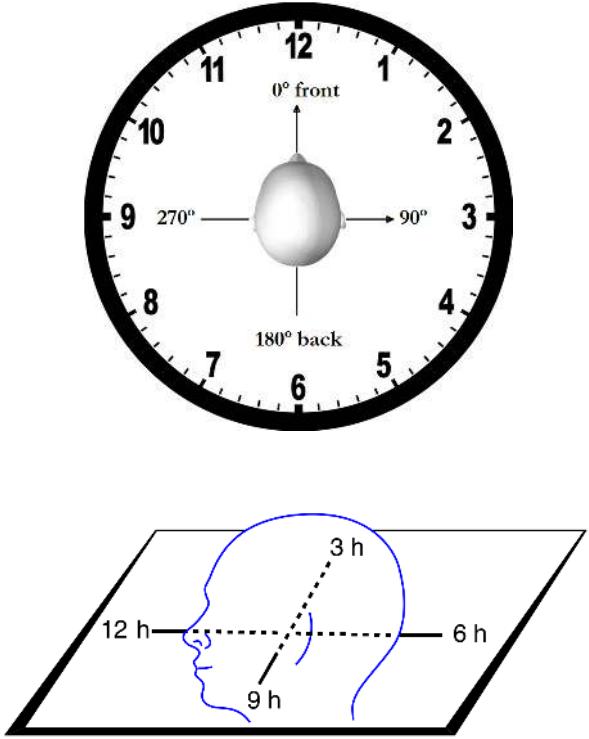


Figure 3: Representation of the head (facing 0° in the azimuth) and the direction of sound according to an analog clock. From above (top) and from a side perspective (bottom).

ters that change the signals according to the effects that the head, torso and pinnae have on the sounds. These filters depend on the source's azimuth, elevation and distance relative to the listener [15].

Navmol modifies the voice synthesizer signal with HRTFs. Instead of generating a mono audio signal with the description of the atom's neighbors, Navmol generates two signals (for the left and right channel) that when heard through headphones give the perception of spatial audio. Here we are using 2D sound, with the sound's direction varying in the plane perpendicular to the head, cutting across the nose and ears, and with 0° ahead of the listener as shown in figure 3.

As an example, let us consider the molecule in figure 2, which has a planar structure, with the center of all its atoms occupying positions in the same plane. When the users are positioned at atom C 1, Navmol indicates that the atom has two neighbors: "atom C 1 at 1 o'clock through a single bond" and "atom C 2 at 9 o'clock through a double bond" (carbon atoms are represented only by the number). If the users wear headphones, they will be immersed in a virtual scene in which they are centered at the atom with the circle (C 1 atom) and will hear those sentences as if coming from 30° and 270° in the horizontal plane, respectively.

4. HEAD RELATED TRANSFER FUNCTIONS SETS

HRTFs can be obtained by recording sounds with microphones inside the ears of dummy heads with pinnae. Different pinnae cause slightly different effects on the sounds as their filtering effect depends on their actual shape. Thus, HRTFs depend on the pinnae used to do the recordings.

As a result, when sound is modified with a given set of HRTFs, not everyone perceives the spatialization in exactly the same manner. Therefore, the set of HRTFs that should be used depends on the listener. It should be the one that approximates better the filtering of the listener's pinnae.

When users start using Navmol, they need to choose the set of HRTFs that works best for them. To make this task easy, Navmol has an accompanying application, the *HRTFs-test-app*, that helps users choose the set of HRTFs. When the users run this application, they will hear the sentence "atom at X o'clock" several times and coming from random directions. The users must introduce the direction from where they think the sound is coming using the clock reference system and then the application calculates the error. The users can run the application with all available sets of HRTFs and choose the one for which they obtained the lowest error.

Many sets of HRTFs exist. We used 53 sets: one set from KEMAR [16], one from CIAIR [17] and 51 from IRCAM [18]. The sets were compiled from the initial HRTFs wave files to the *mhr* format, so as to be in accordance with the OpenAl-Soft implementation of the HRTFs done by JOAL, while also drastically reducing the size of the HRTF measurements.

While Navmol's installation package includes all the 53 sets of HRTFs, it is not convenient for the users to test all the sets. In order to reduce the number of suggested sets to the user, we tried all 53 sets with five volunteers and chose the five sets that worked better for them all, that is, the five best sets for all volunteers. Then we ran a small test with 10 volunteers and these five sets to determine if it was necessary to include more than one HRTFs set in *HRTFs-test-app* sets' suggestion. This procedure is described in section **The HRTFs sets test**.

Following the results of this procedure, Navmol's package suggests the following sets of HRTFs: KEMAR, CIAIR, IRC05, IRC25 and IRC44. The users can run the *HRTFs-test-app* with these five sets and choose the one that works better for them. Afterwards, they can install Navmol with that set of HRTFs and start using it with spatial audio. (If the users do not feel satisfied with these sets, they can run the application with any other set.)

4.1. The HRTFs sets test

The goal of this informal test was to reduce the number of HRTFs sets suggested to the users, so as to make the task of choosing a set of HRTFs easy and not tiring. All the tests described here were made using the same Asus laptop running a 64-bit version of Windows 8 and a Sennheiser HD 202 headset.

We tried the 53 sets of HRTFs in random order with five volunteers. Each volunteer heard the sentence "atom at t o'clock", with t varying from 1 to 12. The sentence was spatialized so as to be heard from the corresponding direction. For instance, if the sentence was "atom at 3 o'clock", the sound's direction was 90° (figure 3).

The volunteers heard this sentence in a clockwise direction twice. The sound would start at 0° in front of the user (12 o'clock)

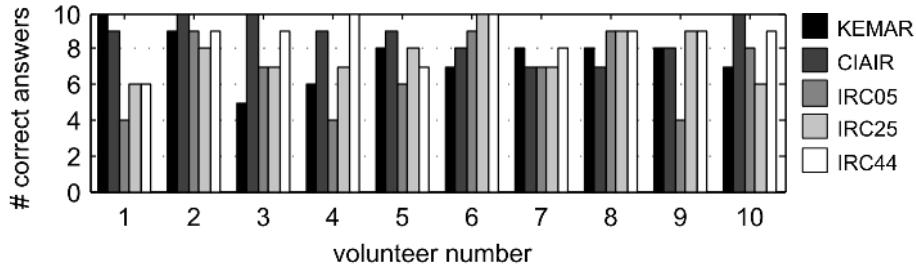


Figure 4: Number of correct answers for each HRTFs set.

and would then proceed for each of the distinct directions, 1 o’clock, 2 o’clock, etc. all the way to 12 o’clock again. The volunteers would then classify the sound spatialization from 1 to 4, with 1 being “I could not determine at all the sound’s directions” and 4 being “I could hear all the directions perfectly”.

Afterwards, the volunteers heard the sentence again but coming from 10 random directions and we asked them to identify the direction of the sound (in terms of hours in an analog clock). The correct answers were counted for each of the 53 sets of HRTFs.

As expected, even though all HRTFs sets had positive scores, different volunteers gave different scores to each HRTFs set. The five best distinct HRTFs for all volunteers were chosen. These were the CIAIR, KEMAR, IRC05, IRC25 and IRC44.

Only five volunteers participated in this procedure because the tasks described took many hours for each person. Afterwards we performed another test with 10 volunteers but using only the five HRTF sets mentioned above. Again, the order of the HRTFs sets was random.

For this second test, we used the sentence “atom at X o’clock” (without replacing X by 1, 2, ..., 12). This test started with a training phase during which the volunteers heard that sentence in clockwise order (for two complete turns) and the direction of the sound varying accordingly.

The main task of the test was performed after the training period. The volunteers heard the sentence 10 times (from random directions) and were asked to identify the direction of the sound. Since humans have a localization accuracy that can diverge up to a few degrees from the correct direction, we considered correct all answers that were displaced by less than 30° to either side of the correct direction (for instance, if the correct direction was 3 o’clock, we considered 2, 3 and 4 o’clock correct and all other answers incorrect) [19].

Figure 4 shows the number of correct answers for each of the five HRTFs set. As it can be observed, while there is at least one set of HRTFs for each subject with more than eight correct answers, as expected the set with the highest number of correct answers varies from subject to subject.

Following these results, Navmol’s installation package includes many sets of HRTFs so that the users can use the set they feel the most comfortable with, without having the need of searching and compiling the measurements by themselves.

5. RESULTS

In order to assess if users can understand the molecules’ structure when Navmol uses sound spatialization, we ran a usability test in which the volunteers had to interpret two molecules using Navmol. Due to the difficulty in having a high number of BVI volunteers to test Navmol, before running the test with BVI people, we ran it with sighted volunteers. This way we could guarantee that the expected outcome was obtained and there were no corrections or modifications to the software that had to be made before running the test with the BVI volunteers.

Before the actual test started, the participants tried the five HRTFs sets to determine which worked better for them. They heard the sentence “atom at X o’clock” from 10 random directions (where X was not replaced by the time) and were asked to estimate the sound’s direction. The HRTFs set with a minimal number of wrong answers (using the same tolerance as above) was chosen and the rest of the test was performed with that set.

After the HRTFs set was chosen, there was a training phase. During training the participants heard the same sentence again but where X o’clock was substituted by the matching direction (1 to 12 o’clock). They heard the sentence for all 12 directions in clockwise order and for two full circles. Then they heard the same sentence again from 10 random directions but without information on the actual direction (*i.e.*, X was not replaced by the corresponding time in hours). They were asked to estimate the sound’s direction and received feedback on their answers, that is, the training application spoke back the correct direction of the spatialized sound.

After the training period, the main task of the test started. This consisted of exploring two molecules with Navmol (figure 5) and reproducing their structure, which for sighted users consisted of drawing them in paper. Before starting the task, the participants were shown a molecule in Navmol’s window and a demonstration on how to explore it. They tried Navmol by themselves so that they could familiarize with the controls, interface and the synthesized

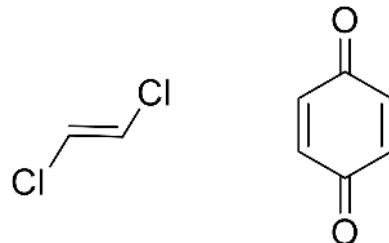


Figure 5: Molecules used in the first test.

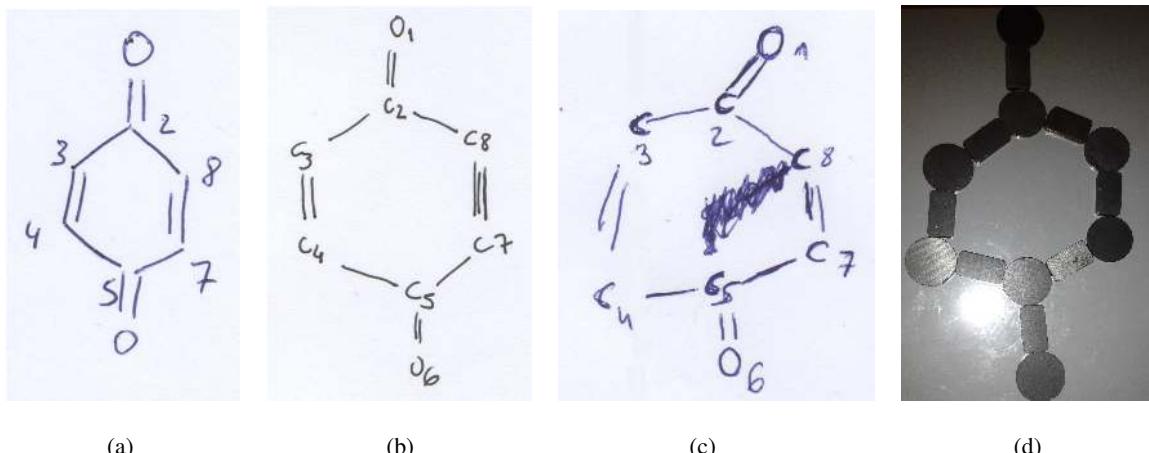


Figure 6: Answers from the usability test. (a)-(c) Answers from three sighted participants. (d) Answer from a blind participant.

voice. After they felt confident on how to use Navmol, the task started. For this, the participants had no access to the screen. They could only use the keyboard to explore the molecule and hear its information.

The point of this task was to see if the participants could understand and reproduce the complete molecules structure from just the spatialized sound cues. All the sentences with the molecule name or constitution were omitted and only the atom bonds were left intact but with the omission of the direction of the bond. For instance, if the sentence for Navmol's normal version was "atom O 2 at 6 o'clock through a double bond" the sentence in the task would only say "atom O 2 at X o'clock through a double bond". This task had two distinct molecules that were given to the participants in a random order.

5.1. Results from sighted volunteers

We ran the test with 10 chemist volunteers and 15 non-chemist volunteers. The chemist volunteers had ages between 22 and 56 (8 women and 2 men, 6 PhD, 3 MSc and 1 BSc). Two of them reported having some hearing problems. The non-chemist volunteers were university students with ages between 18 and 27 and all with normal hearing. There were 10 men and 5 women. They all wore headphones to perform the test and used an external keyboard (so that they did not face the laptop screen when Navmol was running).

It was observed that using only the sound spatialization cues, the participants were able to navigate and understand the molecules of this study. All participants were able to draw the two molecules with only minor errors: on average there were 0.5 errors out of 12 possibilities (12 atoms), being that the chemist participants had an average of 0.9 errors and the other group had an average of 0.3 errors out of the possible 12. The typical errors were a missing bond in a molecule or the representation of a bond not being in the exact correct direction. Figure 6.a-c shows three examples of typical answers from this test (all representing the same molecule). The leftmost and center answers are completely correct. The rightmost answer is also correct but it has only some small orientation errors: the top oxygen atom should have been drawn right on top of the C 2 atom, like in the other two answers, and the C 4 atom has also a small orientation error.

After having drawn the complete molecules or most of the atoms and bonds, most of the chemist volunteers recognized the molecules in question, being that some of them reported that they used this knowledge to finalize drawing them. While this knowledge could have helped them to perform the task, the answers from the non-chemist volunteers show that people who do not have this type of knowledge can also draw the molecules. Whilst the non-chemist volunteers did not recognize the specific molecules of the test (as expected since they did not have the same chemistry background as the first group), they could grasp their structures correctly. This shows that Navmol's sound spatialization can successfully transmit the molecular structure information to users.

As reported above, the chemist volunteers had a slight higher error average than non-chemists. Some of the people from the chemist group recognized the molecules and used this knowledge to help them perform the task. It appears that this can have led them to make more "mistakes" than the people in the other group because reproducing a molecular structure from a chemist's point of view is not necessarily reproducing a drawing. Such a molecular structure is chemically defined by the elements of the atoms, their connectivity and the bond orders, independently of the orientation of the molecule.

5.2. Results from BVI volunteers

In spite of our efforts, we only managed to have four BVI volunteers for this test: two blind male participants and two low vision female participants. The blind participants were 54 and 67 years old. The youngest was blind since he was 2.5 years old, and the oldest was blind from birth. The low vision participants were 22 and 46 years old. The youngest was a university student, who was visually impaired since she was 7 years old, and the oldest had low vision from birth. The three older participants had a bachelor's degree.

The same protocol as described above was used but, due to the vision impairments of the volunteers, some adjustments were made. The instructions were printed both in Braille (for blind volunteers) and with size 16 font (for low vision volunteers). We also read the instructions to them if they preferred it this way.

While the low vision participants were able to draw the molecules, the blind participants used another technique. We provided the



Figure 7: Metal board and magnets for answers from blind participants.

blind volunteers with a set of magnets that they could arrange in a metal board to replicate the molecules' structure (figure 7). There was a set of circular magnets (representing atoms) and rectangular magnets (representing connections between atoms).

All four BVI volunteers correctly represented the two molecules without any errors. Figure 6.d shows the answer from one of the blind volunteers to the same molecule as that of figure 6.a-c. This answer is correct and, as it can be observed, represents the same structure as the drawings in figure 6.a-c. This shows that Navmol's sound spatialization is appropriate to transmit the molecular structure information to BVI users.

Having said that, there is still another question about the sound spatialization that we have not mentioned: is Navmol better suited to transmit the molecular structure information to BVI users with or without spatial audio? While the users can choose to use Navmol with or without spatial audio, according to the youngest low vision volunteer it is easier to interpret the molecules' structure with spatial audio. This volunteer had participated in a previous Navmol test with no spatial audio [9]. As such, she said that the sound spatialization helped very much with the mental conceptualization of the molecules. She referred that with the previous Navmol version, the users had to wait until the end of the sentence that described the atom ("atom ... at ... o'clock...") to know where the atom was positioned. Now, with the sound spatialization, she could identify the direction where the atom was located right from the beginning of the sentence. She mentioned that this helped her immensely to build the correct mental representation of the structure of the molecules.

6. CONCLUSIONS

While spatialized sound has been used before in computer games, here we propose to use it in an application that can make a difference in the lives of BVI students. In order to help these special needs users to understand the molecular structures, we are developing Navmol, a molecular navigator/editor for BVI users that describes molecular structures with synthesized voice. As a contribution to help these users to more easily conceive a mental representation of the structures, we propose to use spatial audio. Thus, we modified the speech synthesizer's signal with HRTFs such that when users hear the molecule description they will immerse in a virtual scene where they stand on the atoms of the molecule and hear the information coming from different angles of the azimuth,

which depend on the directions of the bonds in the molecule.

As observed in the usability test described above, the introduction of spatialized sound to Navmol is a positive addition to its usability, being that by itself, the sound spatialization is enough for the user to realize the correct and complete perception of planar molecular structures. A BVI volunteer who had experimented Navmol with no sound spatialization observed that when the spatial audio is used it is easier to understand the molecules' structure.

The main novelty of this work is the use of spatial audio in conjunction with a polar coordinate system based on the representation of an analogue clock to transmit information about the structure of molecules to BVI people. As a result, this application can have impact in the lives of real people who are typically under-served.

7. ACKNOWLEDGMENTS

We thank all the volunteers who participated in the tests. Special thanks to the team from ACAPO and from Biblioteca Nacional de Portugal, namely to Carlos Ferreira and Maria Aldeguedes, for all the help with the tests with BVI subjects.

This work was funded by Fundação Calouste Gulbenkian under project "Ciência para todos: STEREO+", and Fundação para a Ciência e a Tecnologia through LAQV, REQUIMTE: UID/QUI/50006/2013, and NOVA-LINCS: PEest/UID/CEC/04516/2013.

8. REFERENCES

- [1] *INE Resultados Definitivos Censos 2011*, Instituto Nacional de Estatística, Lisboa, Portugal, 2012.
- [2] *SNR Inquérito Nacional às Incapacidades, Deficiências e Desvantagens - Resultados Globais*, Secretariado Nacional de Reabilitação, Caderno n.8., Retrieved Sep/2008 from <http://www.inr.pt/content/1/117/informacao-estatistica>.
- [3] *INE Resultados Definitivos Censos 2001*, Instituto Nacional de Estatística, Lisboa, Portugal, 2002.
- [4] "APPLICA & CESEP & ALPHAMETRICS, men and women with disabilities in the EU: Statistical analysis of the LFS AD HOC module and the EU-SILC, final report," 2007.
- [5] V. Lounnas, H. Wedler, T. Newman, G. Schaftenaar, J. Harrison, G. Nepomuceno, R. Pemberton, D. Tantillo, and G. Vriend, "Visually impaired researchers get their hands on quantum chemistry: application to a computational study on the isomerization of a sterol," *J. Comput. Aid. Mol. Des.*, vol. 28, no. 11, pp. 1057–1067, 2014.
- [6] V. Lounnas, H. Wedler, T. Newman, J. Black, and G. Vriend, "Blind and Visually Impaired Students Can Perform Computer-Aided Molecular Design with an Assistive Molecular Fabricator," *Lecture Notes in Computer Science*, vol. 9043, pp. 18–29, 2015.
- [7] "MOLinsight web portal," Retrieved Apr/2017 from <http://molinsight.net>.
- [8] F. Pereira, J. Aires-de Sousa, V.D.B. Bonifácio, P. Mata, and A. M. Lobo, "MOLinsight: A Web Portal for the Processing of Molecular Structures by Blind Students," *Journal of Chemical Education*, vol. 88, pp. 361–362, 2011.

- [9] R. Fartaria, F. Pereira, V.D.B. Bonifácio, P. Mata, J. Aires-de Sousa, and A. Lobo, “NavMol 2.0 - A Molecular Structure Navigator/Editor for Blind and Visually Impaired Users,” *European Journal of Organic Chemistry*, vol. 8, pp. 1415–1419, 2013.
- [10] L.C. Widerberg and R. Kaarlela, *Techniques for eating, a guide for blind persons*, Western Michigan University, 1981.
- [11] “Demor, location based 3D audiogame,” Retrieved Apr/2017 from <http://blendid.nl/index8803.html>.
- [12] D. Simões and S. Cavaco, “An orientation game with 3D spatialized audio for visually impaired children,” in *Proceedings of Advances in Computer Entertainment Technology Conference*, 2014.
- [13] “Java Bindings for the OpenAL API,” Retrieved Apr/2017 from <https://jogamp.org/joal/www/>.
- [14] S. Handel, *Listening: An Introduction to the Perception of Auditory Events*, The MIT Press, 1989.
- [15] J. Blauert, *Spatial Hearing*, The MIT Press, 1983.
- [16] B. Gardner and K. Martin, “HRTF measurements of a KEMAR dummy-head microphone,” Tech. Rep., MIT Media Lab Perceptual Computing, 1994.
- [17] T. Nishino, S. Kajita, K. Takeda, and F. Itakura, “Interpolation of head related transfer functions of azimuth and elevation,” *J. Acoust. Soc. Jpn.*, vol. 57, no. 11, pp. 685–692, 2001.
- [18] “Listen HRTF Database,” Retrieved Apr/2017 from <http://recherche.ircam.fr/equipes/salles/listen/index.html>.
- [19] D. Wang and Eds. Brown, G. J., *Computational auditory scene analysis: principles, algorithms and applications*, IEEE Press/Wiley-Interscience, 2006.

PERFORMANCE PORTABILITY FOR ROOM ACOUSTICS SIMULATIONS

Larisa Stoltzfus

School of Informatics
The University of Edinburgh
Edinburgh, UK
larisa.stoltzfus@ed.ac.uk

Alan Gray

EPCC
The University of Edinburgh,
Edinburgh, UK
a.gray@epcc.ed.ac.uk

Christophe Dubach

School of Informatics
The University of Edinburgh
Edinburgh, UK
christophe.dubach@ed.ac.uk

Stefan Bilbao

Acoustics and Audio Group,
The University of Edinburgh
Edinburgh, UK
s.bilbao@ed.ac.uk

ABSTRACT

Numerical modelling of the 3-D wave equation can result in very accurate virtual auralisation, at the expense of computational cost. Implementations targeting modern highly-parallel processors such as NVIDIA GPUs (Graphics Processing Units) are known to be very effective, but are tied to the specific hardware for which they are developed. In this paper, we investigate extending the portability of these models to a wider range of architectures without the loss of performance. We show that, through development of portable frameworks, we can achieve acoustic simulation software that can target other devices in addition to NVIDIA GPUs, such as AMD GPUs, Intel Xeon Phi many-core CPUs and traditional Intel multi-core CPUs. The memory bandwidth offered by each architecture is key to achievable performance, and as such we observe high performance on AMD as well as NVIDIA GPUs (where high performance is achievable even on consumer-class variants despite their lower floating point capability), whilst retaining portability to the other less-performant architectures.

1. INTRODUCTION

The finite difference time domain method (FDTD) is a well-known numerical approach for acoustic modelling of the 3D wave equation [1]. Space is discretised into a three-dimensional grid of points, with data values resident at each point representing the acoustic field at that point. The state of the system evolves through time-stepping: the value at each point is repeatedly updated using *finite differences* of that point in time and space. The so-called *stencil* of points involved in each update is determined by the choice of discretisation scheme for the partial differential operators in the wave equation [2]. This numerical approach is relatively computationally expensive, but amenable to parallelisation. In recent years, there has been good progress in the development of techniques to exploit modern parallel hardware. In particular, NVIDIA GPUs have proven to be a very powerful platform for acoustic modelling [3][4]. Most work in this area has involved writing code using the NVIDIA specific CUDA language, which is tied to this platform. Ideally, however, any software should be able to run in a portable manner across different architectures, such that the performance of alternatives can be explored, and different resources can be exploited both as and when they become available. It is non-trivial,

however, to develop portable application source code that can perform well across different architectures: this issue of *performance portability* is currently of great interest in the general field of High Performance Computing (HPC) [5].

In this paper, we explore the performance portability issue for FDTD numerical modelling of the 3D wave equation. To enable this study, we have developed different implementations of the simulation, each performing the same task, including a CUDA implementation that acts as a “baseline” for use on NVIDIA GPUs, plus other more portable alternatives. This enables us to assess performance across multiple different hardware solutions: NVIDIA GPUs, AMD GPUs, Intel Xeon Phi many-core CPUs and traditional multi-core CPUs. The work also includes the development of a simple, adjustable abstraction framework, where the flexibility comes through the use of templates and macros (for outlining and substituting code fragments) to facilitate different implementation and optimisation choices for a room acoustics simulation. Both basic and advanced versions of an FDTD algorithm that simulates sound propagation in a room (i.e. a cuboid) are explored.

This paper is structured as follows: in Section 2, we give necessary background information on the computational scheme, the hardware architectures under study and the associated programming models; in Section 3, we describe the development of a performant, portable and productive room acoustics simulation framework; in Section 4 we outline the experimental setup for investigating different programming approaches and assessing performance; in Section 5 we present and analyse performance results; and finally we summarise and discuss future work in Section 6.

2. BACKGROUND

This paper is focused on assessing different software implementations of a room acoustics simulation across different types of computing hardware. While this project focuses strictly on single node development, the ideas in this work could easily be extended for use across multi-node platforms by coupling with a message-passing framework. In this section we give some necessary background details. We first describe the FDTD scheme used in this study. We then give details on the hardware architectures that we wish to assess and the native programming methods for these architectures. Finally we describe some pre-existing parallel programming frameworks that provide portability.

2.1. FDTD Room Acoustics Scheme

Wave-based numerical simulation techniques, such as FDTD, are concerned with deriving algorithms for the numerical simulation of the 3D wave equation:

$$\frac{\partial^2 \Psi}{\partial t^2} = c^2 \nabla^2 \Psi \quad (1)$$

Here, $\Psi(\mathbf{x}, t)$ is the dependent variable to be solved for (representing an acoustic velocity potential, from which pressure and velocity may be derived), as a function of spatial coordinate $\mathbf{x} \in \mathcal{D} \in \mathbb{R}^3$ and time $t \in \mathbb{R}^+$.

In standard finite difference time domain (FDTD) constructions, the solution Ψ is approximated by a grid function $\Psi_{l,m,p}^n$, representing an approximation to $\Psi(\mathbf{x} = (lh, mh, ph), t = nk)$, where l, m, p, n are integers, h is a grid spacing in metres, and k is a time step in seconds. The FDTD scheme used for this work is given in [6], and is the standard scheme with a seven-point Laplacian stencil. According to this scheme, updates are calculated as follows:

$$\Psi_{l,m,p}^{n+1} = (2 - 6\lambda^2)\Psi_{l,m,p}^n + \lambda^2 S - \Psi_{l,m,p}^{n-1} \quad (2)$$

where

$$S = \Psi_{l+1,m,p}^n + \Psi_{l-1,m,p}^n + \Psi_{l,m+1,p}^n + \Psi_{l,m-1,p}^n + \Psi_{l,m,p+1}^n + \Psi_{l,m,p-1}^n, \quad (3)$$

The constant $\lambda = ck/h$ is referred to as the Courant number and must satisfy the stability condition $\lambda \leq 1/\sqrt{3}$. It can be seen that each grid value is updated based on a combination of two previous values at the same location, and contributions from each of the six neighbouring points in three dimensions (giving the six terms in S - see Figure 1). The benchmarks used in this study are run over

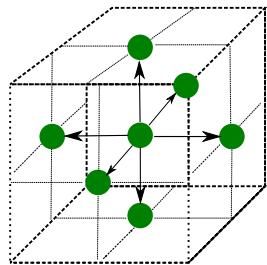


Figure 1: Figure of a 7-point stencil on a three dimensional grid.

13–105 million grid points at a sample rate of 44.1kHz to produce 100ms of sound. The boundary conditions used are frequency-independent impedance boundary conditions. We also include, towards the end of the paper, some results for comparison where a larger stencil is used, as described in [3][7].

2.2. Hardware and Native Programming Methods

The hardware architectures used in this work are multi-core CPU, GPU, and many-core CPU (Intel Xeon Phi) platforms. CPUs have traditionally been the main workhorses for scientific computing and are still targeted for the majority of scientific applications. GPUs have been gaining traction in the scientific computing landscape over the last decade, and can offer significant performance

improvements over traditional CPUs for certain algorithms including wave-based ones like acoustic simulation. Xeon Phi chips are essentially multi-core CPUs, but with a large number of cores each with lower clock speeds and wider vector units. In this section, we describe these architectures in more detail and discuss how they are typically programmed.

2.2.1. Traditional Multi-Core CPUs

Computational simulations (like acoustic models) have historically been run on CPU chips. However, these architectures were originally optimised for sequential codes, whereas scientific applications are typically highly parallel in nature. Since the early 2000s, clock speeds of CPUs have stopped increasing due to physical constraints. Instead, multi-core chips have dominated the market meaning CPUs are now parallel architectures.

OpenMP [8] is one framework designed for running on shared memory platforms like CPUs. It is a multi-threading parallel model which uses compiler directives to determine how an algorithm is split and assigned to different threads, where each thread can utilise one of the multiple available cores. OpenMP emphasises ease of use over control, however there are settings to determine how data or algorithmic splits occur.

2.2.2. GPUs

GPUs were originally designed for accelerating computations for graphics, but have increasingly been re-purposed to run other types of codes [9]. As such, the architecture of GPUs has evolved to be markedly different from CPUs. Instead of having a few cores, they have hundreds or thousands of lightweight cores that operate in a data-parallel manner and can thus do many more operations per second than a traditional CPU. However most scientific applications, including those in this paper, are more sensitive to memory bandwidth: the rate at which data can be loaded and stored from memory. GPUs offer significantly higher memory bandwidths over traditional CPUs because they use graphics memory, though they are not used in isolation but as “accelerators” in conjunction with “host” CPUs. Programming a GPU is more complicated than a CPU as the programmer is responsible for offloading computation to the GPU with a specific parallel decomposition as well as managing the distinct memory spaces.

In this paper we investigate acoustic simulation algorithms using NVIDIA and AMD GPUs. NVIDIA GPUs are most commonly programmed using the vendor-specific CUDA model [10], which extends C, C++ or Fortran. CUDA provides functionality to decompose a problem into multiple “blocks” each with multiple “CUDA threads”, with this hierarchical abstraction designed to map efficiently onto the hardware which correspondingly comprises multiple “Streaming Multiprocessors” each containing multiple “CUDA cores”. CUDA also provides a comprehensive API to allow memory management on the distinct CPU and GPU memory spaces. CUDA is very powerful, but low level and non-portable. AMD GPUs are similar in nature to NVIDIA GPUs, but since there is no equivalent vendor-specific AMD programming model, the most common programming method is to use the cross-platform OpenCL, which we discuss in Section 2.3.1. For simplistic purposes, we will use the same terminology to describe the OpenCL framework as for CUDA.

2.2.3. Intel Xeon Phi Many-core CPU

The Xeon Phi was developed by Intel as a high performance many-core CPU for scientific computing [11]. One of the main benefits of the Xeon Phi is that it uses the same instruction set (X86) as the majority of other mainstream CPUs. This means that, theoretically, codes developed to run on CPUs could be more easily ported to Xeon Phi chips. There are fewer cores on the Xeon Phi than on a GPU, however data does not need to be transferred to and from separate memory. There is also a wider vector instruction set on the Xeon Phi, which means that more instructions can be run in parallel per core than on a CPU or GPU. Depending on the algorithm, this can provide a boost in performance. The Xeon Phi currently straddles both the architecture and the performance of the CPU and GPU. The same languages and frameworks that are used for programming CPUs can be used on Xeon Phis.

2.3. Existing Portable Programming Methods

In this section we review existing portable parallel frameworks and APIs: OpenCL, a low-level API designed for use on heterogeneous platforms; TargetDP, a lightweight framework that abstracts data-parallel execution and memory management syntax in a performance portable manner, and a range of other frameworks offering higher levels of abstraction and programmability. These frameworks and APIs are designed to allow computational codes (like room acoustics) to be portable across different architectures, however they can be difficult to program in and they do not all account for performance across hardware.

2.3.1. OpenCL

OpenCL [12] is a cross-platform API designed for programming heterogeneous systems. It is similar in nature to CUDA, albeit with differing syntax. Whereas CUDA acts as a language extension as well as an API, OpenCL only acts as the latter resulting in the need for more boilerplate code and provides a more low-level programming experience. OpenCL can, however, be used as a portable alternative to CUDA, as it can be executed on NVIDIA, AMD and other types of GPUs, as well as manycore CPUs such as the Intel Xeon Phi. OpenCL is compatible with the C and C++ programming languages.

2.3.2. targetDP

The targetDP programming model [13] is designed to target data-parallel hardware in a platform agnostic manner, by abstracting the hierarchy of hardware parallelism and memory systems in a way which can map on to either GPUs or multi/many-core CPUs (including the Intel Xeon Phi) in a platform agnostic manner. At the application level, targetDP syntax augments the base language (currently C/C++), and this is mapped to either CUDA or OpenMP threads (plus vectorisation in the latter case) depending on which implementation is used to build the code. The mechanism used is a combination of C-preprocessor macros and libraries. As described in [13], the model was originally developed in tandem with the complex fluid simulation package *Ludwig*, where it exploits parallelism resulting from the structured grid-based approach. The lightweight design facilitates integration into complex legacy applications, but the resulting code remains somewhat low-level so it lacks productivity and programmability.

2.3.3. Other Methods

Higher level approaches focus more on distinct layers of abstraction that are far removed from the original codes. These approaches include: parallel algorithmic skeletons, code generators, DSLs (Domain Specific Languages), autotuners, combinations thereof and others. Higher-level frameworks can also provide decoupling layers of functionality, which allows for more flexibility with different implementations and architectures. As many of these frameworks are still in early stages of development, there are limitations in using them with pre-existing code bases. Many of these higher-level frameworks build on the work of skeleton frameworks, which focus on the idea that many parallel algorithms can be broken down into pre-defined building blocks [14]. Thus, an existing code could be embedded into a skeleton framework that already has an abstraction and API built for that algorithm type, such as the stencils found in room acoustics models. These frameworks then simplify the process of writing complex parallel code by providing an interface which masks the low-level syntax and boilerplate. A code generator can either be a type of compiler or more of a source to source language translator. Other higher-level approaches include functional DSLs with auto-tuning [15], rewrite rules [16], skeleton frameworks combined with auto-tuning [17] and many others including the examples below. Liquid Metal is a project started at IBM to develop a new programming language purpose built to tailor to heterogeneous architectures [18]. Exastencils is a DSL developed by a group at the University of Passau that aims to create a layered framework that uses domain specific optimisations to build performant portable stencil applications [19].

3. PORTABLE AND PRODUCTIVE FRAMEWORK DEVELOPMENT

The room acoustics simulation codes used in this work were previously tied to a specific platform (NVIDIA GPUs) through their CUDA implementation. In this study we compare the CUDA (pre-existing), OpenCL, and targetDP frameworks (all with C as a base language). In addition, we introduce the newly developed abstraction framework titled *abstractCL* (with C++ as a base language). In this section we describe our approach in enhancing performance portability and productivity through the development of this new framework for investigating room acoustics simulation codes. First, details about the implementation are discussed followed by the benefits of creating such a framework for the field of acoustics modelling.

3.1. Overview

abstractCL was created to make room simulation kernels on-the-fly, depending on the type of run a user wants to do. The type of variations can be between different data layouts of the grid passed in to represent the room, hardware-specific optimisations or both. This is done through swapping in and out relevant files that include overloaded functions and definitions in the main algorithm itself. The data abstractions and optimisations investigated for this project include: thread configuration settings, memory layouts and memory optimisations. Algorithmic changes can be introduced by adding new classes to the current template for more complicated codes.

3.2. Functionality

abstractCL works through the use of flags which determine what version should be run. Certain functions must always be defined as dictated by a parent class. However, those functions' implementations can be pulled in from different sources and concatenated together to create the simulation kernel before compilation. This framework runs similarly to the other benchmark versions, apart from that the kernel is created before the code is run (which creates more initial overhead). It was developed in C++ (due its built-in functionality for classes, templates, inheritance and strings) as well as OpenCL.

3.3. Advantages

One of the main benefits of creating an abstraction framework in this manner is that room acoustics simulation codes would not need to be rewritten to test out new optimisations. This makes it easier to use than a normal OpenCL implementation. Optimisations can be swapped in and out from the same point, limiting the room for error. Additionally, abstractions and performance can often be at war with each other when developing codes. abstractCL provides the opportunity to explore this tension at the most basic level for these simulations by allowing the data type representing the grid (and grid points) to be implemented in different ways that can be changed easily. For example, when accessing a data point, it could be stored in a number of different places in different memories. Using abstractions that mask implementation, the performance effects of these different implementations can then be investigated and compared. Though the optimisations in this project focus primarily on GPU memories, the framework could be extended to include optimisations specific to other platforms that are swapped in and out on a larger scale or for more complex layouts (ie. combinations of memories used).

4. EXPERIMENTAL SETUP

In this section we describe our setup for investigating alternative implementations of room acoustics benchmarks and our means of assessing their performance on different architectures. First we introduce the environment used in this study, including the separate platforms and benchmarks. Then the analysis including metrics and domain sizes used is described.

4.1. Environment

4.1.1. Hardware

The platforms used for this study are specified in Table 1. Included are two NVIDIA GPUs, two AMD GPUs, an Intel Xeon Phi manycore CPU and a traditional Intel Xeon CPU. Of the two NVIDIA GPUs, the consumer-class GTX-780 has much reduced double precision floating point capability over the high-end K20 variant, but offers higher memory bandwidth. Of the AMD GPUs, the R9 259X2 has higher specifications than the R280 and produces the best results overall.

4.1.2. Memory Bandwidth Reference Benchmark

As described in Section 2, memory bandwidth can be critical to obtaining good performance (and this will be confirmed in Section 5). It is therefore important to assess our results relative to what

is expected given the memory bandwidth capability of a particular architecture. However, the peak values presented in Table 1 are rarely achievable in practice. STREAM [20] is an industry standard benchmark which was run on each architecture to provide a reference instead (through simple operations requiring data access from main memory).

4.2. Analysis

4.2.1. Metrics

The different versions of codes were compared using performance timings (time run in seconds), megavoxels/second, and data throughput (in GB/s). Time is determined by running the application with timing calls in place at key points in the code (to determine how much time is spent in the main computational kernel, the secondary kernel, for data IO and miscellaneous time). Megavoxels per second is found by multiplying the volume of the room by the number of time steps simulated and dividing by the simulation run time in seconds. The data throughput is calculated by multiplying the size of the room by the number of bytes accessed for every grid point.

4.2.2. Acoustical Model Sizes

The “room” used in the comparison runs is a rectangle box. Two different sized boxes (rooms) were used in the simulation runs: 256 x 256 x 202 points and 512 x 512 x 402 points. The purpose of using two room sizes is to see what kind of impact there is from increasing the domain space. These sizes do not indicate the actual size of the room, just the number of points in the grid representing the room. The physical size of the room then scales with the audio sample rate chosen (in this study this is set to 44.10kHz). All versions use double precision as single precision can incur rounding errors in certain cases.

5. RESULTS

In this section we present the results of comparing the different room acoustics model implementations described previously as run on the selected hardware platforms. We first present the best performing results on each architecture to provide an overall assessment of the hardware. We then compare the applicability, performance and portability of these different runs. We go on to show that memory bandwidth is critical to achieving good performance. Finally, we describe the effect of optimisations and extend results to a more complex version of the codes.

5.1. Overall Performance Comparison Across Hardware

In this section we give an overview of performance achieved across the different hardware platforms, to assess the capability of the hardware for this type of problem. Figure 2 shows the time taken for the test case where we choose the best performing existing software on each architecture. It can be seen that the GPUs show similar times, with the AMD R9 295X2 performing fastest. This result translates to 8466 megavoxels updates per second. Another point of interest is that the consumer-class NVIDIA GTX780 is significantly faster for the CUDA implementation than the K20, even though it has many times lower double precision floating point capability. This is because, as discussed further in Section 5.3, memory bandwidth is more important than compute for this type

Table 1: Specification of Different Hardware Architectures Used. Note that the “Ridge Point” is the ratio of Peak GFlops to Peak Bandwidth (in the terminology of the ROOFLINE Model).

Platform	Number of Cores/Stream Processor	Peak Bandwidth (GB/s)	Peak GFlops (Double Precision)	Ridge Point (Flops/Byte))	Memory (MB)
AMD R9 259X2	2816	320	716.67	2.24	4096
AMD R280	2048	288	870	3.02	3072
NVIDIA GTX 780	2304	288.4	165.7	0.57	3072
NVIDIA K20	2496	208	1175	5.65	5120
Xeon Phi 5110P	60	320	1011	3.16	8000
Intel Xeon E5-2620	24	42.6	96	2.25	16000

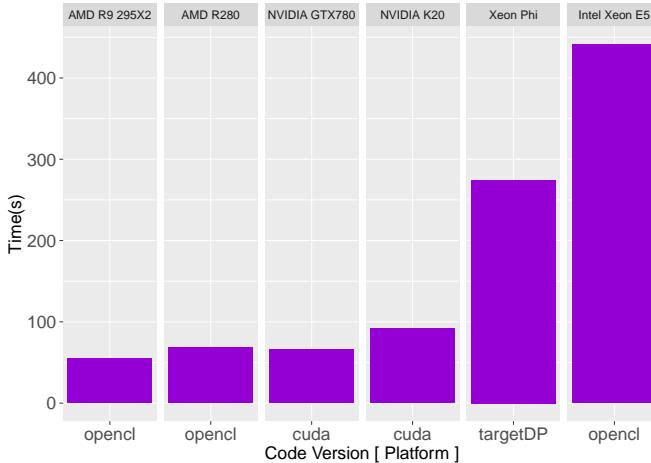


Figure 2: Original fastest (optimised) versions across platforms for simple room acoustics simulation of room size $512 \times 512 \times 404$. The timings shown produce 100ms of sound.

of problem. The Xeon Phi is seen to offer lower performance than the GPUs, but remains faster than the traditional CPU.

5.2. Performance Comparison of Software Versions

In this section we analyse the differences in performance resulting from running different software frameworks on a particular platform. In Figure 3, it can be seen how the performance depends on the software version. The main result is that for each architecture the timings are comparable, in particular in comparison to the *abstractCL* version. On the NVIDIA GPU, there is a small overhead for the portable frameworks (OpenCL, abstractCL and targetDP) relative to use of CUDA, but we see this as a small price to pay for portability to the other platforms. In particular, the newly developed framework *abstractCL* shows comparable performance to the original benchmarks and those written in OpenCL, indicating that it is possible to build performant, portable and productive room acoustics simulations across different hardware.

5.3. Hardware Capability Discussion

In this section, we further analyse the observed performance in terms of the characteristics of the underlying hardware. The ROOFLINE model, developed by Williams et al. [21], can be used to determine for a given application the relative importance of floating

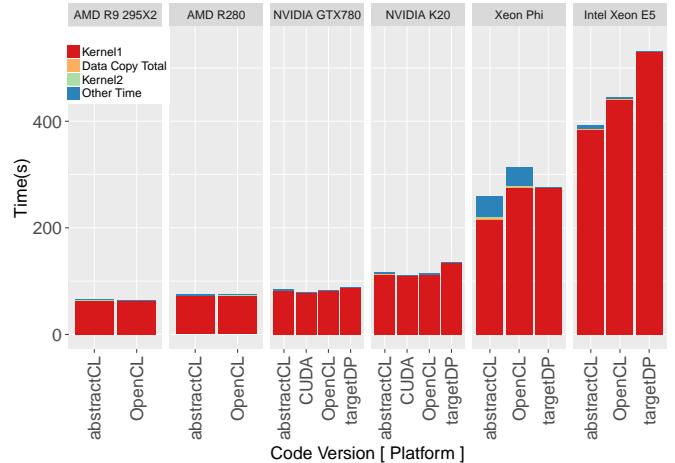


Figure 3: Original timings of the simple room acoustics simulation for room size $512 \times 512 \times 404$ on all architectures tested. The timings shown produce 100ms of sound.

point computation and memory bandwidth capabilities. It uses the concept of “Operational Intensity” (OI): the ratio of operations (in this case double precision floating point operations) to bytes accessed from main memory. The OI, in Flops/Byte, can be calculated for each computational kernel. A similar measure (also given in Flops/Byte and known as the “ridge point”), exists for each processor: the ratio of peak operations per second to the memory bandwidth of the processor. Any kernel which has an OI lower than the ridge point is limited by the memory bandwidth of the processor and any which has an OI higher than the ridge point is limited by the processor’s floating point capability.

The OI for the application studied in this paper is 0.54, which is lower than the ridge points of any of the architectures (given in Table 1), indicating that this simplified version of the application is memory bandwidth bound across the board (and thus not sensitive to floating point capability). This explains why the NVIDIA GTX780 performs so well despite the fact that it has very low floating point capability: its ridge point of 0.57 is still (just) higher than the application OI.

The blue columns in Figure 4 give observed data throughput (volume of data loaded/stored by the application divided by runtime, assuming perfect caching), for each of the architectures. The black lines give the peak bandwidth capability of the hardware (as reported in Table 1). It can be seen that, for all but the Xeon Phi architecture, the measured throughput varies in line with the peak

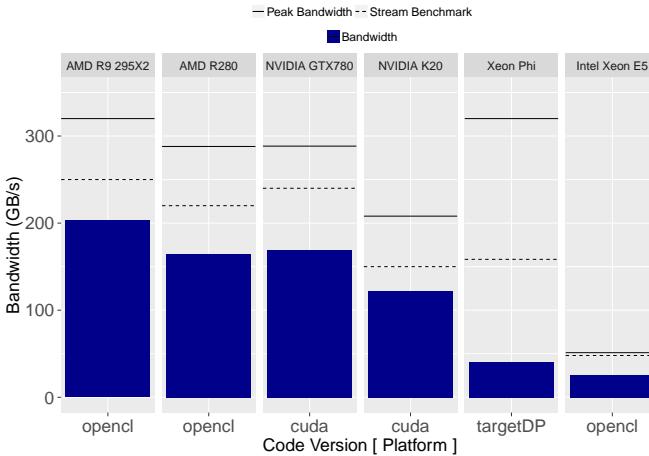


Figure 4: Data Throughput for different versions of the room acoustics benchmark across platforms for room size $512 \times 512 \times 404$.

bandwidth, evidence that the overall performance of this application can largely be attributed to the memory bandwidth of that architecture. Since it is often very difficult to achieve near peak performance, we also include (with dashed horizontal lines) STREAM benchmark results (see Section 4.1.2), which give a more realistic measure of what is achievable. It can be seen that our results are, in general, achieving a reasonable percentage of STREAM but we still have room for improvement. In addition to running STREAM, profiling was also done on a selection of the runs. These results showed higher values than our measured results, indicating that our assumption of perfect caching (see above) is not strictly true and there may be scope to reorganise our memory access patterns to improve the caching. The Xeon Phi is seen to achieve a noticeably lower percentage of peak bandwidth relative to the other architectures, and this warrants further investigation.

5.4. Optimisation

Two optimisation methods were explored for the GPU platforms: shared and texture memory. Texture memory is a read-only or write-only memory that is distinct from global memory and uses separate caching allowing for quicker fetching for specific types of data (in particular, ones that take advantage of locality). The use of texture memory is relatively straightforward in CUDA, requiring only an additional keyword for input parameters. OpenCL is restricted to an earlier version on NVIDIA GPUs which does not support double precision in texture memory, so these results are not included. Shared memory is specific to one of the lightweight “cores” of a GPU (streaming multi-processor on NVIDIA) and can only be shared between threads utilising that core, so can be useful for data re-use. A 2.5D tiling method was used to take advantage of shared memory [6].

Figure 5 shows the results for this experiment, where the optimised versions are the fastest versions found in this project per version per platform. Three different types of codes were run: *sharedtex* uses shared and texture memory (for the CUDA version), *shared* uses only shared memory and *none* uses no memory optimisations. As before, different platforms are indicated by

separate segments of the graphs. The reason for comparing to an optimised thread configuration version instead of the original run was to isolate what effect the memory optimisations had. Both room sizes (large and small) were run, but the large rooms had more significant differences in performance so are the only results shown. Overall the abstractCL version showed the most consistent improvement, however in this graph it is more clear that it is not quite as fast as the OpenCL version due to the overhead of using a more productive framework. All versions showed some improvement with this use of shared memory, but this amount varied per version and per room size across the different architectures. One of the reasons these results do not show more improvement is because the codes are already close to peak bandwidth as is discussed in Section 5.3.

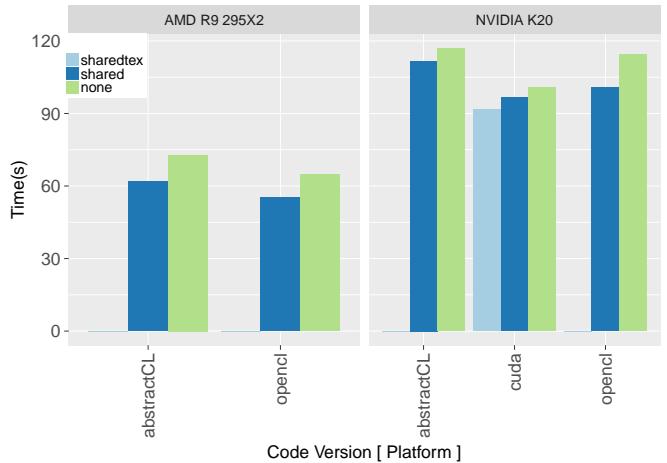


Figure 5: Memory optimisations for CUDA, OpenCL and abstractCL versions of the room acoustics benchmarks run for room size of $512 \times 512 \times 404$ on an NVIDIA and AMD GPU. The timings shown produce 100ms of sound.

5.5. Advanced Simulations

Results in this paper have thus far only been presented for a very simplified problem: wave propagation is assumed lossless and the simplest FDTD scheme (employing a seven-point stencil) is used. It is thus of interest to explore more complex models of wave propagation, as well as improved simulation designs. Two new features were investigated for these so-called “advanced” codes: air viscosity (see [6]) and larger stencil schemes of leggy type (see [3]). The main algorithmic difference in adding viscosity is that another grid is passed into the main kernel and more computations are performed. Schemes operating over leggy stencils require access to grid values beyond the nearest neighbours. For this investigation, 19-point leggy stencils were introduced (three points in each of the six Cartesian directions as well as the central point, see Figure 1 in [3]). The comparison was done on a smaller scale, however, with the intention only of giving a general idea of whether or not the codes performed similarly. Thus, only CUDA and OpenCL versions were tested. The variations were run on the two NVIDIA GPUs, the two AMD GPUs and the Xeon Phi for both small and large room sizes.

Results of the performance of these advanced codes can be discussed in a number of ways including: in comparison to the simpler codes, as a comparison amongst the different advanced versions, as a comparison between versions on the same platform and also comparisons of the same version across platforms. Graphs in Figure 6 show the performance timings and the memory bandwidth of the advanced codes for the various implementations for the larger room size. In these graphs, the following versions are included: `cuda` (the original version), `cuda_adv` (`cuda` with viscosity), `cuda_leggy` (`cuda` with leggy stencils) and `cuda_leggyadv` (`cuda` with leggy stencils and viscosity). The comparable OpenCL counterparts of these versions were also run. These graphs are set up in a similar way as those found in Figures 3 and 4, where the versions of the code run along the x-axis and the performance is on the y-axis (in seconds) and the separated parts of the graph indicate the platform it was run on. Here, the top graph shows performance and the bottom shows bandwidth.

Generally the performance profile of the advanced codes looks similar to what can be seen for the original codes in Section 5.1: the codes still run fastest on AMD R9 259X2 and slowest on the Xeon Phi. The versions run on the AMD R9 259X2 in comparison to the same versions on the K20 hover between being 43%-54% faster. For both large and small rooms, the leggy codes are slower than the viscosity codes for both OpenCL and CUDA on everything except the K20. The combination advanced codes (*_leggyadv) are significantly slower across the board, particularly on the Xeon Phi.

This purpose of this analysis is to see what effect algorithmic changes (ie. number of inputs or floating point operations) in the same benchmark have when run on the same platform. How the performance changes with differences to the models of the rooms can vary quite a bit across platforms, which echoes the results seen when comparing local memory optimisations. When comparing original versions to the leggy, viscosity or combination versions, OpenCL codes are 1.4–6.4x slower for the combination versions. When this is limited to AMD GPUs, the difference is only 1.4x slower - for NVIDIA GPUs, 3–3.6x slower. In comparison, the CUDA version on the NVIDIA GPUs varies from 1.6–2.4x slower for the combination version. For the stand-alone leggy and viscosity versions, this difference is much less pronounced, however the same trend remains: OpenCL versions retain better performance with changes on AMD platforms and significantly worse than CUDA versions on NVIDIA GPUs. These differences cannot wholly be attributed to specification differences given that the difference exists for all these versions between the AMD R280 and NVIDIA GTX 780, which share some similar specifications.

6. SUMMARY AND FUTURE WORK

In this paper we have shown that it is possible to implement room acoustics simulations in a way that allows the same source code to execute with good performance across a range of parallel architectures. Prior to this work, such simulations were predominantly tied to NVIDIA GPUs and we have now extended applicability to other platforms that were previously inaccessible. We have found that the main indicator of how the application will perform on a given architecture is the memory bandwidth offered by that architecture, due to the fact that the algorithm has low operational intensity. The best performing platforms are AMD and NVIDIA GPUs, due to their high memory bandwidth capabilities. The AMD R9 259X2 has the highest peak bandwidth of the GPUs tested, and was corre-

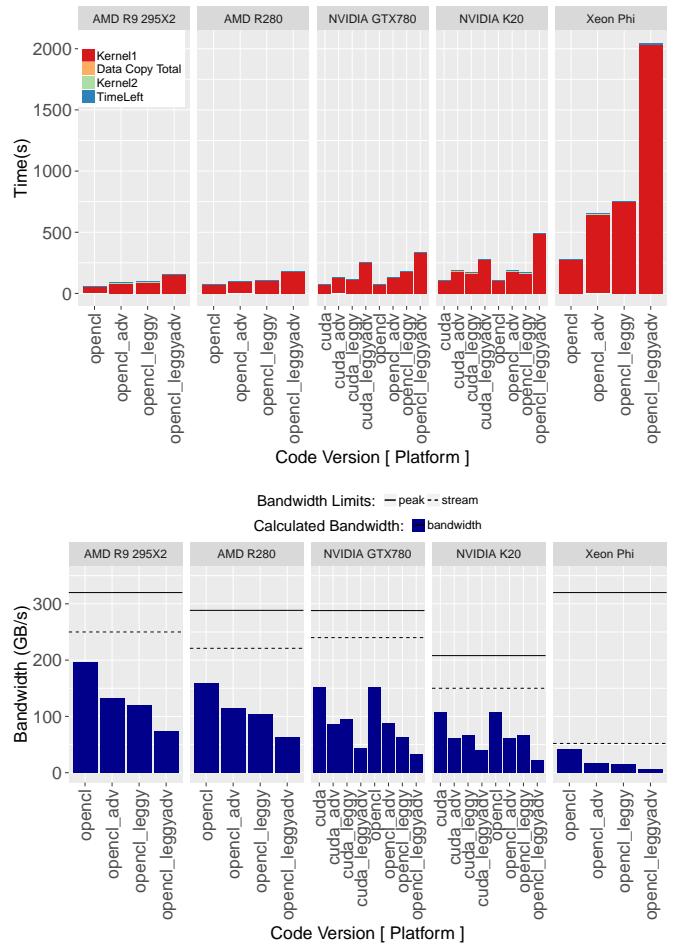


Figure 6: Timing (top) and Bandwidth (bottom) for different versions of the advanced room acoustics benchmarks across platforms for room size 512×512×404. Smaller room results are not included as they show similar pattern.

spondingly found to be the best performing platform for the room acoustics simulation codes. In addition, we found the consumer-class NVIDIA GTX780 outperforms the HPC-specific NVIDIA K20 variant despite the fact it has many times lower floating point capability, due to insensitivity of the application to computational capability, as well as higher memory bandwidth of the former. Traditional CPUs have much lower memory bandwidth than GPUs, and measured performance was correspondingly low. The Intel Xeon Phi platform offers a high theoretical memory bandwidth, but we were unable to achieve a reasonable percentage of this in practice.

Performance-portable frameworks, including the abstractCL framework designed to allow flexibility in implementation options, were able to achieve similar performance to native methods, with only relatively small overheads (that we consider a small price to pay for the benefits that are offered by portability). However, relatively low-level programming is still required for these frameworks (including explicit parallelisation and data management). An ideal framework would offer performance portability, whilst allowing an intuitive definition of the scientific algorithm. Future work will

adapt a higher level framework named LIFT [22], currently under research and development at the University of Edinburgh, to enable it for 3D wave-based stencil computations. This framework aims, through automatic code generation, to allow execution of an application across different hardware architectures in a performance portable and productive manner. Future work will also look into extending abstract frameworks such as LIFT to modelling these codes across multiple parallel devices.

7. ACKNOWLEDGEMENTS

We thank Brian Hamilton and Craig Webb for providing the benchmarks used in this project and for answering questions about acoustics. We would also like to thank Michel Steuwer for providing invaluable advice about GPUs and OpenCL. This work was supported in part by the EPSRC Centre for Doctoral Training in Pervasive Parallelism, funded by the UK Engineering and Physical Sciences Research Council (grant EP/L01503X/1) and the University of Edinburgh.

8. REFERENCES

- [1] Dick Botteldooren, “Finite-Difference Time-Domain Simulation Of Low-Frequency Room Acoustic Problems,” *The Journal of the Acoustical Society of America*, vol. 98, no. 6, pp. 3302–3308, 1995.
- [2] Stefan Bilbao, Brian Hamilton, Alberto Torin, et al., “Large Scale Physical Modeling Sound Synthesis,” in *Stockholm Musical Acoustics Conference (SMAC)*, 2013, pp. 593–600.
- [3] Brian Hamilton, Craig J Webb, Alan Gray, and Stefan Bilbao, “Large Stencil Operations For GPU-Based 3-D Acoustics Simulations,” *Proc. Digital Audio Effects (DAFx)*, (Trondheim, Norway), 2015.
- [4] Niklas Röber, Martin Spindler, and Maic Masuch, “Waveguide-Based Room Acoustics Through Graphics Hardware.,” in *ICMC*, 2006.
- [5] “Compilers and More: What Makes Performance Portable?,” <https://www.hpcwire.com/2016/04/19/compilers-makes-performance-portable/>.
- [6] Craig Webb, *Parallel Computation Techniques for Virtual Acoustics and Physical Modelling Synthesis*, Ph.D. thesis, University of Edinburgh, 2014.
- [7] Jelle Van Mourik and Damian Murphy, “Explicit Higher-Order FDTD Schemes For 3D Room Acoustic Simulation,” *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 22, no. 12, pp. 2003–2011, 2014.
- [8] OpenMP Architecture Review Board, “OpenMP Application Program Interface Version 4.5,” November 2015, <http://www.openmp.org/wp-content/uploads/openmp-4.5.pdf> Accessed: 2016-08-12.
- [9] David B. Kirk and Wen-mei W. Hwu, *Programming Massively Parallel Processors: A Hands-On Approach*, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2nd edition, 2013.
- [10] NVIDIA, “Programming Guide: CUDA Toolkit Documentation,” <http://docs.nvidia.com/cuda/cuda-c-programming-guide/index.html>. Accessed: 2016-08-12.
- [11] James Reinders, “An Overview of Programming for Intel Xeon Processors and Intel Xeon Phi Coprocessor,” https://software.intel.com/sites/default/files/article/330164/an-overview-of-programming-for-intel-xeon-processors-and-intel-xeon-phi-coprocessors_1.pdf. Accessed: 2016-08-05.
- [12] Khronos OpenCL Working Group, “OpenCL Specification. Version 2.2,” 2016, <https://www.khronos.org/registry/OpenCL/specs/opencl-2.2.pdf> Accessed: 2016-08-12.
- [13] Alan Gray and Kevin Stratford, “A Lightweight Approach To Performance Portability With TargetDP,” *The International Journal of High Performance Computing Applications*, p. 1094342016682071, 2016.
- [14] Murray Cole, *Algorithmic Skeletons: Structured Management of Parallel Computation*, Ph.D. thesis, University of Edinburgh, 1989.
- [15] Yongpeng Zhang and Frank Mueller, “Autogeneration and Autotuning Of 3D Stencil Codes On Homogeneous And Heterogeneous GPU Clusters,” *Parallel and Distributed Systems, IEEE Transactions on*, vol. 24, no. 3, pp. 417–427, 2013.
- [16] Franz Franchetti, Frédéric de Mesmay, Daniel McFarlin, and Markus Püschel, “Operator Language: A Program Generation Framework For Fast Kernels,” in *Domain-Specific Languages*. Springer, 2009, pp. 385–409.
- [17] Cédric Nugteren, Henk Corporaal, and Bart Mesman, “Skeleton-based Automatic Parallelization Of Image Processing Algorithms For GPUs,” in *Embedded Computer Systems (SAMOS), 2011 International Conference On*. IEEE, 2011, pp. 25–32.
- [18] Joshua Auerbach, David F Bacon, Perry Cheng, et al., “Growing A Software Language For Hardware Design,” *1st Summit on Advances in Programming Languages (SNAPL 2015)*, vol. 32, pp. 32–40, 2015.
- [19] Christian Lengauer, Sven Apel, Matthias Bolten, et al., “Ex-stencils: Advanced Stencil-Code Engineering,” in *European Conference On Parallel Processing*. Springer, 2014, pp. 553–564.
- [20] John D. McCalpin, “Memory Bandwidth and Machine Balance in Current High Performance Computers,” *IEEE Computer Society Technical Committee on Computer Architecture (TCCA) Newsletter*, pp. 19–25, Dec 1995.
- [21] Samuel Williams, Andrew Waterman, and David Patterson, “Roofline: An Insightful Visual Performance Model For Multicore Architectures,” *Communications of the ACM*, vol. 52, no. 4, pp. 65–76, 2009.
- [22] Michel Steuwer, Christian Fensch, Sam Lindley, and Christophe Dubach, “Generating Performance Portable Code Using Rewrite Rules: From High-Level Functional Expressions To High-Performance Opencl Code,” *ACM SIGPLAN Notices*, vol. 50, no. 9, pp. 205–217, 2015.
- [23] George Teodoro, Tahsin Kurc, Jun Kong, et al., “Comparative Performance Analysis Of Intel Xeon Phi, GPU And CPU,” *arXiv preprint arXiv:1311.0378*, 2013.

ON RESTORING PREMATURELY TRUNCATED SINE SWEEP ROOM IMPULSE RESPONSE MEASUREMENTS

Elliot Kermit 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

When measuring room impulse responses using swept sinusoids, it often occurs that the sine sweep room response recording is terminated soon after either the sine sweep ends or the long-lasting low-frequency modes fully decay. In the presence of typical acoustic background noise levels, perceivable artifacts can emerge from the process of converting such a prematurely truncated sweep response into an impulse response. In particular, a low-pass noise process with a time-varying cutoff frequency will appear in the measured room impulse response, a result of the frequency-dependent time shift applied to the sweep response to form the impulse response.

Here, we detail the artifact, describe methods for restoring the impulse response measurement, and present a case study using measurements from the Berkeley Art Museum shortly before its demolition. We show that while the difficulty may be avoided using circular convolution, nonlinearities typical of loudspeakers will corrupt the room impulse response. This problem can be alleviated by stitching synthesized noise onto the end of the sweep response before converting it into an impulse response. Two noise synthesis methods are described: the first uses a filter bank to estimate the frequency-dependent measurement noise power and then filter synthesized white Gaussian noise. The second uses a linear-phase filter formed by smoothing the recorded noise across perceptual bands to filter Gaussian noise. In both cases, we demonstrate that by time-extending the recording with noise similar to the recorded background noise that we can push the problem out in time such that it no longer interferes with the measured room impulse response.

1. INTRODUCTION

In order to study room acoustics, one must measure accurate impulse responses of the space. These measurements are often challenging and time consuming to acquire. Large spaces frequently exhibit poor background noise levels, so acousticians often employ methods to improve the signal-to-noise ratio (SNR). In particular, linear and logarithmic sine sweep measurements have been shown to be highly effective [1]. Researchers have identified some of the problems and their solutions for using sine sweep measurements to study room reverberation. Specifically, [2] and [3] address issues of time-smearing, clicks/plosives, and pre/post equalization. Further, [4] discusses issues related to clock drift.

This paper addresses another problem that is often encountered when measuring impulse responses with sine sweeps in noisy environments that has not been previously discussed—what happens when the sweep response recording is stopped too early. This is common, as it is challenging to maintain quiet after the room re-

sponse has appeared to decay into the noise floor. Plosives and impulsive noises will be converted into descending sweeps and any such noises in the silence following the recorded sweep will be detrimental to the conversion process. As it turns out, one needs to record a duration of background noise equivalent to the length of the sweep following the response’s decay into the noise floor to ensure that the problem will not be encountered.

If one does not record enough silence (background noise) after the sine sweep response, the resulting impulse response will contain a time-varying low-pass filter characteristic imprinted on its noise floor. This paper addresses the cause of these artifacts and two methods for alleviating the problem in post-production. Related to this problem, [5] and [6] discuss methods for extending impulse responses through the noise floor, however there are implications for how the noise floor is measured based on these new results.

The rest of the paper is organized as follows. In §2 we review the process of converting a sine sweep measurement into an impulse response and introduce the problem associated with stopping the recording prematurely. In §3 we introduce two methods for pre-processing the response sweep by extending the recording with synthesized noise that matches the background noise present in the recording. Next §4 presents examples that demonstrate how our method for preprocessing the sweep response is desirable compared to zero padding or circular convolution. Finally, §5 presents concluding remarks.

2. CONVERTING SINE SWEEPS TO IMPULSE RESPONSES

Linear and logarithmic sine sweeps can be used to measure room impulse responses and work under the principle that the sweep can be viewed as an extremely energetic impulse smeared out in time. A linear sweep, in which the sinusoid frequency increases linearly over time, can be defined as

$$x(t) = \sin\left(\omega_0 t + \frac{\omega_1 - \omega_0}{T} \frac{t^2}{2}\right), \quad t \in [0, T], \quad (1)$$

where ω_0 is the initial frequency in radians, ω_1 the final frequency, and T the total duration. A logarithmic sweep, in which the sinusoid frequency increases exponentially over time, can be defined using the same variables as

$$x(t) = \sin\left(\frac{\omega_0 T}{\ln\left(\frac{\omega_1}{\omega_0}\right)} \left[\exp\left(\frac{\ln\left(\frac{\omega_1}{\omega_0}\right) t}{T}\right) - 1 \right]\right). \quad (2)$$

To convert sine sweep responses to impulse responses, one acyclically convolves an equalized, time-reversed version of the

original sine sweep, \tilde{x} with the recorded response such that

$$h(t) = y(t) * \tilde{x}(t). \quad (3)$$

This works because the group delay is canceled by the time reversal, with the equalization compensating for the relative time spent in each frequency band,

$$\delta(t) = x(t) * \tilde{x}(t). \quad (4)$$

This deconvolution processing aligns the response to the original sine sweep in time effectively forming the impulse response. As a linear sweep traverses any given bandwidth in the same length of time, irrespective of the starting frequency, the equalization is a constant, independent of frequency. For a logarithmic sweep, which spends the same length of time traversing any given octave, an exponential equalization is applied to compensate for the disproportionate amount of low-frequency energy applied to the system. Naturally, the calculation is efficiently computed in the frequency domain¹ as

$$h(t) = \mathcal{F}^{-1} \left(\frac{Y(\omega)}{X(\omega)} \right). \quad (5)$$

One of the benefits to using sine sweep measurements over other integrated-impulse response measurement techniques, such as maximum length sequences, is that many nonlinearities produce only harmonics of an input sinusoid, and the deconvolution process will place the onset of unwanted harmonic component responses before the onset of the linear component of the system response. Using logarithmic sweeps, the time shift of the harmonic distortion components of the response is controlled via the length of the sweep, and the linear response is easily separated from the harmonic distortion response. While [2, 7, 8] and others have shown that useful information can be extracted from the higher order components, that is not the concern of this paper. For acquiring a linear room impulse response, one can simply window out the distortion artifacts that precede the linear response.

The fundamental problem this paper explores is caused by the desire to use acyclic convolution rather than circular convolution to convert the sine sweep response into separate linear "impulse response" and non-linear system responses. The difficulty results from the presence of additive measurement noise, Eq. (3) is actually

$$h(t) = [y(t) + n(t)] * \tilde{x}(t). \quad (6)$$

where $n(t)$ is the measurement noise (often mainly acoustic background) and is assumed to be stationary. In addition to time-aligning the frequencies of the sweep into an impulse response, the background noise is also shifted by the same transformation. Under acyclic convolution, the tail end of the converted impulse response will exhibit a frequency-dependent filter cutoff with the same trajectory as the frequency trajectory of $x(t)$. When the recording of background noise following the sweep response is sufficiently long, the filter artifact will occur after the impulse response has decayed, and can be windowed out. However, when the recording is shut off too early, linear convolution causes this frequency-dependent filter effect to overlap with the impulse response. While a recording with a high SNR may render this effect

¹In this paper, we use t as the argument to functions in the time domain and ω as the argument to functions in the frequency domain.

inaudible, in spaces with poor SNR, this unwanted effect becomes quite clear.

The naïve solution to this problem would be to use circular convolution rather than linear convolution. However, this is not ideal. Circular convolution will reconstruct the noise floor and solve the issue of the filtering effect, shifting the noise corresponding to times before the sweep to the end of the response, but any nonlinearities in the measurement will also be shifted to occur during the desired impulse response. Weak nonlinearities are all but guaranteed when measuring an acoustic space, as loudspeakers are inherently nonlinear at levels useful in impulse response measurement. The effect of circular convolution will corrupt the measurement.

Fig. 1 shows a contrived example that highlights the difference between linear and circular convolution using a linear sweep and noise signal. Under circular convolution, the noise statistics ought to stay constant (i.e., $n(t) \sim n(t) * \tilde{x}(t)$). Because of zero padding, linear convolution causes the noise occupy a longer period of time. Over the course of the beginning of the processed block, the noise starts from the high frequencies, and lower frequencies enter according to the slope of the frequency trajectory of the sweep. At the end of the file, the opposite is true: the high frequencies stop before the lower frequencies. If this noise were the background noise in a space, the effect of the frequencies stopping at different times would be heard as a filter with a time-varying cutoff frequency.

Fig. 2 demonstrates the difference between cyclic and acyclic convolution when there are nonlinearities present. In the acyclic convolution, the nonlinear components precede the desired linear response while in the cyclical convolution they corrupt the linear response.

Ideally, a sufficiently long segment of background noise following the sine sweep is recorded so as to avoid additional processing. In real spaces, low frequencies typically take longer to decay than high frequencies. Ascending sweeps help hide issues caused by the premature ending of a recording, as the low frequency components are provided more time to decay while higher frequencies are being excited in the space. In order to guarantee that the noise roll off problem will not be encountered during the impulse response, an additional amount of background noise of length T must be captured following the decay of the room response to $x(t)$. The longer the sine sweep, the more patient one must be when recording the impulse response. (In addition, transients occurring during the time after the system response has decayed may corrupt the measured impulse response.)

3. NOISE EXTENSION TECHNIQUES

When the recording is cut off too early, our solution is to push the filter cutoff in time so that it no longer interacts with the room response. To do this, we pre-process the sweep response $y(t)$ by extending it with noise that is perceptually similar to the background noise in the physical space. We propose two methods detailed below for synthesizing this noise. In both cases, we analyze a portion of recorded room noise with no other signals present, synthesize additional noise, and splice it onto the end of the response sweep before converting it into an impulse response.

For both methods, we use a 500 ms analysis window and a 50 ms cross-fade. If the room response has decayed sufficiently, it is best to perform the analysis on samples coming from the end of the response audio file as this is where the new samples will

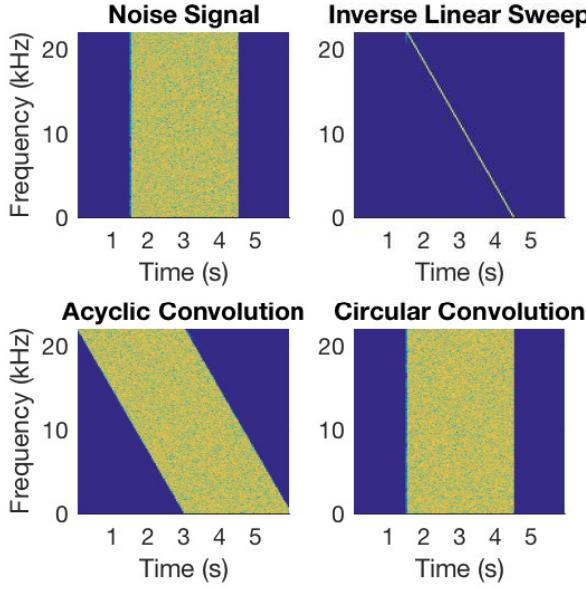


Figure 1: Spectrograms of noise, inverse (linear) sweep, acyclic convolution, and circular convolution.

be spliced. Our goal is to capture both the overall characteristic of the background noise as well as any local features necessary to make an imperceptible transition. If there is a noticeable mismatch between the recorded noise and the synthesized noise, descending-chirps will be introduced into the impulse response by the deconvolution process. If there is not enough isolated noise at the end of the file, it is possible to use a portion of background noise from somewhere else (i.e., proceeding the sweep or another recording taken at the same time).

3.1. Band-pass Filtered Noise

In the first approach, we aim to match the spectrum of the background noise by analyzing the amplitude of the recorded noise in a set of frequency bands, and apply these amplitudes to synthesized white Gaussian noise to correctly color it.

We define the analysis noise $n(t)$ as 500 ms of recorded noise from the recording we intend to match. Additionally, we generate an i.i.d. Gaussian noise sequence, denoted $\gamma(t)$. In this method, we form a synthesis signal $s(t)$ such that

$$|S(\omega)| \sim |N(\omega)|. \quad (7)$$

We use a perfect reconstruction, zero-phase filter bank to split both $n(t)$ and 500 ms of $\gamma(t)$ into K composite frequency bands

$$n(t) = \sum_k n_k(t), k \in [1, 2, \dots, K], \quad (8)$$

and

$$\gamma(t) = \sum_k \gamma_k(t). \quad (9)$$

Our filter bank consists of a cascade of squared 3rd-order Butterworth filters with center frequencies spaced 1/4 octave apart. The

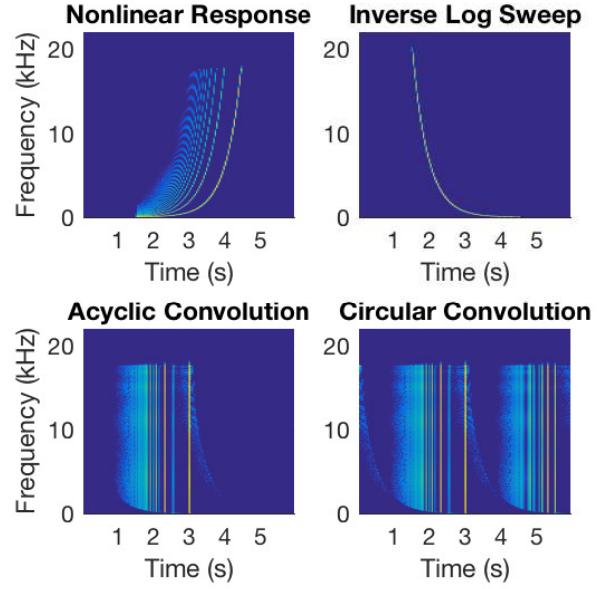


Figure 2: Non-linear sweep response, inverse (logarithmic) sweep, acyclic convolution, and circular convolution.

signals are filtered both in the forwards and backwards directions so that the phase is unaltered. The perfect reconstruction aspect of this filter bank is important because we use it both for analysis and synthesis.

Once separated into bands, we estimate the gain coefficient in each frequency band of both $n(t)$ and $\gamma(t)$ by computing the RMS level on the steady-state portion of the filtered signals. To compute the synthesis noise $s(t)$, we color a sufficiently long² amount of $\gamma(t)$ by scaling each frequency band by the ratio of measured analysis to synthesis gains and sum the result,

$$s(t) = \sum_k \left(\frac{\text{RMS}[n_k(t)]}{\text{RMS}[\gamma_k(t)]} \right) \gamma_k(t). \quad (10)$$

At this point, the steady state portion of $s(t)$ is a block of noise with the same magnitude frequency band response as the analysis signal $n(t)$. An example impulse response and magnitude spectrum resulting from this filter bank can be seen in Fig. 3. We found that 1/4 octave bands were sufficient to match the synthesis and analysis noises' magnitude spectrum. We then use a 50 ms long equal-power cross-fade between the end of $y(t)$ and the beginning of the steady state portion of $s(t)$. After this, it is safe to convert the sweep response into an impulse response as done in Eq. (5).

3.2. ERB-Smoothed Noise

Our second method synthesizes noise that is perceptually similar to the analysis noise via a filter design technique. We define $\gamma(t)$ and $n(t)$ in the same way as above, in § 3.1. We window both noise signals with a Hann window and take their Fourier transforms. In the frequency domain, we smooth both signals on a critical band

²Sufficient here depends on how prematurely the recording was halted.

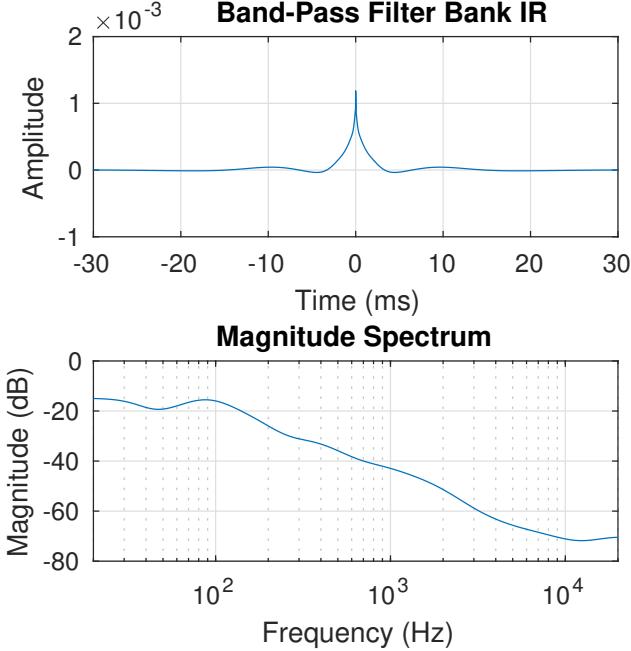


Figure 3: Band-pass filterbank impulse response and magnitude spectrum.

basis by averaging the power within the DFT bins of each critical band such that

$$\hat{N}(\omega) = \sum_{\zeta=f(b(\omega-1/2))}^{f(b(\omega+1/2))} |N(\zeta)|^2 \quad (11)$$

and

$$\hat{\Gamma}(\omega) = \sum_{\zeta=f(b(\omega-1/2))}^{f(b(\omega+1/2))} |\Gamma(\zeta)|^2, \quad (12)$$

where $b(\cdot)$ defines a critical bandwidth. This results in $\hat{N}(\omega)$ and $\hat{\Gamma}(\omega)$, the critical band smoothed versions of $N(\omega)$ and $\Gamma(\omega)$. This processing reduces the complexity and detail of the noise signals' spectra but should not be perceptually audible. We then impart the spectrum of the smoothed analysis noise upon the smoothed synthesis noise in the frequency domain to obtain the transfer function

$$G(\omega) = \frac{\hat{N}(\omega)}{\hat{\Gamma}(\omega)}. \quad (13)$$

We then find a linear-phase version of this transfer function

$$G_{\text{lin}}(\omega) = |G(\omega)| e^{-j\tau\omega}. \quad (14)$$

Returning $G_{\text{lin}}(\omega)$ to the time domain as seen in Fig. 4, we filter $\gamma(t)$ with $g_{\text{lin}}(t)$ such that

$$s(t) = \gamma(t) * g_{\text{lin}}(t). \quad (15)$$

Last, we splice the synthesized noise onto $y(t)$ with a 50 ms equal power cross-fade as described above.

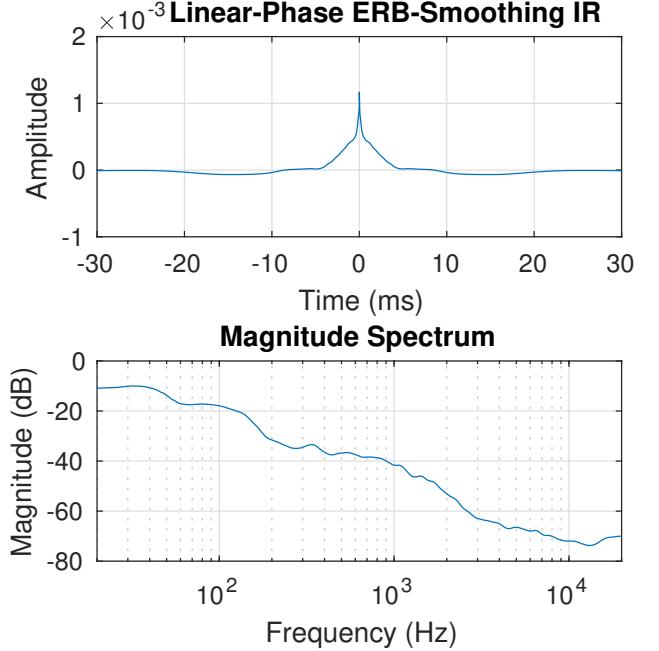


Figure 4: ERB-smoothed noise filter impulse response and magnitude spectrum.

4. EVALUATION

To preserve the acoustics of the Berkeley Art Museum (BAM) at 2626 Bancroft Way in Berkeley, CA, acoustic measurements were taken by Jacqueline Gordon and Zackery Belanger and their team shortly before it was demolished in 2015 [9]. This space, designed by Mario Ciampi, has a very long reverberation time due to its large, concrete structure. Since this space exhibited a high noise floor, a long sine sweep was employed in order to improve the SNR. Circumstances led to the recordings being prematurely ended, and to the discovery of the problem described in this paper. Fig. 5 shows an example recorded (short) response sweep as well as versions extended with noise synthesized with our two methods. Fig. 6 shows the impulse responses computed from these sweeps. Both visually and aurally the noise extended approaches achieve better results than the unprocessed version.

Both approaches for synthesizing noise produce reasonable results, and the power spectrum for the analysis noise and both varieties of synthesis noise can be seen in Fig. 7. On average, the synthesis noise tracks the nature of the room sound well, and both synthesis methods produce perceptually similar results. As it turns out, it is equally important to have a smooth cross-fade between the recorded and synthesized noise as sudden changes will create perceptual artifacts.

5. CONCLUSIONS

In this paper, we discuss how converting a sine sweep response to an impulse response requires a sufficient amount of recording time beyond what is necessary to capture the room response after it has perceptually decayed. While it is naturally best to record this in the physical space, it is not always possible like at the Berkeley Museum of Art. We propose two methods for extending the record-

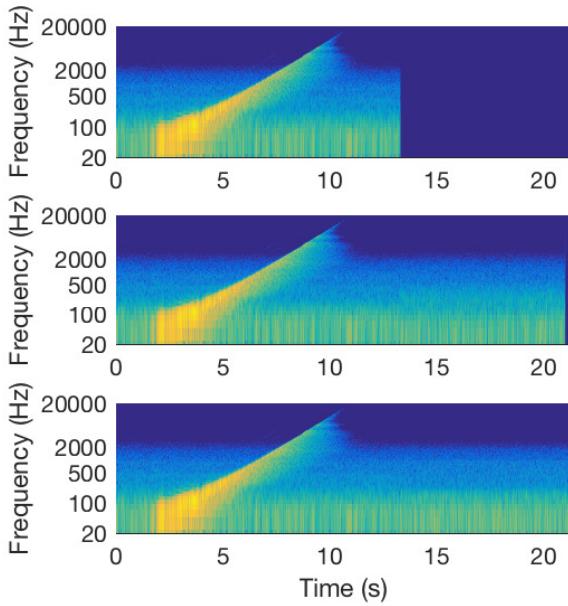


Figure 5: Spectrograms for signal sweep and sweep responses with no treatment, band-passed noise, and ERB smoothed noise.

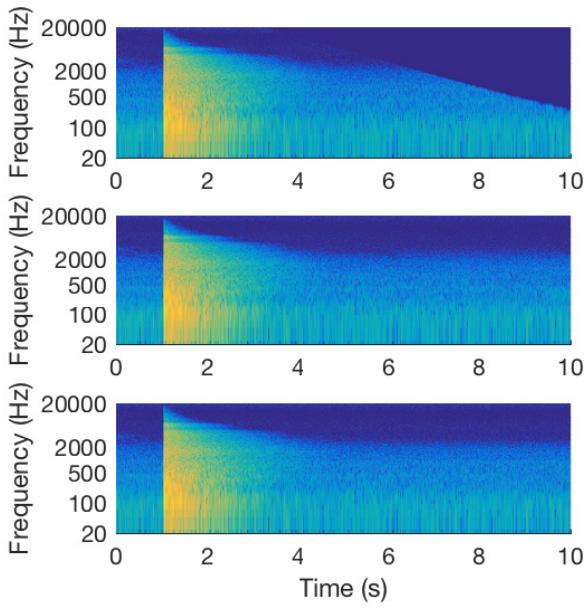


Figure 6: Spectrograms corresponding to the impulse responses calculated from the sweep responses in Fig. 5—no treatment, band-passed noise, and ERB smoothed noise.

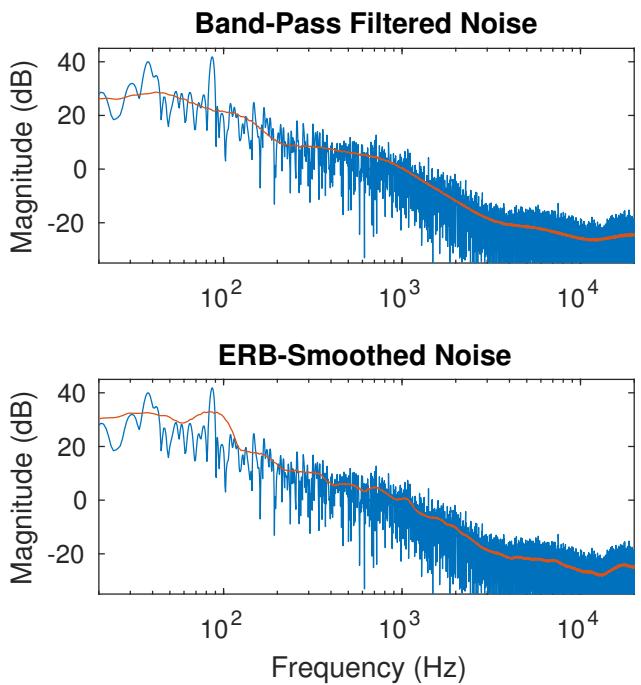


Figure 7: Spectrum of synthesized noise (red) averaged over 500 simulations compared to analyzed noise (blue) for the band-pass filter method (top) and ERB-smoothed method (bottom).

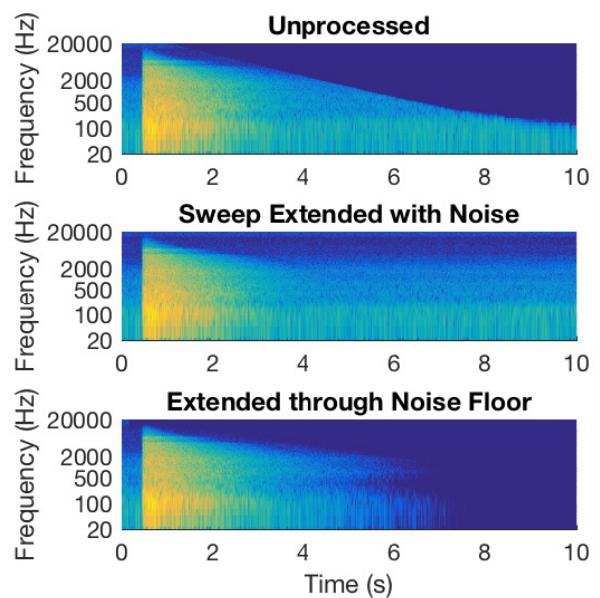


Figure 8: BAM impulse response without processing, extending the sweep with noise, and extending impulse response through the noise floor.

ing after the fact that both depend on analyzing recorded background noise and using this information to color Gaussian noise. One technique measures the frequency-dependent amplitude levels with a bank of band-pass filters while the other involves filtering Gaussian noise with a perceptual-based filter. Both methods work well and eliminate the undesired artifact from the resulting impulse response.

Naturally, an impulse response with a large noise floor is not ideal. In such cases, the techniques described above can be used to prepare an impulse response measurement for further processing, such as described in [6], to extend the measurement through the noise floor. The result of such processing applied to another of the BAM measurements appears in Fig. 8. Note that the extended impulse response shows none of the time-varying low-pass filtering artifacts present in the original measured impulse response.

6. ACKNOWLEDGMENTS

The authors would like to thank Jacqueline Gordon for bringing the BAM acoustics measurement project [9] to our attention, and for setting us onto this work.

7. REFERENCES

- [1] Angelo Farina, “Simultaneous measurement of impulse response and distortion with a swept-sine technique,” in *Audio Engineering Society Convention 108*, Feb 2000.
- [2] Swen Müller and Paulo Massarani, “Transfer-function measurement with sweeps,” *J. Audio Eng. Soc.*, vol. 49, no. 6, pp. 443–71, 2001.
- [3] Angelo Farina, “Advancements in impulse response measurements by sine sweeps,” in *Audio Engineering Society Convention 122*, May 2007.
- [4] Jonathan S. Abel, Nicholas J. Bryan, and Miriam A. Kolar, “Impulse Response Measurements in the Presence of Clock Drift,” in *Audio Engineering Society Convention 129*, Nov 2010.
- [5] 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*, Sep 1997.
- [6] Nicholas J. Bryan and Jonathan S. Abel, “Methods for extending room impulse responses beyond their noise floor,” in *Audio Engineering Society Convention 129*, Nov 2010.
- [7] Antonin Novak, Laurent Simon, Frantisek Kadlec, and Pierrick Lotton, “Nonlinear system identification using exponential swept-sine signal,” *Instrumentation and Measurement, IEEE Transactions on*, vol. 59, no. 8, pp. 2220–9, 2010.
- [8] Jonathan S. Abel and David P. Berners, “A technique for nonlinear system measurement,” in *Audio Engineering Society Convention 121*, Oct 2006.
- [9] Jacqueline Gordon and Zackery Belanger, “Acoustic deconstruction of 2626 Bancroft Way, kickstarter project 632980650,” 2015.

CONSTRAINED POLE OPTIMIZATION FOR MODAL REVERBERATION

Esteban Maestre

CAML / CIRMMT
Schulich School of Music
McGill University
Montréal, Canada
esteban@music.mcgill.ca

Jonathan S. Abel, Julius O. Smith

CCRMA
Music Department
Stanford University
Stanford, USA
abel@ccrma.stanford.edu
jos@ccrma.stanford.edu

Gary P. Scavone

CAML / CIRMMT
Schulich School of Music
McGill University
Montréal, Canada
gary@music.mcgill.ca

ABSTRACT

The problem of designing a modal reverberator to match a measured room impulse response is considered. The modal reverberator architecture expresses a room impulse response as a parallel combination of resonant filters, with the pole locations determined by the room resonances and decay rates, and the zeros by the source and listener positions. Our method first estimates the pole positions in a frequency-domain process involving a series of constrained pole position optimizations in overlapping frequency bands. With the pole locations in hand, the zeros are fit to the measured impulse response using least squares. Example optimizations for a medium-sized room show a good match between the measured and modeled room responses.

1. INTRODUCTION

Modal synthesis has long been used in computer music to simulate large resonating acoustic structures [1]. It was arguably first understood by Daniel Bernoulli circa 1733 [2], when he realized that acoustic vibrations could be seen as a superposition of pure harmonic (sinusoidal) vibrations. Constituting a flexible and efficient approach as compared to convolution techniques, modal structures have been recently proposed for implementing reverberation: [3, 4] suggest synthesizing late-field room reverberation using randomly generated modes; [5, 6] describe how to use measurements to design room reverberation and electromechanical reverberators from spectral peak picking, and implement them as a parallel sum of resonators in a structure termed a “modal reverberator.”

The modal reverberator relies on a numerically robust parallel structure [7] that provides computational advantages, as it can be efficiently computed, and only audible modes need to be implemented. The parallel decomposition leads to precise control over decay rate, equalization, and other reverberation features, as they can be individually adjusted on a mode by mode basis, and are easily slewed over time. Another advantage of the modal reverberator structure is that separate parameters control the spatial and temporal features of the reverberation: The mode frequencies and dampings are properties of the room or resonating object, describing the mode oscillation frequencies and mode decay times. The mode amplitudes are determined by the source and listener positions according to the mode spatial patterns (given room dimensions and boundary conditions). In this way, the poles of the resonant filters are fixed according to the room, and the zeros are derived from the source and listener positions within the room. In this work, we are concerned with designing a modal reverberator—that is, finding the poles and zeros of each resonant mode filter—so that its output approximates a given measured room response.

In the context of adaptive acoustic echo cancellation, [8, 9] proposed methods for estimating the poles and zeros of moderately low-order transfer functions used to represent the low-frequency region of room responses: using multiple impulse responses obtained for different source and listener positions in the same room, a set of “common acoustical poles” are first estimated using a least-squares technique; then, different sets of zeros are estimated and interpolated to model the localization of the source and listener.

In [10], low-frequency modes were identified in the room impulse response spectrogram from local peaks in estimated reverberation time as a function of frequency. This approach can successfully find modes and their associated poles and zeros, though it is only applicable to long-lasting, low-frequency modes.

In [6], the resonant filter parameters were estimated from a measured room impulse by first finding the mode frequencies from peaks in the room impulse response magnitude spectrum. The number of modes was given by the number of spectral peaks above a given threshold relative to the critical-band-smoothed magnitude spectrum. The mode dampings were estimated from the reverberation time in a band of frequencies about the respective mode frequencies. Finally, the mode amplitudes were found via least-squares fit to the measured impulse response, given the estimated mode frequencies and dampings. While this approach produced a good match between the measured and modeled impulse responses, and could be implemented in such a way as to generate audio effects such as pitch shifting and distortion [11, 12], it was thought that further optimization could noticeably improve the result. Consider a pair of modes that are close in frequency. The stronger mode would likely bias the peak frequency of the weaker mode, as its spectral peak would be on the sloping shoulder of the stronger resonance. In addition, while it is common for adjacent modes to have dampings that differ by a fair amount, the damping assigned to adjacent modes using the approach of [6] would be nearly identical.

In this work, we use an approach similar to that of [13, 14, 15, 16] to optimize initial estimates of the mode pole parameters. As a medium-sized room could have upwards of a couple thousand modes to be modeled, our approach optimizes the mode parameters in a set of overlapping frequency bands so that any given band optimization has a manageable number of parameters. Once the mode pole parameters (equivalently, the mode frequencies and dampings) are estimated, a linear least-squares fit to the measured impulse response is used to find the mode filter zeros.

The rest of the paper is organized as follows. Section 2 introduces a modal reverberator parallel structure and outlines the procedure for pole-zero design from an impulse response measurement. Section 3 describes the procedure used for pole initialization, and Section 4 describes the pole optimization algorithm. Finally,

Section 5 presents some preliminary results and Section 6 discusses potential improvements and applications.

2. MODAL REVERBERATOR DESIGN

Given a reverberation impulse response measurement $h(t)$ and an input signal $x(t)$, one can obtain the *reverberated* output signal $y(t)$ from $x(t)$ by $y(t) = h(t) * x(t)$, where '*' denotes convolution. The *modal synthesis* approach [1] approximates $h(t)$ by a sum of M parallel components $h_m(t)$, each corresponding to a resonant mode of the reverberant system $h(t)$. In the z -domain, each parallel term can be realized as a recursive second-order system $H_m(z)$. This is expressed as

$$Y(z) = \hat{H}(z)X(z) = \sum_{m=1}^M H_m(z)X(z), \quad (1)$$

where $\hat{H}(z)$ is a $2M$ -order digital approximation of $h(t)$, and each m -th parallel term $H_m(z)$ is

$$H_m(z) = (g_{0,m} + g_{1,m}z^{-1})R_m(z) \quad (2)$$

with real numerator coefficients $g_{0,m}$ and $g_{1,m}$, and

$$R_m(z) = \frac{1}{1 + a_{1,m}z^{-1} + a_{2,m}z^{-2}} \quad (3)$$

is a resonator with real coefficients defined by a pair of complex-conjugate poles p_m and p_m^* . Denominator coefficients are related to pole angle and radius by $a_{1,m} = -2|p_m| \cos \angle p_m$ and $a_{2,m} = |p_m|^2$, and define the m -th modal frequency f_m and bandwidth β_m via $f_m = f_s \angle p_m / 2\pi$ and $\beta_m = -f_s \log|p_m|/\pi$ respectively, where f_s is the sampling frequency in Hz. Numerator coefficients $g_{0,m}$ and $g_{1,m}$ are used to define the complex gain of the m -th mode [17].

2.1. Design problem

The problem of designing $\hat{H}(z)$ from a given measurement $h(t)$ involves a modal decomposition, and it can be posed as

$$\underset{\mathbf{p}, \mathbf{g}_0, \mathbf{g}_1}{\text{minimize}} \quad \varepsilon(\hat{H}, H), \quad (4)$$

where \mathbf{p} is a set of M complex poles inside the upper half of the unit circle on the z -plane, \mathbf{g}_0 and \mathbf{g}_1 are two sets of M real coefficients, and $\varepsilon(\hat{H}, H)$ is an error measure between the model and the measurement. To find good approximations of dense impulse responses, one needs to face decompositions on the order of hundreds or thousands of highly overlapping modes. In this work we solve this non-linear problem in two steps: first, we find a convenient set of M modes via constrained optimization of complex poles \mathbf{p} ; second, we obtain the modal complex gains by solving a linear problem to find coefficients $\mathbf{g}_0, \mathbf{g}_1$ as described in Section 2.2.

2.2. Estimation of modal gains

Given a target frequency response $H(e^{j\omega})$ and M complex poles $p_1 \cdots p_m \cdots p_M$, it is straightforward to solve for the numerator coefficients by formulating a linear problem. Let vector $\mathbf{h} = [h_1 \cdots h_k \cdots h_K]^T$ contain K samples of $H(e^{j\omega})$ at normalized angular frequencies $0 \leq \omega_k < \pi$, i.e., $h_k = H(e^{j\omega_k})$. Likewise, let vector $\mathbf{r}_m^0 = [r_{m,1}^0 \cdots r_{m,k}^0 \cdots r_{m,K}^0]^T$ sample the frequency

response of $R_m(z)$ with $r_{m,k}^0 = R_m(e^{j\omega_k})$, and vector $\mathbf{r}_m^1 = [r_{m,1}^1 \cdots r_{m,k}^1 \cdots r_{m,K}^1]^T$ the frequency response of $z^{-1}R_m(z)$ with $r_{m,k}^1 = e^{-j\omega_k}R_m(e^{j\omega_k})$. Next, let \mathbf{Q} be the $K \times 2M$ matrix of basis vectors constructed as $\mathbf{Q} = [\mathbf{r}_1^0 \cdots \mathbf{r}_M^0, \mathbf{r}_1^1 \cdots \mathbf{r}_M^1]$. Finally, let vector \mathbf{g} contain the numerator coefficients arranged as $\mathbf{g} = [g_{0,1} \cdots g_{0,M}, g_{1,1} \cdots g_{1,M}]^T$. Now we can solve the least-squares projection problem

$$\underset{\mathbf{g}}{\text{minimize}} \quad \|\mathbf{Q}\mathbf{g} - \mathbf{h}\|^2. \quad (5)$$

2.3. Pole optimization

Recently we proposed methods for modal decomposition of string instrument bridge admittance measurements (e.g., [14, 16]) using constrained pole optimization via quadratic programming as described in [15], with complexities in the order of dozens of modes. In such methods, a quadratic model is used at each step to numerically approximate the gradient of an error function that spans the full frequency band. Because in this work we deal with decompositions of much higher order, using this technique to simultaneously optimize hundreds or thousands of poles is impractical. Therefore, we propose here an extension of the method to allow for several pole optimizations to be carried out separately, for different overlapping frequency bands. Optimized poles are then collected and merged into a single set from which numerator coefficients are estimated by least-squares as described above leading to (5).

3. INITIALIZATION PROCEDURE

A trusted initial set of pole positions is essential for nonlinear, non-convex optimization. Once the order of the system is imposed, pole initialization comprises two main steps: modal frequency estimation and modal bandwidth estimation. Modal frequency estimation is based on spectral peak picking from the frequency response measurement, while modal bandwidths are estimated from analyzing the energy decay profile as obtained from a time-frequency representation of the impulse-response measurement.

To impose a modal density given the model order M , we assume that the decreasing frequency-resolution at higher frequencies in human hearing makes it unnecessary to implement the ever-increasing modal density of reverberant systems. To that end, we devised a peak-picking procedure to favor a uniform modal density over a warped frequency axis approximating a Bark scale [18].

3.1. Estimation of modal frequencies

Estimation of modal frequencies is based on analysis of the log-magnitude spectrum $\Upsilon(f)$ of the impulse response. From $\Upsilon(f)$, all N peaks in $f_{\min} \leq f < f_{\max}$, with $0 \leq f_{\min} < f_{\max} < f_s/2$, are found by collecting all N local maxima after smoothing of $\Upsilon(f)$. For each peak, a corresponding peak salience descriptor is computed by frequency-warping and integration of $\Upsilon(f)$ around each peak, as described in [16].

Next, a total of B adjacent bands are defined between f_{\min} and f_{\max} , each spanning from delimiting frequencies f_l^b to f_r^b and following a Bark scale. This is carried out by uniformly dividing a Bark-warped frequency axis ϖ into B portions spread between warped frequencies ϖ_{\min} and ϖ_{\max} , and then mapping delimiting warped frequencies ϖ_l^b and ϖ_r^b to their linear frequency counterparts. To map between linear and warped frequencies, we use the arctangent approximation of the Bark scale as described in [18].

From all N initial peaks and corresponding annotated frequencies and saliences, picking of M modal frequencies is carried out via an iterative procedure that starts from B lists of candidate peaks, each containing the initial N_b peaks that lie inside band b . First, we define the following variables: P as the total number of picked peaks, N as the total number of available peaks, and N^b as the number of available peaks in the b -th band. The procedure for peak picking is detailed next:

```

1: if  $N \leq M$  then
2:   pick all  $N$  peaks
3: else
4:    $P \leftarrow 0$ 
5:   while  $N - (M - P) > 0$  &  $M > P$  do
6:      $A \leftarrow \lfloor (N - P)/B \rfloor$ 
7:      $C \leftarrow (N - P) \bmod B$ 
8:      $b \leftarrow 1$ 
9:     while  $b \leq B$  do
10:     $D \leftarrow A + C \bmod b$ 
11:     $E \leftarrow \min(D, N^b)$ 
12:    pick the highest-salience  $E$  peaks in list  $b$ ;
        remove those  $E$  peaks from list  $b$ .
13:     $N_b \leftarrow N^b - E$ 
14:     $N \leftarrow N - E$ 
15:     $P \leftarrow P + E$ 
16:   end while
17: end while
18: end if
```

Once the M peaks have been selected, parabolic interpolation of $\Upsilon(f)$ around each m -th peak is used for defining the m -th modal frequency f_m .

3.2. Estimation of modal bandwidths

Modal bandwidth estimation starts from computing a spectrogram of the impulse response, leading to a time-frequency log-magnitude representation $\Upsilon(t, f)$ from which we want to estimate the decay rate of all L frequency bins within $f_{\min} \leq f < f_{\max}$. Several methods have been proposed for decay rate estimation in the context of modal reverberation (see [3] and references therein). Since in our case we only aim at providing a trusted initial estimate of the bandwidth of each mode, we employ a simple method based on linear interpolation of a decaying segment of the magnitude envelope $\Upsilon(t, f_l)$ of each l -th frequency bin, as follows.

The decaying segment for the l -th bin is delimited by start time t_s^l and end time t_e^l . First, t_s^l is defined as the time at which $\Upsilon(t, f_l)$ has fallen 10 dB below the envelope maximum. Second, t_e^l is defined as the time at which $\Upsilon(t, f_l)$ has reached a noise floor threshold Υ_{th} . To set Υ_{th} , we first estimate the mean μ_n and standard deviation σ_n of the noise floor magnitude, and then define $\Upsilon_{\text{th}} = \mu_n + 2\sigma_n$.

By linear interpolation of the decay segment of each l -th frequency bin, the decay rate τ_l is obtained from the estimated slope, and the bandwidth β_l is computed as $\beta_l = 1/\pi\tau_l$, leading to the bandwidth profile $\beta(f_l)$. Finally, each modal bandwidth β_m is obtained by interpolation of $\beta(f_l)$ at its corresponding modal frequency f_m .

3.3. Pole positioning

Since optimization is carried out in the z -plane, we use estimated frequencies and bandwidths to define M initial pole positions inside

the unit circle on the z -plane by computing their angle ω_m and radius $|p_m|$ as $\omega_m = 2\pi f_m/f_s$ and $|p_m| = e^{-\pi\beta_m/f_s}$. We use \mathbf{p}^x to denote the vector of initial pole positions.

4. POLE OPTIMIZATION ALGORITHM

Pole optimization is carried out by dividing the problem into many optimization subproblems, each focused on one of B frequency bands and dealing with a subset of the initial poles. In each b -th subproblem, we optimize only those poles for which corresponding modal frequencies lay inside the b -th frequency band. To help with mode interaction around the band edges, we configure the bands to be partially overlapping. After optimization, we collect optimized poles only from the non-overlapping regions of the bands.

Prior to segmenting into bands or performing any optimization, we use the set \mathbf{p}^x of initial M poles to solve problem (5). This leads to initial gain coefficient vectors \mathbf{g}_0^x and \mathbf{g}_1^x . These, together with initial poles \mathbf{p}^x , are used to support band pre-processing and optimization as detailed below.

4.1. Band preprocessing

We first uniformly divide the Bark-warped frequency axis ϖ into B adjacent bands between ϖ_{\min} and ϖ_{\max} . Band edges of the b -th band are ϖ_r^b and ϖ_l^b . Next, an overlapping region is added to each side of each band, extending the edges from ϖ_r^b and ϖ_l^b to ϖ_L^b and ϖ_R^b respectively. The outer edges are defined as $\varpi_L^b = \varpi_r^b - \delta^b(\varpi_r^b - \varpi_l^b)$ and $\varpi_R^b = \varpi_r^b + \delta^b(\varpi_r^b - \varpi_l^b)$, with δ^b being a positive real number. The outer edges define the b -th (extended) optimization frequency band, on which the b -th optimization problem is focused. This is illustrated in Figure 1, where each optimization frequency band is conformed by three subbands: a center subband matching the original, non-overlapping band, and two side subbands.

Once the optimization bands have been configured on the warped frequency axis ϖ , all band edges are mapped back to the linear frequency axis f . Then, for each b -th band, two sets of poles \mathbf{p}^b and $\mathbf{p}^{\neg b}$ are created from the initial set \mathbf{p}^x as follows. Set \mathbf{p}^b includes all U poles whose modal frequencies f_u are in $f_L^b \leq f_u < f_R^b$, while set $\mathbf{p}^{\neg b}$ includes the remaining O poles (with $O = M - U$), i.e., those poles whose modal frequency f_o is not in $f_L^b \leq f_o < f_R^b$. Moreover, from initial gain vectors \mathbf{g}_0^x and \mathbf{g}_1^x we retrieve the gains corresponding to poles in \mathbf{p}^b and to poles in $\mathbf{p}^{\neg b}$, leading to two pairs of gain vectors $\mathbf{g}_0^b, \mathbf{g}_1^b$ and $\mathbf{g}_0^{\neg b}, \mathbf{g}_1^{\neg b}$. Finally, with arranged sets of poles and gains, we construct two models $H^b(z)$ and $H^{\neg b}(z)$ of the form of $\hat{H}(z)$ in (1). The first model,

$$H^b(z) = \sum_{u=1}^U (g_{0,u}^b + g_{1,u}^b z^{-1}) R_u^b(z) \quad (6)$$

with resonators $R_u^b(z)$ constructed from poles \mathbf{p}^b , will be optimized to solve the b -th band subproblem (see Section 4.2). The second model,

$$H^{\neg b}(z) = \sum_{o=1}^O (g_{0,o}^{\neg b} + g_{1,o}^{\neg b} z^{-1}) R_o^{\neg b}(z) \quad (7)$$

with resonators $R_o^{\neg b}(z)$ constructed from poles $\mathbf{p}^{\neg b}$, presents fixed coefficients and is used to pre-synthesize a frequency response to be used as a constant offset during optimization of the model (6) above

(see Section 4.3). We include this offset response to account for how (fixed) off-band modes contribute to the frequency response of model (6) during optimization.

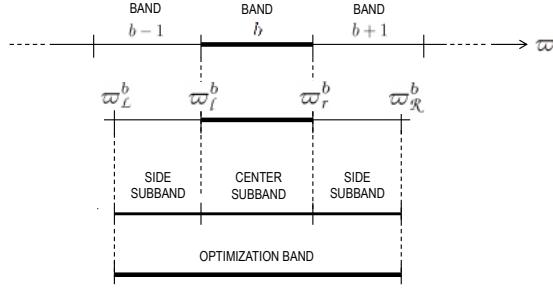


Figure 1: Optimization preprocessing: partition into overlapping frequency bands on a Bark-warped frequency axis $\bar{\omega}$ (depicted is band b).

4.2. Band optimization

We parametrize the initial set of U modes inside the b -th band by representing each u -th mode as a z -plane complex pole pair p_u^b in terms of two parameters: an angle parameter $w_u^b = |\angle p_u^b|$ and a radius parameter $s_u^b = -\log(1 - |p_u^b|)$. This leads to two parameter sets: a set $\mathbf{w}^b = \{w_1^b \dots w_U^b\}$ of angle parameter values, and a set $\mathbf{s}^b = \{s_1^b \dots s_U^b\}$ of radius parameter values. With this parametrization, we state the b -th band optimization problem as

$$\begin{aligned} & \underset{\mathbf{w}^b, \mathbf{s}^b}{\text{minimize}} && \varepsilon(\hat{H}^b, H^b) \\ & \text{subject to} && \mathbf{C}^b, \end{aligned} \quad (8)$$

where \mathbf{C}^b is a set of linear constraints specific to band b and $\varepsilon(\hat{H}^b, H^b)$ is an error function, also specific to band b , computed as described in Section 4.3. Note that numerator coefficients have been left out as they are not exposed as variables in the optimization (see [15]). Constraints \mathbf{C}^b are used to restrict the position and arrangement of poles inside the b -th unit circle sector used to represent the b -th band on the z -plane. We have schematically represented the optimization process in Figure 2. We map band edge frequencies f_L^b and f_R^b to sector edge angles ω_L^b and ω_R^b via $\omega_L^b = 2\pi f_L^b / f_s$ and $\omega_R^b = 2\pi f_R^b / f_s$ respectively.

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 b -th unit circle sector is preserved during optimization, therefore reducing the number of crossings over local minima (see [15]). Elements in sets \mathbf{w}^b and \mathbf{s}^b are jointly sorted as pairs (each pair corresponding to a complex-conjugate pole) by ascending angle parameter w_u .

From ordered sets \mathbf{w}^b and \mathbf{s}^b , linear constraints \mathbf{C}^b are defined as follows. First, stability is enforced by $0 < s_u^b$. Then, poles are constrained to stay in the b -th sector via $\omega_L^b \leq w_u^b < \omega_R^b$. Next, to aid convergence we constrain the pole sequence order in set \mathbf{w}^b to be respected. This is expressed by $w_{u-1}^b < w_u^b < w_{u+1}^b$. 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 $w^- < w_u^b < w^+$ and $s^- < s_u^b < s^+$, where ' $-$ ' and ' $+$ ' superscripts are used to indicate lower and upper bounds, respectively.

We solve this problem by means of sequential quadratic programming [19]. At each step, the error surface is quadratically approximated by successive evaluations of the band approximation error function described in Section 4.3.

4.3. Band error computation

At each i -th step of the optimization, given a set of poles $\mathbf{p}^b|_i$ defined from current values in parameter sets \mathbf{w}^b and \mathbf{s}^b , the error $\varepsilon(\hat{H}^b|_i, H^b)$ is computed by solving a linear problem similar to 5, but restricted to the b -th frequency band.

First, let vector $\mathbf{h} = [h_1 \dots h_K \dots h_K]^T$ contain K samples of the measurement $H(e^{j\omega})$ at normalized angular frequencies $\omega_L^b \leq \omega_k < \omega_R^b$, i.e., $h_k = H(e^{j\omega_k})$. Similarly, let vector $\mathbf{v} = [v_1 \dots v_K \dots v_K]^T$ contain K samples of the frequency response of model $\hat{H}^{-b}(z)$, i.e., $v_k = \hat{H}^{-b}(e^{j\omega_k})$. Then, we obtain U frequency response vectors $\mathbf{r}_u^0|_i = [r_{u,1}^0|_i \dots r_{u,K}^0|_i \dots r_{u,K}^0|_i]^T$ by using $\mathbf{p}^b|_i$ to evaluate each $R_u^b(z)$ in the same frequency range, i.e., $r_{u,k}^0|_i = R_u^b(e^{j\omega_k})|_i$. Likewise, we obtain U vectors $\mathbf{r}_u^1|_i = [r_{u,1}^1|_i \dots r_{u,k}^1|_i \dots r_{u,K}^1|_i]^T$ by evaluating each $z^{-1}R_u^b(z)$, i.e., $r_{u,k}^1|_i = e^{-j\omega_k} R_u^b(e^{j\omega_k})|_i$. Next, all $2U$ vectors $\mathbf{r}_u^0|_i$ and $\mathbf{r}_u^1|_i$ are arranged to form a basis matrix $\mathbf{Q}^b|_i$ of size $K \times 2U$ as $\mathbf{Q}^b|_i = [\mathbf{r}_1^0|_i \dots \mathbf{r}_U^0|_i, \mathbf{r}_1^1|_i \dots \mathbf{r}_U^1|_i]$. Finally, let vector \mathbf{g}^b contain the numerator coefficients of model $H^b(z)$ arranged as $\mathbf{g}^b = [g_{0,1}^b \dots g_{0,U}^b, g_{1,1}^b \dots g_{1,U}^b]^T$. With all these, we solve the least-squares problem

$$\underset{\mathbf{g}^b}{\text{minimize}} \quad \|\mathbf{Q}^b|_i \mathbf{g}^b + \mathbf{v} - \mathbf{h}\|^2 \quad (9)$$

and use obtained vector \mathbf{g}^b to compute the i -th step error as

$$\varepsilon(H^b|_i, \hat{H}^b) = \|\mathbf{Q}^b|_i \mathbf{g}^b + \mathbf{v} - \mathbf{h}\|^2. \quad (10)$$

4.4. Pole collection and problem solution

Once the poles of the B overlapping frequency bands have been optimized, the final set poles \mathbf{p} is constructed by collecting all optimized poles inside each of the B center subsectors, i.e., all poles whose angle parameter w satisfy $\omega_L^b \leq w < \omega_R^b, \forall b \in \{1 \dots B\}$. Using collected poles, we solve problem (5) of Section 2.2.

5. EXAMPLE RESULTS

We carried out a set of test examples to model an impulse response measurement taken from a medium-sized room at a sampling frequency of 48000 Hz. In our models, we explored different orders $M = 800, 1200, 1600, 1800$, with modes in the region between 30 Hz and 20 kHz. In all cases, we chose to use a sufficiently large number of bands $B = 200$, with a constant overlapping $\delta^b = 1.0$. Example synthesized impulse responses are available online¹.

In Figure 3 we display the spectrogram of the target impulse response plus three example models of order $M = 800, 1200, 1800$. While we observe an overall good match both in frequency and time, it is clear how the response of lower order models (e.g., $M = 800$) present a more sparse modal structure, especially in the high end.

¹<http://ccrma.stanford.edu/~esteban/modrev/dafx2017>

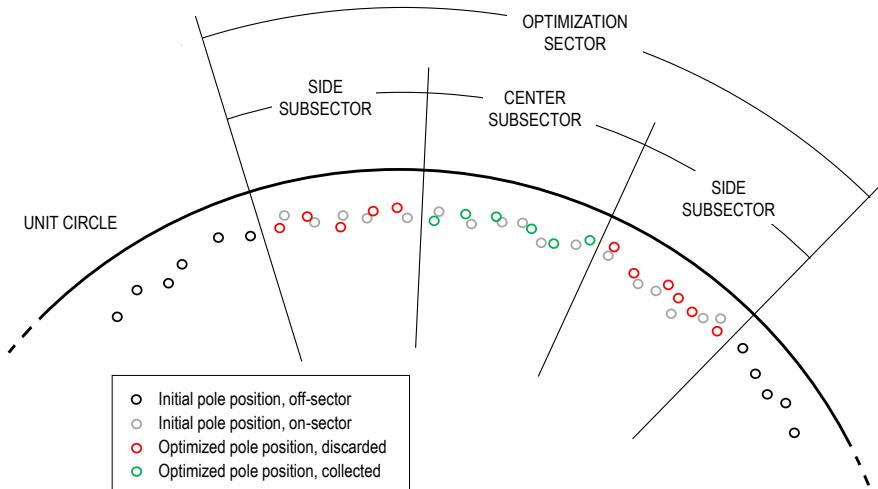


Figure 2: Schematic representation of pole optimization inside a frequency band, which is mapped to a sector of the unit circle on the z -plane.

As it can be perceived from listening to the modeled responses, this leads to a metallic character in the sound.

To get an idea of the how the modeling error is reduced during optimization, we compare the error $H(\omega) - \hat{H}(\omega)$ obtained before and after optimization. This is featured in Figure 4 for $M = 1600$, where it is possible to observe the coarse envelope of the error to decrease by 5 to 10 dB in all frequency regions.

For a detail of how optimization improves the modeling accuracy, in Figure 5 appear the magnitude responses of the target measurement (middle), initial (bottom) and optimized models (top) of $M = 1600$ for three different frequency regions. With regard to the time domain, initial and optimized impulse response models are compared to the target measurement in Figure 6, where it is possible to observe how optimization helps to significantly reduce the pre-onset ripple of the model.

6. CONCLUSION

We have presented a frequency-domain pole-zero optimization technique to fit the coefficients of a modal reverberator, applied to model a mid-sized room impulse-response measurement. Our method, which includes a pole initialization procedure to favor a constant density of modes over a Bark-warped frequency axis, is based on constrained optimization of pole positions within a number of overlapping frequency bands. Once the pole locations are estimated, the zeros are fit to the measured impulse response using linear least squares. Our initial explorations with example models of a medium-sized room display a good agreement between the measured and modeled room responses, demonstrating how our pole optimization technique can be of practical use in modeling problems that require thousands of modes to accurately simulate dense impulse responses.

Our initial results show a promising path for improving the accuracy of efficient modal reverberators. At the same time, optimization could lead to a reduction of the computational cost given a required accuracy. Besides carrying out a more exhaustive exploration of model orders and parameter values (e.g. frequency-dependent overlapping factor), gathering data from subjective tests

could provide a good compromise between perceptual quality and computational cost. In terms of constraint definition, upper and lower bounds are still set by hand—further exploration could lead to methods for designing bounds for pole radii by attending to a statistical analysis of observed modal decays and therefore avoid an excess in pole selectivity. A potential development aims at making the whole process to be iterative: use the optimized solution as an initial point, perform again the optimization, and repeat until no improvement is obtained; this could perhaps have a positive effect in the vicinity of band edges, though its significance would need to be tested.

In conclusion, we note that *modal synthesis* supports a *dynamic* representation of a space, at a fixed cost, so that one may simulate walking through the space or listening to a moving source by simply changing the coefficients of the fixed modes being summed. The modal approach may thus be considered competitive with three-dimensional physical models such as Scattering Delay Networks [20, 21, 22] in terms of psychoacoustic accuracy per unit of computational expense. In that direction, an imminent extension of our method deals with simultaneously using N impulse response measurements of the same space (a set of measurements taken for different combinations of source and receiver locations, as previously proposed in [8] for low-order equalization models of the low frequency region) to optimize a common set of poles: at each step, poles in each frequency band are optimized via minimization of a global error function that simultaneously accounts for the approximation error of N models (one per impulse response) constructed from the same set of poles. A common set of poles suffices for each fixed spatial geometry, or coupled geometries.

7. REFERENCES

- [1] J. M. Adrien, “The missing link: Modal synthesis,” in *Representations of Musical Signals*, G. De Poli, A. Piccialli, and C. Roads, Eds., pp. 269–267. MIT Press, 1991.
- [2] O. Darrigol, “The acoustic origins of harmonic analysis,” *Archive for History of the Exact Sciences*, vol. 61, no. 4, 2007.

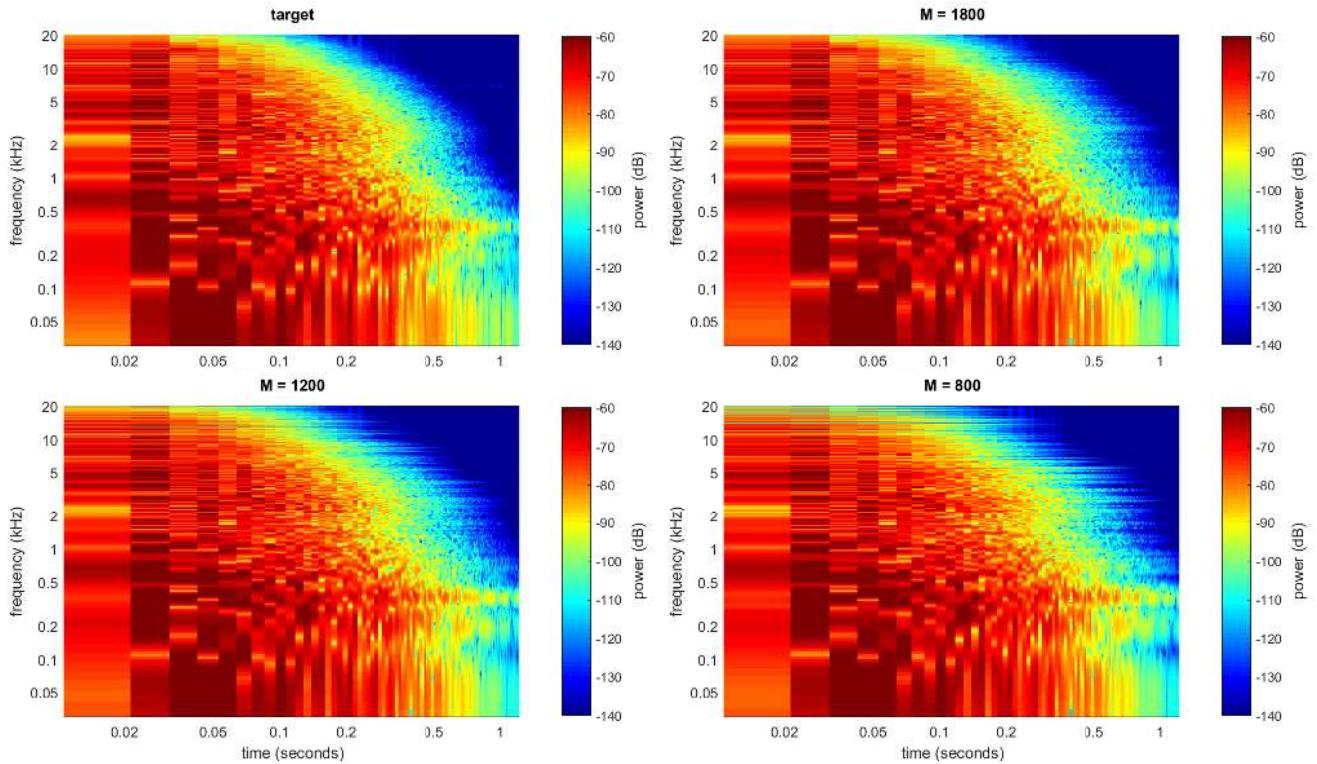


Figure 3: Spectrogram of the target impulse response plus three example models with $M = 800, 1200, 1800$.

- [3] M. Karjalainen, P. Antsalo, A. Mäkivirta, T. Peltonen, and V. Välimäki, “Estimation of modal decay parameters from noisy response measurements,” *Journal of the Audio Engineering Society*, vol. 50:11, pp. 867–878, 2002.
- [4] M. Karjalainen and H. Jarvelainen, “More about this reverberation science: Perceptually good late reverberation,” in *Audio Engineering Society Convention 111*, New York, U.S.A., 2001.
- [5] J. S. Abel, S. A. Coffin, and K. S. Spratt, “A modal architecture for artificial reverberation,” *The Journal of the Acoustical Society of America*, vol. 134:5, pp. 4220, 2013.
- [6] J. S. Abel, S. A. Coffin, and K. S. Spratt, “A modal architecture for artificial reverberation with application to room acoustics modeling,” in *Proc. of the Audio Engineering Society 137th Convention*, 2014.
- [7] B. Bank, “Perceptually motivated audio equalization using fixed-pole parallel second-order filters,” *IEEE Signal Processing Letters*, vol. 15, pp. 477–480, 2008.
- [8] Y. Haneda, S. Makino, and Y. Kaneda, “Common acoustical pole and zero modeling of room transfer functions,” *IEEE Transactions on Speech and Audio Processing*, vol. 2:2, pp. 320–328, 1994.
- [9] Y. Haneda, Y. Kaneda, and N. Kitawaki, “Common-acoustical-pole and residue model and its application to spatial interpolation and extrapolation of a room transfer function,” *IEEE Transactions on Speech and Audio Processing*, vol. 7:6, pp. 709–717, 1999.
- [10] A. Mäkivirta, P. Antsalo, M. Karjalainen, and V. Välimäki, “Low-frequency modal equalization of loudspeaker-room responses,” in *Audio Engineering Society 111th Convention*, 2001.
- [11] J. S. Abel and K. J. Werner, “Distortion and pitch processing using a modal reverberator,” in *Proc. of the International Conference on Digital Audio Effects*, 2015.
- [12] K. J. Werner and J. S. Abel, “Modal processor effects inspired by Hammond tonewheel organs,” *Applied Sciences*, vol. 6(7), pp. 185, 2016.
- [13] E. Maestre, G. P. Scavone, and J. O. Smith, “Modeling of a violin input admittance by direct positioning of second-order resonators,” *The Journal of the Acoustical Society of America*, vol. 130, pp. 2364, 2011.
- [14] 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.
- [15] 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.
- [16] 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/ACM Transactions on Audio, Speech and Language Processing*, vol. 25:5, pp. 1128–1139, 2017.

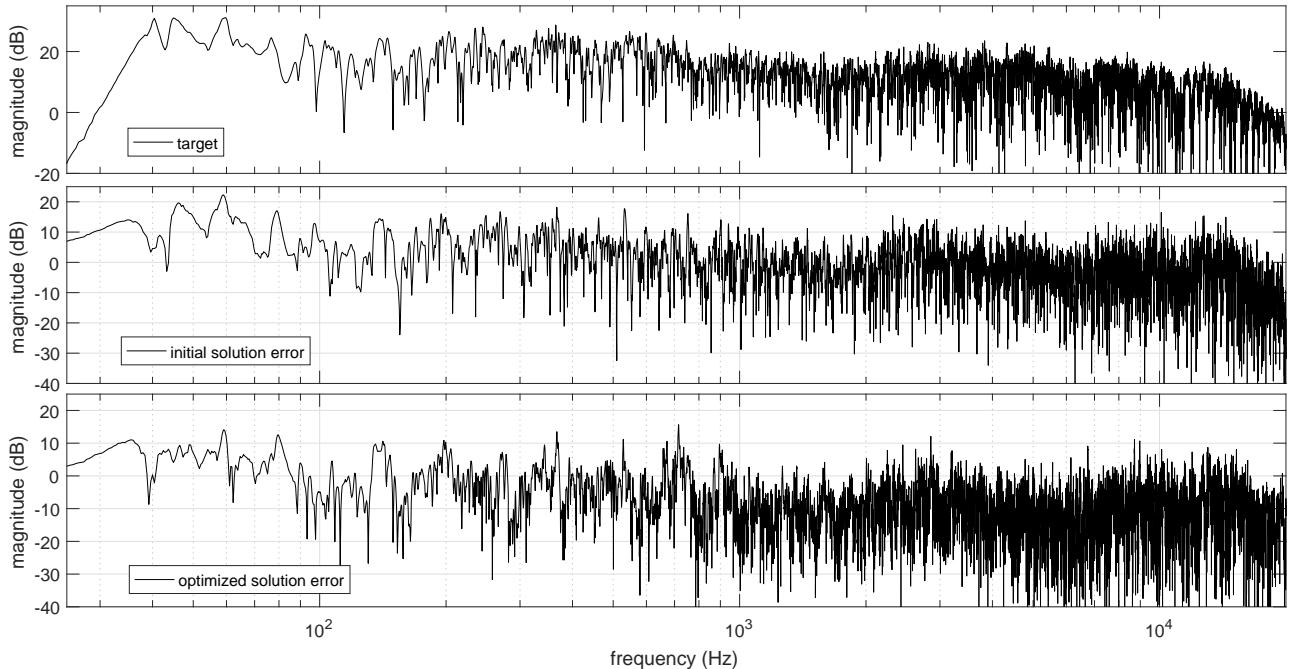


Figure 4: Comparison of the approximation error as a function of frequency, for $M = 1600$.

- [17] J.O. Smith, *Physical Audio Signal Processing*, W3K Publishing, 2004, online book: <http://ccrma.stanford.edu/~jos/pasp/>.
- [18] J. O. Smith and J. S. Abel, “Bark and ERB bilinear transforms,” *IEEE Transactions on Speech and Audio Processing*, vol. 7:6, pp. 697–708, 1999.
- [19] J. Nocedal and S. J. Wright, *Numerical Optimization*, Springer, 2006.
- [20] E. De Sena, H. Hacihabiboglu, Z. Cvetkovic, and J.O. Smith, “Efficient synthesis of room acoustics via scattering delay networks,” *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 23, no. 9, pp. 1478–1492, 2015.
- [21] H. Hacihabiboglu, E. De Sena, Z. Cvetkovic, J. Johnston, and J. O. Smith, “Perceptual spatial audio recording, simulation, and rendering,” *IEEE Signal Processing Magazine*, pp. 36–54, 2017.
- [22] A. Meacham, L. Savioja, S. R. Martín, and J. O. Smith, “Digital waveguide network reverberation in non-convex rectilinear spaces,” in *Audio Engineering Society 141st Convention*, 2016.

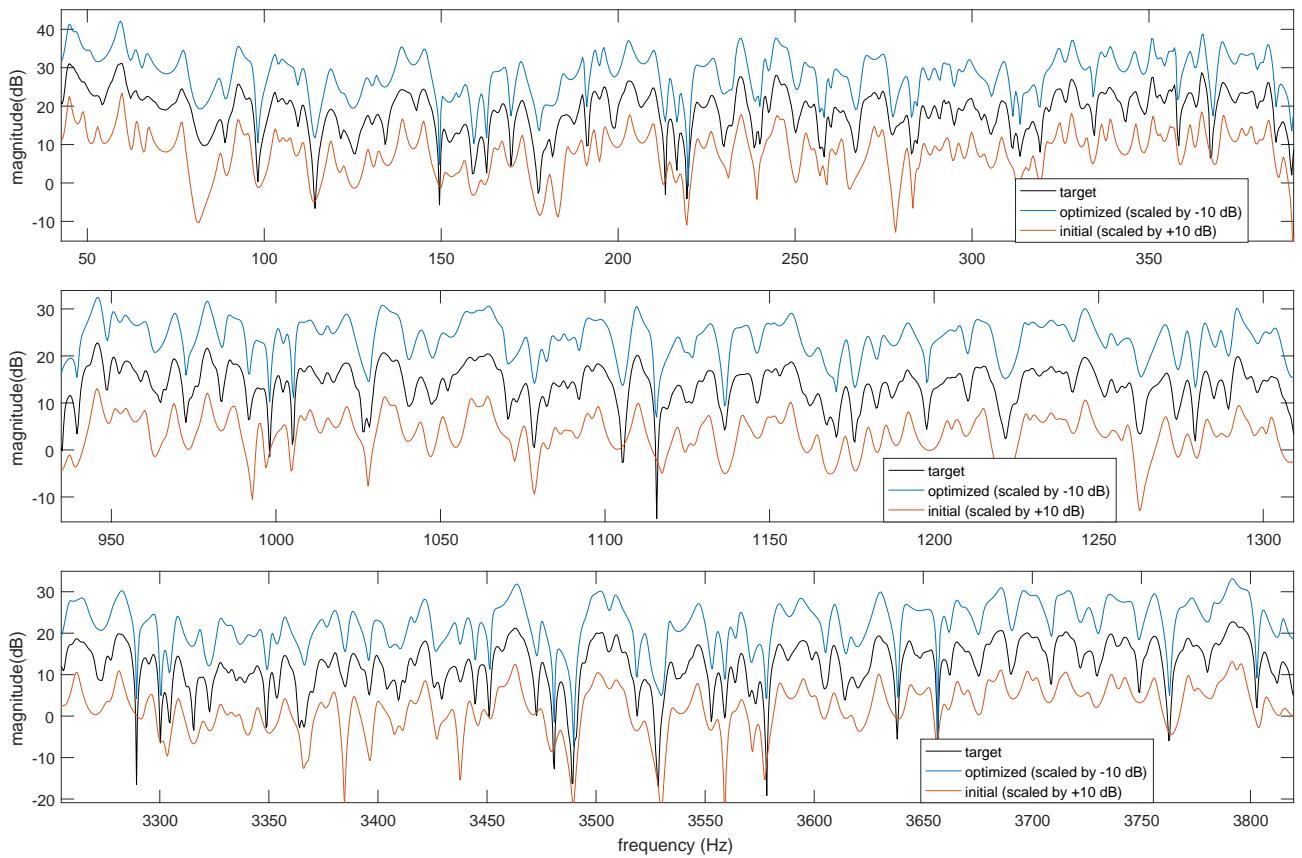


Figure 5: Magnitude response of an example model with $M = 1600$, displayed for three different frequency regions.

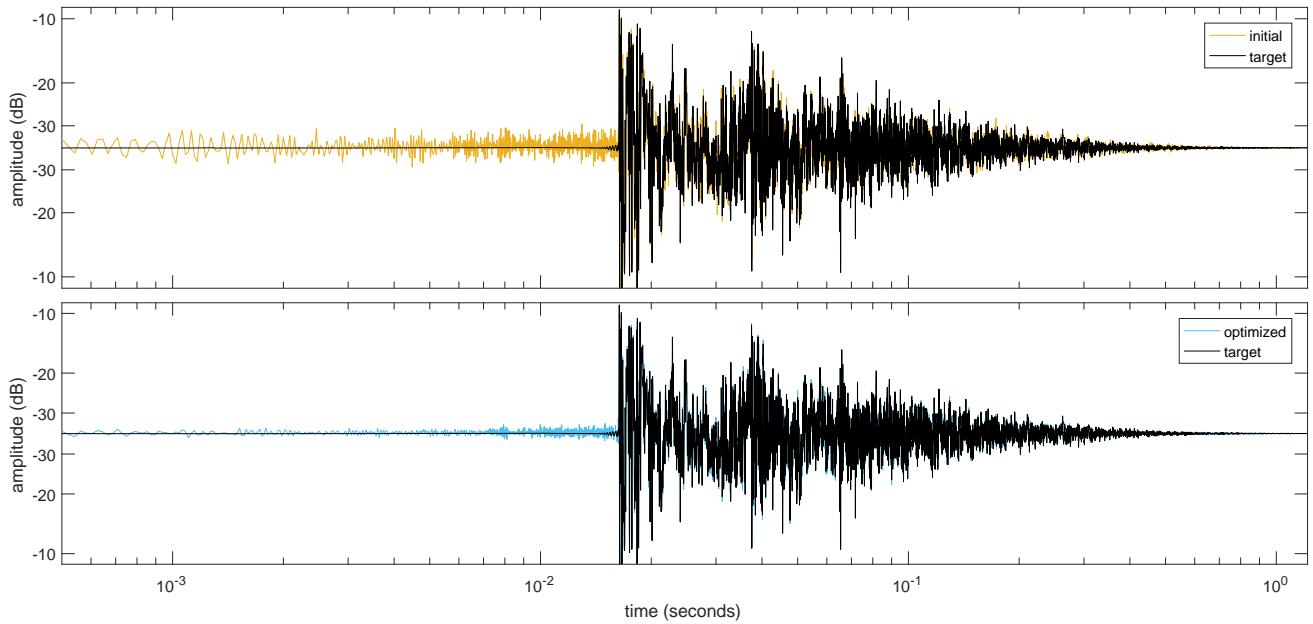


Figure 6: Initial (top) and optimized (bottom) impulse response example models, for $M = 1600$.

DIFFUSE-FIELD EQUALISATION OF FIRST-ORDER AMBISONICS

Thomas McKenzie, Damian Murphy, Gavin Kearney

Audio Lab,
Communication Technologies Research Group
Department of Electronic Engineering
University of York
York, UK
ttm507@york.ac.uk

ABSTRACT

Timbre is a crucial element of believable and natural binaural synthesis. This paper presents a method for diffuse-field equalisation of first-order Ambisonic binaural rendering, aiming to address the timbral disparity that exists between Ambisonic rendering and head related transfer function (HRTF) convolution, as well as between different Ambisonic loudspeaker configurations. The presented work is then evaluated through listening tests, and results indicate diffuse-field equalisation is effective in improving timbral consistency.

1. INTRODUCTION

With the recent increased interest in virtual reality due to the development of high resolution head mounted displays (HMDs) that utilise low latency head tracking, it is desirable to have a matching aural experience that is more realistic than stereophony. Commercial spatial audio systems need to be suitable for a wide audience, be portable and require minimal configuration and calibration. Binaural audio is a spatial audio approach that aims to reproduce the natural localisation cues that allow humans to discern the position of sound sources: primarily the interaural time and level differences (ITD and ILD, respectively) and spectral cues produced by the shape, position and orientation of the ears, head and body. These cues can be recorded and stored as head related transfer functions (HRTFs).

The most convincing binaural systems are the most natural sounding ones [1]. As spectral changes are the biggest differentiating factor between simulation and reality [2], timbre is a vital consideration for binaural reproduction. All parts of the binaural simulation chain affect timbre, from the transducers and equipment used in the recording and reproduction stages to signal processing. A timbrally transparent binaural experience therefore requires consideration of each part of the process.

Binaural reproduction of a source at any angle to the head can be reproduced through interpolation of a dense grid of HRTFs or through ‘virtualisation’ of loudspeaker arrays using HRTFs at the loudspeaker angles. The naturalness of reproduction for the latter case is therefore dependent on the spatial audio rendering scheme used, such as wave field synthesis [3], vector base amplitude panning [4] or Ambisonics [5].

This paper presents a method for diffuse-field equalisation of three first-order Ambisonic virtual loudspeaker configurations. This study aims to answer the following:

- Whether diffuse-field equalisation increases timbral consistency between Ambisonic binaural rendering and direct HRTF convolution.

- Whether diffuse-field equalisation improves timbral consistency across different first-order Ambisonic virtual loudspeaker configurations.

Though the methods and ideas presented have been implemented for binaural reproduction of Ambisonics, they could be applied to loudspeaker reproduction of Ambisonics too. This paper is structured as follows. Section 2 presents a brief background of theoretical and practical approaches relevant to this study. Section 3 describes the methodology for diffuse-field simulation and equalisation of Ambisonic virtual loudspeakers. Evaluation of the method is presented in Section 4 through listening tests, results and discussion. Finally, conclusions and future directions for the work are summarised in Section 5.

2. BACKGROUND

2.1. Ambisonics

Ambisonics is a spatial audio approach that allows recording, storing and reproduction of a 3D sound field, first introduced by Michael Gerzon in the 1970s [5–7]. Ambisonics is based on spatial sampling and reconstruction of a sound field using spherical harmonics [8]. Ambisonics has many advantages over other surround sound approaches. Whereas for most surround sound systems each channel of the recording is the specific signal sent to an individual loudspeaker, the number and layout of loudspeakers for reproduction of Ambisonic format sound does not need to be considered in the encoding or recording process. Furthermore, the sound field can be easily rotated and transformed once in Ambisonic format.

Ambisonics can be rendered binaurally over headphones using a virtual loudspeaker approach by convolving each loudspeaker signal with a HRTF corresponding to the loudspeaker’s position. The convolved binaural signals from every loudspeaker in the configuration are then summed at the left and right ears to produce the overall headphone mix. The binaural Ambisonic approach to spatial audio is popular with virtual reality applications and used in conjunction with HMDs, as head rotations can be compensated through counter-rotations of the sound field before it is decoded for the loudspeaker configuration [9, 10], thus removing the need for computationally intensive interpolation across a large dataset of HRTFs [11].

The number of channels in an Ambisonic format is determined by the Ambisonic order. First-order Ambisonics has 4 channels: one with an omnidirectional polar pattern (W channel) and three with figure-of-eight polar patterns facing in the X, Y and Z directions (X, Y and Z channels). In reproduction, each loudspeaker is fed an amount of the W, X, Y and Z channels depending on

its position. A regular arrangement of loudspeakers in a sphere can produce an accurate representation (at low frequencies) of the original sound field at the centre, also known as the ‘sweet spot’ [10] of the sphere. Increasing the Ambisonic order expands the number of channels, introducing greater spatial discrimination of sound sources. Higher-order Ambisonics requires more loudspeakers for playback, but the sound field reproduction around the head is accurate up to a higher frequency [12].

A major issue with Ambisonics is timbre. Depending on the order of Ambisonics used, reproduction above the spatial aliasing frequency, relative to the head size, is inaccurate and timbral inconsistencies exist which are noticeable when listening. This inconsistency is caused by spectral colouration above the spatial aliasing frequency due to comb filtering inherent in the summation of coherent loudspeaker signals with multiple delay paths to the ears. Timbre between different loudspeaker layouts also varies substantially, even without changing the Ambisonic order. The unnatural timbre of Ambisonics therefore makes the produced spatial audio easily distinguishable from natural sound fields. This poses significant issues for content creators who desire a consistent timbre between different playback scenarios.

2.2. Diffuse-Field Equalisation

A diffuse-field response refers to the direction independent frequency response, or the common features of a frequency response at all angles of incidence. This can be obtained from the root-mean-square (RMS) of the frequency responses for infinite directions on a sphere [13]. A diffuse-field response of a loudspeaker array requires an averaging of the frequency responses produced by the loudspeaker array when playing sound arriving from every point on a sphere (including between loudspeaker positions), however a sufficient approximate diffuse-field response can be calculated from a finite number of measurements. It is important to sample the sound field evenly in all directions to not introduce bias in any direction.

Diffuse-field equalisation, also referred to as diffuse-field compensation, can be employed to remove the direction-independent aspects of a frequency response introduced by the recording and reproduction signal chain. A diffuse-field equalisation filter is calculated from the inverse of the diffuse-field frequency response. Diffuse-field compensation of a loudspeaker array is achieved by convolving the output of the array with its inverse filter.

2.3. Individualisation

Though individualised HRTFs produce more accurate localisation cues and more natural timbre than non-individualised HRTFs [14–17], the measurement process is lengthy and requires specific hardware, stringent set up and an anechoic environment. For wide use individualised HRTFs are therefore not practical, and generic HRTFs produced from dummy heads are utilised.

2.4. Headphone Equalisation

The transfer function between a headphone and eardrum (HpTF) is highly individual [18, 19]. It also varies depending on the position of the headphones on the head: even small displacements of the headphone on the ear can produce large changes in the HpTF.

Headphone equalisation has been shown to improve plausibility of binaural simulations when correctly implemented [1], however equalisation based on just one measurement can produce worse

results than no equalisation at all [20]. Therefore, headphone equalisation should always be calculated from an average of multiple measurements. This will smooth out the deep notches and reduce sharp peaks in the inverse filter, which are more noticeable than troughs [21, 22].

Non-individual headphone equalisation can also be detrimental to timbre, unless the non-individual headphone equalisation is performed using the same dummy head as that used for binaural measurements, which has been shown to produce even greater naturalness than individual headphone compensation [23].

3. METHOD

This section presents a method for simulating an approximate diffuse-field response and equalisation of three Ambisonic virtual loudspeaker configurations (see Table 1). As this study used virtual loudspeakers, calculations were made using Ambisonic gains applied to head related impulse responses (HRIRs): the time-domain equivalent of HRTFs. No additional real-world measurements were necessary.

The three loudspeaker configurations utilised in this study were the octahedron, cube and bi-rectangle (see Table 1). In this paper, all sound incidence angles are referred to in spherical coordinates of azimuth (denoted by θ) for angles on the horizontal plane, and elevation (denoted by ϕ) for angles on the vertical plane, with $(0^\circ, 0^\circ)$ representing a direction straight in front of the listener at the height of the ears. Changes in angles are positive moving anticlockwise in azimuth and upwards in elevation.

Table 1: *First-Order Ambisonic Loudspeaker Layouts.*

Loudspeaker number	Loudspeaker location (θ°, ϕ°)		
	Octahedron	Cube	Bi-rectangle
1	0, 90	45, 35	90, 45
2	45, 0	135, 35	270, 45
3	135, 0	225, 35	45, 0
4	225, 0	315, 35	135, 0
5	315, 0	45, -35	225, 0
6	0, -90	135, -35	315, 0
7		225, -35	90, -45
8		315, -35	270, -45

3.1. Even distribution of points on a sphere

The approximate diffuse-field responses were calculated using 492 regularly spaced points on a sphere. The points were calculated by dividing each of the 20 triangular faces on an icosahedron into 7^2 sub-triangles, resulting in a polygon with 1470 equally sized faces and 492 vertices. The vertices were then projected onto a sphere, producing 492 spherical coordinates (see Figure 1) [24].

3.2. Ambisonic encoding and decoding

The 492 spherical coordinates were encoded into first-order Ambisonic format, with N3D normalisation and ACN channel ordering. The Ambisonic encode and decode MATLAB functions used in this study were written by Archontis Politis [25]. The MATLAB build used in this study was version 9.0.0 - R2016a. For each of the 492 directions, gains for the W, Y, Z and X channels were produced.

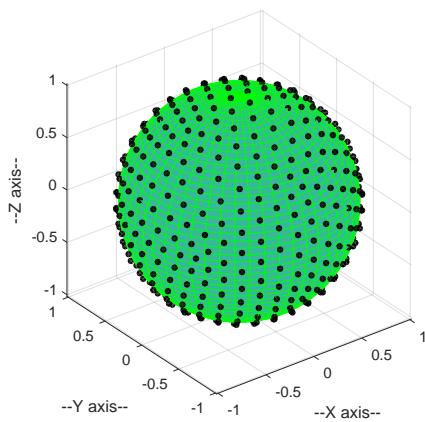


Figure 1: Distribution of 492 points on a sphere utilised in calculation of diffuse-field response, from [24].

The decode matrices for the three loudspeaker configurations were then calculated, again using N3D normalisation and ACN channel ordering. For each loudspeaker in each configuration, this produced the gain of the four Ambisonic channels (W, Y, Z and X) as determined by the loudspeaker's position.

A dual-band decode method [26] was utilised in this study to optimise the accuracy of localisation produced by the virtual loudspeaker configurations. As ITD is the most important azimuthal localisation factor at low frequencies and no elevation cues exist below 700 Hz [27,28], a mode-matching decode, which is optimised for the velocity vector [6], was used for this frequency range. Above 700 Hz, the wavelength of sounds become comparable or smaller than the size to the human head (which on average has a diameter of 17.5 cm [29]) and ILD is the most important localisation factor in this frequency range. Above 700 Hz therefore, mode-matching with maximised energy vector (MaxRe) [11] was used, which is optimised for energy [30].

The decode matrices for each loudspeaker layout were therefore calculated twice: with and without MaxRe weighting, for the high and low frequencies respectively. The dual-band decode was achieved through a crossover between the two decode methods at 700 Hz. The filters used were 32nd order linear phase high pass and low pass filters with Chebyshev windows.

The encoded Ambisonic channel gains were then applied to the loudspeaker decode matrices for each loudspeaker configuration, resulting in a single value of gain for each loudspeaker for each of the 492 directions. As this study used virtual loudspeakers for binaural reproduction of Ambisonics, the gain for each loudspeaker was then applied to a HRIR pair measured at the corresponding loudspeaker's location. The HRIRs used in this study were of the Neumann KU 100 dummy head at 44.1 kHz and 16-bit resolution, from the SADIE HRTF database [31].

The resulting HRIRs for each virtual loudspeaker in the configuration were then summed, leaving a single HRIR pair for each of the 492 directions. The HRIR pairs represent the transfer function of the Ambisonic virtual loudspeaker configuration for each direction of sound incidence. This was repeated for the three Ambisonic loudspeaker configurations.

3.3. Diffuse-Field Calculation and Equalisation

An approximate diffuse-field response was then generated as an average of the power spectrum responses of the 492 HRIR pairs (the equivalent of 492 measurements for different sound source directions in the loudspeaker arrays) for each virtual loudspeaker configuration. The contribution to the diffuse-field response of each HRIR pair was based on the solid angle of incidence to ensure no direction contributed more or less depending on the number of measurements from that direction.

Simulated diffuse-field frequency responses of the three loudspeaker configurations are presented in Figure 2. The differences between the three configurations are clearly visible, from the differing bass responses to the broadband deviations above 1 kHz. The cube and bi-rectangle feature large boosts at 3 kHz and 2.5 kHz respectively, while the octahedron is slightly attenuated between 1 kHz and 4 kHz. Above 6 kHz, all three configurations vary significantly.

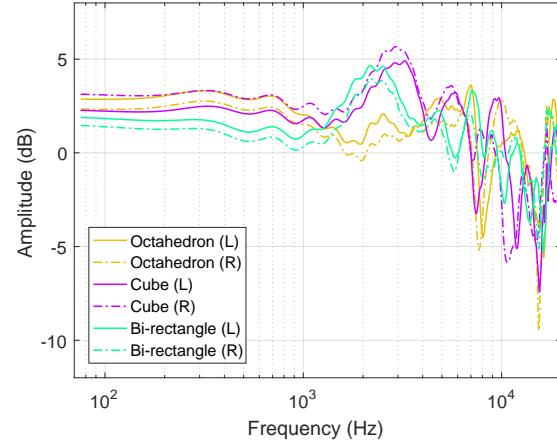


Figure 2: Comparison of the diffuse-field responses of three first-order Ambisonic loudspeaker configurations.

Inverse filters were calculated from the diffuse-field responses in the frequency range of 20 Hz - 20 kHz, using Kirkeby's least-mean-square (LMS) regularisation method [32], which is perceptually preferred to other regularisation methods [1]. To avoid sharp notches and peaks in the inverse filter, 1/4 octave smoothing was used. By convolving the Ambisonic loudspeaker configurations with their inverse filters, the diffuse-field responses are equalised to within ± 1.5 dB of unity in the range of 20 Hz - 15 kHz. The diffuse-field responses, inverse filters and resultant diffuse-field compensated (DFC) frequency responses of the three Ambisonic loudspeaker configurations are presented in Figure 3.

4. EVALUATION

To measure the effectiveness of diffuse-field equalisation of first-order Ambisonics, two listening tests were conducted. The first assessed whether diffuse-field equalisation improves timbral consistency between Ambisonic binaural rendering and diffuse-field equalised HRTF convolution, and the second assessed whether the use of diffuse-field equalisation improves timbral consistency between different first-order Ambisonic virtual loudspeaker config-

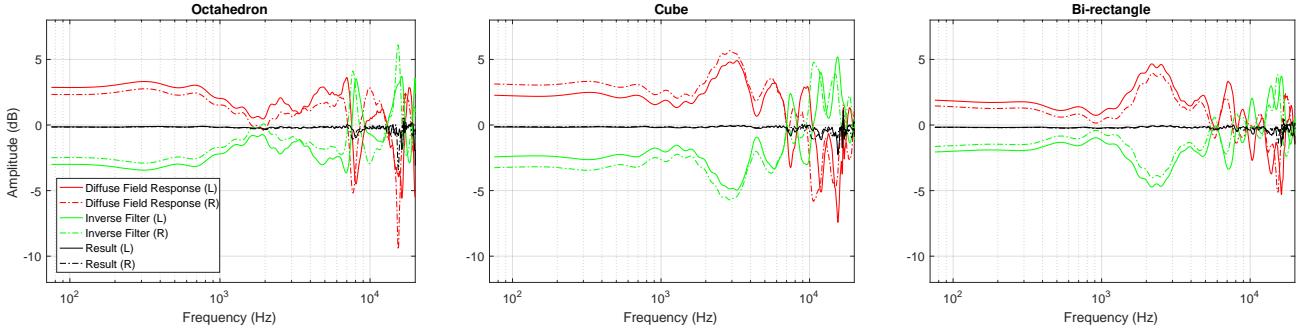


Figure 3: Diffuse-field responses, inverse filters and resultant DFC responses of three first-order Ambisonic loudspeaker configurations.

urations. The tests were conducted in a quiet listening room using an Apple Macbook Pro with a Fireface 400 audio interface, which has software controlled input and output levels. A single set of Sennheiser HD 650 circumaural headphones were used for all tests. These are free-air equivalent coupling (FEC). FEC headphones offer more realistic reproduction of binaural sounds over non-FEC headphone types, such as closed-back and in-ear headphones, as they do not alter the acoustical impedance of the ear canal when placed over the ears [18, 33].

The listening tests were conducted using 22 participants, aged from 20 - 57. All had at least some experience in audio engineering or music production, which was deemed a necessary prerequisite as the task of assessing and comparing different timbres involves critical listening. All participants had normal hearing, which was assessed prior to the listening tests through completion of an online hearing test [34]. Timbre was defined to participants as the tonal qualities and spectral characteristics of a sound, independent of pitch and intensity.

4.1. Test Stimuli

The stimulus used in the tests was one second of monophonic pink noise at a sample rate of 44.1 kHz, windowed by an onset and offset Hanning ramp of 5 ms [35] to avoid unwanted audible artefacts. The Ambisonic sound samples were created by encoding the pink noise into first-order Ambisonics (N3D normalisation and ACN channel numbering) and decoding to the three virtual loudspeaker layouts using the same dual-band decode technique and HRTFs as subsection 3.2. The direct HRTF renders were created by convolving the pink noise stimulus with a diffuse-field compensated HRTF pair of the corresponding sound source direction. The HRTFs used for the direct convolutions were from the same HRTF database as used in subsection 3.2. Overall, there were 7 different test configurations: virtual loudspeakers in octahedron, cube and bi-rectangle arrangements, all with and without diffuse-field equalisation, and direct HRTF convolution. To minimise the duration of the experiment, sound source directions were limited to the horizontal plane. For each test configuration, test sounds were generated for four sound source directions ($\theta = 0^\circ, 90^\circ, 180^\circ$ and 270°).

4.2. Level Normalisation

Each test sound was normalised relative to a frontal incident sound ($0^\circ, 0^\circ$) at a level of -23 dBFS. RMS amplitude X_{RMS} of signal X was calculated as

$$X_{\text{RMS}} = \sqrt{\frac{1}{n}(x_1^2 + x_2^2 + \dots + x_n^2)}, \quad (1)$$

where x_n is the value of sample n (x_1, x_2, \dots, x_n). The required RMS amplitude for -23 dBFS RMS was calculated from the following formula:

$$\text{dBFS}_{\text{RMS}} = 20 \log_{10} Y_{\text{RMS}}, \quad (2)$$

where Y_{RMS} is the absolute value of RMS amplitude. To produce an RMS level of -23 dBFS, Y_{RMS} is therefore $10^{-1.15}$. The normalisation constant K , to which each test sound was multiplied, was calculated as

$$K = \frac{2(10^{-1.15})}{L_{\text{RMS}} + R_{\text{RMS}}}, \quad (3)$$

where L_{RMS} and R_{RMS} are the left and right RMS amplitudes of frontal incidence, calculated for each test configuration. Finally, each test sound was multiplied by the normalisation constant of its corresponding test configuration.

4.3. Headphone Level Calibration

The listening tests were run at an amplitude of 60 dBA, chosen in accordance with Hartmann and Rakerd [36] who found that high sound pressure levels increase error in localisation.

Headphone level was calibrated using the following method: a Genelec 8040 B loudspeaker was placed inside an anechoic chamber, emitting pink noise at an amplitude that was adjusted until a sound level meter at a distance of 1.5 m and incidence of $(0^\circ, 0^\circ)$ displayed a loudness of 60 dBA. The sound level meter was then replaced with a Neumann KU 100 dummy head at the same 1.5 m distance facing the loudspeaker at $(0^\circ, 0^\circ)$, and the input level of the KU 100's microphones was measured.

The loudspeaker was then removed, and the Sennheiser HD 650 headphones to be used in the listening tests were placed on the dummy head. Pink noise convolved with KU 100 HRTFs at $(0^\circ, 0^\circ)$ from the SADIE database, which were recorded at a distance of 1.5 m [31], was played through the headphones. The convolved pink noise was normalised to a level of -23 dBFS RMS: the same loudness as the test sounds (subsection 4.2). The output level of the headphones was then adjusted on the audio interface until the input from the KU 100 matched the same loudness as from the loudspeaker. This headphone level was kept constant for all listening tests.

4.4. Headphone Equalisation

The Sennheiser HD 650 headphones used in the listening tests were equalised relative to the Neumann KU 100 dummy head (the same as that used for the Ambisonic virtual loudspeakers and HRTF rendering). The HD 650 headphones were placed on the KU 100 dummy head, and a 10 second 20 Hz - 20 kHz sine sweep was played through the headphones with the output of the KU 100 microphones being recorded. The HpTF was then calculated by deconvolving the recording with an inverse of the original sweep [37]. The HpTF measurement process was repeated 20 times, with the headphones being removed and repositioned between each measurement (see Figure 4).

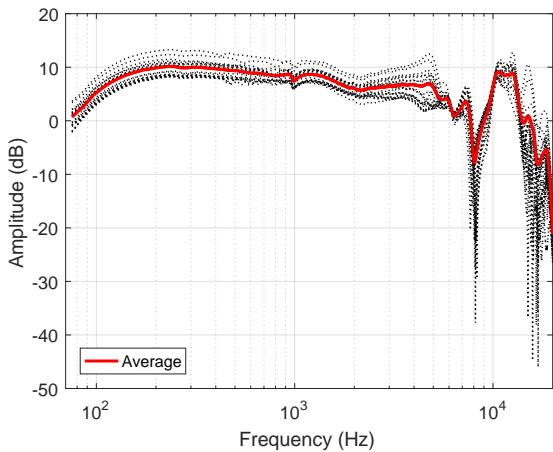


Figure 4: 20 measurements of the headphone transfer function of Sennheiser HD 650 headphones on the Neumann KU 100 dummy head (left ear). Average response in red.

An average of the 20 HpTF measurements was then calculated for the left and right ears by taking the power average of the 20 frequency responses. An inverse filter was computed using Kirkeby regularization [38], with the range of inversion from 200 Hz - 16 kHz and 1/2 octave smoothing. The average HpTFs, inverse filters and resultant frequency responses (produced by convolving the averages with their inverse filters) are presented in Figure 5.

4.5. Listening Test 1 - ABX

The first listening test followed the ABX paradigm [39], whereby three stimuli (A, B and X) were presented consecutively to the participants, who were instructed to answer which of A or B was closest to X in timbre. This differs from the modern definition of the ABX test where X would be one of A or B: here X was employed as a reference sound (HRTF convolution), and A and B were the Ambisonic binaural renders - one of which was diffuse-field equalised. The test was forced choice: the participant had to answer either A or B. The null hypothesis is that diffuse-field equalisation has no effect on the similarity of timbre between Ambisonic binaural rendering and HRTF convolution.

There were 12 test conditions in total: four sound source directions (as in subsection 4.1), across the three Ambisonic loudspeaker configurations. Every test condition was repeated with the order of DFC and non-DFC Ambisonic rendering reversed, to avoid bias towards any particular arrangement, resulting in a total of 24 test

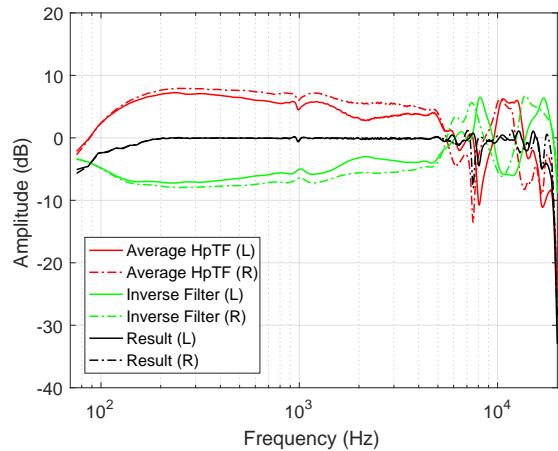


Figure 5: Average HpTFs and inverse filters of Sennheiser HD 650 headphones on the Neumann KU 100 dummy head.

files. The presentation of test files was double blinded and the order randomised separately for each participant.

4.5.1. Results

The data from the first listening test is non-parametric, and results are binomial distributions with 44 trials across subjects of each condition. For results to be statistically significant at less than 5% probability of chance, the cumulative binomial distribution must be greater than or equal to 61.36%: therefore the DFC Ambisonics needs to have been chosen a minimum of 27 times out of the 44 trials of that condition.

An average of results across all conditions shows that diffuse-field equalised Ambisonic rendering was perceived as closer in timbre to HRTF convolution for 65.5% of tests. Results for the separate conditions of the first listening test are presented in Figure 6 as the percentage that timbre of DFC Ambisonic rendering was perceived as closer to HRTF convolution than non-DFC Ambisonics for 44 trials across all participants. A higher percentage demonstrates a clearer indication that the DFC Ambisonics was closer to the HRTF reference, with values at or above 61.36% statistically significant.

The results are statistically significant ($p < 0.05$) for 9 conditions, therefore the null hypothesis can be rejected for 9 of the 12 tested conditions: diffuse-field equalisation does in fact have an effect on the perceived timbre of Ambisonic binaural rendering. The three conditions that were below statistical significance are for rear-incidence ($\theta = 180^\circ$). This suggests that diffuse-field equalisation has a different effect on the timbre of Ambisonics depending on the angle of sound incidence. Friedman's analysis of variance (ANOVA) tests were conducted to test whether this effect is statistically significant (see Table 2).

The Friedman's ANOVA tests showed that for the cube (Chi-sq = 17.36; $p = 0.0006$), the effect that diffuse-field equalisation has on timbre varies significantly depending on the sound source direction, but not for the octahedron and bi-rectangle ($p > 0.05$). Post-hoc analysis to determine which direction produced the statistical significance in the cube's results was conducted using a Wilcoxon signed rank test, which showed the outlying results were from ($\theta = 180^\circ$).

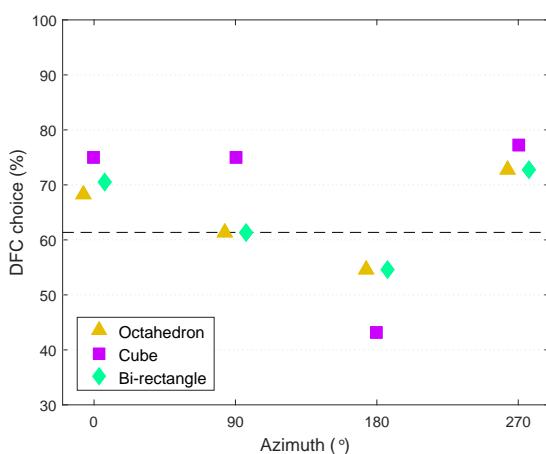


Figure 6: Results across subjects for which DFC Ambisonic rendering was considered more consistent in timbre to HRTF convolution than non-DFC Ambisonics, for each condition of the ABX test. Dashed line at 61.36% shows the boundary for statistical significance. Small horizontal offsets applied to results for visual clarity.

Table 2: Friedman's ANOVA tests to determine whether direction of sound incidence had a significant effect on results for the 3 tested loudspeaker configurations.

Loudspeaker Configuration	Chi-sq	p
Octahedron	4.05	0.2566
Cube	17.36	0.0006
Bi-rectangle	4.32	0.2293

The results presented in Figure 6 also suggest that, in general, diffuse-field equalisation had a larger effect on timbre for the cube than the other loudspeaker configurations. To test for statistical significance of this, Friedman's ANOVA tests were conducted on the different loudspeaker configurations across all 4 directions (see Table 3). These tests showed that, for all directions of sound incidence, this difference in results is not statistically significant ($p > 0.05$).

Table 3: Friedman's ANOVA tests to determine whether the virtual loudspeaker configurations produced significantly different results from each other for the 4 tested directions of sound incidence.

Azimuth (°)	Chi-sq	p
0	0.67	0.7165
90	2.67	0.2636
180	1.61	0.4464
270	0.36	0.8338

4.6. Listening Test 2 - AB

The second listening test was an AB comparison. Two sets of three stimuli were presented consecutively to the participants, who were instructed to answer which of set A or set B had the most consistent timbre. Each set consisted of the three Ambisonic binaural renders (octahedron, cube and bi-rectangle): one set with diffuse-field

equalisation and the other without. The test was forced choice: the participant had to answer either A or B. The null hypothesis is that diffuse-field equalisation has no effect on the consistency of timbre between different Ambisonic virtual loudspeaker configurations.

There were 4 conditions in total: one for each of the four sound source directions (as in subsection 4.1). Every test file was repeated with the order of DFC and non-DFC Ambisonic rendering reversed, to avoid bias towards any particular arrangement, resulting in a total of 8 test files. As in the first listening test, presentation of test files was double blinded and the order randomised separately for each participant.

4.6.1. Results

As for the first listening test, data from the second listening test is non-parametric and results are binomial distributions, with 44 trials across subjects of each condition. For results to be statistically significant at less than 5% probability of chance, the cumulative binomial distribution must be greater than or equal to 61.36%; therefore the DFC Ambisonics needs to have been chosen a minimum of 27 times out of the 44 trials of that condition.

An average of results across all directions shows that Ambisonic virtual loudspeakers were perceived as more consistent in timbre when diffuse-field equalised for 74.4% of tests. Results for the separate conditions of the second listening test are presented in Figure 7 as the percentages that the three DFC loudspeaker configurations were perceived as more consistent in timbre than non-DFC configurations. A higher percentage demonstrates a clearer indication that diffuse-field compensated Ambisonics was more consistent in timbre across different loudspeaker configurations, with values at or above 61.36% statistically significant.

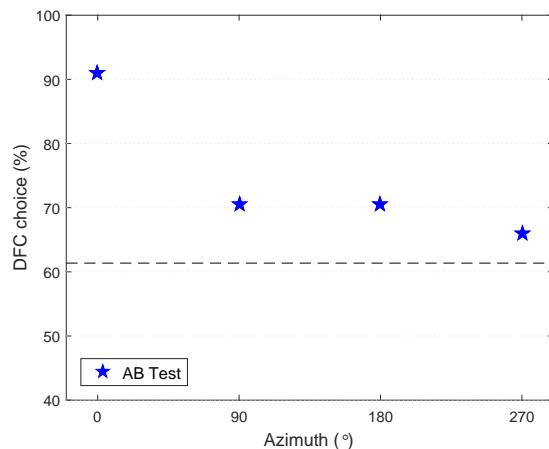


Figure 7: Results across subjects for which DFC Ambisonic rendering was considered more consistent in timbre across different virtual loudspeaker configurations, for each condition of the AB test. Dashed line at 61.36% shows the boundary for statistical significance from chance at a confidence level of $p < 0.05$.

Results for all conditions are statistically significant, as the DFC Ambisonic loudspeaker configurations were considered more consistent for more than 61.36% of results. Therefore the null hypothesis can be rejected: diffuse-field equalisation does have an effect on the consistency of timbre between different Ambisonic

virtual loudspeaker configurations. Frontal incidence ($\theta = 0^\circ$) produced the most notable result: the three virtual loudspeaker configurations were considered more consistent when diffuse-field equalised for 91% of tests in this condition. The other sound source directions ($\theta = 90^\circ, 180^\circ$ and 270°) favoured DFC Ambisonics as more consistent for 70%, 70% and 66% of the tests, respectively. As in the first listening test, direction of sound incidence appears to have had an effect on the results. To determine the significance of this effect, a Friedman's ANOVA test was conducted which showed statistical significance (Chi-sq = 9.19; $p = 0.0269$). Post-hoc analysis to determine which condition produced the statistical significance was conducted using a Wilcoxon signed rank test, which showed the outlying results were from ($\theta = 0^\circ$).

4.7. Discussion

The results from the two listening tests clearly indicate the applicability of diffuse-field equalisation for reducing the timbral differences between Ambisonic binaural rendering and HRTF convolution, as well as improving timbral consistency between different Ambisonic virtual loudspeaker configurations.

One observation from the results is that direction of incidence had a substantial effect on people's judgement, with DFC causing minimal change to timbre for rear incident ($\theta = 180^\circ$) sounds in the first test but a clear trend for other directions. This change is statistically significant for the results of the cube in the first test. Direction of incidence also had a significant effect on results for the second test: the three virtual loudspeaker configurations were perceived to be more consistent when diffuse-field equalised for all sound incidence, but the trend was significantly more pronounced for frontal incidence ($\theta = 0^\circ$).

A possible explanation for the variation in results depending on direction of incidence is that, in general, the diffuse-field equalisation filters calculated in this study boosted high frequencies and attenuated low frequencies (see Figure 3). As the direction of incidence changes the frequency content of a sound, with rear-incident sounds typically subject to attenuation in the high frequencies due to pinnae shadowing, this may explain the variation in significance of results.

In this study, when calculating the diffuse-field responses of the loudspeaker configurations, every direction was weighted evenly. However, to address the direction-dependent influence of diffuse-field equalisation on timbre, an approach by changing the weighting based on the solid angle could be taken by increasing the weighting for rear-incident sounds when calculating the diffuse-field responses. This could produce a more even effect on timbre for different directions of sound incidence. This theory requires further investigation, which is currently in progress.

5. CONCLUSIONS

This paper has demonstrated the timbral variation that exists between different Ambisonic loudspeaker configurations above the spatial aliasing frequency, due to comb filtering produced by the summing of multiple sounds at the ears. A method to address this timbral disparity through diffuse-field equalisation has been presented, and the effectiveness of the method in improving the consistency of timbre between different spatial audio rendering methods and between different first-order Ambisonic loudspeaker configurations has been evaluated.

The conducted subjective listening tests show that diffuse-field equalisation of Ambisonics is successful in improving timbral consistency between Ambisonic binaural rendering and HRTF convolution, as well as between different first-order Ambisonic loudspeaker configurations. However, results differ depending on sound incidence, and a theory to address this by changing the directional weighting in the diffuse-field calculation has been proposed which is currently being investigated.

Future work will look at diffuse-field equalisation of higher-order Ambisonics, as well as assessing the effect that diffuse-field equalisation has on localisation accuracy in Ambisonic rendering. If it can be shown that diffuse-field equalisation either improves or has no effect on localisation accuracy, then diffuse-field equalisation will be a clear recommendation for achieving more natural sounding Ambisonics.

6. ACKNOWLEDGEMENT

Thomas McKenzie was supported by a Google Faculty Research Award.

7. REFERENCES

- [1] Zora Schärer and Alexander Lindau, "Evaluation of equalization methods for binaural signals," in *Proc. AES 126th Convention*, 2009, pp. 1–17.
- [2] Alexander Lindau, Torben Hohn, and Stefan Weinzierl, "Binaural resynthesis for comparative studies of acoustical environments," in *Proc. AES 122nd Convention*, Vienna, 2007, pp. 1–10.
- [3] A. J. Berkout, D. de Vries, and P. Vogel, "Acoustic control by wave field synthesis," *J. Acoust. Soc. Am.*, vol. 93, no. 5, pp. 2764–2778, 1993.
- [4] Ville Pulkki, "Virtual sound source positioning using vector base amplitude panning," *J. Audio Eng. Soc.*, vol. 45, no. 6, pp. 456–466, 1997.
- [5] Michael A. Gerzon, "Peripherony: with-height sound reproduction," *J. Audio Eng. Soc.*, vol. 21, no. 1, pp. 2–10, 1973.
- [6] Michael A. Gerzon, "Criteria for evaluating surround-sound systems," *J. Audio Eng. Soc.*, vol. 25, no. 6, pp. 400–408, 1977.
- [7] Michael A. Gerzon, "Ambisonics in multichannel broadcasting and video," *J. Audio Eng. Soc.*, vol. 33, no. 11, pp. 859–871, 1985.
- [8] Pierre Lecomte, Philippe Aubert Gauthier, Christophe Langrenne, Alexandre Garcia, and Alain Berry, "On the use of a lebedev grid for ambisonics," in *Proc. AES 139th Convention*, 2015, pp. 1–12.
- [9] Adam McKeag and David McGrath, "Sound field format to binaural decoder with head-tracking," in *Proc. AES 6th Australian Regional Convention*, 1996, pp. 1–9.
- [10] Markus Noisternig, A. Sontacchi, T. Musil, and R. Höldrich, "A 3D ambisonic based binaural sound reproduction system," in *Proc. AES 24th International Conference on Multichannel Audio*, 2003, number March, pp. 1–5.
- [11] Gavin Kearney and Tony Doyle, "Height perception in ambisonic based binaural decoding," in *Proc. AES 139th Convention*, 2015, pp. 1–10.

- [12] Jérôme Daniel, *Représentation de champs acoustiques, application à la transmission et à la reproduction de scènes sonores complexes dans un contexte multimédia*, Ph.D. thesis, l’Université Paris, 2001.
- [13] Aaron J. Heller and Eric M. Benjamin, “Calibration of sound-field microphones using the diffuse-field response,” in *Proc. AES 133rd Convention*, San Francisco, 2012, pp. 1–9.
- [14] Elizabeth Wenzel, Marianne Arruda, Doris Kistler, and Frederic L. Wightman, “Localization using nonindividualized head-related transfer functions,” *J. Acoust. Soc. Am.*, vol. 94, no. 1, pp. 111–123, 1993.
- [15] Henrik Møller, Michael. F. Sørensen, Clemen B. Jensen, and Dorte Hammershøi, “Binaural technique: do we need individual recordings?”, *J. Audio Eng. Soc.*, vol. 44, no. 6, pp. 451–469, 1996.
- [16] Adelbert W. Bronkhorst, “Localization of real and virtual sound sources,” *J. Acoust. Soc. Am.*, vol. 98, no. 5, pp. 2542–2553, 1995.
- [17] CJ Tan and Woon Seng Gan, “Direct concha excitation for the introduction of individualized hearing cues,” *J. Audio Eng. Soc.*, vol. 48, no. 7, pp. 642–653, 2000.
- [18] Henrik Møller, Dorte Hammershøi, Clemen B. Jensen, and Michael F. Sørensen, “Transfer characteristics of headphones measured on human ears,” *J. Audio Eng. Soc.*, vol. 43, no. 4, pp. 203–217, 1995.
- [19] David Griesinger, “Accurate timbre and frontal localization without head tracking through individual eardrum equalization of headphones,” in *Proc. AES 141st Convention*, 2016, pp. 1–8.
- [20] Abhijit Kulkarni and H. Steven Colburn, “Variability in the characterization of the headphone transfer-function,” *J. Acoust. Soc. Am.*, vol. 107, no. 2, pp. 1071–1074, 2000.
- [21] Roland Bücklein, “The audibility of frequency response irregularities,” *J. Audio Eng. Soc.*, vol. 29, no. 3, pp. 126 – 131, 1981.
- [22] Bruno Masiero and Janina Fels, “Perceptually robust headphone equalization for binaural reproduction,” in *Proc. AES 130th Convention*, 2011, pp. 1–7.
- [23] Alexander Lindau and Fabian Brinkmann, “Perceptual evaluation of headphone compensation in binaural synthesis based on non-individual recordings,” *J. Audio Eng. Soc.*, vol. 60, no. 1-2, pp. 54–62, 2012.
- [24] John Burkardt, “SPHERE_GRID - points, lines, faces on a sphere,” Available at http://people.sc.fsu.edu/~jburkardt/datasets/sphere_grid/sphere_grid.html, accessed February 09, 2017.
- [25] Archontis Politis, *Microphone array processing for parametric spatial audio techniques*, Ph.D. thesis, Aalto University, 2016.
- [26] Aaron J. Heller, Richard Lee, and Eric M. Benjamin, “Is my decoder ambisonic?”, in *Proc. AES 125th Convention*, San Francisco, 2008, pp. 1–22.
- [27] Mark B. Gardner, “Some monaural and binaural facets of median plane localization,” *J. Acoust. Soc. Am.*, vol. 54, no. 6, pp. 1489–1495, 1973.
- [28] V. Ralph Algazi, Carlos Avendano, and Richard O. Duda, “Elevation localization and head-related transfer function analysis at low frequencies,” *The Journal of the Acoustical Society of America*, vol. 109, no. 3, pp. 1110–1122, 2001.
- [29] George F. Kuhn, “Model for the interaural time differences in the azimuthal plane,” *J. Acoust. Soc. Am.*, vol. 62, no. 1, pp. 157–167, 1977.
- [30] Michael A. Gerzon and Geoffrey J. Barton, “Ambisonic decoders for HDTV,” in *Proc. AES 92nd Convention*, 1992, number 3345, pp. 1–42.
- [31] Gavin Kearney and Tony Doyle, “A HRTF database for virtual loudspeaker rendering,” in *Proc. AES 139th Convention*, 2015, pp. 1–10.
- [32] Ole Kirkeby and Philip A. Nelson, “Digital filter design for inversion problems in sound reproduction,” *J. Audio Eng. Soc.*, vol. 47, no. 7/8, pp. 583–595, 1999.
- [33] Henrik Møller, “Fundamentals of binaural technology,” *Applied Acoustics*, vol. 36, no. 3-4, pp. 171–218, 1992.
- [34] Stéphane Pigeon, “Hearing test and audiogram,” Available at <https://hearingtest.online/>, accessed March 15, 2017.
- [35] David Schonstein, Laurent Ferré, and Brian Katz, “Comparison of headphones and equalization for virtual auditory source localization,” *J. Acoust. Soc. Am.*, vol. 123, no. 5, pp. 3724–3729, 2008.
- [36] W M Hartmann and B Rakerd, “Auditory spectral discrimination and the localization of clicks in the sagittal plane.,” *J. Acoust. Soc. Am.*, vol. 94, no. 4, pp. 2083–2092, 1993.
- [37] Angelo Farina, “Simultaneous measurement of impulse response and distortion with a swept-sine technique,” in *Proc. AES 108th Convention*, Paris, 2000, number I, pp. 1–23.
- [38] Ole Kirkeby, Philip A. Nelson, Hareo Hamada, and Felipe Orduna-Bustamante, “Fast deconvolution of multichannel systems using regularization,” *IEEE Transactions on Speech and Audio Processing*, vol. 6, no. 2, pp. 189–194, 1998.
- [39] W. A. Munson and Mark B. Gardner, “Standardizing auditory tests,” *J. Acoust. Soc. Am.*, vol. 22, no. 5, pp. 675, 1950.

IMPROVING ELEVATION PERCEPTION WITH A TOOL FOR IMAGE-GUIDED HEAD-RELATED TRANSFER FUNCTION SELECTION

Michele Geronazzo

Dept. of Neurological, Biomedical and Movement Sciences,
University of Verona
Verona, Italy
michele.geronazzo@univr.it

Enrico Peruch, Fabio Prandoni, and Federico Avanzini

Dept. of Information Engineering
University of Padova
Padova, Italy
avanzini@dei.unipd.it

ABSTRACT

This paper proposes an image-guided HRTF selection procedure that exploits the relation between features of the pinna shape and HRTF notches. Using a 2D image of a subject's pinna, the procedure selects from a database the HRTF set that best fits the anthropometry of that subject. The proposed procedure is designed to be quickly applied and easy to use for a user without previous knowledge on binaural audio technologies. The entire process is evaluated by means of an auditory model for sound localization in the mid-sagittal plane available from previous literature. Using virtual subjects from a HRTF database, a virtual experiment is implemented to assess the vertical localization performance of the database subjects when they are provided with HRTF sets selected by the proposed procedure. Results report a statistically significant improvement in predictions of localization performance for selected HRTFs compared to KEMAR HRTF which is a commercial standard in many binaural audio solutions; moreover, the proposed analysis provides useful indications to refine the perceptually-motivated metrics that guides the selection.

1. INTRODUCTION

Our auditory system continuously captures everyday acoustic scenes and acquires spatial information by processing temporal and spectral features of sound sources related to both the environment and the listener himself. Knowledge of such a complex process is needed in order to develop accurate and realistic artificial sound spatialization in several application domains, including music listening, entertainment (e.g. gaming), immersive virtual reality, sensory substitution devices (e.g. for visually-impaired users), tele-operation, tele-conferencing, and so on [1].

Many of the above mentioned scenarios require spatial sound to be delivered through headphones. This usually involves the use of *binaural room impulse responses* (BRIRs), which are the combination of two components: the *room impulse response* (RIR), and the *head-related impulse response* (HRIR), which accounts for the acoustic transformations produced by the listener's head, pinna, torso and shoulders. Having a set of HRIRs (or Head-Related Transfers Functions - HRTFs, their Laplace transforms) measured over a discrete set of spatial locations allows to spatially render a dry sound by convolving it with the desired HRIR pair. Moving sound sources can also be rendered by suitably interpolating spatially neighboring HRIRs.

The ability to localize sound sources is important in several everyday activities. Accordingly, localization accuracy is a relevant auditory quality even in *Virtual Auditory Displays* (VADs) [2]. This paper deals in particular with elevation localization cues, which are mainly provided by monaural spectral features of the HRTF.

Specifically, the scattering of acoustic waves in the proximity of the pinna creates a complex and individual topography of pressure nodes which is not completely understood [3, 4], and results in elevation- and listener-dependent peaks and notches that appear in the HRTF spectrum in the range [3, 16] kHz. This monaural information complements binaural cues such as *interaural time difference* (ITD) and *interaural level difference* (ILD), which are mainly related to localization in the horizontal plane and are almost constants with varying elevations.

Individual anthropometric features of the human body have a key role in shaping individual HRTFs (see the discussion in Sec. 2 below). This paper proposes an image-guided HRTF selection technique that builds on previous work on the relation between features of the pinna shape and HRTF notches [5]. Using a 2D image of a subject's pinna, the procedure selects from a database the HRTF set that best fits the anthropometry of that subject. One of the challenging issues with this approach is the trade off between handiness of pinna feature acquisition and localization performance in elevation; since the procedure in [5] relied on expert operators for the extraction of anthropometric information, this work provides an easy to use tool for a user without previous knowledge on pinna acoustics and spatial hearing.

Auditory localization performance with HRTF sets is usually assessed through psychoacoustic experiments with human subjects. However, an attractive alternative approach consists in using computational auditory models able to simulate the human auditory system. If the auditory model is well calibrated to the reality, a perceptual metric can be developed to predict the perceptual performance of a VAD. The proposed HRTF selection procedure is here validated on subjects from the CIPIC subjects [6] for whom HRTFs and side-pictures of the pinna are available. The applicability of the proposed notch distance metric are also discussed in terms of individual HRTF identification from images. Performances in elevation perception are evaluated by means of an auditory model for sound localization in the mid-sagittal plane [7] (i.e., the vertical plane dividing the listener's head in left and right halves) provided by the Auditory Modeling Toolbox¹. Using virtual subjects from the CIPIC database, we present a virtual experiment that assesses the vertical localization performance of CIPIC subjects when they are provided with HRTF sets selected by the proposed procedure.

2. RELATED WORKS

One of the main limitations of binaural audio technologies for commercial use is the hard work behind the creation of the in-

¹<http://amtoolbox.sourceforge.net/>

dividual HRTFs that capture all of the physical effects creating a personal perception of immersive audio. The measurement of a listener's own HRTFs in all directions requires a special measuring apparatus and a long measurement time, often a too heavy task to perform for each subject involved in every-day application. That is the main reason why alternative ways are preferred to provide HRTFs giving to listeners a personalized, but not individually created, HRTF set: a trade off between quality and cost of the acoustic data for audio rendering [8, 9].

2.1. Individual / Own HRTFs

The standard setup for individual HRTF measurement is in an anechoic chamber with a set of loudspeakers mounted on a geodesic sphere (with a radius of at least one meter in order to avoid near-field effects) at fixed intervals in azimuth and elevation. The listener, seated in the center of the sphere, has microphones in his/her ears. After subject preparation, HRIRs are measured playing analytic signals and recording responses collected at the ears for each loudspeaker position in space (see [9] for a systematic review on this topic).

The main goal is to extract the set of HRTFs for every listener thus providing him/her the individual/own transfer function. In addition to the above mentioned high demanding requirements (time and equipment), there are some other critical aspects in HRTF measurements; listener's pose is usually limited to a few positions (standing or sitting), relatively few specific locations around his/her body, and own intrinsic characterization without considering that pinna shape is one of the human body part that always grows during lifetime [10]. Moreover, repeatability of HRTF measurements are still a delicate issue [11].

2.2. Personalized / Generalized HRTFs

The personalized HRTFs are chosen among available HRTFs of a dataset instead of doing individual measurements. This procedure is based on a match between external subjects (the one without individual HRTFs) and internal, i.e. belonging to a database, subjects with already stored information (both acoustics and anthropometry). The most interesting and important part, is the method of how is selected a specific set of HRTFs to an external subject. Researchers are finding different ways to deal with this issue and there are a variety of alternatives using common hardware and/or software tools. The main benefit of this approach is that a user can be guided to a self selection of their best HRTF set without needing a special equipment or knowledge. It has to be noted that the personalized HRTF can not guarantee the same performance as their own HRTF but they usually provide better performance than the generic dummy-head HRTFs such those of Knowles Electronic Manikin for Acoustic Research (KEMAR) [12].

In the following, we summarize three main approaches to HRTF selection.

- **DOMISO[13]**²

In this technique, subjects can choose their most suitable HRTFs from among many, taken from a database following tournament-style listening tests. The database (corpus) is built using different subjects, storing 120 sets of HRTFs, one set per listener.

²DOMISO: Determination method of OptimuM Impulse-response by Sound Orientation

Performances of this technique were evaluated by Yukio Iwaya that proved that the personalized DOMISO HRTFs results were similar to individualized HRTFs ones but very different from the away condition (totally random HRTF, that could not win the tournament).

- **Two steps selection[14, 15]**

This is a technique based on two different steps. Usually the first step selects one subset from a complete initial pool of HRTF sets, removing worse HRTFs from a perceptual point of view. The second step refines the selection in order to obtain the best match among generic HRTFs of a dataset which is reduced in size compared to the complete database.

- **Matching anthropometric ear parameters[16, 17]**

This method is based on finding the best match HRTF in the anthropometric domain, matching the external ear shape of a subject using anthropometric measurements available in the database.³

3. IMAGE-GUIDED HRTF SELECTION

Another approach to HRTF selection problem consists in mapping anthropometric features into the HRTF domain, following a ray-tracing modeling of pinna acoustics [18, 19]. The main idea is to draw pinna contours on a image. Distances from the ear canal entrance define reflections on pinna borders generating spectral notches in the HRTF. Accordingly, one can use such anthropometric distances and corresponding notch parameters to choose the best match among a pool of available HRTFs [5].

3.1. Notch distance metrics

The extraction of HRTFs using reflections and contours is based on an approximate description of the acoustical effects of the pinna on incoming sounds. In particular, the distance d_c between a reflection point on the pinna and the entrance of the ear canal (the "focus point" hereafter) is given by:

$$d_c(\phi) = \frac{ct_d(\phi)}{2}, \quad (1)$$

where $t_d(\phi)$ is elevation-dependent temporal delay between the direct and the reflected wave and c is the speed of sound.

The corresponding notch frequency depends on the sign of the reflection. Assuming the reflection coefficient to be positive, a notch is created at all frequencies such that the phase difference between the reflected and the direct wave is equal to π :

$$f_n(\phi) = \frac{2n + 1}{2t_d(\phi)} = \frac{c(2n + 1)}{4d_c(\phi)}, \quad (2)$$

where $n \in \mathbb{N}$. Thus, the first notch frequency is found when $n = 0$, giving the following result:

$$f_0(\phi) = \frac{c}{4d_c(\phi)}. \quad (3)$$

In fact, a previous study [19] on the CIPIC database [6] proved that almost 80% of the subjects in the database exhibit a clear negative reflection in their HRIRs. Under this assumption, notches are

³see section 3.2 for further details.

found at full-wavelength delays, resulting in the following equation:

$$f_n(\phi) = \frac{n+1}{t_d(\phi)} = \frac{c(n+1)}{2d_c(\phi)}, \quad (4)$$

where $n \in \mathbb{N}$ and

$$f_0(\phi) = \frac{c}{2d_c(\phi)}. \quad (5)$$

In particular it has been shown [5] that the first and most prominent notch in the HRTF is typically associated to the most external pinna contour on the helix border (the “ C_1 ” contour hereafter).

Now, assume that N estimates of the C_1 contour and K estimates of the focus point have been traces on a 2D picture of the pinna of a subject (the meaning of N and K is explained later in Sec. 3.2). We define the basic notch distance metric in the form of a mismatch function between the corresponding notch frequencies, and the notch frequencies of a HRTF:

$$m_{(k,n)} = \frac{1}{N_\varphi} \sum_{\varphi} \frac{|f_0^{(k,n)}(\varphi) - F_0(\varphi)|}{F_0(\varphi)}, \quad (6)$$

where $f_0^{(k,n)}(\varphi) = c/[2d_c^{(k,n)}(\varphi)]$ are the frequencies extracted from the image and contours of the subject, and F_0 are the notch frequencies extracted from the HRTF with *ad-hoc* algorithms such as those developed in [18, 20, 17]; (k, n) with $(0 \leq k < K)$ and $(0 \leq n < N)$ refers to a one particular pair of traced C_1 contour and focus point; φ spans all the $[-45^\circ, +45^\circ]$ elevation angles for which the notch is present in the corresponding HRTF; N_φ is the number of elevation angles on which the summation is performed. Extracted notches need to be grouped into a single track evolving through elevation consistently, a labeling algorithm (e.g. [19]) performed such computation along subsequent HRTFs.

If the notches extracted from the subject’s pinna image are to be compared with a set of HRTFs taken from a database, various notch distance metrics can be defined based on this mismatch function, to rank database HRTFs in order of similarity. In particular, we define three metrics:

- **Mismatch:** each HRTF is assigned a similarity score that corresponds exactly to increasing values of the mismatch function calculated with Eq. (6) (for a single (k, n) pair).
- **Ranked position:** each HRTF is assigned a similarity score that is an integer corresponding to its ranked position taken from the previous mismatch values (for a single (k, n) pair).
- **Top- M appearance:** for a given integer M , for each HRTF, a similarity score is assigned according to the number of times (for all the (k, n) pairs) in which that HRTF ranks in the first M positions.

3.2. A HRTF selection tool

Based on the concepts outlined above, we propose a tool for selecting from a database a HRTF set that best fits the image of a subject’s pinna. The C_1 contour and the focus point are traced manually on the pinna image by an operator, and then the HRTF sets in the database are automatically ranked in order of similarity with the subject. The tool is implemented in Matlab.

Graphical user interface. Figure 1 provides a screenshot of the main GUI which is responsible for managing subjects and organizing them in a list (on the left of the screen). The list can be

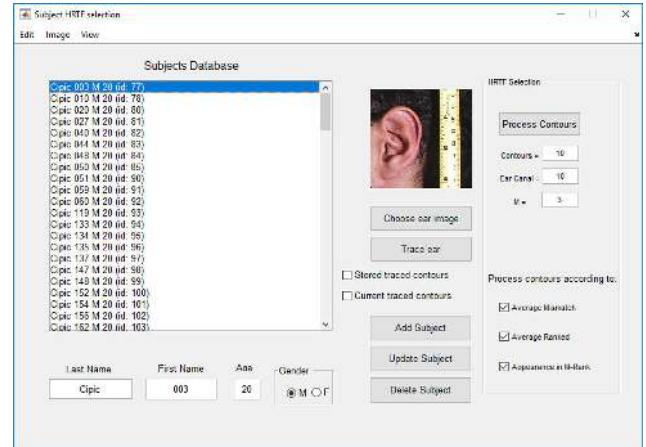


Figure 1: A tool for HRTF selection: main Graphical User Interface.

managed efficiently using the three buttons, “Add Subject”, “Update Subject” and “Delete Subject”, as well as some text-fields used to assign to each subject their own information. For each subject stored in the list, an image of the left pinna can be assigned with the button “Choose ear image”: the image will be shown in the middle of the GUI when a name from the list is clicked.

After loading the pinna image of a subject, the main pinna contour C_1 and the focus point can be traced manually by clicking on the “Trace Ear” button. Two parameters N and K can be specified, which are the number of estimates that the operator will trace for the C_1 contour and the focus point, respectively.

Two checkboxes under the “Trace Ear” button aid the usability of the tracing task: the first one is the “Stored traced contours” that shows the already drawn contours in the previous drawing session. The second one, called “Current traced contours” is about visualizing on pinna image the contours drawn in the current session.⁴

One last parameter to be set, M , refers to the top- M appearance metrics discussed above. By clicking on the “Process Contours”, the application returns the ranked positions of the database HRTFs according to the three metrics.

Database of generic HRTFs and extracted F_0 . The public database used for our purpose is the CIPIC [6]. The first release provided HRTFs for 43 subjects (plus two dummy-head KEMAR HRTF sets with small and large pinnae, respectively) at 25 different azimuths and 50 different elevations, to a total of 1250 directions. In addition, this database includes a set of pictures of external ears and anthropometric measures for 37 subjects. Information of the first prominent notch in each HRTF were extracted with the *structural decomposition algorithm* [20, 9] and F_0 tracks were labeled with the algorithm in [19] and then stored in a custom data structure;

Guidelines for contour tracing. In the trace-ear GUI, the user has to draw by hand N estimates of the C_1 contours on top of a pinna image. After that, the user has to point K positions of the

⁴The default tracing procedure allows drawing a single contour/focus point at a time, that visually disappears once traced; for every estimate, our tool shows pinna images clean from traced information.

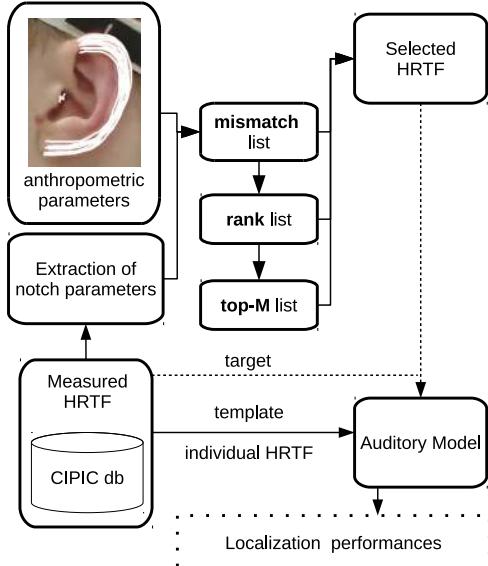


Figure 2: Schematic view of the proposed validation procedure with auditory model predictions.

ear canal entrance. The rationale behind this is that by averaging over repeated attempts we aim at reducing errors due to operator's mistakes and inherent ambiguities of the tracing task (as an example, the true location of the ear canal entrance is not known from the 2D image and must be guessed). By working on the application, we have derived some empirical guidelines for the tracing task which will be very useful for future non-expert operators. In particular, the most effective way to trace the C_1 contour from the image is to cover the area of C_1 with N curves, starting from the internal edge to the external edge of C_1 and vice versa, while the most effective way to trace the focus point is to guess the ear canal entrance with K points in the most likely area. In other words, the tracing procedure is a simplified version of the *optimal focus* estimation procedure proposed in [19] where a minimization problem was solved by searching in a wide area near the pinna *tragus* tracing several specific contours. On the other hand, real case applications with a physical human ear allow the operator to easily localize where the ear canal is, reducing also uncertainty for the estimation of external pinna contours.

4. VALIDATION

The main aim of the proposed validation procedure is to verify the effectiveness of our HRTF selection tool in providing a subject with a HRTF that is reasonably close to his/her individual HRTF, by only using a picture of his/her external ear. Strengths and limits of such an approach are discussed also with the support of an auditory model to predict performance in elevation perception. Figure 2 depicts a schematic view of the entire validation process.

4.1. Data acquisition and analysis

Our experimental subjects were taken from the CIPIC database. In particular, we selected the 22 CIPIC subjects for which a complete set of data was available (HRTF, anthropometry and ear pictures). We chose to draw $N = 10$ estimates of the C_1 contour

and $K = 10$ estimates of the focus point, a good trade off that guarantees enough accuracy and fast completion of the selection procedure. The parameter M was set to 3. The entire procedure of creating a subject, retrieving the picture and anthropometric measures, and drawing the contours and focus points, takes about 5 minutes for each subject. Data processing time is negligible. With these settings each subject has $N \times K = 100$ pairs of contours and focus points ready to be processed.

The results of the computation are three rankings of 43 HRTF sets (CIPIC's dummy heads were excluded for homogeneity) derived from our metrics:

- **Average mismatch:** CIPIC subjects are sorted according to their mismatch values (averaged over the $N \times M$ estimates), in increasing order of mismatch.
- **Average rank:** CIPIC subjects are sorted according to their rank in the previous ranking (averaged over the $N \times M$ estimates), in increasing order of rank.
- **Top-M appearance:** CIPIC subjects are sorted according to the number of their occurrences of the in the top-3 positions for each (n, k) pair of estimates, in decreasing order of occurrence count.

For each metrics, we defined three *best fitting HRTFs* by choosing the HRTFs ranking first in each ranking: best average mismatch (**best m**), best average rank (**best r**), and best top-3 rank (**best top3**) selected HRTFs.

A preliminary analysis on data distributions of mismatch and rank values showed that normality assumption was violated according to a Shapiro-Wilk test; thus, two Kruskal Wallis nonparametric one-way ANOVAs with three levels of feedback condition (individual, dummy-head KEMAR, best m) and (individual, dummy-head KEMAR, best r) were performed to assess the statistical significance of mismatch and rank metrics, respectively, on all traced pinna contours and ear-canal points. Pairwise *post-hoc* Wilcoxon tests for paired samples with Holm-Bonferroni correction procedures on p-values provided statistical significances in performance between conditions.

4.2. Auditory model simulations

Using the predictions of an auditory model, we simulated a virtual experiment where every CIPIC listener would be asked to provide an absolute localization judgment about spatialized auditory stimulus. We adopted a recent model [7], that follows a “*template-based*” paradigm implementing a comparison between the internal representation of an incoming sound at the eardrum and a reference template. Spectral features from different HRTFs correlate with the direction of arrival, leading to a spectro-to-spatial mapping and a perceptual metric for elevation performances.

The model is based on two processing phases. During peripheral processing, an internal representation of the incoming sound is created and the *target* sound (e.g. a generic HRTF set) is converted into a *directional transfer function* (DTF). In the second phase, the new representation is compared with a *template*, i.e. individual DTFs computed from individual HRTFs, thus simulating the localization process of the auditory system (see previous works [21] for further details on this methodology).

For each target angle, the probability that the virtual listener points to a specific angle defines the *similarity index* (SI). The index value results from the distance (in degrees) between the target angle and the response angle which is the argument of a Gaussian

distribution with zero-mean and standard deviation, called *uncertainty*, U . The lower the U value, the higher the sensitivity of the virtual listener in discriminating different spectral profiles resulting in a measure of probability rather than a deterministic value.

The virtual experiment was conducted simulating listeners with all analyzed CIPIC HRTFs, using an uncertainty value $U = 1.8$ which is similar to average human sensitivity [7]. We predicted elevation performance for every virtual subject when listening with his/her own individual HRTFs, with those of CIPIC subject 165 (the KEMAR), and the best m / best r / best top3 selected HRTFs. The precision for the j -th elevation response close to the target position is defined in the *local polar RMS error* (PE):

$$PE_j = \sqrt{\frac{\sum_{i \in L} (\phi_i - \varphi_j)^2 p_j[\phi_i]}{\sum_{i \in L} p_j[\phi_i]}},$$

where $L = \{i \in N : 1 \leq i \leq N_\phi, |\phi_i - \varphi_j| \bmod 180^\circ < 90^\circ\}$ defines local elevation responses within $\pm 90^\circ$ w.r.t. the local response ϕ_i and the target position φ_j , and $p_j[\phi_i]$ denotes the prediction, i.e. probability mass vector.

The average PE was computed considering only elevation responses φ_j between $[-45^\circ, +45^\circ]$, where inter-subject variability in human spatial hearing emerges [22], thus providing a single number that quantifies localization performance [21].⁵ In order to verify statistically significant differences between predicted average PE s, paired t-tests were performed between pairs of localization performances using different HRTFs.

5. RESULTS AND DISCUSSION

A preliminary analysis on data distribution of rank values derived from mismatches between $f_0^{(k,n)}(\varphi)$ and individual HRTF's $F_0(\varphi)$ (22×100 observations) was conducted in order to identify the existence of outliers for our metrics. Samples in the last quartile of this distribution were considered cases of limited applicability for the proposed notch distance metric, showing a rank position greater than 27.25 of a total of 43.

Leaving aside for a moment the discussion on applicability of our metrics, we considered the last quartile value as a threshold for the average rank position of each individual HRTF in order to discard CIPIC subjects which can not be classified according to our criteria and for which no firm conclusions can be drawn. After the application of such threshold, the same analysis was performed on 17×100 observations, i.e. 5 subjects were removed; the 75% of the observations had a rank position less than 18 which is in the first half of the available positions. Moreover, the median value for rank position is 8, which suggests data convergence to the first rank positions.

Figure 3 depicts the three typical tracing scenarios: (a) a consistent trace-notch correspondence, (b) a systematic lowering in notch frequency of traces, and (c) an irregular notch detection. In the first case, traced contours and individual HRTF notches are in the same range resulting in the ideal condition of applicability for the proposed metric. The latter situation occasionally occurred

⁵We focused on local polar error in the frontal median plane, where individual elevation-dependent HRTF spectral features perceptually dominate; on the contrary, front-back confusion rate (similar to quadrant error rate QE in [7]) derives from several concurrent factors, such as dynamic auditory cues, visual information, familiarity with sound sources and training [23], thus it was not considered in this study.

due to irregularities of HRTF measurements or erroneous track label assignment of $F_0(\varphi)$ evolving through elevation (in 2 of the 5 subjects which were previously removed).⁶ On the other hand, the case where a systematic lowering in notch frequency of traces occurred (in 3 of the 5 subjects previously removed) deserves a more careful consideration: from one of our previous studies [19], we identified a 20% of CIPIC subjects for whom a positive reflection coefficient better models the acoustic contribution of the pinna. Accordingly, it is reasonable to think that those three ex-

⁶Repeatability of HRTF measurements are still a delicate issue, suggesting a high variability in spectral details [11].

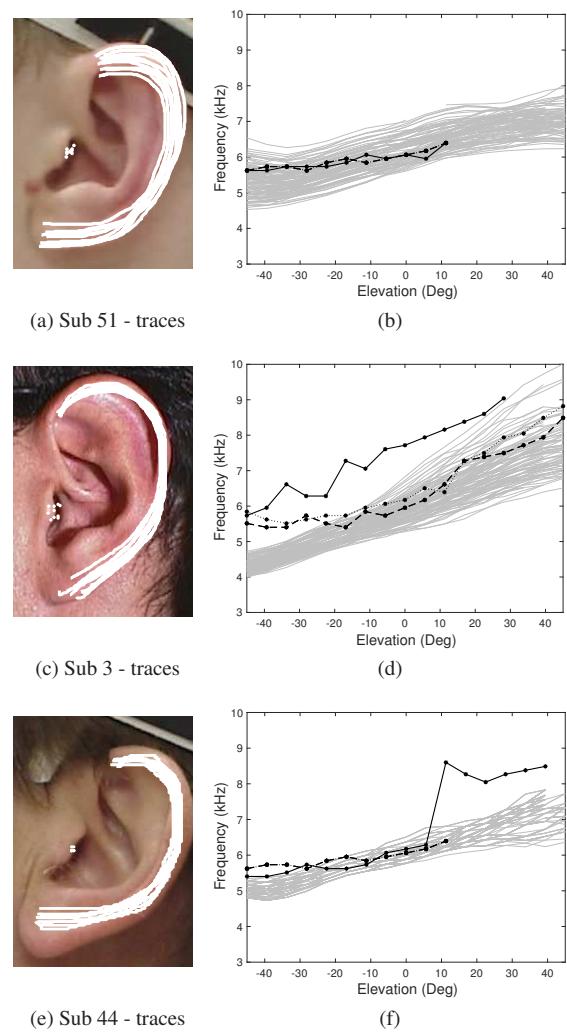


Figure 3: (a,c,e) Examples of traced C_1 /focus points for three CIPIC subjects; (b,d,f) corresponding $f_0^{(k,n)}(\varphi)$ (light gray lines) with $F_0(\varphi)$ values of individual HRTFs (black solid line), best selection according to mismatch/rank metric (black dotted line), and best selection according to Top-3 metric (black dash-dotted line). In this examples, best HRTF selection according to mismatch and rank metrics do not differ significantly.

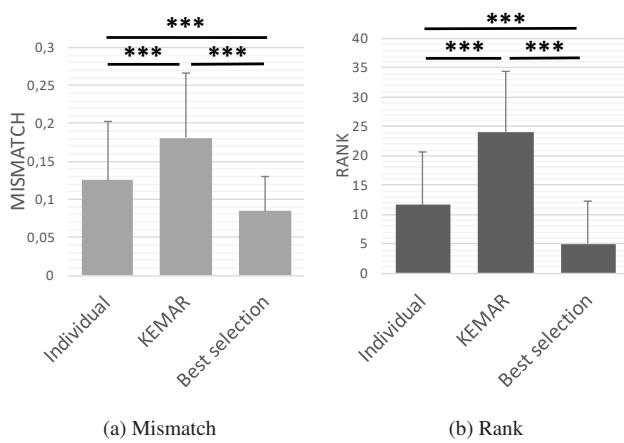


Figure 4: Global statistics (average + standard deviation) for metric assessment on (a) mismatch, (b) rank, grouped by HRTF condition. Asterisks and bars indicate, where present, a significant difference (*: $p < 0.05$, **: $p < 0.01$, ***: $p < 0.001$ at post-hoc test).

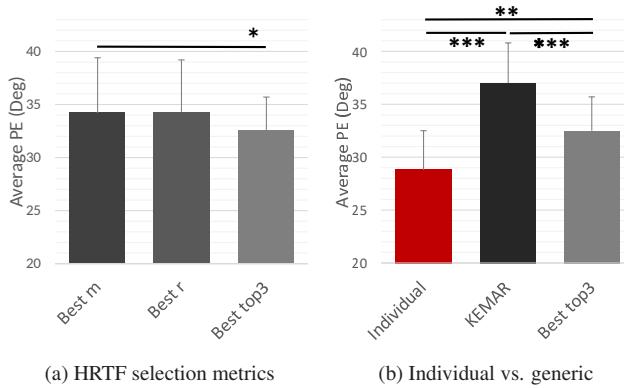


Figure 5: Global statistics (average + standard deviation) for localization prediction in elevation on average PE for (a) metrics based on notch distance mismatch, (b) individual vs. generic (KEMAR) vs. personalized (best top3). Asterisks and bars indicate, where present, a significant difference (*: $p < 0.05$, **: $p < 0.01$, ***: $p < 0.001$ at post-hoc test).

cluded subjects can be assigned to this special group.⁷

Our metrics based on notch distance clearly distinguish the three sets of HRTF, i.e. individual, KEMAR, and best selected, in terms of mismatch and rank (Fig. 4 clearly shows this aspect); Kruskal Wallis nonparametric one-way ANOVAs with three levels of feedback condition (individual HRTF, KEMAR, best m) provided a statistically significant result for mismatch [$\chi^2(2)=1141.8$, $p \ll 0.001$]; Pairwise post-hoc Wilcoxon tests for paired samples with Holm-Bonferroni correction revealed statistical differences among all pairs of conditions ($p \ll 0.001$). The same anal-

⁷Unfortunately, we were not able to directly compare our current study with [19] because different CIPIC subjects were considered.

ysis with (individual HRTF, KEMAR, best r) as levels of feedback condition provided a statistically significant result for rank [$\chi^2(2)=2362.3$, $p \ll 0.001$] with statistical differences among all pairs of conditions ($p \ll 0.001$). At first glance, since we were trying to select the individual HRTFs from pinna contours, these results appear to be counter intuitive because we always selected a generic HRTF which differed from the individual HRTF in terms of both mismatch and rank. However, this evidence can not be misleading because we already know from our previous study in [5] that the notch associated to the pinna helix border is not enough to describe elevation cues for all listeners. Moreover, biometric recognition studies [24] show that the pinna concha wall is also relevant in order to uniquely identify a person. Finally, multiple contours tracing highly contributes to the uncertainty of notch frequency matches, providing a good average rank anyway (11), though preventing the individual HRTFs to be chosen as best selection.

Surprisingly, localization predictions from auditory model simulations provided average local polar RMS error (average PE) which has a statistically significant difference between best m and best top3 metrics, $t(16) = 2.134$, $p < 0.05$ (see Fig. 5.a for a graphical representation). This results suggest that best top3 yields better localization performances than best m, and with a similar compared to best r (not proven to be statistically significant in this study). Intuitively, best top3 metric is more robust to contour uncertainty because of the M-constraint in its definition, that allowed us to filter out variability due to HRTF set with sparse appearances in our rankings.

Finally, localization predictions were computed also for individual HRTFs and KEMAR virtual listening. Pairwise t-tests reveal significant differences in average PEs between individual listening condition and KEMAR ($t(16) = -7.79$, $p \ll 0.001$), and between individual listening condition and best top3 HRTF ($t(16) = -4.13$, $p < 0.01$), reporting a better performance with individual HRTFs. Moreover, pairwise t-test reports significant differences in average PEs between best top3 HRTF and KEMAR ($t(16) = 5.590$, $p \ll 0.001$), with a better performance of the selected HRTF compared to dummy-head listening condition. This final result further confirms the gap between individual HRTF listening and the proposed HRTF selection based on pinna image; on the other hand, best top 3 criteria selected generic HRTFs that outperformed KEMAR listening condition.

6. CONCLUSIONS

The proposed distance metrics considering the C_1 contour provides insufficient features in order to unambiguously identify an individual HRTF from the corresponding side picture of the listener. Moreover, multiple tracing of C_1 and of the focus point adds further variability to the procedure resulting in extra uncertainty for the validation. On the other hand, our final result confirms that our image-guide HRTF selection procedure provides a useful tool in terms of:

- **personalized dataset reduction:** since individual HRTF rank is on average the 12th position, one can compute a personalized short list of ≈ 12 best candidate HRTFs for a given pinna picture, in which finding with high probability a generic HRTF reasonably close to the individual one. Accordingly, a subsequent refinement of the HRTF selection procedure might be required through subjective selection procedures or additional data analysis on the reduced

dataset.

- **better performance than KEMAR:** confirming our previous findings in psychoacoustic evaluation [5], auditory model predictions reported a statistically significant improvement in localization performance with generic HRTFs selected based on top3 metric compared to KEMAR; this result has important practical implications for binaural audio applications requiring HRTF personalization: our tool allows a user without specific expertise to choose a generic HRTF in a few minutes; this selection outperforms localization performance with KEMAR HRTFs, which are usually default solutions for commercial applications in VADs.

Further research is still needed in order to increase the applicability of our notch distance metrics; CIPIC subjects can be also analyzed applying Eqs. (2) and (3) (notches caused by positive reflections) to Eq. (6), and localization predictions with both reflection signs can be compared. Contours associated to antihelix and concha reflections can be traced, and the mismatch definition can be modified accordingly by combining the contributions of each contour with different weights [5]. Furthermore, notch distance metrics, i.e. mismatch, rank, and top-M metrics, can be hierarchically applied in the HRTF selection process in order to refine the selection: as an example, starting from the top M metric one can disambiguate similar HRTF sets looking at mismatch and rank metrics. In particular, the influence of the M parameter on HRTF appearance in the rank metric has to be investigated in more detail.

An alternative approach, which is currently being investigated, amounts to estimating the first pinna notch directly via acoustic measurements, through a so-called “acoustic selfie” which roughly acquires individual HRTFs using a smartphone loudspeaker as sound source and binaural microphones as receivers [25]. In this way, the frequencies $f_0^{(k,n)}(\varphi)$ could be directly computed in the acoustic domain, further reducing manual intervention.

Finally, it is indisputable that experimental validation with massive participation of human subjects will be highly relevant in terms of reliability of any HRTF selection procedure. A new research framework for binaural audio reproduction in web browsers is currently in development phase [26] with the goal of overcoming common limitations in HRTF personalization studies, such as low number of participants (e.g. [17]), coherence in simplifications of localization experiment (e.g. [15]), and reliability of the predictions with computational auditory models [27].

7. REFERENCES

- [1] J Blauert, *The Technology of Binaural Listening*, Modern Acoustics and Signal Processing. Springer Berlin Heidelberg, 2013.
- [2] A Lindau, V Erbes, S Lepa, H.-J Maempel, F Brinkman, and S Weinzierl, “A Spatial Audio Quality Inventory (SAQI),” *Acta Acustica united with Acustica*, vol. 100, no. 5, pp. 984–994, Sept. 2014.
- [3] H Takemoto, P Mokhtari, H Kato, R Nishimura, and K Iida, “Mechanism for generating peaks and notches of head-related transfer functions in the median plane,” *The Journal of the Acoustical Society of America*, vol. 132, no. 6, pp. 3832–3841, 2012.
- [4] S Prepelija, M Geronazzo, F Avanzini, and L Savioja, “Influence of Voxelization on Finite Difference Time Domain Simulations of Head-Related Transfer Functions,” *The Journal of the Acoustical Society of America*, vol. 139, no. 5, pp. 2489–2504, May 2016.
- [5] M Geronazzo, S Spagnol, A Bedin, and F Avanzini, “Enhancing Vertical Localization with Image-guided Selection of Non-individual Head-Related Transfer Functions,” in *IEEE Int. Conf. on Acoust. Speech Signal Process. (ICASSP 2014)*, Florence, Italy, May 2014, pp. 4496–4500.
- [6] V. R Algazi, R. O Duda, D. M Thompson, and C Avendano, “The CIPIC HRTF Database,” in *Proc. IEEE Work. Appl. Signal Process., Audio, Acoust.*, New Paltz, New York, USA, Oct. 2001, pp. 1–4.
- [7] R Baumgartner, P Majdak, and B Laback, “Assessment of Sagittal-Plane Sound Localization Performance in Spatial-Audio Applications,” in *The Technology of Binaural Listening*, J Blauert, Ed., Modern Acoustics and Signal Processing, pp. 93–119. Springer Berlin Heidelberg, Jan. 2013.
- [8] M Geronazzo, S Spagnol, and F Avanzini, “Mixed Structural Modeling of Head-Related Transfer Functions for Customized Binaural Audio Delivery,” in *Proc. 18th Int. Conf. Digital Signal Process. (DSP 2013)*, Santorini, Greece, July 2013, pp. 1–8.
- [9] M Geronazzo, *Mixed structural models for 3D audio in virtual environments*, Ph.D. Thesis, University of Padova, Padova, Italy, Apr. 2014.
- [10] C Sforza, G Grandi, M Binelli, D. G Tommasi, R Rosati, and V. F Ferrario, “Age- and sex-related changes in the normal human ear,” *Forensic Science International*, vol. 187, no. 1–3, pp. 110.e1–110.e7, May 2009.
- [11] A Andreopoulou, D Begault, and B Katz, “Inter-Laboratory Round Robin HRTF Measurement Comparison,” *IEEE Journal of Selected Topics in Signal Processing*, vol. PP, no. 99, pp. 1–1, 2015.
- [12] W. G Gardner and K. D Martin, “HRTF Measurements of a KEMAR,” *The Journal of the Acoustical Society of America*, vol. 97, no. 6, pp. 3907–3908, June 1995.
- [13] Y Iwaya, “Individualization of head-related transfer functions with tournament-style listening test: Listening with other’s ears,” *Acoustical science and technology*, vol. 27, no. 6, pp. 340–343, 2006.
- [14] L Sarlat, O Warusfel, and I Viaud-Delmon, “Ventriloquism aftereffects occur in the rear hemisphere,” *Neuroscience Letters*, vol. 404, no. 3, pp. 324–329, Sept. 2006.
- [15] B. F. G Katz and G Parseihian, “Perceptually based Head-Related Transfer Function Database Optimization,” *The Journal of the Acoustical Society of America*, vol. 131, no. 2, pp. EL99–EL105, Feb. 2012.
- [16] D Zotkin, R Duraiswami, and L Davis, “Rendering localized spatial audio in a virtual auditory space,” *IEEE Transactions on Multimedia*, vol. 6, no. 4, pp. 553 – 564, Aug. 2004.
- [17] K Iida, Y Ishii, and S Nishioka, “Personalization of head-related transfer functions in the median plane based on the anthropometry of the listener’s pinnae,” *The Journal of the Acoustical Society of America*, vol. 136, no. 1, pp. 317–333, July 2014.

- [18] V. C Raykar, R Duraiswami, and B Yegnanarayana, “Extracting the Frequencies of the Pinna Spectral Notches in Measured Head Related Impulse Responses,” *The Journal of the Acoustical Society of America*, vol. 118, no. 1, pp. 364–374, July 2005.
- [19] S Spagnol, M Geronazzo, and F Avanzini, “On the Relation between Pinna Reflection Patterns and Head-Related Transfer Function Features,” *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 21, no. 3, pp. 508–519, Mar. 2013.
- [20] M Geronazzo, S Spagnol, and F Avanzini, “Estimation and Modeling of Pinna-Related Transfer Functions,” in *Proc. of the 13th Int. Conference on Digital Audio Effects (DAFx-10)*, Graz, Austria, Sept. 2010, pp. 431–438.
- [21] M Geronazzo, A Carraro, and F Avanzini, “Evaluating vertical localization performance of 3d sound rendering models with a perceptual metric,” in *2015 IEEE 2nd VR Workshop on Sonic Interactions for Virtual Environments (SIVE)*, Arles, France, Mar. 2015, pp. 1–5, IEEE Computer Society.
- [22] J Blauert, *Spatial Hearing: The Psychophysics of Human Sound Localization*, MIT Press, Cambridge, MA, USA, 1983.
- [23] E. M Wenzel, M Arruda, D. J Kistler, and F. L Wightman, “Localization using nonindividualized head-related transfer functions,” *The Journal of the Acoustical Society of America*, vol. 94, no. 1, pp. 111–123, 1993, 00940.
- [24] E González, L Alvarez, and L Mazorra, “Normalization and Feature Extraction on Ear Images,” in *Proc. IEEE 46th Int. Carnahan Conf. Security Tech.*, Boston, MA, USA, Oct. 2012, pp. 97–104.
- [25] M Geronazzo, J Fantin, G Sorato, G Baldovino, and F Avanzini, “Acoustic Selfies for Extraction of External Ear Features in Mobile Audio Augmented Reality,” in *Proc. 22nd ACM Symposium on Virtual Reality Software and Technology (VRST 2016)*, Munich, Germany, Nov. 2016, pp. 23–26.
- [26] M Geronazzo, J Kleimola, and P Majdak, “Personalization Support for Binaural Headphone Reproduction in Web Browsers,” in *Proc. 1st Web Audio Conference*, Paris, France, Jan. 2015.
- [27] R Baumgartner, P Majdak, and B Laback, “Modeling sound-source localization in sagittal planes for human listeners,” *The Journal of the Acoustical Society of America*, vol. 136, no. 2, pp. 791–802, 2014.

VELVET-NOISE DECORRELATOR

Benoit Alary, Archontis Politis, and Vesa Välimäki *

Acoustics Lab, Dept. of Signal Processing and Acoustics
Aalto University
Espoo, Finland
Benoit.Alary@aalto.fi

ABSTRACT

Decorrelation of audio signals is an important process in the spatial reproduction of sounds. For instance, a mono signal that is spread on multiple loudspeakers should be decorrelated for each channel to avoid undesirable comb-filtering artifacts. The process of decorrelating the signal itself is a compromise aiming to reduce the correlation as much as possible while minimizing both the sound coloration and the computing cost. A popular decorrelation method, convolving a sound signal with a short sequence of exponentially decaying white noise which, however, requires the use of the FFT for fast convolution and may cause some latency. Here we propose a decorrelator based on a sparse random sequence called velvet noise, which achieves comparable results without latency and at a smaller computing cost. A segmented temporal decay envelope can also be implemented for further optimizations. Using the proposed method, we found that a decorrelation filter, of similar perceptual attributes to white noise, could be implemented using 87% less operations. Informal listening tests suggest that the resulting decorrelation filter performs comparably to an equivalent white-noise filter.

1. INTRODUCTION

Decorrelation is a useful operation in audio signal processing, as it can reduce correlation properties of partially correlated signals or can generate versions of a monophonic signal that are as little correlated as possible [1]. Leaving aside applications where it may be used internally by another method to improve its performance, such as echo cancellation, decorrelation is usually associated with multichannel processing and modification of the perceived spatial properties of sounds [1], [2], [3].

Reverberation, spatially extended sources, and ambience with diffuse sound properties all produce binaural signals that have a low correlation. Conversely, when delivering two binaural signals to a listener in a spatial simulation environment the results can vary. The sound is perceived centrally when both signals are fully correlated but, when partially correlated, the spatial image is extended. Two fully incoherent signals may be perceived as two separated lateral events [4]. These properties make decorrelation a useful tool for tasks such as spatial source spreading, artificial reverberation, spatial audio coding, source distance rendering, and headphone externalization [1], [2], [5].

Various decorrelation methods have been proposed in the literature. Generally, most methods aim to a) minimize correlation between the input and the output signal, b) preserve the magnitude

spectrum of the signal as much as possible, c) have a short decorrelation impulse response, if time-invariant, and d) be as computationally efficient as possible. Conditions b) and c) attempt to guarantee spatial modification of the sound without changing its spectral character or adding perceived reverberation.

A basic approach is to convolve the audio signal with a short white-noise sequence, in the range of 20–30 ms [1]. Variants of this apply an exponentially decaying envelope, or different decaying envelopes at different bandwidth decreasing time constants at higher frequencies, mimicking late reverberation properties. Such sequences have been studied by Hawksford and Harris [6]. The same work also proposes a *temporally diffuse impulse*, a sum of decaying sinusoids equalized with their minimum-phase equivalent to be flat. Xie *et al.* have studied the use of pseudo-random binary sequences for decorrelation [7]. Instead of synthetic signals, short late segments of room impulse responses are used directly by Faller in a spatial audio coding framework [5]. All these methods require convolution with the generated filters, making them costly in applications with multiple decorrelators applied in parallel, such as source spreading, or on the outputs of a large multichannel system. The fast convolution method is typically used to reduce the computational load, as it implements the convolution as a multiplication of spectra in the frequency domain. The efficiency then comes from the use of the fast Fourier transform (FFT) algorithm. Unfortunately, the fast convolution method causes latency, which is undesirable in real-time audio processing.

A more efficient method, which guarantees a flat magnitude response, is based on cascaded allpass filters [1], [8]. Alternatively, some methods operate in the time-frequency domain, using the short-time Fourier transform or a filter bank, and apply delays on each bin or subband. Such an approach was first introduced in [9], with fixed delays, drawn randomly, applied in perceptually motivated critical bands. This approach has been used with success in parametric spatial audio reproduction methods, e.g. in [10], to decorrelate diffuse/ambient components at the output channels. A more elaborate subband decorrelation approach has been presented by Vilkamo *et al.* [11] for artificial reverberation with convenient spectral control. Although more efficient, the filter-bank method also introduces latency.

This paper proposes the use of a velvet-noise sequence for decorrelating audio signals. Velvet noise consists of samples of values of either -1, 0, or 1 [12], [13]. The spacing between non-zero elements is also randomized to satisfy a density parameter. With sufficient density, these sparse sequences are perceived as smoother than Gaussian white noise [12], [13]. One of the useful applications of velvet-noise sequences is simulating room impulse responses. The late reverberation has been re-created using exponentially decaying filtered noise [14], [15]. Velvet-noise sequences offer a computationally efficient alternative for this purpose [12],

* This work was supported by the Academy of Finland (ICHO project, grant no. 296390).

[16], [17], [18], [19], [20], [21]. Whereas previous work on velvet noise has focused on its perceptual qualities and suitability to artificial reverberation, we investigate how to design a velvet-noise decorrelator (VND).

This paper is organized in the following way. In Section 2, we discuss velvet noise and how signals can be efficiently convolved with it. Section 3 introduces various decorrelation methods using velvet noise. Section 4 presents our results and compares the proposed method with a white-noise-based decorrelation method. Section 5 concludes this paper.

2. VELVET NOISE

2.1. Velvet-Noise Sequence

The main goal of using velvet-noise sequences (VNS) is to create spectrally neutral signals, comparable to white noise, while using as few non-zero values as possible. By taking advantage of the sparsity of the signal, we can efficiently convolve it in the time domain with another signal without any latency [19], [21]. The first step in the generation of velvet noise is to create a sequence of evenly spaced impulses at the desired density [12], as shown in Fig. 1(a). The sign of the impulses is then randomized (see Fig. 1(b)). Finally, the spacing between each impulse is also randomized within the space available to satisfy the density parameter, which is illustrated in Fig. 1(c).

For a given density ρ and sampling rate f_s , the average spacing between two impulses is

$$T_d = f_s / \rho, \quad (1)$$

which is called the grid size. The total number of impulses is

$$M = L_s T_d, \quad (2)$$

where L_s is the total length in samples. The sign of each impulse is

$$s(m) = 2 \text{ round}(r_1[m]) - 1, \quad (3)$$

where m is the impulse index, the round function is the rounding operation, and $r_1(m)$ is a random number between 0 and 1. The location of the impulse is calculated from

$$k(m) = \text{round}[mT_d + r_2[m](T_d - 1)], \quad (4)$$

where $r_2(m)$ is also a random number between 0 and 1.

2.2. Velvet-Noise Convolution

To convolve a VNS with another signal x , we take advantage of the sparsity of the sequence. Indeed, by storing the velvet-noise sequence as a series of non-zero elements, all mathematical operations involving zero can be skipped. For further optimization, the location of the positive and negative impulses can be stored in separate arrays, k_+ and k_- , which removes the need for multipliers in the convolution [19], [21]

$$x * k = \sum_{m=0}^{M+} x[n - k_+[m]] - \sum_{m=0}^{M-} x[n - k_-[m]]. \quad (5)$$

For a sequence with a density of a 1000 impulses per second, which has been found sufficient for decorrelation, and a sample

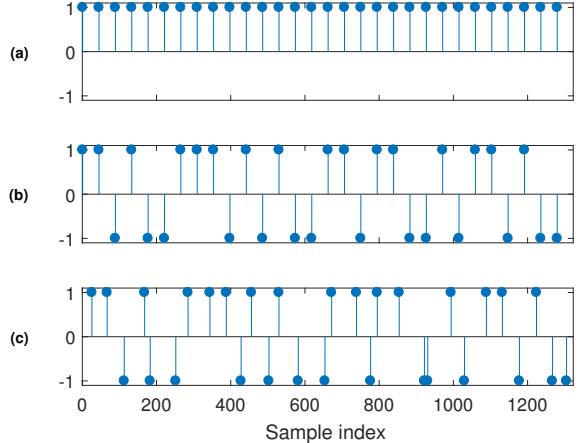


Figure 1: Steps to create a velvet-noise sequence: (a) a sequence of evenly spaced impulses, (b) the values of impulses are randomized to either 1 or -1 , and (c) the spacing between the impulses is randomized.

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 spectral domain convolution using an FFT for an equivalent white-noise sequence. Furthermore, this sparse time-domain convolution offers the benefit of being latency-free.

3. VELVET-NOISE DECORRELATION

3.1. Exponential Decay

When using a sequence of white noise for decorrelation, adding an exponentially decaying amplitude envelope to the signal is recommended to minimize the potentially audible smearing of transients [6].

The decay characteristics of the envelope are designed to achieve a desired attenuation and target length. The envelope is given by

$$D(m) = e^{-\alpha m} \quad (6)$$

based on a decay constant α given by

$$\alpha = \frac{-\ln 10^{-L_{\text{dB}}/20}}{M}, \quad (7)$$

where L_{dB} is the target total decay, while m represent the index of a specific impulse and M is the total number of non-zeros element in the sequence. The attenuation of an impulse is set by its index m . For a given set of parameters, every VNS generated has the same decay attenuation $D(m)$ for the same index m , only the sign varies. This distribution method ensures the energy is consistently distributed amongst the impulses regardless of their positions. The energy could also be distributed over time, depending on an impulse location. However, since the impulse locations vary based on a random value, a variation of distributed power would be obtained, which would lead to unbalanced decorrelation filters, unsuitable for multichannel purposes.

To generate an exponentially decaying velvet-noise filter (Figure 2), we combine the decay envelope of Eq. (6) to the random sign sequence:

$$s_e(m) = e^{-\alpha m} s(m). \quad (8)$$

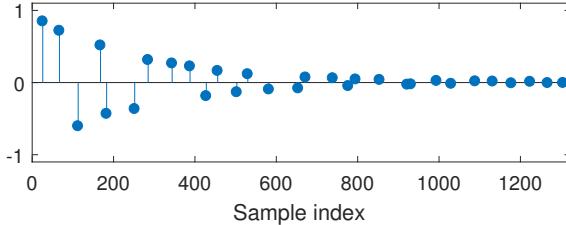


Figure 2: Velvet-noise sequence with an exponentially decaying temporal envelope.

The velvet-noise convolution equation also needs to be updated to include the attenuation factors:

$$x * k = \sum_{m=0}^M x[n - k[m]] s_e(m). \quad (9)$$

3.2. Segmented Decay

Unfortunately, the decaying amplitude envelope requires multiplications and thus leads to more operations for the convolution. However, since the benefits of a smoothly decaying envelope are not really perceivable with this application, the process can be simplified while still attenuating impulses based on their location. Indeed, audible decorrelation artifacts can be caused by velvet-noise sequences that contain large impulses near the end of the sequence. Therefore, to prevent these artifacts while minimizing the number of multiplications required by the sparse convolution, we can simply limit the attenuation factors to a fixed set of attenuation coefficients, as illustrated in Fig. 3. Storing each segment in a separate list allows for a segmented convolution, applying each coefficient to a partial sum

$$x * k = \sum_{i=0}^I s_i \left(\sum_{m=0}^{M_i+} x[n - k_+[m]] - \sum_{m=0}^{M_i-} x[n - k_-[m]] \right). \quad (10)$$

The specific number of segments I and coefficient s_i values can be set manually. While testing with this approach, we have learned that a small number of segments can sound satisfying, e.g., $I = 4$.

3.3. Logarithmic Impulse Distribution

Since the loudness of later impulses needs to be minimized, the samples near the end of the sequence contribute very little to the convolved signal. To this end, the impulses can be concentrated at the beginning of the sequence, where loud impulses do not cause as much smearing of the transients in the decorrelated signals, could be beneficial. For this purpose, we can concentrate the impulses at the beginning of the sequence by distributing their location logarithmically. The grid size between each impulses is

$$T_{\text{dl}}(m) = \frac{T_{\text{T}}}{100} 10^{\frac{2T_{\text{d}} m}{M}} \quad (11)$$

by distributing the spacing logarithmically based on a total length T_{T} in samples and the density parameter T_{d} . This will also require the use of an updated location equation

$$k(m) = \text{round}\left(r_2[m](T_{\text{dl}}(m) - 1)\right) + \sum_{i=0}^m T_{\text{dl}}(i). \quad (12)$$

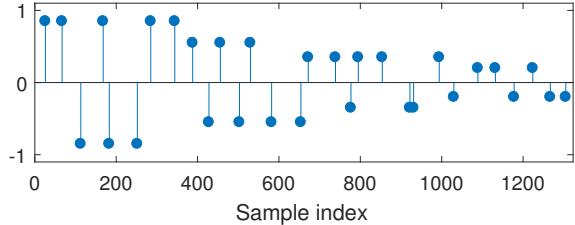


Figure 3: Velvet-noise with a segmented, staircase-like decay envelope.

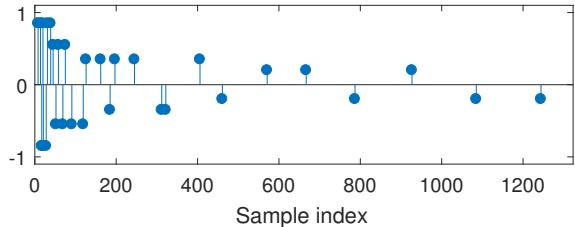


Figure 4: Velvet-noise sequence having a logarithmically widening grid size combined with a segmented decay envelope.

Figure 4 shows an example in which the impulse locations are distributed logarithmically and the segmented decay envelope is also applied. Since the impulses are not independent and identically distributed, this distribution may not preserve a flat power spectrum. However, given the low impulse density, this does not appear to have a significant impact on the spectral envelope in practice (see Section 4.2).

4. RESULTS

To evaluate the proposed methods, we first compared the computing cost of convolving velvet noise against the FFT-based white-noise method using different signal lengths and impulse density parameters. We analyzed the spectral envelope of the generated filters to ensure a spectral smoothness comparable to an exponentially decaying white-noise sequence and we calculated the cross-correlation after convolving a sine-sweep signal with the generated filters. The signal coherence was calculated over the audible frequency range. Finally, to complement the numerical evaluation methods, a preliminary perceptual test was conducted to validate the potential of the proposed method to externalize stereo sounds over headphones.

4.1. Computing Resources

We compared the total number of arithmetic operations required by the white-noise decorrelation to the VND method. To estimate the cost of the FFT-based convolution required by the white noise, we chose the radix-2 algorithm. However, a low-latency version would require extra operations to segment the input signal into multiple parts. The radix-2 algorithm has a complexity of $N \log_2(N)$ additions and $\frac{N}{2} \log_2(N)$ complex multiplications. We only counted the operations required to compute the FFT of the input signal, since the white-noise filter itself only needs to be converted into the spectral domain once [17]. The cost per sample was calculated using an $N/2$ window size for the input signal.

The chosen amplitude decay or the impulse distribution method do not have any impact on the required convolution, only the use of segments does. Therefore, the computational cost of the proposed methods can be regrouped into two categories: regular VND and segmented VND. In Fig. 5, the proposed methods are shown to outperform FFT-based convolution well below the required length of a decorrelation filter. Furthermore, for a length of 1024 samples, approximately 0.23 ms at a 44.1 kHz sampling rate, the proposed methods remain more efficient when the impulse density is kept below 6000 impulses per second or below 13000 impulses per second when using the segmented method with four segments.

In our sound examples, the decorrelation filters were chosen to be of length 30 ms, which leads to a signal length of 1323 samples for a sampling rate of 44.1 kHz. An impulse density of 1000 impulses per seconds was selected for the VND. The input signal was segmented into windows of 2048 samples. Since the convolution result has the length $2M - 1$, convolving two signals of length M requires a zero-padded FFT of 4096 samples to convolve with a filter of the same length. We can see from Table 1 that, for a length of 30 ms and a density of 1000 impulses per seconds, the VND with exponential decay envelope will require 76% less mathematical operations than fast convolution, whereas a segmented version of the algorithm with four segments will yield 87% fewer operations than fast convolution.

4.2. Spectral Envelope

In the context of decorrelation, a filter should be as spectrally flat as possible while randomizing the phase. Figure 6 compares the power spectra of a 30 ms white-noise signal with a VNS of the same length, both using an exponential decay envelope. Figure 7 compares the power spectra of a segmented VNS to another one that is similarly segmented but logarithmically distributed. The solid line shows the spectra of one randomly generated sequence. Since they are of short length, the randomization process does not lead to a perfectly flat spectrum. However, they both exhibit similar spectral behavior and they both have a comparable standard deviation over multiple randomized instances, as shown with the dashed lines.

4.3. Cross-Correlation

Although assessing the perceptual quality of a decorrelator numerically is difficult, the cross-correlation of two signals decorrelated with a VND should result in a similar plot when compared with a white-noise decorrelation. Assuming that the original input signal is a and its decorrelated version is $b = h * a$, where h is the decorrelation filter, the correlation between them is

$$r_{ab}(l) = \sum_n a(n + l)b(n), \quad (13)$$

where l is the correlation lag. For decorrelation purposes, the zero-lag case $r_{ab}(0)$ is of practical interest, since it corresponds to the maximum similarity between the two sequences. The normalized zero-lag correlation is given by

$$\rho_{ab} = \frac{\sum_n a(n)b(n)}{\sqrt{\sum_n a^2(n) \sum_n b^2(n)}}. \quad (14)$$

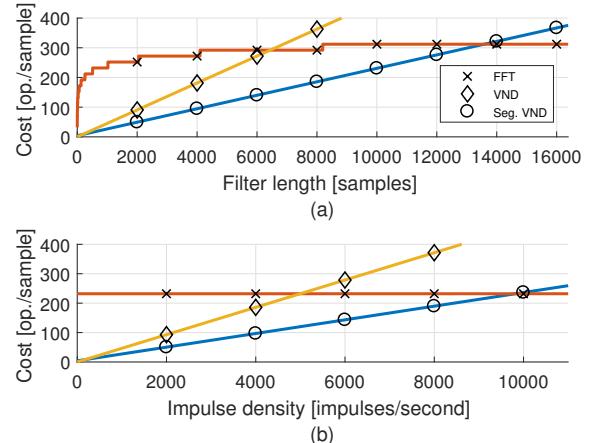


Figure 5: Comparison of computational cost between block FFT-based convolution (cross), VND (diamond), and segmented VND using four segments (circle): (a) the number of operations required to process each input sample for different filter lengths when the impulse density ρ is set to 1000 impulses per second and (b) the cost for various impulse density settings when the filter length is set to $L_s = 1024$ samples.

Table 1: Computational cost per output sample in FFT-based fast convolution, in exponentially decaying velvet-noise convolution, and in segmented velvet-noise convolution. The VNS has 30 non-zero elements (30 ms at a density of 1000 impulses/seconds at 44.1 kHz).

	Fast convolution	Exp. VND	Seg. VND
ADD	148	30	30
MUL	104	30	4
Total	252	60	34

The normalized cross-correlation coefficient is bounded between $\rho_{ab} \in [0, 1]$, and a lower maximum value indicates a more effective decorrelation.

We used a sine sweep to compare the correlation of two channels decorrelated either by a white-noise sequence or a segmented VNS (Fig. 8). Short windows of 20 ms were taken from both signals to calculate the cross-correlation values at different lag distances and generate a cross-correlogram. Figure 8(a) shows the auto-correlation patterns of the input signal, while Figs. 8(b) and (c) both display a comparable amount of decorrelation after applying the filters on each channels.

4.4. Coherence

Apart from the time-domain correlation, another useful metric is the cross-correlation in different frequency bands, called coherence. Normally, a 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 can use an octave or third-octave filterbank. The signals for the

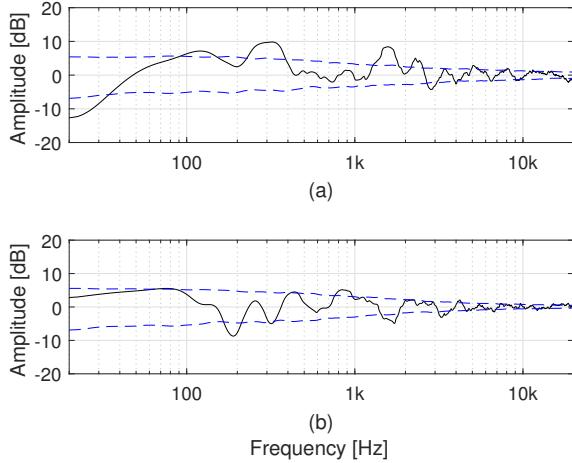


Figure 6: Spectral envelope of (a) an exponentially decaying white-noise sequence and (b) an exponentially decaying VNS. A third-octave smoothing has been applied to these magnitude responses. The dashed lines represent the average standard deviation of randomly generating 500 sequences.

q^{th} band are denoted as a_q and b_q and the correlation coefficient as

$$\rho_{ab}^{(q)} = \frac{\sum_n a_q(n)b_q(n)}{\sqrt{\sum_n a_q^2(n) \sum_n b_q^2(n)}}. \quad (15)$$

Using this equation, we calculated the coherence of both white-noise decorrelation and VND methods. In Fig. 9, the decorrelation was applied to a white-noise input signal, and the tests were run five hundred times to accumulate the averaged results. Figure 10 shows the same calculation comparing a segmented VNS to another that is similarly segmented but logarithmically distributed. The exponentially decaying VNS shows similar results as the white-noise sequence. However, the segmented version shows better decorrelation when the impulses are linearly distributed in time (see Fig. 10(a)), whereas it shows less decorrelation when distributed logarithmically (see Figure 10(b)). Based on these results, the segmented version shows the most promising decorrelation potential.

4.5. Perceptual Testing

Perceptual testing of decorrelation filters is important, since it is difficult to numerically evaluate the impact of the decorrelator on the stereo image. During an informal perceptual test, several subjects found the proposed VND to lead to a similar or better decorrelation in comparison to the exponentially decaying white-noise decorrelator. The spectral coloration and spreading were comparable, and the transients did not have artifacts, provided that latter impulses were sufficiently attenuated in the VNS. However, a more thorough listening test is required before drawing definitive conclusions. The VND could also be compared to other decorrelation methods, such as the allpass filter and filter-bank methods.

The preliminary listening suggests that the VND can provide a satisfactory externalization of stereo sounds over headphones at a lower computational cost than the white-noise decorrelation method. No extra perceivable artifacts were detected when using

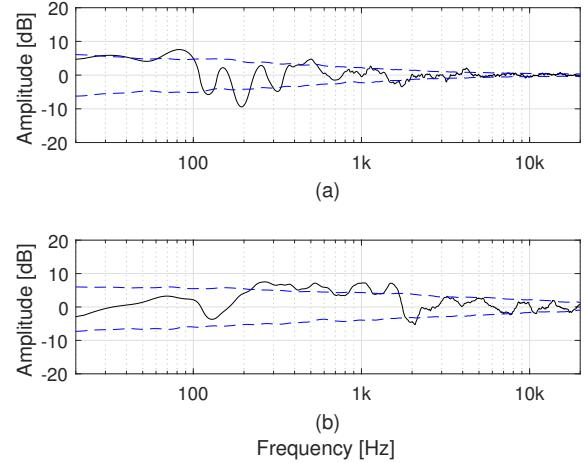


Figure 7: Spectral envelope of (a) a segmented VNS and (b) a segmented and logarithmically distributed VNS. Both use four segments of equal length and with values 0.85, 0.55, 0.35, and 0.20, respectively. A third-octave smoothing has been applied to these magnitude responses. The dashed lines represent the average standard deviation of randomly generating 500 sequences.

the segmented decay envelope instead of the exponential decay. The tests subjects did detect some low-frequency filtering and a spectrally fluctuating stereo image. However, these are common by-products of decorrelation filters. Audio examples are available at <http://research.spa.aalto.fi/publications/papers/dafx17-vnd/>.

5. CONCLUSION

The decorrelation of audio signals by convolving them with a short VNS was proposed. Since a VNS is a spectrally flat signal, it can create decorrelated signals in the same way as white noise. The proposed VND method allows a latency-free, time-domain convolution at favorable computing costs. A VNS having an appropriate density and a decaying temporal envelope provides a comparable decorrelation with a smaller computational cost when compared to a white-noise sequence. When used to replace white noise in our test scenario, the VND reduces the number of arithmetic operations by 76% to 87%, depending on the configuration. Several parameters can be used to alter the decorrelation itself, such as density, decaying envelope, segmentation, and impulse distribution. According to this study, the VNS with a segmented decay envelope appears to be the best option, since it produces as good a decorrelation as white noise without latency and using 87% less operations. Future work may study the impact of these various parameters as well as conduct a formal perceptual evaluation and a thorough comparison with other well-established decorrelation methods.

6. ACKNOWLEDGMENT

The authors would like to thank Luis Costa for proofreading this paper.

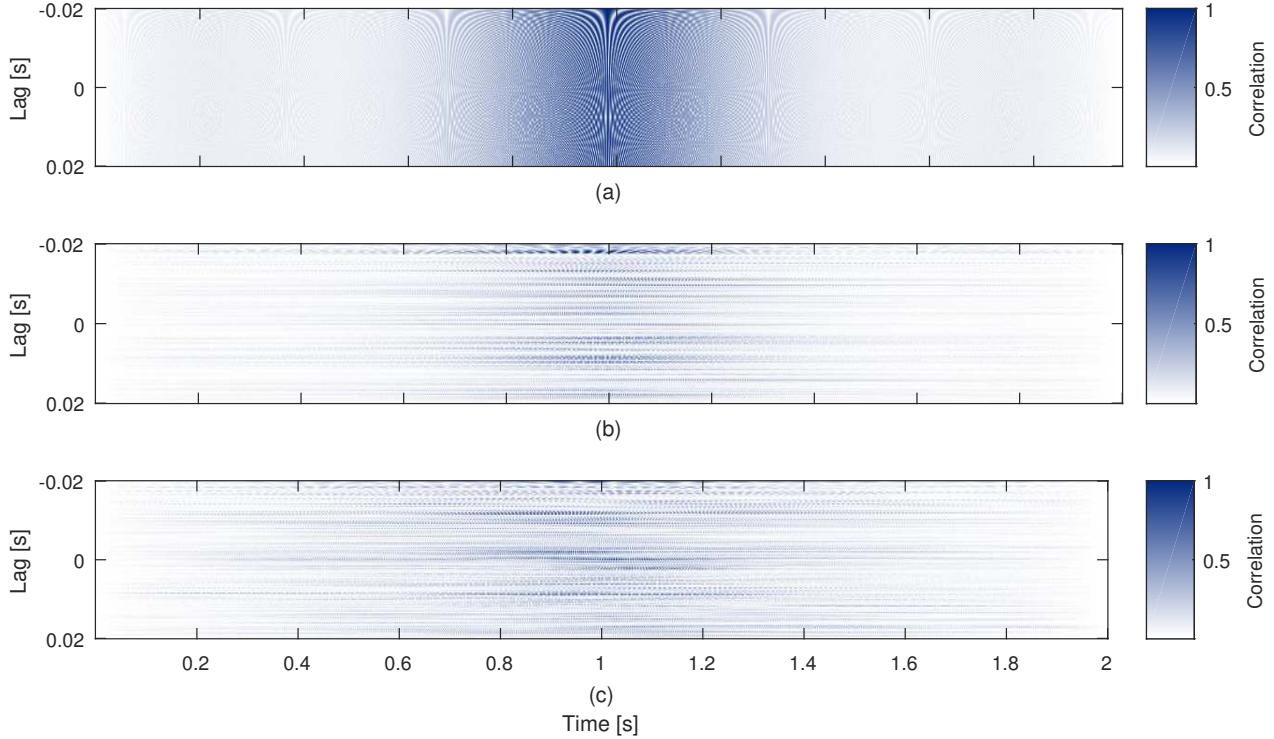


Figure 8: (a) Auto-correlogram of the sine sweep signal. (b) Cross-correlogram of both channels using the white-noise sequence for decorrelation. (c) Cross-correlogram of both channels using the segmented VND for decorrelation. The correlation values were normalized for visualization purposes.

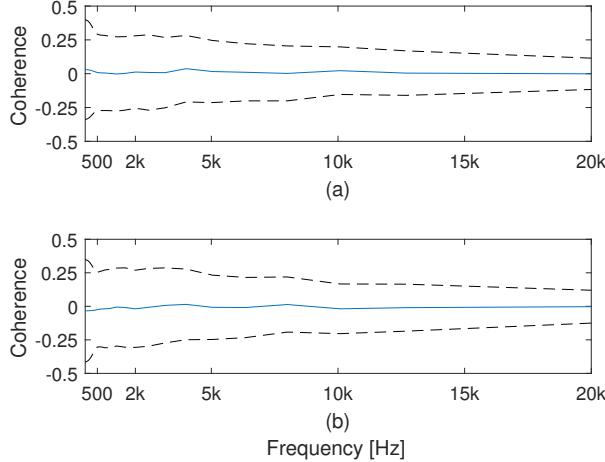


Figure 9: Coherence graph, from 20 Hz to 20 kHz, between left and right channels when using (a) white noise and (b) velvet noise. The dashed lines represent the average standard deviation of the data after running the algorithm 500 times.

7. REFERENCES

- [1] 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.

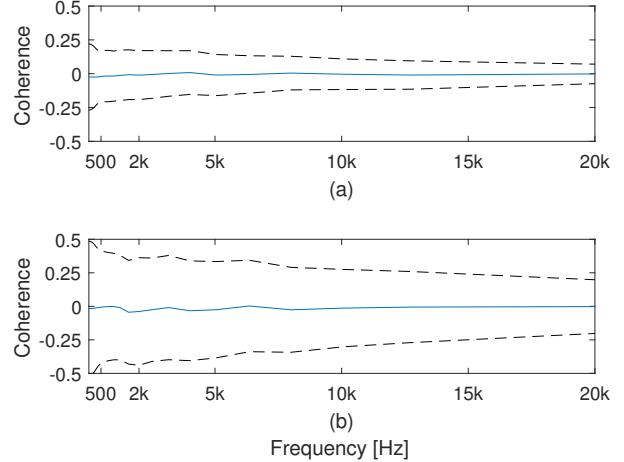


Figure 10: Coherence graph, from 20 Hz to 20 kHz, between left and right channels when using (a) a segmented VNS and (b) a segmented and logarithmically distributed VNS. The dashed lines represent the average standard deviation of the data after running the algorithm 500 times.

- [2] G. Potard and I. Burnett, “Decorrelation techniques for the rendering of apparent sound source width in 3D audio displays,” in *Proc. Int. Conf. Digital Audio Effects (DAFx-04)*, Naples, Italy, Oct. 2004, pp. 280–284.

- [3] 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.
- [4] J. Blauert, *Spatial Hearing: The Psychophysics of Human Sound Localization*, MIT press, 1997.
- [5] 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.
- [6] M. J. Hawksford and N. Harris, “Diffuse signal processing and acoustic source characterization for applications in synthetic loudspeaker arrays,” in *Proc. Audio Eng. Soc. 112nd Conv.*, Munich, Germany, May 2002.
- [7] B. Xie, B. Shi, and N. Xiang, “Audio signal decorrelation based on reciprocal-maximal length sequence filters and its applications to spatial sound,” in *Proc. Audio Eng. Soc. 133rd Conv.*, San Francisco, CA, USA, Oct. 2012.
- [8] E. Kermit-Canfield and J. Abel, “Signal decorrelation using perceptually informed allpass filters,” in *Proc. 19th Int. Conf. Digital Audio Effects (DAFx-16)*, Brno, Czech Republic, Sept. 2016, pp. 225–231.
- [9] M. Bouéri and C. Kyriakakis, “Audio signal decorrelation based on a critical band approach,” in *Proc. Audio Eng. Soc. 117th Conv.*, San Francisco, CA, USA, Oct. 2004.
- [10] A. Politis, J. Vilkamo, and V. Pulkki, “Sector-based parametric sound field reproduction in the spherical harmonic domain,” *IEEE J. Selected Topics in Signal Processing*, vol. 9, no. 5, pp. 852–866, Aug. 2015.
- [11] J. Vilkamo, B. Neugebauer, and J. Plogsties, “Sparse frequency-domain reverberator,” *J. Audio Eng. Soc.*, vol. 59, no. 12, pp. 936–943, Dec. 2012.
- [12] M. Karjalainen and H. Järveläinen, “Reverberation modeling using velvet noise,” in *Proc. 30th Int. Conf. Audio Eng. Soc.: Intelligent Audio Environments*, Saariselkä, Finland, Mar. 2007.
- [13] 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.
- [14] P. Rubak and L. G. Johansen, “Artificial reverberation based on a pseudo-random impulse response,” in *Proc. Audio Eng. Soc. 104th Conv.*, Amsterdam, The Netherlands, May 1998.
- [15] P. Rubak and L. G. Johansen, “Artificial reverberation based on a pseudo-random impulse response II,” in *Proc. Audio Eng. Soc. 106th Conv.*, Munich, Germany, May 1999.
- [16] 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.
- [17] 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.
- [18] S. Oksanen, J. Parker, A. Politis, and V. Välimäki, “A directional diffuse reverberation model for excavated tunnels in rock,” in *Proc. IEEE Int. Conf. Acoustics, Speech, and Signal Processing (ICASSP)*, Vancouver, Canada, May 2013, pp. 644–648.
- [19] B. Holm-Rasmussen, H.-M. Lehtonen, and V. Välimäki, “A new reverberator based on variable sparsity convolution,” in *Proc. 16th Int. Conf. Digital Audio Effects (DAFx-13)*, Maynooth, Ireland, Sept. 2013, pp. 344–350.
- [20] V. Välimäki, J. D. Parker, L. Savioja, J. O. Smith, and J. S. Abel, “More than 50 years of artificial reverberation,” in *Proc. Audio Eng. Soc. 60th Int. Conf. Dereverberation and Reverberation of Audio, Music, and Speech*, Leuven, Belgium, Feb. 2016.
- [21] 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.

PARAMETRIC ACOUSTIC CAMERA FOR REAL-TIME SOUND CAPTURE, ANALYSIS AND TRACKING

Leo McCormack, Symeon Delikaris-Manias and Ville Pulkki *

Acoustics Lab, Dept. of Signal Processing and Acoustics
School of Electrical Engineering
Aalto University, Espoo, FI-02150, Finland
leo.mccormack@aalto.fi

ABSTRACT

This paper details a software implementation of an acoustic camera, which utilises a spherical microphone array and a spherical camera. The software builds on the Cross Pattern Coherence (CroPaC) spatial filter, which has been shown to be effective in reverberant and noisy sound field conditions. It is based on determining the cross spectrum between two coincident beamformers. The technique is exploited in this work to capture and analyse sound scenes by estimating a probability-like parameter of sounds appearing at specific locations. Current techniques that utilise conventional beamformers perform poorly in reverberant and noisy conditions, due to the side-lobes of the beams used for the power-map. In this work we propose an additional algorithm to suppress side-lobes based on the product of multiple CroPaC beams. A Virtual Studio Technology (VST) plug-in has been developed for both the transformation of the time-domain microphone signals into the spherical harmonic domain and the main acoustic camera software; both of which can be downloaded from the companion web-page.

1. INTRODUCTION

Acoustic cameras are tools developed in the spatial audio community which are utilised for the capture and analysis of sound fields. In principle, an acoustic camera imitates the audiovisual aspect of how humans perceive a sound scene by visualising the sound field as a power-map. Incorporating this visual information within a power-map can significantly improve the understanding of the surrounding environment, such as the location and spatial spread of sound sources and potential reflections arriving from different surfaces. Essentially, any system that incorporates a video of the sound scene in addition to a power-map overlay can be labelled as an acoustic camera, of which several commercial systems are available today.

Capturing, analysing and tracking the position of sound sources is a useful technique with application in a variety of fields, which include reflection tracking in architectural acoustics [1, 2, 3], sonar navigation and object detection [4, 5], espionage and the military [6, 7]. An acoustic camera can also be used to identify sound insulation issues and faults in electrical and mechanical equipment. This is due to the fact that in many of these scenarios, the fault can be identified as an area that emits the most sound energy. Therefore, calculating the energy in multiple directions and subsequently generating a power-map that depicts their energies relative to each-other is an effective method of identifying

the problem area. There have also been instances which incorporate a spherical video. One example is in [8], where a parabolic mirror with a single camera sensor is utilised and the image is then obtained after unwrapping. For a study using single and multiple cameras, the reader is directed to [9].

One approach to developing an acoustic camera is to utilise a rectangular or circular microphone array and then apply beamforming in several directions to generate the power-map. However, a more common approach is to use a spherical microphone array, as it conveniently allows for the decomposition of the sound scene into individual spatial components, referred to as spherical harmonics [10]. Using these spherical harmonics, it is possible to carry out beamforming with similar spatial resolution for all directions on the sphere. Therefore, these are preferred for use-cases in which a large field of view has to be analysed. The most common signal-independent beamformer in the spherical harmonic domain is based on the plane wave decomposition (PWD) algorithm, which (as the name would suggest) relies on the assumption that the sound sources are received as plane-waves, which makes it suited only for far-field sound sources [11]. These beam patterns can be further manipulated using in-phase [12], Dolph-Chebyshev [10] or maximum energy weightings [13].

Signal-dependent beamformers can also be utilised in acoustic cameras with the penalty of higher computational cost. A common solution is the minimum-variance distortion-less response (MVDR) algorithm [14]. This approach takes into account the inter-channel dependencies between the microphone array signals, in an attempt to enhance the beamformers performance by placing nulls to the interferers. However, the performance of such an algorithm is relatively sensitive in scenarios where high background noise and/or reverberation are present in the sound scene [15]. An alternative approach, proposed in [16], is to apply pre-processing in order to separate the direct components from the diffuse field. This subspace-based separation has been shown to dramatically improve the performance of existing super-resolution imaging algorithms [17]. Another popular subspace-based approach is the multiple signal classification (MUSIC) algorithm [18], which has been orientated as a multiple speaker localisation method in [19], by incorporating a direct-path dominance test.

A recent spatial filtering technique, which can potentially be applied to spherical microphone arrays, is the cross-pattern coherence (CroPaC) algorithm [20]. It is based on measuring the correlation between coincident beamformers and providing a post filter that will suppress noise, interferers and reverberation. The advantage of CroPaC, when compared to other spatial filtering techniques, is that it does not require the direct estimation of the microphone noise. The algorithm has recently been extended in the spherical harmonic domain for arbitrary combinations of beam-

* The research leading to these results has received funding from the Aalto ELEC school.

formers in [21].

The purpose of this work is to detail a scalable acoustic camera system that utilises a spherical microphone array and a spherical video camera, which are placed in a near-coincident fashion. Several different static and adaptive beamforming techniques have been implemented within the system and care has been taken to ensure that the proposed system is accessible to a wide range of acoustic practitioners. Additionally, we investigate the use of a coherence-based parameter to generate the power-maps. The main contributions of this work can be summarised as:

- The capture and analysis of a sound scene using a microphone array and subsequently estimating a parameter to determine sound source activity in specific directions.
- The development of a real-time VST plug-in, for spatially encoding the microphone array signals into spherical harmonic signals.
- Devising a real-time acoustic camera, also implemented as a VST plug-in, by utilising a commercially available spherical microphone array and spherical camera.
- The use of vector-base amplitude panning (VBAP) [22], in order to interpolate the power-map grid.
- Optimal side-lobe suppression of the CroPaC spatial filters for analysis purposes.

This paper is organised as follows. In Section 2 we provide the necessary background on spherical microphone array processing, which includes the encoding of the microphone signals into spherical harmonic signals and common signal-dependent and signal-independent beamforming techniques. In Section 3 we elaborate the proposed theoretical background for generating the power-maps. In Section 4 the details of the hardware and software are shown in detail. Finally, in Section 5 we present our conclusions.

2. SPHERICAL MICROPHONE ARRAY PROCESSING

Spherical microphone arrays (SMA) are commonly utilised for sound field analysis for three-dimensional (3-D) spaces, as they provide a similar performance in all directions when sensors are placed uniformly or nearly-uniformly on the sphere. In this section we provide a brief overview of how to estimate the spherical harmonic signals from the microphone signals and how to perform adaptive and non-adaptive beamforming in the spherical harmonic domain. Only the details required for the current implementation are included here. For a detailed overview of these methods, the reader is referred to [12, 23, 10, 24, 25].

Note that matrices, \mathbf{M} , have been denoted using bold upper-case letters and vectors, \mathbf{v} , are denoted with bold lower-case letters.

2.1. Spatial encoding

The SMA may be denoted with Q sensors at $\Omega_q = (\theta, \phi, r)$ locations with $\theta \in [-\pi/2, \pi/2]$ denoting elevation angle, $\phi \in [-\pi, \pi]$ azimuthal angle and r the radius. A common approach is to decompose the microphone input signal, $\mathbf{x} \in \mathbb{C}^{Q \times 1}$, into a set of spherical harmonic signals for each frequency. The accuracy of this decomposition depends on the microphone distribution on the sphere, the type of the array and the radius [10]. The total number of microphones defines the highest order of spherical harmonic

signals L that can be estimated. Please note that the frequency and time indexes are omitted for the brevity of notation.

The spherical harmonic signals can be estimated as

$$\mathbf{s} = \mathbf{W}\mathbf{x}, \quad (1)$$

where

$$\mathbf{s} = [s_{00}, s_{1-1}, s_{10}, \dots, s_{L-1}, s_L]^T \in \mathbb{C}^{(L+1)^2 \times 1}, \quad (2)$$

are the spherical harmonic signals and $\mathbf{W} \in \mathbb{C}^{(L+1)^2 \times Q}$ is the frequency-dependent spatial encoding matrix. For uniform or nearly-uniform microphone arrangements, it can be calculated as

$$\mathbf{W} = \alpha_q \mathbf{W}_l \mathbf{Y}^\dagger, \quad (3)$$

where α_q are the sampling weights, which depend on the microphone distribution on the sphere [10]. The sampling weights can be calculated as $\alpha_q = \frac{4\pi}{Q}$. Furthermore, $\mathbf{W}_l \in \mathbb{C}^{(L+1)^2 \times (L+1)^2}$ is an equalisation matrix that eliminates the effect of the sphere, defined as

$$\mathbf{W}_l = \begin{bmatrix} w_0 & & & & \\ & w_1 & & & \\ & & w_1 & & \\ & & & w_1 & \\ & & & & \ddots \\ & & & & & w_L \end{bmatrix}, \quad (4)$$

where

$$w_l = \frac{1}{b_l} \frac{|b_l|^2}{|b_l|^2 + \lambda^2}, \quad (5)$$

where b_l are frequency and order-dependent modal coefficients, which contain the information of the type of the array, open or rigid, and the type of sensors, omnidirectional or directional. Lastly, λ is a regularisation parameter that influences the microphone noise amplification. For details of some alternative options for calculating the equalisation matrix \mathbf{W}_l , the reader is referred to [26, 27], or for a signal-dependent encoder [28]. $\mathbf{Y}(\Omega_q) \in \mathbb{R}^{Q \times (L+1)^2}$ is a matrix containing the spherical harmonics

$$\mathbf{Y}(\Omega_q) = \begin{bmatrix} Y_{00}(\Omega_1) & Y_{00}(\Omega_2) & \dots & Y_{00}(\Omega_Q) \\ Y_{-11}(\Omega_1) & Y_{-11}(\Omega_2) & \dots & Y_{-11}(\Omega_Q) \\ Y_{10}(\Omega_1) & Y_{10}(\Omega_2) & \dots & Y_{10}(\Omega_Q) \\ Y_{11}(\Omega_1) & Y_{11}(\Omega_2) & \dots & Y_{11}(\Omega_Q) \\ \vdots & \vdots & \vdots & \vdots \\ Y_{LL}(\Omega_1) & Y_{LL}(\Omega_2) & \dots & Y_{LL}(\Omega_Q) \end{bmatrix}^T, \quad (6)$$

where Y_{lm} are the individual spherical harmonics of order $l \geq 0$ and degree $m \in [-l, l]$.

2.2. Generating power-maps and pseudo-spectrums in the spherical harmonic domain

A power-map can be generated by steering beamformers in multiple directions, as dictated by some form of pre-defined grid. The energy of these beamformed signals can then be calculated and subsequently plotted with an appropriate colour gradient.

Static beamformers in the spherical harmonic domain can be generated using

$$y(\Omega_j) = \mathbf{w}_{\text{PWD}}^H \mathbf{s}, \quad (7)$$

where y denotes the output signal for direction Ω_j and $\mathbf{w}_{\text{PWD}} \in \mathbb{C}^{(L+1)^2 \times 1}$ is a vector containing the beamforming weights, calculated as

$$\mathbf{w}_{\text{PWD}} = \mathbf{y}(\Omega_j) \odot \mathbf{d}, \quad (8)$$

where $\mathbf{y}(\Omega_j) \in \mathbb{C}^{1 \times (L+1)^2}$ are the spherical harmonics for direction Ω_j , \odot denotes the Hadamard product and \mathbf{d} is a vector of weights, defined as

$$\mathbf{d} = [d_0, d_1, d_1, d_1, \dots, d_L]^T \in \mathbb{R}^{(L+1)^2 \times 1}. \quad (9)$$

The weights \mathbf{d} can be adjusted to synthesise different types of axis symmetric beamformers: regular [10], in-phase [12], maximum energy [13, 23] and Dolph-Chebyshev [10]. A comparison of the performance of these beamformers as DOA estimators is given in [21]. The spherical harmonic signals or a spherical harmonic-domain beamformer can be steered to an arbitrary angle. Steering matrices for rotationally symmetric functions can be obtained using real multipliers [29]. Rotations for arbitrary angles can also be performed by utilising the Wigner-D weighting [30], or by utilising projection methods [31].

Adaptive beamformers may also be utilised for generating power-maps. Typically, these signal-dependent methods operate on the covariance matrix of the spherical harmonic signals $\mathbf{C}_{lm} \in \mathbb{C}^{(L+1)^2 \times (L+1)^2}$, which can be estimated as

$$\mathbf{C}_{lm} = \mathbf{W} \mathbf{C}_x \mathbf{W}^H, \quad (10)$$

where $\mathbf{C}_x = \mathbb{E} [\mathbf{x} \mathbf{x}^H] \in \mathbb{C}^{Q \times Q}$ is the covariance matrix of the microphone input signals and $\mathbb{E} [\cdot]$ represents a statistical expectation operator. The covariance matrix can be estimated using an average over finite time frames, typically in the range of tens of milliseconds, or by employing recursive schemes.

A popular signal-dependent beamforming approach is to solve the MVDR minimisation problem, which aims to synthesise a beam that adaptively changes according to the input signal. The response of this beamformer is constrained to have unity gain in the look direction, while the variance of the output is minimised [10]. This minimisation problem is defined as

$$\begin{aligned} & \text{minimise } \mathbf{w} \mathbf{C}_{lm} \mathbf{w}^H \\ & \text{subject to } \mathbf{y}(\Omega_j) \mathbf{w}^H = 1, \end{aligned} \quad (11)$$

which can be solved to obtain the beamforming weights using

$$\mathbf{w} = \frac{\mathbf{y}(\Omega_j) \mathbf{C}_{lm}^{-1}}{\mathbf{y}(\Omega_j) \mathbf{C}_{lm}^{-1} \mathbf{y}^H(\Omega_j)}. \quad (12)$$

The main advantage of applying the MVDR algorithm in the spherical harmonic domain, instead of utilising the microphone signals directly, is that the steering vectors are simply the spherical harmonics for different angles.

Alternatively, instead of generating a traditional power-map using beamformers, a pseudo-spectrum may be obtained by utilising subspace methods, such as the MUSIC algorithm described in [19]. First, the signal $\mathbf{U}_s \in \mathbb{C}^{1 \times 1}$ and noise $\mathbf{U}_n \in \mathbb{C}^{(L+1)^2-1 \times (L+1)^2-1}$ subspaces are obtained via a singular-value decomposition (SVD) of the spherical harmonic covariance matrix

$$\mathbf{C}_{lm} = \mathbf{U} \boldsymbol{\Sigma} \mathbf{U}^H = [\mathbf{U}_s \mathbf{U}_n] \begin{bmatrix} \boldsymbol{\Sigma}_s & \mathbf{0} \\ \mathbf{0} & \boldsymbol{\Sigma}_n \end{bmatrix} \begin{bmatrix} \mathbf{U}_s \\ \mathbf{U}_n \end{bmatrix}, \quad (13)$$

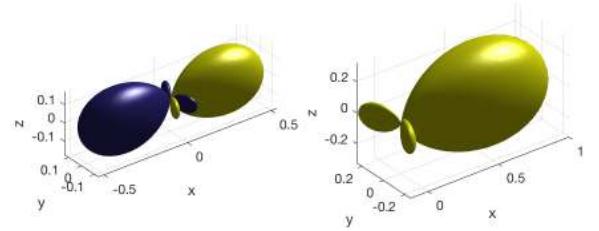


Figure 1: Visualisation of a CroPaC beam for $L = 2$ before and after half-wave rectification, shown in the left and right plots, respectively.

where $\boldsymbol{\Sigma}$ denotes the singular values and \mathbf{C}_{lm} is of unit effective rank.

A direct-path dominance test is then performed, in order to ascertain which time-frequency bins provide a significant contribution to the direct path of a sound source. These time-frequency bins are selected by determining whether the first singular value, σ_1 of matrix $\boldsymbol{\Sigma}$ is significantly larger than the second singular value, σ_2

$$\frac{\sigma_1}{\sigma_2} > \beta, \quad (14)$$

where $\beta \geq 1$ is a threshold parameter.

Essentially, this subspace method is based on the assumption that the direct path of a sound source will be characterised with higher energy than the reflecting path [19]. However, unlike the PWD and MVDR approaches, where a power-map is generated by depicting the relative energy of beamformers, the MUSIC pseudo-spectrum is obtained as

$$S_{\text{MAP}}(\Omega_j) = \frac{1}{\mathbf{y}(\Omega_j) (\mathbf{I} - \mathbf{U}_s \mathbf{U}_s^H) \mathbf{y}^H(\Omega_j)}, \quad (15)$$

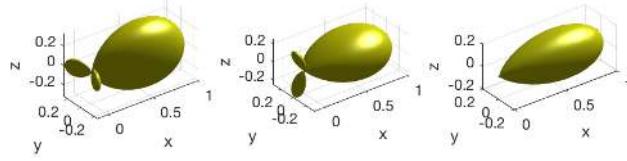
where S_{MAP} is the pseudo-spectrum value for direction Ω_j , and \mathbf{I} is an identity matrix.

3. COHERENCE-BASED SOUND SOURCE TRACKING

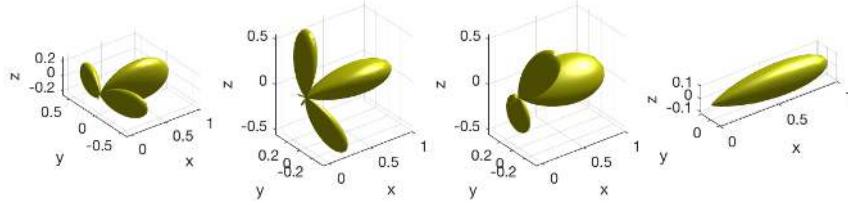
In this work, instead of utilising beamformers to generate an energy-based power-map or utilising subspace methods to generate a pseudo-spectrum, we estimate a parameter using the cross spectrum of different beamformers. This parameter, the cross pattern coherence (CroPaC), has been utilised for spatial filtering applications, where it has been shown to be effective in noisy and reverberant conditions [20, 32]. In this section we propose a generalisation of the algorithm presented in [20] for SMAs, using static beamformers and microphone arrays that define an arbitrary order L . A novel approach of suppressing the side-lobes of CroPaC beams is also explored.

3.1. Cross-spectrum-based parameter estimation

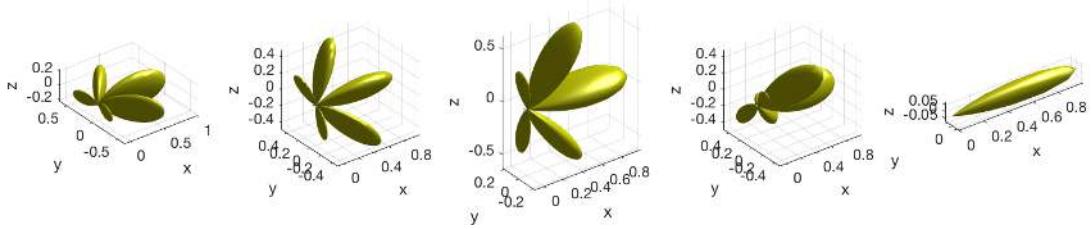
The CroPaC algorithm estimates the probability of a sound source emanating from a specific direction in a 3-D space. The time-domain microphone signals are initially transformed into the spherical harmonic domain according to the formulation shown in Section 2.1, up to order L . The spherical harmonic signals are



(a) Side-lobe suppression for order $L = 1$. The two beams on the left show the rotating beam patterns and the beam on the far-right is the resulting beam pattern.



(b) Side-lobe suppression for order $L = 2$. The three beams on the left show the rotating beam patterns and the beam on the far-right is the resulting beam pattern.



(c) Side-lobe suppression for order $L = 3$. The four beams on the left show the rotating beam patterns and the beam on the far-right is the resulting beam pattern.

Figure 2: Visualisation of side-lobe cancellation for $L = 1, 2, 3$.

then transformed into the time-frequency domain and a parameter is estimated for each frequency, k , and time index, n . The cross spectrum is then calculated between two spherical harmonic signals of orders L and $L + 1$ and the same degree m

$$G(\Omega_j, k, n) = \lambda \frac{\Re[\mathbf{s}_L(\Omega_j, k, n)^* \mathbf{s}_{L-1}(\Omega_j, k, n)]}{\sum_{L=1}^{L+1} |\mathbf{s}_L(\Omega_j, k, n)|^2}, \quad (16)$$

where \Re denotes the real operator, \mathbf{s}_L and \mathbf{s}_{L-1} are the spherical harmonic signals for a look direction Ω_j and the same degree m , $*$ denotes the complex conjugate and λ is an order-dependent normalisation factor to ensure that $G_{\text{MAP}} \in [0, 1]$. The normalisation factor can be calculated as

$$\lambda = \frac{(L+1)^2 - (L-1)^2 + 1}{2} = \frac{4L+1}{2}. \quad (17)$$

The power-map is then estimated for a grid of look directions $\Omega = (\Omega_1, \Omega_2, \dots, \Omega_J)$, averaged across frequencies and subjected to a

half-wave rectifier. The resulting power-map is then given by

$$G_{\text{MAP}}(\Omega_j, n) = \max \left[0, \frac{1}{K} \sum_{k=1}^K G(\Omega_j, k, n) \right]. \quad (18)$$

The half-wave rectification process ensures that only sounds arriving from the look direction are analysed. An illustration of the effect of the half-wave rectification process to the directional selectivity of the CroPaC beams is depicted in Fig. 1.

3.2. Side-lobe suppression

The calculation of the spectrum between different orders of beam-formers results in the creation of unwanted side-lobes that exhibit different shapes depending on the order. A visual depiction of these aberrations, in Fig. 2, have been generated by multiplying the following spherical harmonics together: $Y_{LL} Y_{(L+1)(L+1)}$ for

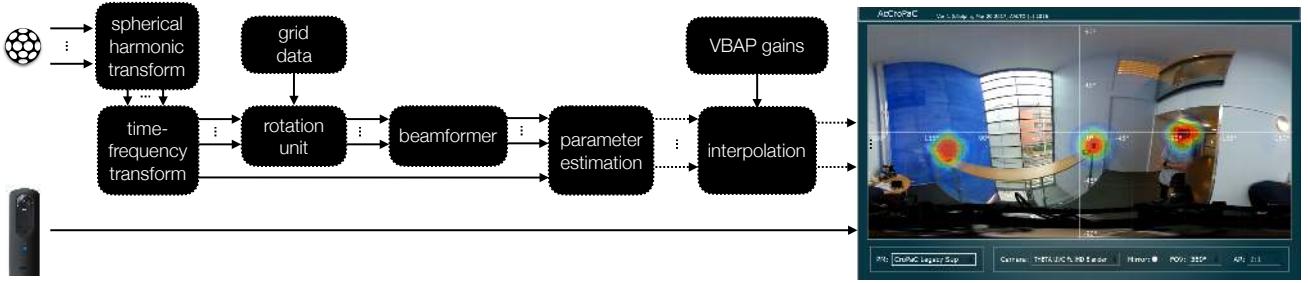


Figure 3: Block diagram of the proposed parametric acoustic camera system. The microphone array signals are processed in the time-frequency domain using the proposed parametric analysis. The output is then averaged across frequencies and the grid is interpolated with VBAP for visualisation. The resulting power-map is projected on top of the spherical video.

$L = 1$ (Fig. 2, (a), left), $L = 2$ (Fig. 2, (b), left) and $L = 3$ (Fig. 2, (c), left).

These side-lobes can potentially introduce biases in the power-map. Therefore, in this sub-section we propose a technique to suppress these side-lobes by multiplying rotated versions of the estimated beams. The number of estimated beams is determined by the order L . The side-lobe suppressing parameter G_{SUP} is estimated as

$$G_{\text{SUP}}(\Omega_j, n) = \begin{cases} G_{\text{MAP}}(\Omega_j, n) & , \text{if } L = 1 \\ \prod_{i=1}^L G_{\text{MAP}}^{\rho_i}(\Omega_j, n) & , \text{if } L > 1, \end{cases} \quad (19)$$

where ρ_i is a parameter that defines an axis symmetric roll on the direction of the beam of $\frac{\pi}{L}$ radians. Such a roll can successfully suppress side-lobes that are generated by the multiplication of the different spherical harmonics. This concept is illustrated in Fig. 2. Each row illustrates the side-lobe suppression for different orders. In the top row $L = 1$, which results in a single roll of $\frac{\pi}{2}$ in (19). For $L = 2$ (middle row) and $L = 3$ (bottom row) three and four rolls of $\frac{\pi}{3}$ and $\frac{\pi}{4}$ are applied, respectively. The resulting enhanced beam patterns $G_{\text{SUP}} \in [0, 1]$, which are derived from the product of multiple $G_{\text{MAP}} \in [0, 1]$, are shown on the right-hand side of the figures.

4. IMPLEMENTATION DETAILS

In order to make the software easily accessible, the acoustic camera was implemented as a virtual studio technology (VST) audio plug-in¹ using the open-source JUCE framework. The motivation for selecting the VST architecture, is the wide range of digital audio workstations (DAWs) that support them. This enables an acoustic practitioner to select their DAW of choice, which will in turn act as the bridge between real-time microphone signal capture and the subsequent visualisation of the energy distribution in the sound scene. The rationale behind the selection of JUCE is the many useful classes it offers; most notably of which is camera support, which allows for real-time video to be placed behind the corresponding power-map. Additionally, the framework is developed with cross-platform support in mind and can also produce other audio plug-in formats, provided that the corresponding source development kits are linked to the project.

¹The VST plug-ins are available for download on the companion web-page: <http://research.spa.aalto.fi/publications/papers/acousticCamera/>

The algorithms within the acoustic camera have been generalised to support spherical harmonic signals up to the 7th order. These signals can be optionally generated by using the accompanying Mic2SH VST, which accepts input signals from spherical microphone arrays such as A-format microphones (1st order) or the Eigenmike (up to 4th order). In the case of the Eigenmike, Mic2SH will also perform the necessary frequency-dependent equalisation, described in Section 2.1, in order to mitigate the radial dependency incurred when estimating the pressure on a rigid sphere. Different equalisation strategies have been implemented that are common in the literature, such as the Tikhonov-based regularised inversion [23] and soft limiting [33].

In order to optimise the linear algebra operations, the code within the audio plug-in has been written to conform to the basic linear algebra library (BLAS) standard. Other operations such as the lower-upper (LU) factorisation and SVD are addressed by the linear algebra package (LAPACK) standard; for which Apple’s accelerate framework and Intel’s MKL are supported for the Mac OSX and Windows versions, respectively.

The overall block diagram of the proposed system is shown in Fig. 3. The time-domain microphone array signals are initially transformed into spherical harmonic signals using the Mic2SH audio plug-in, which are then transformed into the time-frequency domain by the acoustic camera. For computational efficiency reasons, the spherical harmonic signals are rotated after the time-frequency transform towards the points defined by the pre-computed spherical grid. These signals are then fed into a beamformer unit, which forms the two beams that are required to compute the cross-spectrum based parameter for each grid point. Note that when the side-lobe suppression mode is enabled, one parameter is estimated per roll and the resulting parameters are multiplied, as defined in (19). For visualisation, the parameter value at each of the grid points is interpolated using VBAP and projected on top of the spherical video.

The user-interface for the acoustic camera consists of a view window and a parameter editor (see Fig. 3). The view window displays the camera feed and overlays the user selected power-map in real-time. The field-of view (FOV) and the aspect ratio are user definable in the parameter editor, which allows the VST to accommodate a wide range of different web-cam devices. Additionally, the image frames from the camera can be optionally mirrored using an appropriate affine transformation (left-right, or up-down); in order to accommodate for a variety of different camera orientations.

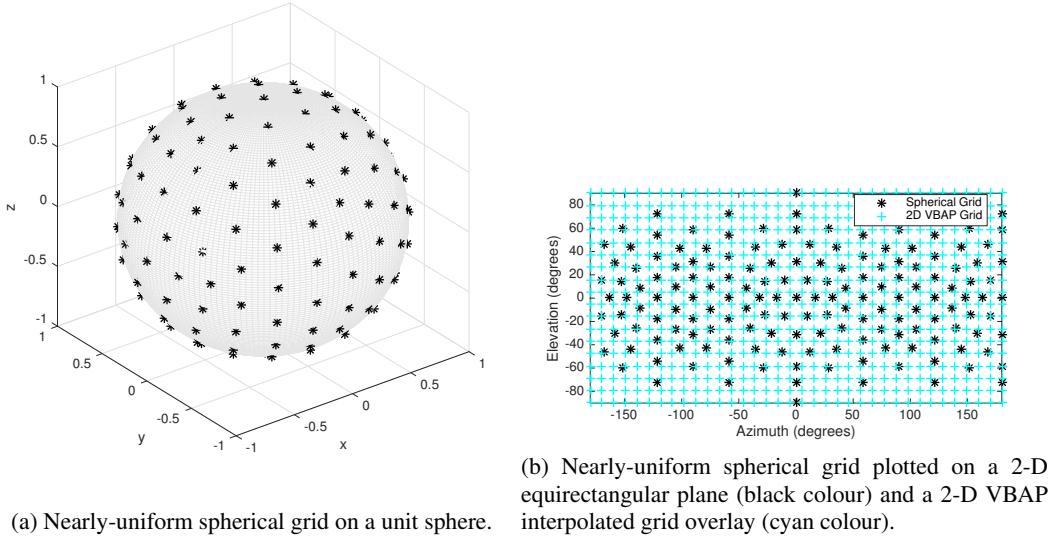


Figure 4: Spherical and interpolated grids.

4.1. Time-frequency transform

The time-frequency transform utilised in this work is filter-bank-based and was implemented originally for the study in [21]². The filter-bank was configured to use a hop size of 128 samples and an FFT size of 1024. Additionally, the optional hybrid filtering mode offered by the filter-bank was enabled, which allows for more resolution in the low frequency region by dividing the lowest four bands into eight; thus, attaining 133 frequency bands in total. A sampling rate of 48 kHz was chosen, and the power-map analysis utilises frequency bands with centre frequencies between [140, 8000] Hz. The upper limit of 8000 Hz was selected due to the spatial aliasing of the microphone array used [34].

4.2. Power-map modes and sampling grids

The power-map is generated by sampling the sphere with a spherical grid. A precomputed almost-uniform spherical grid was chosen that provides 252 nearly-uniformly distributed data points on the sphere. The grid is based on the 3LD library [35], where the points are generated by utilising geodesic spheres. This is performed by tessellating the facets of a polyhedron and extending them to the radius of the original polyhedron. The intersection points between them are the points of the spherical grid. Two different power-map modes and two pseudo-spectrum methods were implemented in the spherical harmonic domain: conventional signal-independent beamformers (PWD, minimum side-lobe, maximum energy and Dolph-Chebyshev); MVDR beamformers; multiple signal classification (MUSIC); and the proposed cross-spectrum-based with the additional side-lobe suppression. The power-map/pseudo-spectrum values are then summed over the analysis frequency bands and averaged over time slots using a one-pole filter

$$\hat{G}_{\text{SUP}}(\Omega_j, n) = \alpha \hat{G}_{\text{SUP}}(\Omega_j, n) + (1 - \alpha) G_{\text{SUP}}(\Omega_j, n - 1), \quad (20)$$

²<https://github.com/jvilkamo/afSTFT>

where $\alpha \in [0, 1]$ is the smoothing parameter. The spherical power-map values are then interpolated to attain a two-dimensional (2-D) power-map, using pre-computed VBAP gains. The spherical and interpolated grids are shown in Fig. 4. These 2-D power-maps are then further interpolated using bi-cubic interpolation depending on the display settings and are normalised such that $\hat{G}_{\text{SUP}} \in [0, 1]$. The pixels that correspond to the 2-D interpolated results are then coloured appropriately, such that red indicates high energy and blue indicates low energy. Additionally, the transparency factor is gradually increased for the lower energy valued beams to ensure that they do not unnecessarily detract from the video stream.

4.3. Example power-maps

Power-maps examples are shown in Fig. 5 for four different modes: the basic PWD beamformer, the adaptive MVDR beamformer, the subspace MUSIC approach, and the proposed technique. The recordings were performed by utilising the Eigenmike microphone array and a RICOH Theta S spherical camera. Fourth order spherical harmonic signals were generated using the accompanying Mic2SH VST plugin, which were then used by all four power-map modes. The video was unwrapped using the software provided by RICOH and then combined with the calculated power-map to complete the acoustic camera system. However, since the camera may not be facing the same look direction as the microphone array, a calibration process is required in order to align the power-map with the video stream. However, it should be noted that since the two devices do not share a common origin, sources that are very close to the array may not be correctly aligned. The resulting power-maps are shown for two different recording scenarios: a staircase of high reverberation time of approximately 2 seconds (Fig. 5, bottom) and a corridor of approximately 1.5 seconds (Fig. 5, top).

It can be seen from Fig. 5(top) that there is one direct source and at least one prominent early reflection. However, in the case of PWD, the distinction between the two paths is the least clear, and also erroneously indicates that the sources are spatially larger

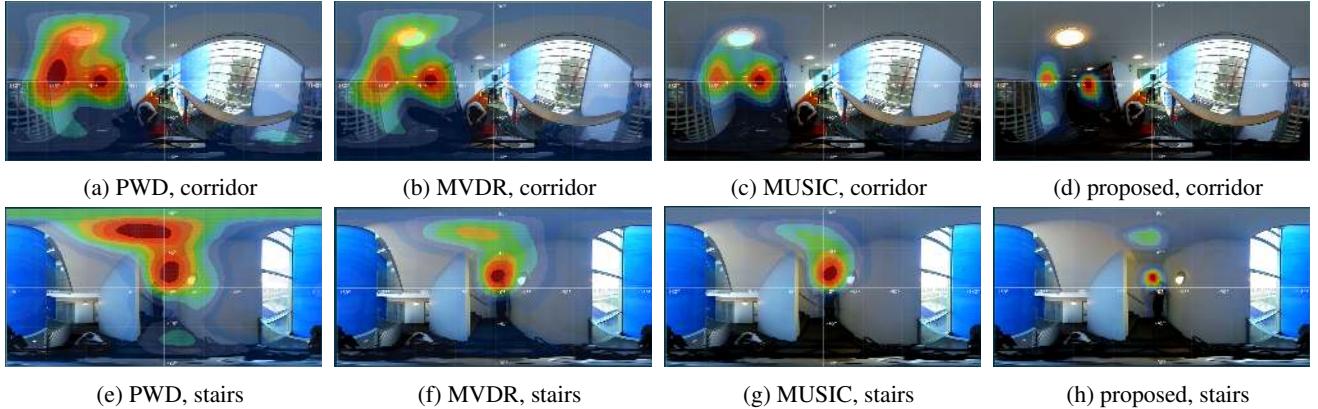


Figure 5: Images of the acoustic camera VST display, while using fourth order spherical harmonic signals and the four processing modes in reverberant environments.

than they actually are. The distinction between the two paths is improved slightly when using MVDR beamformers, which is improved further when utilising the MUSIC algorithm. However, in the case of the proposed technique, the two paths are now completely isolated and a second early reflection with lower energy is now visible; which is not as evident in the other three methods. PWD also indicates a sound source that is likely the result of the side-lobes pointing towards the real sound source; thus, highlighting the importance of side-lobe suppression for acoustic camera applications. Fig. 5(bottom) indicates similar performance; however, in the case of MUSIC, the ceiling reflection is more difficult to distinguish as a separate entity.

5. CONCLUSIONS

This paper has presented an acoustic camera that is easily accessible as a VST plug-in. Among the possible power-map modes available, is the proposed coherence-based parameter, which can be tuned to this particular use case via additional suppression of the side-lobes. This method presents an intuitive approach to attaining a power-map, and is potentially easier and computationally cheaper to implement than MVDR or MUSIC, as it does not rely on lower-upper decompositions, Gaussian Elimination, or singular value decompositions. It is also demonstrated that in the simple recording scenarios, the proposed method can be inherently tolerant to reverberation.

6. REFERENCES

- [1] Adam O'Donovan, Ramani Duraiswami, and Dmitry Zotkin, “Imaging concert hall acoustics using visual and audio cameras,” in *Acoustics, Speech and Signal Processing, 2008. ICASSP 2008. IEEE International Conference on*. IEEE, 2008, pp. 5284–5287.
- [2] Angelo Farina, Alberto Amendola, Andrea Capra, and Christian Varani, “Spatial analysis of room impulse responses captured with a 32-capsule microphone array,” in *Audio Engineering Society Convention 130*. Audio Engineering Society, 2011.
- [3] Lucio Bianchi, Marco Verdi, Fabio Antonacci, Augusto Sarti, and Stefano Tubaro, “High resolution imaging of acoustic reflections with spherical microphone arrays,” in *Applications of Signal Processing to Audio and Acoustics (WASPAA), 2015 IEEE Workshop on*. IEEE, 2015, pp. 1–5.
- [4] GC Carter, “Time delay estimation for passive sonar signal processing,” *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 29, no. 3, pp. 463–470, 1981.
- [5] H Song, WA Kuperman, WS Hodgkiss, Peter Gerstoft, and Jea Soo Kim, “Null broadening with snapshot-deficient covariance matrices in passive sonar,” *IEEE journal of Oceanic Engineering*, vol. 28, no. 2, pp. 250–261, 2003.
- [6] RK Hansen and PA Andersen, “A 3d underwater acoustic camera: properties and applications,” in *Acoustical Imaging*, pp. 607–611. Springer, 1996.
- [7] Russell A Moursund, Thomas J Carlson, and Rock D Peters, “A fisheries application of a dual-frequency identification sonar acoustic camera,” *ICES Journal of Marine Science: Journal du Conseil*, vol. 60, no. 3, pp. 678–683, 2003.
- [8] Leonardo Scopuce, Angelo Farina, and Andrea Capra, “360 degrees video and audio recording and broadcasting employing a parabolic mirror camera and a spherical 32-capsules microphone array,” *IBC 2011*, pp. 8–11, 2011.
- [9] Adam O'Donovan, Ramani Duraiswami, and Jan Neumann, “Microphone arrays as generalized cameras for integrated audio visual processing,” in *Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference on*. IEEE, 2007, pp. 1–8.
- [10] Boaz Rafaely, *Fundamentals of spherical array processing*, vol. 8, Springer, 2015.
- [11] Miljko M Erić, “Some research challenges of acoustic camera,” in *Telecommunications Forum (TELFOR), 2011 19th*. IEEE, 2011, pp. 1036–1039.
- [12] J Daniel, *Représentation de champs acoustiques, application à la reproduction et à la transmission de scènes sonores complexes dans un contexte multimédia*, Ph.D. thesis, Ph. D. thesis, University of Paris 6, Paris, France, 2000.
- [13] Franz Zotter, Hannes Pomberger, and Markus Noisternig, “Energy-preserving ambisonic decoding,” *Acta Acustica united with Acustica*, vol. 98, no. 1, pp. 37–47, 2012.

- [14] Michael Brandstein and Darren Ward, *Microphone arrays: signal processing techniques and applications*, Springer Science & Business Media, 2013.
- [15] Michael D Zoltowski, “On the performance analysis of the mvdr beamformer in the presence of correlated interference,” *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 36, no. 6, pp. 945–947, 1988.
- [16] Nicolas Epain and Craig T Jin, “Super-resolution sound field imaging with sub-space pre-processing,” in *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*. IEEE, 2013, pp. 350–354.
- [17] Tahereh Noohi, Nicolas Epain, and Craig T Jin, “Direction of arrival estimation for spherical microphone arrays by combination of independent component analysis and sparse recovery,” in *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*. IEEE, 2013, pp. 346–349.
- [18] Ralph Schmidt, “Multiple emitter location and signal parameter estimation,” *IEEE transactions on antennas and propagation*, vol. 34, no. 3, pp. 276–280, 1986.
- [19] O Nadiri and B Rafaely, “Localization of multiple speakers under high reverberation using a spherical microphone array and the direct-path dominance test,” *IEEE/ACM Transactions on Audio, Speech and Language Processing (TASLP)*, vol. 22, no. 10, pp. 1494–1505, 2014.
- [20] Symeon Delikaris-Manias and Ville Pulkki, “Cross pattern coherence algorithm for spatial filtering applications utilizing microphone arrays,” *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 21, no. 11, pp. 2356–2367, 2013.
- [21] Symeon Delikaris-Manias, Despoina Pavlidi, Ville Pulkki, and Athanasios Mouchtaris, “3d localization of multiple audio sources utilizing 2d doa histograms,” in *Signal Processing Conference (EUSIPCO), 2016 24th European*. IEEE, 2016, pp. 1473–1477.
- [22] Ville Pulkki, “Virtual sound source positioning using vector base amplitude panning,” *Journal of the Audio Engineering Society*, vol. 45, no. 6, pp. 456–466, 1997.
- [23] Sébastien Moreau, Jérôme Daniel, and Stéphanie Bertet, “3d sound field recording with higher order ambisonics—objective measurements and validation of a 4th order spherical microphone,” in *120th Convention of the AES*, 2006, pp. 20–23.
- [24] Thushara D Abhayapala, “Generalized framework for spherical microphone arrays: Spatial and frequency decomposition,” in *Acoustics, Speech and Signal Processing, 2008. ICASSP 2008. IEEE International Conference on*. IEEE, 2008, pp. 5268–5271.
- [25] Heinz Teutsch, *Modal array signal processing: principles and applications of acoustic wavefield decomposition*, vol. 348, Springer, 2007.
- [26] David L Alon, Jonathan Sheaffer, and Boaz Rafaely, “Robust plane-wave decomposition of spherical microphone array recordings for binaural sound reproduction,” *The Journal of the Acoustical Society of America*, vol. 138, no. 3, pp. 1925–1926, 2015.
- [27] Stefan Lösler and Franz Zotter, “Comprehensive radial filter design for practical higher-order ambisonic recording,” *Fortschritte der Akustik, DAGA*, pp. 452–455, 2015.
- [28] Franz Zotter Christian Schörkhuber and Robert Höldrich, “Signal-dependent encoding for first-order ambisonic microphones,” *Fortschritte der Akustik, DAGA*, 2017.
- [29] J. Meyer and G. Elko, “A highly scalable spherical microphone array based on an orthonormal decomposition of the soundfield,” in *2002 IEEE International Conference on Acoustics, Speech, and Signal Processing*, May 2002, vol. 2, pp. II–1781–II–1784.
- [30] Boaz Rafaely and Maor Kleider, “Spherical microphone array beam steering using wigner-d weighting,” *IEEE Signal Processing Letters*, vol. 15, pp. 417–420, 2008.
- [31] J. Atkins, “Robust beamforming and steering of arbitrary beam patterns using spherical arrays,” in *2011 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*, Oct 2011, pp. 237–240.
- [32] Symeon Delikaris-Manias, Juha Vilkamo, and Ville Pulkki, “Signal-dependent spatial filtering based on weighted-orthogonal beamformers in the spherical harmonic domain,” *IEEE/ACM Transactions on Audio, Speech and Language Processing (TASLP)*, vol. 24, no. 9, pp. 1507–1519, 2016.
- [33] Benjamin Bernschütz, Christoph Pörschmann, Sascha Spors, Stefan Weinzierl, and Begrenzung der Verstärkung, “Soft-limiting der modalen Amplitudenverstärkung bei sphärischen mikrofonarrays im plane wave decomposition verfahren,” *Proceedings of the 37. Deutsche Jahrestagung für Akustik (DAGA 2011)*, pp. 661–662, 2011.
- [34] Boaz Rafaely, Barak Weiss, and Eitan Bachmat, “Spatial aliasing in spherical microphone arrays,” *IEEE Transactions on Signal Processing*, vol. 55, no. 3, pp. 1003–1010, 2007.
- [35] Florian Hollerweger, *Periphonic sound spatialization in multi-user virtual environments*, Citeseer, 2006.

ESTIMATING PICKUP AND PLUCKING POSITIONS OF GUITAR TONES AND CHORDS WITH AUDIO EFFECTS

Zulfadhl Mohamad*, Simon Dixon,

Centre for Digital Music,
Queen Mary University of London
London, United Kingdom
z.b.mohamad@qmul.ac.uk
s.e.dixon@qmul.ac.uk

Christopher Harte,

Melodient Limited
United Kingdom
chris@melodient.com

ABSTRACT

In this paper, we introduce an approach to estimate the pickup position and plucking point on an electric guitar for both single notes and chords recorded through an effects chain. We evaluate the accuracy of the method on direct input signals along with 7 different combinations of guitar amplifier, effects, loudspeaker cabinet and microphone. The autocorrelation of the spectral peaks of the electric guitar signal is calculated and the two minima that correspond to the locations of the pickup and plucking event are detected. In order to model the frequency response of the effects chain, we flatten the spectrum using polynomial regression. The errors decrease after applying the spectral flattening method. The median absolute error for each preset ranges from 2.10 mm to 7.26 mm for pickup position and 2.91 mm to 21.72 mm for plucking position estimates. For strummed chords, faster strums are more difficult to estimate but still yield accurate results, where the median absolute errors for pickup position estimates are less than 10 mm.

1. INTRODUCTION

The popularity of the electric guitar began to rise in the mid 1950s and it soon became, and has since remained, one of the most important instruments in Western popular music. Well known guitar players can often be recognised by the distinctive electric guitar tone they create. Along with the player's finger technique, the unique tone they achieve is produced by their choice of electric guitar model, guitar effects and amplifiers. Some popular musicians prefer using one or two choices of guitar models for recordings and live performances. Some guitar enthusiasts are keen to know how their favourite guitar players produce their unique tones. Indeed, some will go as far as to purchase the same electric guitar model, effects and amplifiers to replicate the sounds of their heroes.

The tones produced by popular electric guitar models such as Fender and Gibson are clearly distinguishable from each other, with different pickup location, width, and sensitivity, along with circuit response of the model leading to different timbres. The pickup location of an electric guitar contributes to the sound significantly, where it produces a comb-filtering effect on the spectrum. The tonal differences can be heard just by switching the pickup configuration of the guitar. Thus, if the type of electric guitar is known, the estimated pickup position can help distinguish which pickup configuration is selected. Equally, where the guitar model on a recording is not known, pickup position estimates could be

useful in deducing which type of electric guitar could have produced a particular tone.

In 1990, research on synthesising electric guitar sound was proposed by Sullivan whereby the Karplus-Strong algorithm is extended to include a pickup model with distortion and feedback effects [1]. Commercial firms such as Roland and Line 6 produce guitar synthesisers that use hexaphonic pickups allowing them to process sound from each string separately to mimic the sound of many other popular guitars [2, 3]. They model the pickups of popular electric guitars including the effects of pickup position, height and magnetic apertures. Lindroos et al. [4] introduced a parametric electric guitar synthesis where conditions that affect the sound can be changed such as the force of the pluck, plucking point and pickup position. Further details of modelling the magnetic pickup are studied by Paiva et al. [5], which include the effect of pickup width, nonlinearity and circuit response.

Since electric guitar synthesis requires the pickup and plucking positions to be known, a method to estimate these parameters could be useful when trying to replicate the sound of popular guitarists from a recording. Several papers propose techniques to estimate the plucking point on an acoustic guitar, using either a frequency domain approach [6, 7] or a time domain approach [8]. A technique to estimate the plucking point and pickup position of an electric guitar is proposed by Mohamad et al. [9] where direct input electric guitar signals are used in the experiments.

Our previous work has dealt with recordings of isolated guitar tones. The first major contribution of the current paper is to extend the previous work to estimate the pickup and plucking locations of electric guitar signals that are processed through a combination of emulated electric guitar effect, amplifier, loudspeaker and microphone. This can bring us closer to estimating pickup and plucking locations of real-world electric guitar recordings. We also introduce a technique to flatten the spectrum before calculating the autocorrelation of the spectral peaks and finding the two minima of the autocorrelation. The second major contribution is to investigate the performance of our method on strummed chords and propose modifications to mitigate the effects of overlapping tones. We perform experiments with chords strummed at different speeds and with different bass notes to determine optimal parameters for our method.

In Sec. 2, we explain the comb-filtering effects produced in electric guitar tones. Sec. 3 describes the method for estimating the pickup and plucking position of an electric guitar. Sec. 4 explains the datasets that are used in this paper. There are two datasets where one is for testing the effects of different combinations of electric guitar effects, amplifier, loudspeaker and microphone and the other is for testing the effects of various chords. We eval-

* Zulfadhl Mohamad is supported by the Malaysian government agency, Majlis Amanah Rakyat (MARA)

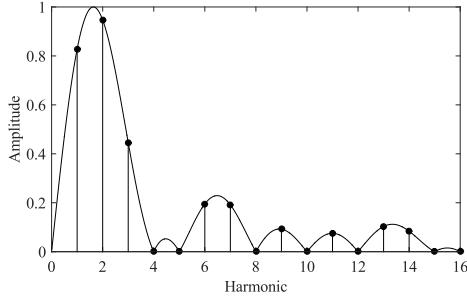


Figure 1: The spectral envelope of the electric guitar model plucked at one quarter of the string length with pickup situated at one-fifth of the string length from the bridge.

ate our method for various chains of effects in Sec. 5 and various chords in Sec. 6. The conclusion is presented in Sec. 7.

2. ELECTRIC GUITAR MODEL

When a string is plucked, two waves travel in opposite directions propagating away from the plucking point. The waves are then reflected back from the nut and bridge of the electric guitar producing a standing wave in the string. The vibrations of the strings are sensed by a pickup at a certain distance along the string thus certain harmonics (those with nodes at the pickup location) cannot be sensed. Similarly, harmonics with nodes at the plucking position are not excited. This means that depending on the locations of the pluck and pickup, certain harmonics are suppressed, resulting in two simultaneous comb-filtering effects. The spectral envelope of an electric guitar string model, \hat{X}_k plucked at a point ρ , with a vertical displacement a , sensed at a point d is calculated as:

$$\hat{X}_k = A_x \frac{S_\rho S_d}{k} \quad (1)$$

where $A_x = \frac{-2ac}{\pi L R_\rho (1 - R_\rho)}$, c is the velocity of transverse waves in the string, k is the harmonic number, $S_\rho = \sin(k\pi R_\rho)$, $S_d = \sin(k\pi R_d)$, R_d is the ratio between distance d and string length L and R_ρ is the ratio between distance ρ and string length L . Note that the comb filters have a -6 dB/octave slope.

An example using Eq. (1) is shown in Fig. 1 where the electric guitar is plucked at a quarter of the string length and is sensed by a pickup situated at one-fifth of the string length; every 4th and 5th harmonic is suppressed.

3. METHOD

The overview of the method for estimating the pickup and plucking positions of an electric guitar is shown in Fig. 2.

3.1. Onset and Pitch Detection

First, the spectral flux of the electric guitar signal is used to estimate the onset time [10]. The power spectrum for a frame is compared against the previous frame, where positive changes in the magnitude in each frequency bin are summed. Peaks in the spectral flux suggest possible onset times. We set the window size to be 40 ms with 10 ms overlapping windows to find the onset times.

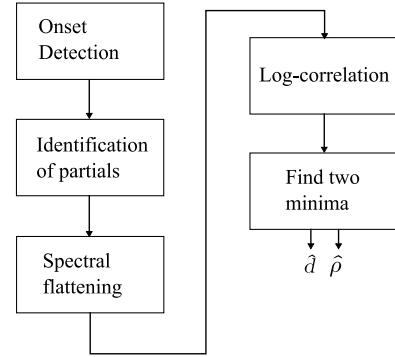


Figure 2: Block diagram for estimating pickup position and plucking point of an electric guitar

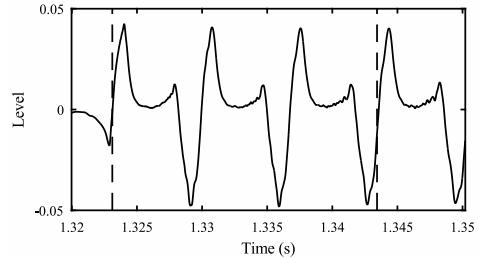


Figure 3: An excerpt of an electric guitar tone where the string is plucked on the open 4th string at 110 mm from the bridge with a pickup located at 159 mm from the bridge. The dashed vertical lines mark the detected beginnings of the 1st and 4th periods.

Due to the window overlap, the initial estimated onset time typically comes before the plucking noise, thus it is necessary to refine the estimate to be closer to the plucking event. Starting from the initial onset time estimate, the peaks of the signal are detected and peaks less than 30% of the maximum peak are discarded to avoid unwanted small peaks at the beginning due to plucking noise. Starting from the first peak and working backwards, the first zero-crossing is taken to be the start time of the tone. Fig. 3 shows an excerpt of an electric guitar tone plucked on the open 4th (D) string at 110 mm from the bridge with a pickup located at 159 mm from the bridge, which starts from the initial onset time (around 1.32 s) and the first vertical dashed line represents the estimated start time of the tone.

After estimating the onset time, the fundamental frequency f_0 is estimated using YIN [11], where we set the window size to be 46 ms.

3.2. Identification of Partials

From the onset time of the tone, the STFT of the signal is performed using a hamming window and zero padding factor of 4 to analyse the first few periods of the waveform. In this example, the first 3 periods are taken for STFT analysis as shown in Fig. 3. The advantage of taking such a small window is that time modulation effects such as reverb and delay will not be prevalent during the first few periods of the signal.

Each spectral peak is found in windows of ± 30 cents around estimated partial frequencies calculated based on typical inhar-

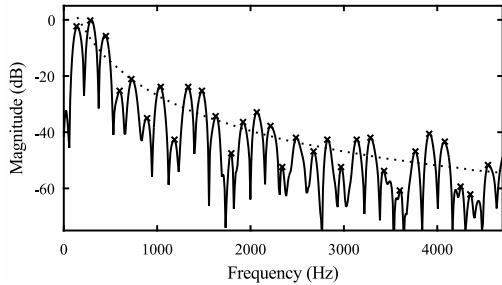


Figure 4: Spectrum of the electric guitar tone in Fig. 3 with the detected spectral peaks (crosses) and slope of the spectrum (dotted line).

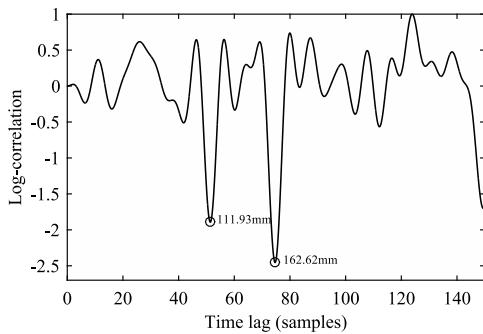


Figure 5: The log-correlation of an electric guitar plucked on the open 4th string at 100 mm from the bridge with a pickup located at 159 mm from the bridge.

monicity coefficients for each string [12]. Quadratic interpolation is used to refine the magnitudes and frequencies of the spectral peaks [13]. From each pair of estimated partial frequencies, the inharmonicity B can be determined. The median of all B values is taken as our estimated inharmonicity coefficient \hat{B} .

Some spectral peaks may be falsely detected, for instance, some estimated partial frequencies may be located on top of or close to each other. This is mainly because the initial inharmonicity coefficient that we set might be more or less than the actual inharmonicity coefficient of the string. Therefore, we need to set a threshold to identify any falsely estimated partial frequencies. We calculate the target frequencies using the inharmonicity coefficient \hat{B} estimated earlier:

$$f_k = k f_0 \sqrt{1 + \hat{B} k^2} \quad (2)$$

Then, any estimated partial frequencies deviating by more than ± 30 cents from their target frequencies are identified as false. The corrected spectral peak is found in the revised window, and refined using quadratic interpolation. If no peak is found in the window, the corrected partial frequency is set equal to its target frequency.

Fig. 4 shows the spectrum of the electric guitar signal in Fig. 3 with the detected spectral peaks.

3.3. Spectral flattening

In this paper, we introduce an approach to flatten the spectrum of the observed data X_k . We do this because the ideal model of Sec. 2

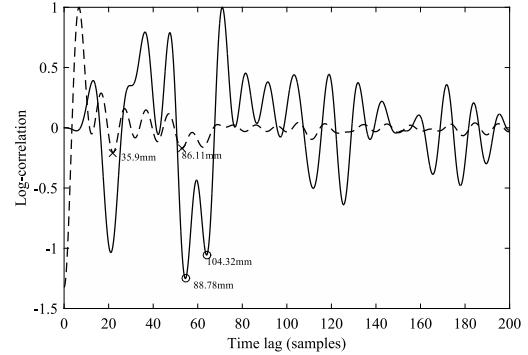


Figure 6: The log-correlation of an electric guitar tone with spectral flattening (solid line) and without spectral flattening (dashed line). The electric guitar is plucked on the open 5th string at 90 mm from the bridge with a pickup located at 102 mm from the bridge.

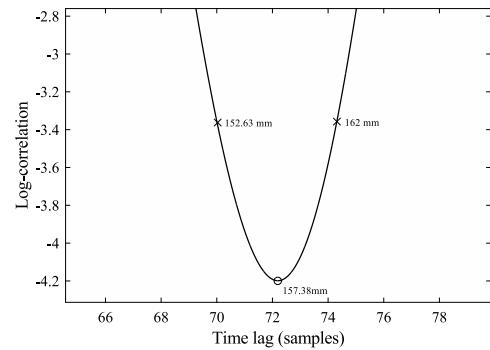


Figure 7: The log-correlation of an electric guitar tone where the string is plucked on the open 4th string at 150 mm from the bridge with a pickup located at 159 mm from the bridge, resulting in the two troughs merging into a single trough.

ignores the low-pass filtering effect of the finite widths of the plectrum and pickup. Flattening the spectrum reverses this effect by increasing the level of higher harmonics.

The best fitting curve for the log magnitude X_k in the log-frequency domain is calculated using polynomial regression. We compare linear and third-order polynomial regression to approximate the frequency response produced by the guitar signal chain. Matlab's `polyfit` function is used to retrieve the coefficients of the polynomial $p(x)$. The polynomial regression curve for the spectrum in Fig. 4 is shown as a dotted line.

Then, the spectrum X_k can be flattened as follows:

$$\bar{X}_k = \frac{X_k}{e^{p(\log(k))}} \quad (3)$$

where \bar{X}_k is the flattened spectrum of the observed data X_k .

3.4. Log-correlation

Since the plucking point ρ , and pickup position d , produce two comb-filtering effects as shown in Eq. (1), the delay of the comb filters can be estimated using the autocorrelation of the log magnitude of the spectral peaks. The log-correlation as described by

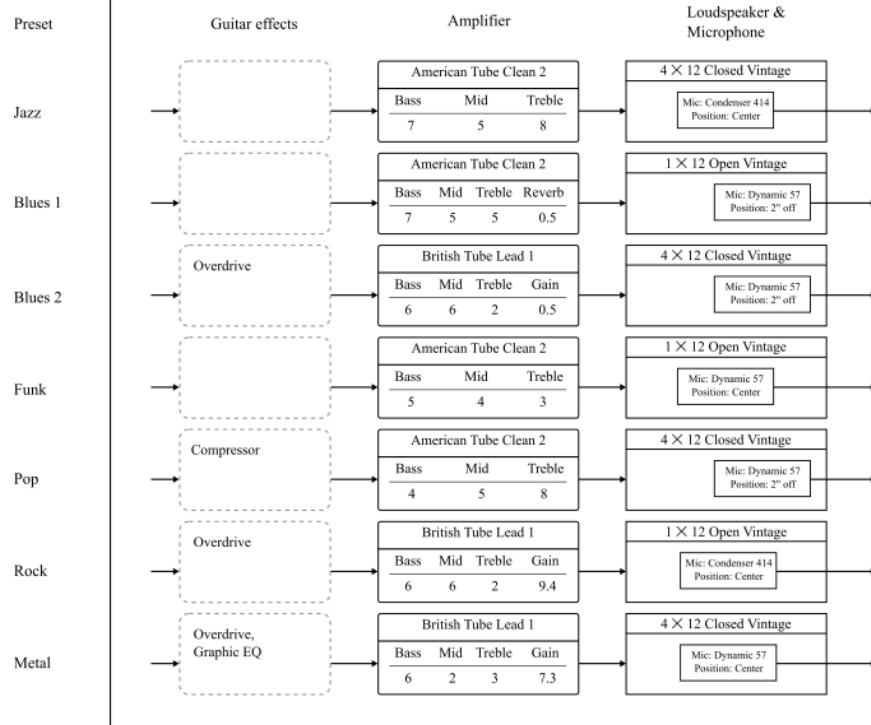


Figure 8: Seven combinations of emulated electric guitar effects, amplifier, loudspeaker and microphone.

Traube and Depalle [7] is calculated as:

$$\Gamma(\tau) = \sum_{k=1}^K \log(\bar{X}_k^2) \cos\left(\frac{2\pi k \tau}{T}\right) \quad (4)$$

where T is the period of the signal. For an electric guitar, it is expected that we would see two minima in the log-correlation where the time lag of one trough τ_d indicates the position of the pickup and the time lag of the other τ_p indicates the position of the plucking event. Therefore, the estimated pickup position, \hat{d} and plucking point \hat{p} can be calculated by finding the time lags, $\hat{\tau}_d$ and $\hat{\tau}_p$ in the log-correlation using trough detection, where $\hat{d} = \frac{\hat{\tau}_d L}{T}$ and $\hat{p} = \frac{\hat{\tau}_p L}{T}$.

3.5. Find two minima of the log-correlation

The method for finding the two minima of the log-correlation can be described by an example where the electric guitar tone in Fig. 3 is taken for analysis. The log-correlation is calculated until T samples ($f_0 = 147.59$ Hz) where the time lag resolution is 0.01 samples. The log-correlation of the electric guitar tone is shown in Fig. 5, where the lowest two troughs that correspond to the pickup and plucking locations are visible where one time lag is 74.66 samples (or 162.62 mm) and the other is 51.38 samples (or 111.93 mm). We are only interested in the lowest two troughs located in the first half of the log-correlation but it is not possible to determine which represents the pickup and which is the pluck point from this information alone. Instead, given that we already know what model guitar was used to produce the sounds, we take the trough that is closest to a known pickup location as our estimated

pickup position and the other as the estimated plucking point. In this example, the absolute errors for the pickup position and plucking point estimates are 3.62 mm and 1.93 mm respectively.

Without flattening the spectrum, the troughs are not as apparent and it could be difficult to detect the two minima. An example is given in Fig. 6, where the electric guitar is played on the open 5th string, plucked 90 mm from the bridge with the pickup situated at 102 mm from the bridge. It shows two log-correlations of the spectral peaks where one is with spectral flattening (solid line) and the other is without spectral flattening (dashed line). We can see that the troughs corresponding to the pickup and plucking positions are emphasised if the spectrum has been flattened. The plucking point estimate is also closer to the known plucking point.

There are cases where the plucking point is at or near the pickup position, causing the two troughs to merge together. We need to set a threshold to define whether that is the case. If the second lowest trough detected is above 40% of the lowest trough, then only the lowest trough is selected. By taking an example of an electric guitar plucked at 150 mm from the bridge with the pickup located at 159 mm from the bridge, Fig. 7 shows the only trough detected where the time lag is 72.2 samples (or 157.38 mm). Since the two troughs are merged together, it will be less accurate if we assume that the plucking point is at the pickup location. Thus, we take the time lags where both are at 80% of the minimum value. This also applies to plucks that are at the pickup position, where the width of the trough is thinner. Fig. 7 shows the estimated pickup position and plucking point which are 162 mm and 152.63 mm respectively.

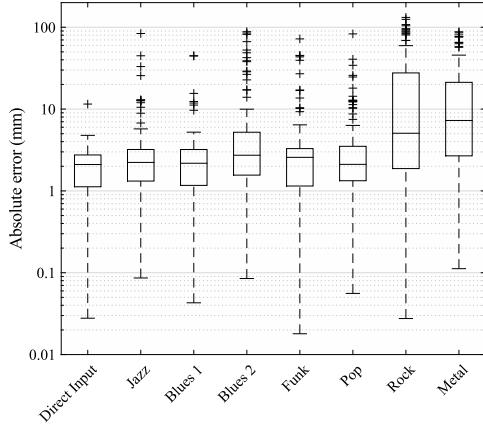


Figure 9: Box plot of absolute error in pickup position estimates for each preset. Note that the y-axis is in log scale.

4. AUDIO SAMPLES

In this paper, we use 144 direct input electric guitar signals recorded by Mohamad et al. [9] with sampling rate, f_s 44.1 kHz. The signals are recorded from a Squier Stratocaster and consist of moderately loud (*mezzo-forte*) plucks on 6 open strings at 8 plucking positions using 3 single pickup configurations. The strings are plucked at 30 mm to 170 mm from the bridge at 20 mm intervals. The pickup configurations consist of neck, middle and bridge pickup where their distances are measured around 160 mm, 100 mm and 38 – 48 mm (slanted pickup) from the bridge respectively. The length of the strings is around 650 mm. Note that the offsets of the measurements due to different adjustments of each saddle bridge are taken into account.

All the signals are processed through 7 combinations of digitally emulated electric guitar effects, amplifier, loudspeaker cabinet and microphone in Reaper [14], resulting in 1152 audio samples in total (8×144 samples, where direct input signals are also included). We select 7 presets that are freely available in AmpliTube Custom Shop by IK Multimedia, where each of them produces a tone for a certain style [15]. Common styles are selected which are Jazz, Blues, Funk, Pop, Rock and Metal. As shown in Fig. 8, there are 3 emulated guitar effects, 2 emulated amplifiers, 2 emulated loudspeaker cabinets and 2 emulated microphones. Note that each preset has different equipment settings and microphone placement.

Additionally, a second dataset is used to test the accuracy of the estimation on various downstroke chords. We recorded 81 direct input electric guitar signals which consist of 3 chords (E major, A major and G major) strummed at 3 positions (on top of each pickup) with 3 different speeds and 3 pickup configurations (neck, middle and bridge pickup). The same electric guitar was used for this dataset and these signals are also processed through the 7 combinations of effects discussed earlier.

5. RESULTS: ELECTRIC GUITAR EFFECTS, AMPLIFIER, LOUDSPEAKER AND MICROPHONE

In this section, we examine the effects of different combinations of emulated electric guitar effects, amplifier, loudspeaker and microphone on the estimates. We test on the 8 combinations mentioned

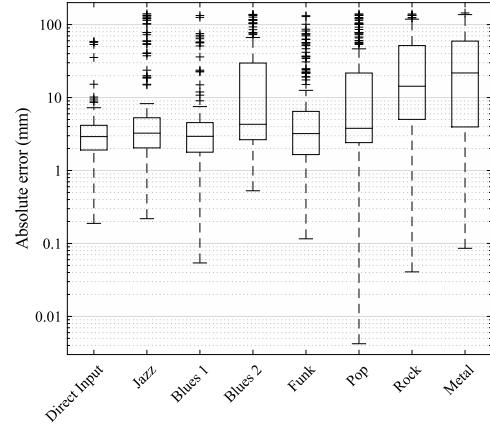


Figure 10: Box plot of absolute error in plucking point estimates for each preset. Note that the y-axis is in log scale.

Table 1: Median absolute error for pickup position and plucking point estimates for each preset, comparing between no spectral flattening (NSF), linear regression spectral flattening (LRSF) and polynomial regression spectral flattening (PRSF).

Preset	Median absolute error (mm)					
	Pickup position			Plucking point		
	NSF	LRSF	PRSF	NSF	LRSF	PRSF
Direct Input	2.56	2.03	2.10	4.14	2.87	2.91
Jazz	2.44	2.17	2.23	3.61	3.42	3.25
Blues 1	2.32	2.28	2.18	4.25	3.04	2.94
Blues 2	3.22	3.62	2.73	4.52	6.90	4.30
Funk	2.55	2.03	2.57	4.61	3.06	3.21
Pop	2.34	2.25	2.11	4.05	3.90	3.79
Rock	5.58	4.17	5.07	17.01	14.75	14.29
Metal	9.54	8.84	7.26	34.60	31.08	21.72
Average	3.82	3.42	3.28	9.60	8.63	7.05

in Sec. 4, where the emulated equipment used is shown in Fig. 8. The method described in Sec. 3 is used to find the estimates, where the spectrum is flattened using polynomial regression. The first 10 cycles of the tones are taken for the STFT analysis for all presets. For clean tones i.e. Direct Input, Jazz, Blues 1, Funk and Pop presets, the total number of harmonics K is set to 40. For overdriven and distorted tones i.e. Blues 2, Rock and Metal presets, the total number of harmonics K is set to 30.

Fig. 9 and 10 show the absolute errors for pickup position and plucking point estimates respectively. The line inside each box is the median and the outliers are represented by cross symbols (+). Overall, the median absolute errors for pickup position estimates are less than 8 mm ranging from 2.10 mm – 7.26 mm. The median absolute errors for plucking point estimates are less than 30 mm ranging from 2.91 mm – 21.72 mm.

In Fig. 9, the third quartiles for most presets are less than 10 mm which suggests that the pickup position estimates are robust to most presets. The errors for pickup position estimates increase as the signal gets heavily distorted. The errors for plucking point estimates also show a similar trend as shown in Fig. 10, where errors increase as the electric guitar signal is more distorted.

Finally, we compare the two spectral flattening methods described previously which are linear regression spectral flattening (LRSF) and polynomial regression spectral flattening (PRSF). Ta-

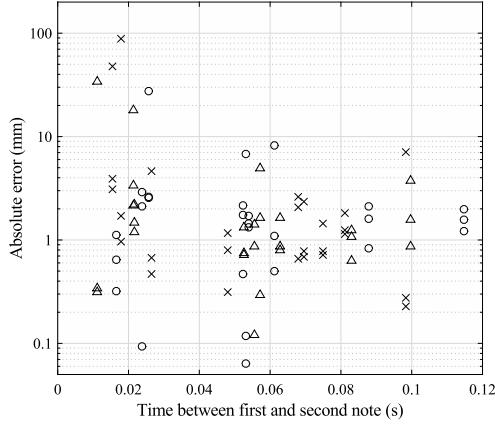


Figure 11: Absolute errors for pickup position estimates of chords. The crosses represent E major chords, circles represent A major chords and triangles represent G major chords. Note that the y-axis is in log scale.

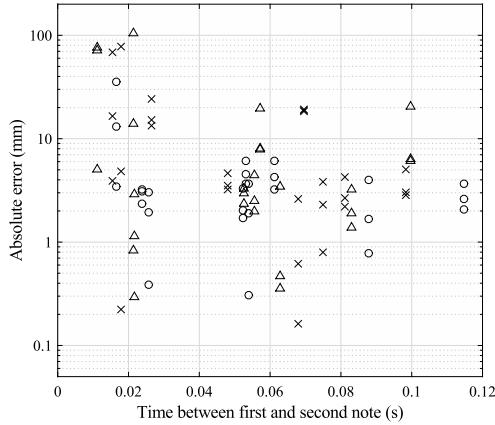


Figure 12: Absolute errors for plucking point estimates of chords. The crosses represent E major chords, circles represent A major chords and triangles represent G major chords. Note that the y-axis is in log scale.

ble 1 shows the median absolute errors for pickup and plucking position estimates. The average median absolute error across all presets for pickup and plucking position estimates decrease by 0.14 mm and 1.58 mm respectively using PRSF compared to LRSF. Overall, the median absolute errors decrease when the spectral flattening methods are applied. This suggests that we improved the method by introducing a technique to flatten the spectrum.

6. RESULTS: CHORDS

In this section, we test the accuracy of our method on strummed chords. The chords played are E major, A major and G major, where the first string to be struck is the 6th string for all chords (downstrokes). Each chord is strummed at 3 different speeds and 3 positions. Since the pickup and pluck positions are unlikely to change during the strum, and our method only requires the first few pitch periods of the electric guitar tone, it should be possible to estimate the pickup and plucking positions where the second

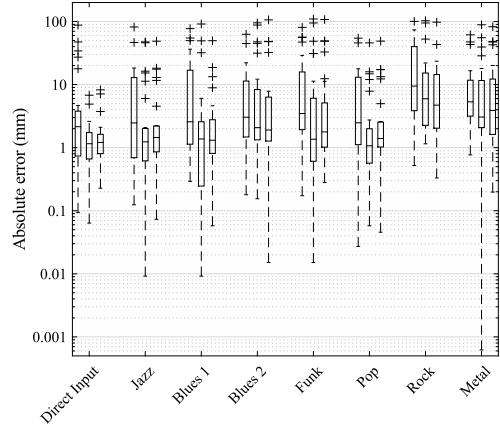


Figure 13: Box plot of absolute error in pickup position estimates for chords for each preset. Each preset has three box plots, where the boxes (from left to right) are for fast, slow and very slow strums respectively. Note that the y-axis is in log scale.

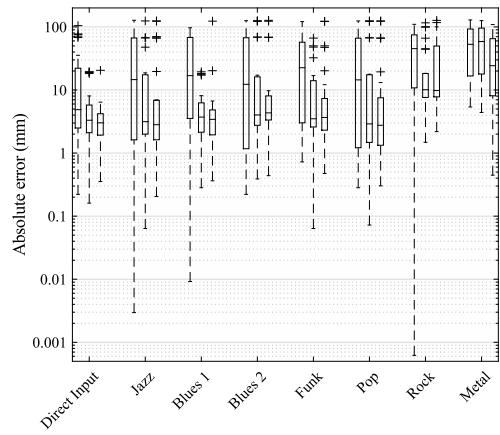


Figure 14: Box plot of absolute error in plucking point estimates for chords for each preset. Each preset has three box plots, where the boxes (from left to right) are for fast, slow and very slow strums respectively. Note that the y-axis is in log scale.

note is plucked after a few cycles of the first note. Furthermore, we manually measure the time between the first and second note, t_c . For our method to be unaffected by the strum, the shortest time allowed between the first and second note would be 36.4ms (3 cycles of note 82.41 Hz) for the worst case scenario of the first pitch being E2, the lowest pitch on the guitar. However, natural strumming of a guitar leads to values of t_c of 80ms for slow strums and 20ms for fast strums.

The method described in Sec. 3 is used to estimate the pickup and plucking positions of each chord, where the spectrum is flattened using PRSF. The fundamental frequencies of the first note struck on each chord are known in advance, which are 82.41 Hz (E major and A major) and 98.00 Hz (G major). To present the worst case scenario, the A major chord is played in second inversion (i.e. with a low E in bass). Multiple pitch estimation could be used to estimate the fundamental frequency of the first note [16]. The total number of harmonics K is set to 40 for Direct Input, Jazz,

Blues 1, Funk and Pop presets and 30 for Blues 2, Rock and Metal presets. The first 2 cycles are taken for the STFT analysis when t_c is shorter than 40ms and the first 3 cycles are taken when t_c is longer than 40ms. A shorter window is needed for faster strums so that less of the second note is included in the STFT analysis.

Fig. 11 and 12 show the absolute errors for pickup and plucking position estimates respectively for direct input signals. The absolute errors increase for faster strums, nevertheless, most of the errors are less than 20 mm for both pickup and plucking position estimates even though the second note starts to bleed into the window. In Fig. 11 and 12, the shortest t_c for E major, G major and A major chords are 15ms, 11ms and 17ms respectively. This means that the second note for each chord overlaps 38%, 45% and 10% of the analysed window respectively. Chords with later second notes (i.e. a smaller overlap) yield more accurate results, for example, in Fig. 11 shows that the pickup estimates are all less than 2 mm error for A major chord at $t_c = 17$ ms. Furthermore, the accuracy of the pickup and plucking position estimates increases when t_c is more than 20ms.

Fig. 13 and 14 show the box plots of absolute errors in pickup and plucking position estimates respectively for each preset. Similar to single note guitar tones, the errors increase as the signal gets more harmonically distorted. Errors also increase for fast strums. Nevertheless, the median absolute errors for pickup position estimates are less than 10 mm.

7. CONCLUSIONS

We have presented a technique to estimate the pickup position and plucking point on an electric guitar recorded through a typical signal processing chain i.e. electric guitar effects, amplifier, loudspeaker and microphone with different settings for each equipment. For each preset, the median absolute errors for pickup position and plucking point estimates are 2.10 mm – 7.26 mm and 2.91 mm – 21.72 mm, where errors increase when signals are more distorted. The other aspects of the signal chain appear to have little effect on our results. Pickup position estimates can be used to distinguish which pickup is selected for a known electric guitar. For an unknown electric guitar, the estimates can be used to distinguish between typical electric guitar models.

The method can reliably estimate the pickup position of most clean, compressed and slightly overdriven tones i.e. with the Jazz, Blues 1, Blues 2, Pop and Funk presets, where 89% – 99% of the errors are less than 10 mm. For Rock and Metal presets, 57% and 63% of the errors are less than 10 mm respectively. Nevertheless, the median absolute errors for both presets are less than 8 mm. We also introduced a flattening method using linear and polynomial regression, where the errors decrease after applying the spectral flattening method. The median absolute errors for pickup and plucking position estimates decrease by 0.14 mm and 1.58 mm respectively across all presets compared to linear regression flattening method.

Furthermore, we evaluate our method for various downstroke chords. Pickup and pluck positions for most chords are detected correctly, with errors similar to those observed for single notes. A small number of outliers are observed which are mostly caused by overlapping tones disturbing our analysis method. The pickup position estimates are quite robust to downstroke chords where the median absolute errors for each preset are less than 10 mm. The errors increase for faster strums (t_c less than 30ms). This may suggest that upstroke chords would pose less of a problem, where

the first string struck has a higher pitch (or shorter period).

Further investigation could look into distinguishing between pickup position and plucking point estimates. Plucking positions vary constantly while the pickup position almost always remains fixed in one place; distinguishing the estimates from each other might therefore be achieved by determining which estimates deviate more frequently over a sequence of notes. An electric guitar commonly has an option to mix two pickups together, so in our further work, we are looking into estimating the pickup positions and plucking point of a mixed pickup signal.

8. REFERENCES

- [1] C. R. Sullivan, “Extending the Karplus-Strong algorithm to synthesize electric guitar timbres with distortion and feedback,” *Computer Music Journal*, vol. 14, no. 3, pp. 26–37, 1990.
- [2] A. Hoshiai, “Musical tone signal forming device for a stringed musical instrument,” 22 November 1994, US Patent 5,367,120.
- [3] P. J. Celi, M. A. Doidic, D. W. Fruehling, and M. Ryle, “Stringed instrument with embedded dsp modeling,” 7 September 2004, US Patent 6,787,690.
- [4] N. Lindroos, H. Penttilä, and V. Välimäki, “Parametric electric guitar synthesis,” *Computer Music Journal*, vol. 35, no. 3, pp. 18–27, 2011.
- [5] R. C. D. Paiva, J. Pakarinen, and V. Välimäki, “Acoustics and modeling of pickups,” *Journal of the Audio Engineering Society*, vol. 60, no. 10, pp. 768–782, 2012.
- [6] C. Traube and J. O. Smith, “Estimating the plucking point on a guitar string,” in *Proceedings of the Conference on Digital Audio Effects (DAFx)*, 2000, pp. 153–158.
- [7] C. Traube and P. Depalle, “Extraction of the excitation point location on a string using weighted least-square estimation of a comb filter delay,” in *Proceedings of the Conference on Digital Audio Effects (DAFx)*, 2003.
- [8] H. Penttilä and V. Välimäki, “A time-domain approach to estimating the plucking point of guitar tones obtained with an under-saddle pickup,” *Applied Acoustics*, vol. 65, no. 12, pp. 1207–1220, 2004.
- [9] Z. Mohamad, S. Dixon, and C. Harte, “Pickup position and plucking point estimation on an electric guitar,” in *IEEE International Conference on Acoustics, Speech, and Signal Processing, 2017 (ICASSP)*, 2017, pp. 651–655.
- [10] P. Masri, *Computer modelling of sound for transformation and synthesis of musical signals.*, Ph.D. thesis, University of Bristol, 1996.
- [11] A. De Cheveigné and H. 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.
- [12] S. Dixon, M. Mauch, and D. Tidhar, “Estimation of harpsichord inharmonicity and temperament from musical recordings,” *The Journal of the Acoustical Society of America*, vol. 131, no. 1, pp. 878–887, 2012.
- [13] J. O. Smith, “Spectral audio signal processing,” Available at: <https://ccrma.stanford.edu/jos/sasp/>, accessed April 8, 2017.

- [14] Cockos, Inc., “Reaper: Digital audio workstation,” Available at <http://www.reaper.fm/>, accessed April 8, 2017.
- [15] IK Multimedia, “Amplitube custom shop,” Available at <https://www.ikmultimedia.com/products/amplitubecs/>, accessed April 8, 2017.
- [16] A. P. Klapuri, “Multiple fundamental frequency estimation based on harmonicity and spectral smoothness,” *IEEE Transactions on Speech and Audio Processing*, vol. 11, no. 6, pp. 804–816, 2003.

UNSUPERVISED TAXONOMY OF SOUND EFFECTS

David Moffat

Centre for Digital Music,
Queen Mary University of London
London, UK
d.j.moffat@qmul.ac.uk

David Ronan

Centre for Intelligent Sensing,
Queen Mary University of London
London, UK
d.m.ronan@qmul.ac.uk

Joshua D. Reiss

Centre for Digital Music,
Queen Mary University of London
London, UK
joshua.reiss@qmul.ac.uk

ABSTRACT

Sound effect libraries are commonly used by sound designers in a range of industries. Taxonomies exist for the classification of sounds into groups based on subjective similarity, sound source or common environmental context. However, these taxonomies are not standardised, and no taxonomy based purely on the sonic properties of audio exists. We present a method using feature selection, unsupervised learning and hierarchical clustering to develop an unsupervised taxonomy of sound effects based entirely on the sonic properties of the audio within a sound effect library. The unsupervised taxonomy is then related back to the perceived meaning of the relevant audio features.

1. INTRODUCTION

Sound designers regularly use sound effect libraries to design audio scenes, layering different sounds in order to realise a design aesthetic. For example, numerous explosion audio samples are often combined to create an effect with the desired weight of impact. A large part of this work involves the use of foley, where an artist will perform sound with a range of props. A key aspect of foley is that the prop being used may not match the object in the visual scene, but is capable of mimicking its sonic properties. An example would be the use of a mechanical whisk, which becomes a convincing gun rattle sound effect when combined in a scene with explosions and shouting.

Sound designers are less interested in the physical properties or causes of a sound, and more interested in their sonic properties. Despite this, many sound effect libraries are organised into geographical or physical categories. In [1] a sound search tool based on sonic properties is proposed, considering loudness, pitch and timbral attributes. A similar tool for semantic browsing of a small library of urban environmental sounds has also been proposed [2]. No other known classification methods for sound effects based on their sonic attributes exist, instead most previous work focuses either on perceptual similarity or the context and source of the sound.

Given that the practical use for a sound sample is often abstracted from its original intention, source or semantic label, categorisation based on this information is not always desirable. Furthermore, no standard exists for the labelling of recorded sound, and the metadata within a sound effect library can be highly inconsistent. This makes the task of searching and identifying useful sounds extremely laborious, and sound designers will often resort to recording new sound effects for each new project.

The aim of this paper is to produce a hierarchical taxonomy of sound effects, based entirely on the sonic properties of the audio samples, through the use of unsupervised learning. Such an approach offers an alternative to standard categorisation, in the hope

that it will aid the search process by alleviating dependence on hand written labels and inconsistent grouping of sounds.

Different approaches to developing taxonomies of sound are discussed in Section 2. Section 3 presents the dataset, feature selection technique and unsupervised learning method undertaken to produce a hierarchy within a sound effect library. The taxonomy we produced is presented in Section 4. The evaluation of the presented taxonomy is undertaken in Section 4.4 and discussed in Section 5. Finally, the validity of the taxonomy and future work is discussed in Section 6.

2. BACKGROUND

There are a number of examples of work attempting to create a taxonomy of sound. In [3], the author classified sounds by acoustics, psychoacoustics, semantics, aesthetics and referential properties. In [4], the authors classified "noise-sound" into six groups: roars, hisses, whispers, impactful noises, voiced sounds and screams. This is further discussed in [5].

Production of a taxonomy of sounds heard in a cafe or restaurant were produced, basing the grouping on the sound source or context [6, 7].

In [8] the authors presented a classification scheme of sounds based on the state of the physical property of the material. The sound classifications were vibrating solids, liquids and aerodynamic sounds (gas). A series of sub-classifications based on hybrid sounds were also produced along with a set of properties that would impact the perception of the sound. This was developed further by attempting to understand how participants would arbitrarily categorise sounds [9]. In [10] the authors asked participants to identify how similar sounds are to each other along a series of different dimensions. They then performed hierarchical cluster analysis on the results, to produce a hierarchical linkage structure of the sounds. Furthermore, in [11] the authors performed a similar study where participants were asked how alike sets of sounds were. Audio features were then correlated to a likeness measure and a hierarchical cluster was produced on the set of selected features.

In [12] the authors asked participants to rate the similarity of audio samples, and performed hierarchical cluster analysis to demonstrate the related similarity structure of the sounds. Acoustic properties of sound walk recordings were taken and unsupervised clustering performed in [13]. These clusters were identified and related back to some semantic terms. Similarly, sound walks and interviews were used to identify appropriate words as sound descriptors [14]. Classification of sound effects by asking individuals to identify suitable adjectives to differentiate different sound samples was performed in [15] and similarly in [16] where the authors define classes of sound descriptor words that can be used to

Reference	Type of Sound	Classification properties	Quantitative Analysis	Qualitative Analysis	Word Classification	Audio Feature Analysis	Hierarchical Cluster
[3]	Environmental	Acoustics	N	N	N	Y	N
[3]	Environmental	Aesthetics	N	N	N	N	N
[3]	Environmental	Source/context	N	N	N	N	Y
[4, 5]	Environmental	Subjective	N	N	N	N	N
[6]	Cafe sounds	Source or context	N	N	N	N	Y
[7]	Restaurant	Subjective ‘liking’ score	N	Y	Y	N	N
[7]	Restaurant	Word occurrence	N	Y	Y	N	Y
[8]	Environmental	Physical properties	Y	N	N	N	N
[9]	Environmental	Subjective grouping	Y	N	N	N	Y
[10]	Environmental	Subjective ratings	Y	N	N	Y	Y
[11]	Environmental	Subjective ratings	Y	N	N	N	Y
[12]	Environmental	Subjective ratings	Y	N	Y	N	Y
[13]	Sound walks	Low level audio features	Y	Y	N	Y	N
[14]	Sound walks	Semantic words	Y	N	Y	N	N
[15]	Soundscape	Subjective free text word recurrence	N	Y	Y	N	N
[16]	‘Perceptual attribute’ words	Definition of word	N	Y	Y	N	N
[17]	Broadcast objects	Predefined word list	Y	Y	Y	N	Y
[18]	Urban sounds	Source	N	N	N	N	Y
[19]	Synthesised sounds	Control parameters	N	N	N	N	Y
[20]	Field recordings	Labels/audio features	Y	N	N	Y	N

Table 1: Summary of literature on sound classification

relate the similarity of words. In an extension to this, [17] asked participants to perform a sorting and labelling task on broadcast audio objects, again yielding a hierarchical cluster.

[18] produced a dataset of urban sounds, and a taxonomy for the dataset, where sounds are clustered based on the cause of the audio, rather than the relative similarity of the audio sample themselves. They then used this dataset for unsupervised learning classification [21, 22]. In the context of synthesised sounds, [19] grouped sounds by their control parameters.

There is no clear standard method for grouping sounds such as those found in a sound effect library. It becomes clear from the literature that there is limited work utilising audio features to produce a taxonomy of sound. It can be seen in Table 1 that a large range of relevant work structures sound grouping based on either subjective rating or word clustering. It is also apparent there is little work clustering the acoustic properties of individual samples. There is a discussion of sound classification based on the acoustic properties of samples [3], but only a high level discussion is presented and is not pursued further.

3. METHODOLOGY

We used unsupervised machine learning techniques to develop an inherent taxonomy of sound effects. This section will detail the various development stages of the taxonomy, as presented in Figure 1. The Adobe sound effects library was used. A set of audio features were extracted, feature selection that was performed using Random Forests and a Gaussian Mixture Model was used to predict the optimal number of clusters in the final taxonomy. From the reduced feature set, unsupervised hierarchical clustering was performed to produce the number of clusters as predicted using the Gaussian Mixture Model. Finally the hierarchical clustering results are interpreted. All software is available online¹.

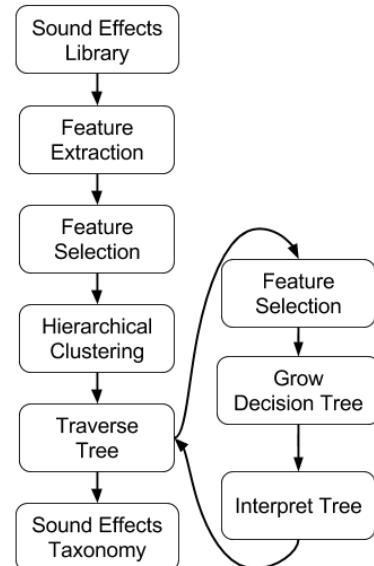


Figure 1: Flow Diagram of unsupervised sound effects taxonomy system.

3.1. Dataset

A dataset containing around 9,000 audio samples from the Adobe sound effect library² is used. This sound effects library contains a range of audio samples. All input audio signals were downmixed to mono, downsampled to 44.1 kHz if required, and had the initial and final silence removed. All audio samples were loudness normalised using ReplayGain [23]. Each sound effect was placed in a different folder, describing the context of the original sound effect. The original labels from the sound effect library can be found in Table 2, along with the number of samples found in each folder.

¹<https://goo.gl/9aWhTX>

²<https://goo.gl/TzQgsB>

Class Name	Quantity of Samples	Class Name	Quantity of Samples
Ambience	92	Animals	173
Cartoon	261	Crashes	266
DC	6	DTMF	26
Drones	75	Emergency Effects	158
Fire and Explosions	106	Foley	702
Foley Footsteps	56	Horror	221
Household	556	Human Elements	506
Impacts	575	Industry	378
Liquid-Water	254	Multichannel	98
Multimedia	1223	Noise	43
Production Elements	1308	Science Fiction	312
Sports	319	Technology	219
Tones	33	Transportation	460
Underwater	73	Weapons	424
Weather	54		

Table 2: *Original label classification of the Adobe Sound Effects Dataset. DC are single DC offset component signals. DTMF is Dual Tone Multi Frequency - a set of old telephone tones.*

3.2. Feature Extraction

The dataset described in Section 3.1 was used. We used Essentia Freesound Extractor to extract audio features [24], as Essentia allows for extraction of a large number of audio features, is easy to use in a number of different systems and produced the data in a highly usable format [25]. 180 different audio features were extracted, and all frame based features were calculated using a frame size of 2048 samples with a hop size of 1024, with the exception of pitch based features, which used a frame size of 4096 and the hop size 2048. The statistics of these audio features were then calculated, to summarise frame based features over the audio file. The statistics used are the mean, variance, skewness, kurtosis, median, mean of the derivative, the mean of the second derivative, the variance of the derivative, the variance of the second derivative, the maximum and minimum values. This produced a set of 1450 features, extracted from each file. Sets of features were removed if they provided no variance over the dataset, thus reducing the original feature set to 1364 features. All features were then normalised to the range [0, 1].

3.3. Feature Selection

We performed feature selection using a similar method to the one described in [26], where the authors used a Random Forest classifier to determine audio feature importance.

Random forests are an unsupervised classification technique where a series of decision trees are created, each with a random subset of features. The out-of-bag (OOB) error was then calculated, as a measure of the random forests classification accuracy. From this, it is possible to allocate each feature with a Feature Importance Index (FII), which ranks all audio features in terms of importance by evaluating the OOB error for each tree grown with a given feature, to the overall OOB error [27].

In [26] the authors eliminated the audio features from a Random Forest that had an FII less than the average FII and then grew a new Random Forest with the reduced audio feature set. This elimination process would repeat until the OOB error for a newly grown Random Forest started to increase.

Here, we opted to eliminate the 1% worst performing audio features on each step of growing a Random Forest, similar to but

more conservative than the approach in [28]. In order to select the correct set of audio features that fit our dataset we chose the feature set that provided us with lowest mean OOB error over all the feature selection iterations.

On each step of the audio feature selection process, we cluster the data using a Gaussian Mixture Model (GMM). GMM's are an unsupervised method for clustering data, on the assumption that data points can be modelled by a gaussian. In this method, we specify the number of clusters and get a measure of GMM quality using the Akaike Information Criterion (AIC). The AIC is a measure of the relative quality of a statistical model for a given dataset. We keep increasing the number of clusters used to create each GMM, while performing 10-fold cross-validation until the AIC measure stops increasing. This gives us the optimal number of clusters to fit the dataset.

3.4. Hierarchical Clustering

There are two main methods for hierarchical clustering. Agglomerative clustering is a bottom up approach, where the algorithm starts with singular clusters and recursively merges two or more of the most similar clusters. Divisive clustering is a top down approach, where the data is recursively separated out into a fixed number of smaller clusters.

Agglomerative clustering was used in this paper, as it is frequently applied to problems within this field [10, 11, 12, 13, 26, 17]. It also provides the benefit of providing cophenetic distances between different clusters, so that the relative distances between nodes of the hierarchy are clear. Agglomerative clustering was performed, on the feature reduced dataset, by assigning each individual sample in the dataset as a cluster. The distance was then calculated for every cluster pair based on Ward's method [29],

$$d(c_i, c_j) = \sqrt{\frac{2n_{c_i}n_{c_j}}{n_{c_i} + n_{c_j}}} euc(x_{c_i}, x_{c_j}) \quad (1)$$

where for clusters c_i and c_j , x_c is the centroid of a cluster c , n_c is the number of elements in a cluster c and $euc(x_{c_i}, x_{c_j})$ is the euclidean distance between the centroids of clusters c_i and c_j . This introduces a penalty for clusters that are too large, which reduces the chances of a single cluster containing the majority of the dataset and that an even distribution across a hierarchical structure is produced. The distance is calculated for all pairs of clusters, and the two clusters with the minimum distance d are merged into a single cluster. This is performed iteratively until we have a single cluster. This provides us with a full structure of our data, and we can visualise our data from the whole dataset, down to each individual component sample.

3.5. Node Semantic Context

In order to interpret the dendrogram produced from the previous step, it is important to have an understanding of what is causing the separation at each of the node points within the dendrogram. Visualising the results of machine learning algorithms is a challenging task. According to [30] decision trees are the only classification method which provides a semantic explanation of the classification. This is because a decision tree facilitates inspection of individual features and threshold values, allowing interpretation of the separation of different clusters. This is not possible with any other classification methods. As such, we undertook feature

selection and then grew a decision tree to provide some semantic meaning to the results.

Each node point can be addressed as a binary classification problem. For each node point, every cluster that falls underneath one side is put into a single cluster, and everything that falls on the other side of the node is placed in another separate cluster. Everything that does not fall underneath the node is ignored. This produces two clusters, which represent the binary selection problem at that node point. From this node point, a random forest is grown to perform the binary classification between the two sets and feature selection is then performed as described in Section 3.3. The main difference here is that only the five most relevant features, based on the FII are selected at each stage.

A decision tree is grown with this reduced set of 5 audio features, to allow manual visualisation of the separation of data at each node point within the hierarchical structure. The decision tree is constructed by minimising the Gini Diversity Index (GDI), at each node point within the decision tree, which is calculated as:

$$GDI = 1 - \sum_i p(i)^2 \quad (2)$$

where i is the class and $p(i)$ is the fraction of objects within class i following the branch. The decision trees are grown using the CART algorithm [31]. To allow for a more meaningful visualisation of the proposed taxonomy, the audio features and values were translated to a semantically meaningful context based on the audio interpretation of the audio feature. The definitions of the particular audio features were investigated and the authors identified the perceptual context of these features, providing relevant semantic terms in order to describe the classification of sounds at each node point.

4. RESULTS AND EVALUATION

4.1. Feature Extraction Results

Figure 2 plots the mean OOB error for each Random Forest that is grown for each iteration of the audio feature selection process. In total there were 325 iterations of the feature selection process, where the lowest OOB error occurred at iteration 203 with a value of 0.3242. This reduced the number of audio features from 1450 to 193.

Figure 3 depicts the mean OOB error for each Random Forest feature selection iteration against the optimal amount of clusters, where the optimal amount of clusters was calculated using the AIC for each GMM created. We found the optimal amount of clusters to be 9, as this coincides with the minimum mean OOB error in Figure 3.

4.2. Hierarchical Clustering Results

Having applied agglomerative hierarchical clustering to the reduced dataset, the resultant dendrogram can be seen in Figure 4. The dotted line represents the cut-off for depth analysis, chosen based on the result that the optimal choice of clusters is 9.

The results of the pruned decision trees are presented in Figure 5. Each node point identified the specific audio feature provides the best split in the data, to create the structure as presented in Figure 4.

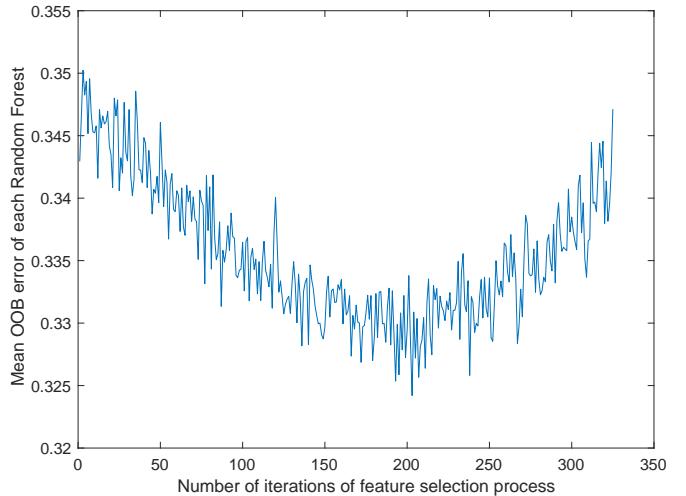


Figure 2: Mean OOB Error for each Random Forest grown plotted against number of feature selection iterations

4.3. Sound Effects Taxonomy Result

The audio features used for classification were related to their semantic meanings by manual inspection of the audio features used and the feature definitions. This is presented in Figure 6. As can be seen, the two key factors that make a difference to the clustering are periodicity and dynamic range.

Periodicity is calculated as the relative weight of the tallest peak in the beat histogram. Therefore strongly periodic signals have a much higher relative peak weight than random signals, which we expect to have near-flat beat histograms. Dynamic range is represented by the ratio of analysis frames under 60dB to the number over 60dB as all audio samples were loudness normalised and all leading and trailing silence was removed, as discussed in Section 3.2. Further down the taxonomy, it is clear that periodicity stands out as a key factor, in many different locations, along with the metric structure of periodicity, calculated as the weight of the second most prominent peak in the beat histogram. Structured music with beats and bars will have a high metrical structure, whereas single impulse beats or ticks will have a high beat histogram at one point but the rest of the histogram should look flat.

4.4. Evaluation

To evaluate the results of the produced sound effect taxonomy, as presented in Figure 6, the generated taxonomy was compared to the original sound effect library classification scheme, as presented in Section 3.1. The purpose of this is to produce a better understanding of the resulting classifications, and how it compares to more traditional sound effects library classifications. It is not expected that our clusters will appropriately represent an pre-existing data clusters, but that it may give us a better insight into the representation of the data.

Each of the 9 clusters identified in Figures 5 and 6 were evaluated by comparing the original classification labels found in Table 2 to the new classification structure. This is presented in Figure 7, where each cluster has a pie chart representing the distribution of original labels from the dataset. Only labels that make up more than 5% of the dataset were plotted.

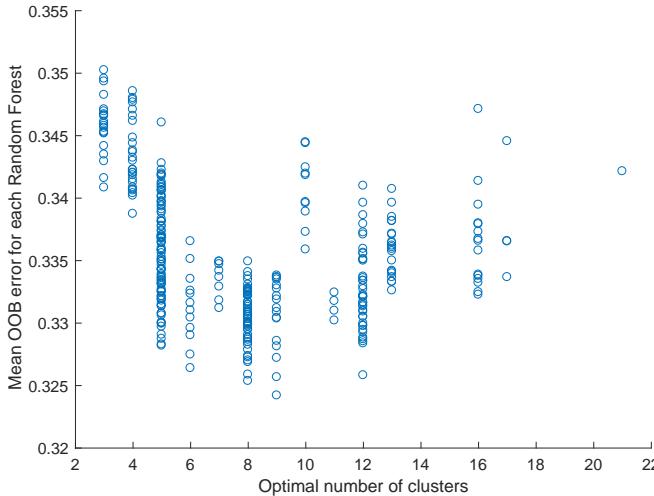


Figure 3: Mean OOB Error for each Random Forest grown plotted against optimal number of clusters for each feature selection iteration

In cluster 1, which has quick, periodic, high dynamic range sounds with a gradual decay, the majority of the results are from a range of production elements which are highly reverberant repetitive sounds, such as slide transition sounds. Many of these sounds are artificial or reverberant in nature, which follows the intuition of the cluster identification.

Cluster 2 contains a combination of foley sounds and water-splashing sounds. These sounds are somewhat periodic, such as lapping water, but do not have the same decay as in cluster 1.

Cluster 3 is very mixed. Impacts, household sounds and foley make up the largest parts of the dataset, but there is also contribution from crashes, production elements and weapon sounds. It is clear from the distribution of sounds that this cluster contains mostly impactful sounds. It is also evident that a range of impactful sounds from across the sound effect library have been grouped together.

In cluster 4, most of the samples are from the production elements label. These elements are moderately periodic at a high rate, such as clicking and whooshing elements, which are also similar to the next category of multimedia.

Cluster 5 contains a spread of sound labels, which includes transport and production elements as the two largest components. In particular, the transport sounds will be a periodic repetition of engine noises or vehicles passing, while remaining at a consistent volume.

There is a large range of labels within cluster 6. The three most prominent are human, multimedia and production elements, though cartoon and emergency sounds also contribute to this cluster. Human elements are primarily speech sounds, so the idea that periodic sounds that do not have a lot of high mid seems suitable, as the human voice fundamental frequency is usually between 90Hz and 300Hz.

Cluster 7 is entirely represented by the science fiction label. These fairly repetitive, constant volume sounds have an unnaturally large amount of high mid frequency.

Within cluster 8, the largest group of samples is multimedia, which consists of whooshes and swipe sounds. These are aperi-

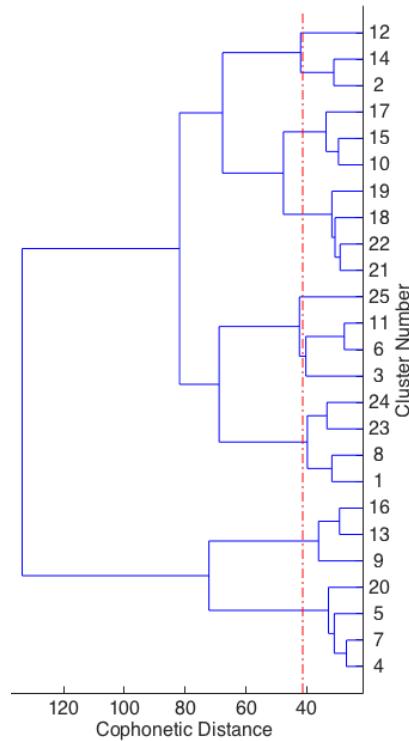


Figure 4: Dendrogram of arbitrary clusters - The dotted line represents the cut-off for the depth of analysis (9 clusters)

odic, and their artificial nature suggests a long reverb tail or echo. A low dynamic range suggests that the samples are consistent in loudness, with very few transients.

Finally, cluster 9 consists of a range of aperiodic impactful sounds from the impact, foley, multimedia and weapon categories.

5. DISCUSSION

The 9 inferred clusters were compared to the 29 original labels. It is clear that some clusters relate to intuition, and that this structure may aid a sound designer and present a suitable method for finding sounds, such as impactful sounds in cluster 9. Despite this, there are some clusters that do not make intuitive sense, or are difficult to fully interpret. We suspect that this is due to the depth of analysis on the dataset. Despite the GMM predicting 9 clusters within the data, we believe that a greater depth of analysis and clustering could aid in providing more meaningful, interpretable results, as many of the clusters are currently too large.

As can be seen from Figure 6 and discussed in Section 4, dynamic range and periodic structure are the key factors that separate this dataset. It is surprising that no timbral attributes and only one spectral attribute appears in the top features for classification within the dataset, and that seven of the eight features are time domain features.

Cluster 7 was described entirely as ‘Science Fiction’ in Section 4.4. This set of sound effects is entirely artificial, created using synthesisers and audio production. We believe that that the grouping using this audio feature is an artefact of the artificial nature of the samples and the fact they all come from a single source. This

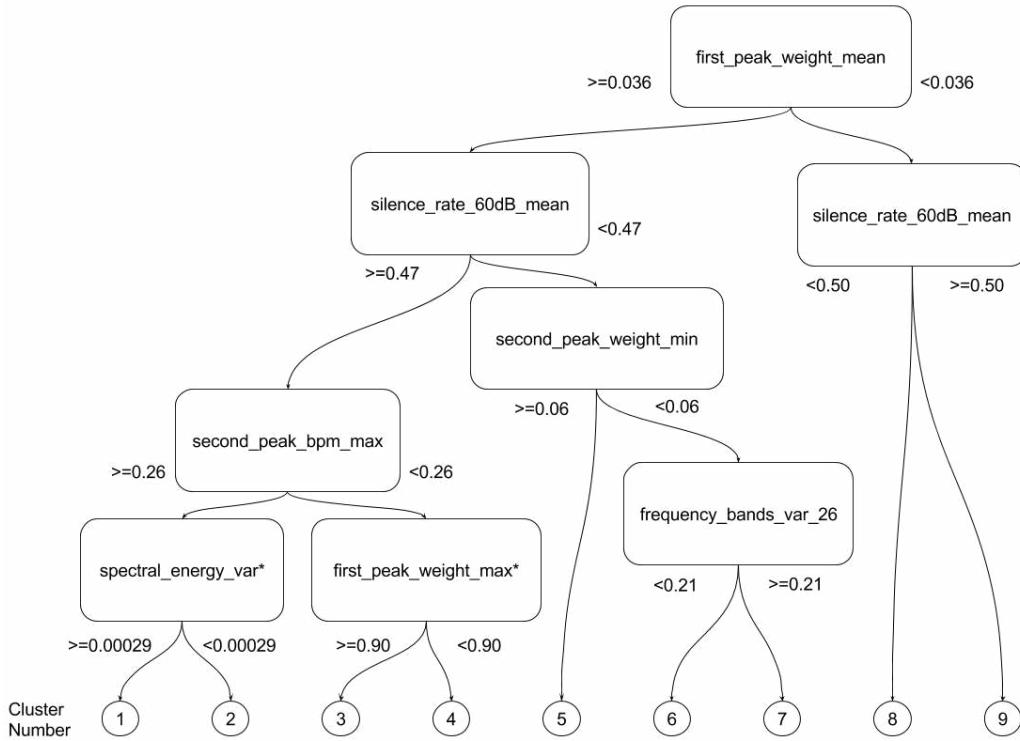


Figure 5: Machine learned structure of sound effects library, where clusters are hierarchical clusters. The single audio feature contributing to the separation is used as the node point, with normalised audio feature values down each branch to understand the impact the audio feature has on the sound classification. The * represents a feature separation where the classification accuracy is less than 80%, never less than 75%.

is also caused by the analysis and evaluation of a single produced sound effect library. This artefact may be avoided with a large range of sound effects from different sources.

Section 4.4 shows that the current classification system for sound effects may not be ideal, especially since expert sound designers often know what sonic attributes they wish to obtain. This is one of the reasons that audio search tools have become so prominent, yet many audio search tools only work using tag metadata and not the sonic attributes of the audio files.

Our produced taxonomy is very different from current work. As presented in Section 2, most literature bases a taxonomy on either audio source, environmental context or subjective ratings.

6. CONCLUSION

Given a commercial sound effect library, a taxonomy of sound effects has been learned using unsupervised learning techniques.

At the first level, a hierarchical structure of the data was extracted and presented in Figure 4. Following from this, a decision tree was created and pruned, to allow for visualisation of the data, as in Figure 5. Finally a semantically relevant context was applied to data, to produce a meaningful taxonomy of sound effects which is presented in Figure 6. A semantic relationship between different sonic clusters was identified.

The hierarchical clusters of the data provide deeper understanding of the separating attributes of sound effects, and gives us an insight into relevant audio features for sound effect classifica-

tion. We demonstrated the importance of the periodicity, dynamic range and spectral features for classification. It should be noted that although the entire classification was performed in an unsupervised manner, there is still a perceptual relevance to the results and there is a level of intuition provided by the decision tree and our semantic descriptors. Furthermore, the clustering and structure will be heavily reliant on the sound effects library used.

We also demonstrated that current sound effect classification and taxonomies may not be ideal for their purpose. They are both non-standard and often place sonically similar sounds in very different categories, potentially making it challenging for a sound designer to find an appropriate sound. We have proposed a direction for producing new sound effect taxonomies based purely on the sonic content of the samples, rather than source or context metadata.

In future work, validation of the results on larger sound effect datasets could aid in evaluation. By using the hierarchical clustering method, one can also produce a cophenetic distance between two samples. This would allow identification of how the distance can correlate with perceived similarity and may provide some interesting and insightful results. Further development of the evaluation and validation of the results, perhaps through perceptual listening tests, would be of beneficial to this field of research. It is also possible to look at the applications of hierarchical clustering towards other types of musical sounds, such as musical instrument classification. Hierarchical clustering is able to provide more information and context than many other unsupervised clus-

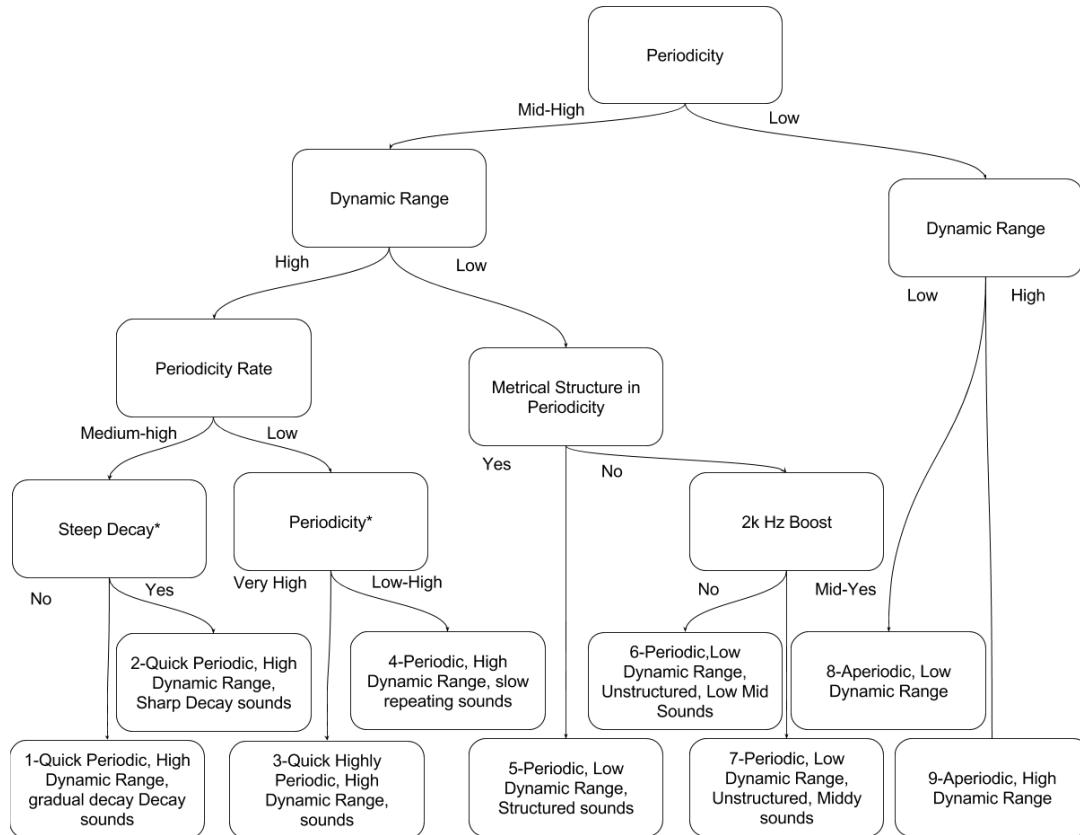


Figure 6: Machine learned taxonomy, where each node separation point is determined by hierarchical clustering and text within each node is an semantic interpretation of the most contributing audio feature for classification. Each final cluster is given a cluster number and a brief semantic description. The * represents a feature separation where the classification accuracy is less than 80%, never less than 75%.

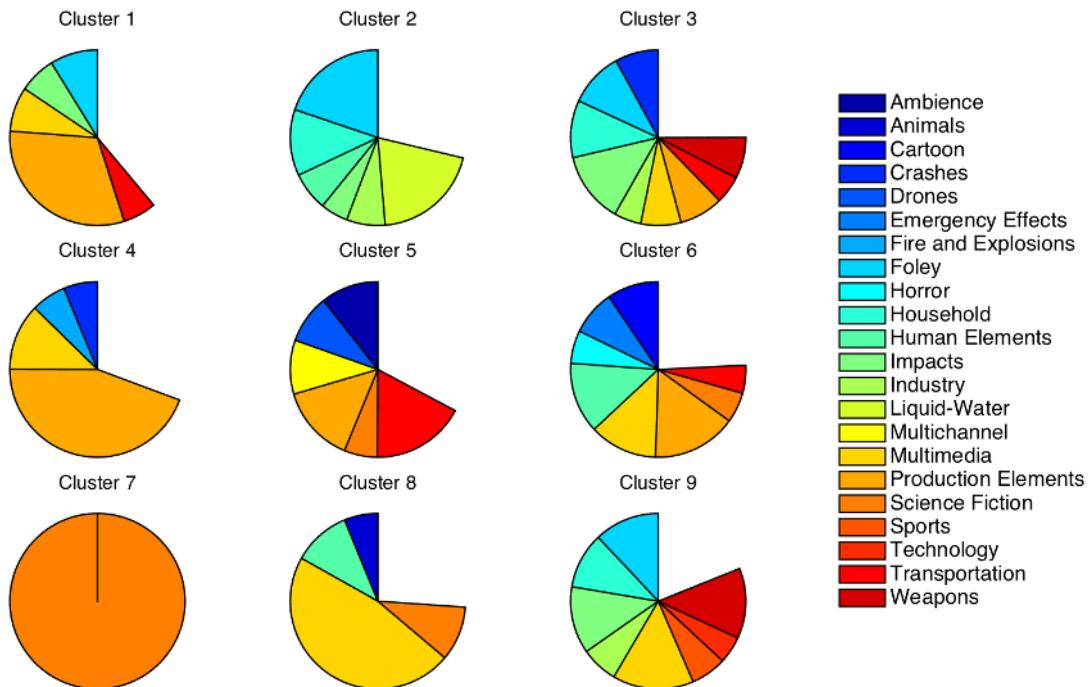


Figure 7: Dataset labels per cluster, where all labels that make up more than 5% of the dataset were plotted

tering methods. Further evaluation of clusters produced could be undertaken, as well as a deeper analysis into each of the identified clusters, to produce a deeper taxonomy.

7. ACKNOWLEDGEMENTS

This work was supported by EPSRC. Many thanks to anonymous internal reviews.

8. REFERENCES

- [1] Erling Wold et al., “Content-based classification, search, and retrieval of audio,” *IEEE multimedia*, vol. 3, no. 3, pp. 27–36, 1996.
- [2] Grégoire Lafay, Nicolas Misdariis, Mathieu Lagrange, and Mathias Rossignol, “Semantic browsing of sound databases without keywords,” *Journal of the Audio Engineering Society*, vol. 64, no. 9, pp. 628–635, 2016.
- [3] R Murray Schafer, *The soundscape: Our sonic environment and the tuning of the world*, Inner Traditions/Bear & Co, 1993.
- [4] Luigi Russolo and Francesco Balilla Pratella, *The art of noise:(futurist manifesto, 1913)*, Something Else Press, 1967.
- [5] Luigi Russolo, “The art of noises: Futurist manifesto,” *Audio culture: Readings in modern music*, pp. 10–14, 2004.
- [6] Ian Stevenson, “Soundscape analysis for effective sound design in commercial environments,” in *Sonic Environments Australasian Computer Music Conference*. Australasian Computer Music Association, 2016.
- [7] PerMagnus Lindborg, “A taxonomy of sound sources in restaurants,” *Applied Acoustics*, vol. 110, pp. 297–310, 2016.
- [8] 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.
- [9] Olivier Houix et al., “A lexical analysis of environmental sound categories.,” *Journal of Experimental Psychology: Applied*, vol. 18, no. 1, pp. 52, 2012.
- [10] James A Ballas, “Common factors in the identification of an assortment of brief everyday sounds.,” *Journal of experimental psychology: human perception and performance*, vol. 19, no. 2, pp. 250, 1993.
- [11] Brian Gygi, Gary R Kidd, and Charles S Watson, “Similarity and categorization of environmental sounds,” *Perception & psychophysics*, vol. 69, no. 6, pp. 839–855, 2007.
- [12] Kirsteen M Aldrich, Elizabeth J Hellier, and Judy Edworthy, “What determines auditory similarity? the effect of stimulus group and methodology,” *The Quarterly Journal of Experimental Psychology*, vol. 62, no. 1, pp. 63–83, 2009.
- [13] Monika Rychtáříková and Gerrit Vermeir, “Soundscape categorization on the basis of objective acoustical parameters,” *Applied Acoustics*, vol. 74, no. 2, pp. 240–247, 2013.
- [14] William J Davies et al., “Perception of soundscapes: An interdisciplinary approach,” *Applied Acoustics*, vol. 74, no. 2, pp. 224–231, 2013.
- [15] Iain McGregor et al., “Sound and soundscape classification: establishing key auditory dimensions and their relative importance,” in *12th International Conference on Auditory Display*, London, UK, June 2006.
- [16] Torben Holm Pedersen, *The Semantic Space of Sounds*, Delta, 2008.
- [17] James Woodcock et al., “Categorization of broadcast audio objects in complex auditory scenes,” *Journal of the Audio Engineering Society*, vol. 64, no. 6, pp. 380–394, 2016.
- [18] Justin Salamon, Christopher Jacoby, and Juan Pablo Bello, “A dataset and taxonomy for urban sound research,” in *Proceedings of the 22nd ACM international conference on Multimedia*. ACM, 2014, pp. 1041–1044.
- [19] Davide Rocchesso and Federico Fontana, *The sounding object*, Mondo estremo, 2003.
- [20] Edgar Hemery and Jean-Julien Aucouturier, “One hundred ways to process time, frequency, rate and scale in the central auditory system: a pattern-recognition meta-analysis,” *Frontiers in computational neuroscience*, vol. 9, 2015.
- [21] Justin Salamon and Juan Pablo Bello, “Unsupervised feature learning for urban sound classification,” in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2015, pp. 171–175.
- [22] Justin Salamon and Juan Pablo Bello, “Deep convolutional neural networks and data augmentation for environmental sound classification,” *IEEE Signal Process. Lett.*, vol. 24, no. 3, pp. 279–283, 2017.
- [23] David J. M. Robinson and Malcolm J. Hawksford, “Psychoacoustic models and non-linear human hearing,” in *109th Audio Engineering Society Convention*, Los Angeles, CA, USA, September 2000.
- [24] Dmitry Bogdanov et al., “Essentia: An audio analysis library for music information retrieval,” in *ISMIR*, 2013, pp. 493–498.
- [25] David Moffat, David Ronan, and Joshua D. Reiss, “An evaluation of audio feature extraction toolboxes,” in *Proc. 18th International Conference on Digital Audio Effects (DAFx-15)*, November 2015.
- [26] David Ronan, David Moffat, Hatice Gunes, and Joshua D. Reiss, “Automatic subgrouping of multitrack audio,” in *Proc. 18th International Conference on Digital Audio Effects (DAFx-15)*. DAFX-15, November 2015.
- [27] Leo Breiman, “Random forests,” *Machine learning*, vol. 45, no. 1, pp. 5–32, 2001.
- [28] Robin Genuer, Jean-Michel Poggi, and Christine Tuleau-Malot, “Variable selection using random forests,” *Pattern Recognition Letters*, vol. 31, no. 14, pp. 2225–2236, 2010.
- [29] Joe H Ward Jr, “Hierarchical grouping to optimize an objective function,” *Journal of the American statistical association*, vol. 58, no. 301, pp. 236–244, 1963.
- [30] David Baehrens et al., “How to explain individual classification decisions,” *Journal of Machine Learning Research*, vol. 11, no. Jun, pp. 1803–1831, 2010.
- [31] Leo Breiman et al., *Classification and regression trees*, CRC press, 1984.

THE MIX EVALUATION DATASET

Brecht De Man

Joshua D. Reiss

Centre for Digital Music,
School of Electronic Engineering and Computer Science,
Queen Mary University of London
London, UK
{b.deman,joshua.reiss}@qmul.ac.uk

ABSTRACT

Research on perception of music production practices is mainly concerned with the emulation of sound engineering tasks through lab-based experiments and custom software, sometimes with unskilled subjects. This can improve the level of control, but the validity, transferability, and relevance of the results may suffer from this artificial context. This paper presents a dataset consisting of mixes gathered in a real-life, ecologically valid setting, and perceptual evaluation thereof, which can be used to expand knowledge on the mixing process. With 180 mixes including parameter settings, close to 5000 preference ratings and free-form descriptions, and a diverse range of contributors from five different countries, the data offers many opportunities for music production analysis, some of which are explored here. In particular, more experienced subjects were found to be more negative and more specific in their assessments of mixes, and to increasingly agree with each other.

1. INTRODUCTION

Many types of audio and music research rely on multitrack audio for analysis, training and testing of models, or demonstration of algorithms. For instance, music production analysis [1], automatic mixing [2], audio effect interface design [3], instrument grouping [4], and various types of music information retrieval [5] all require or could benefit from a large number of raw tracks, mixes, and processing parameters. This kind of data is also useful for budding mix engineers, audio educators, and developers, as well as creative professionals in need of accompanying music or other audio where some tracks can be disabled [6].

Despite this, multitrack audio is scarce. Existing online resources of multitrack audio content typically have a relatively low number of songs, offer little variation, are restricted due to copyright, provide little to no metadata, or lack mixed versions and corresponding parameter settings. An important obstacle to the widespread availability of multitrack audio and mixes is copyright, which restricts the free sharing of most music and their components. Furthermore, due to reluctance to expose the unpolished material, content owners are unlikely to share source content, parameter settings, or alternative versions of their music. While there is no shortage of mono and stereo recordings of single instruments and ensembles, any work concerned with the study or processing of multitrack audio therefore suffers from a severe lack of relevant material.

This impedes reproduction or improvement of previous studies where the data cannot be made public, and comparison of different works when there is no common dataset used across a wider community. It further limits the generality, relevance, and quality of

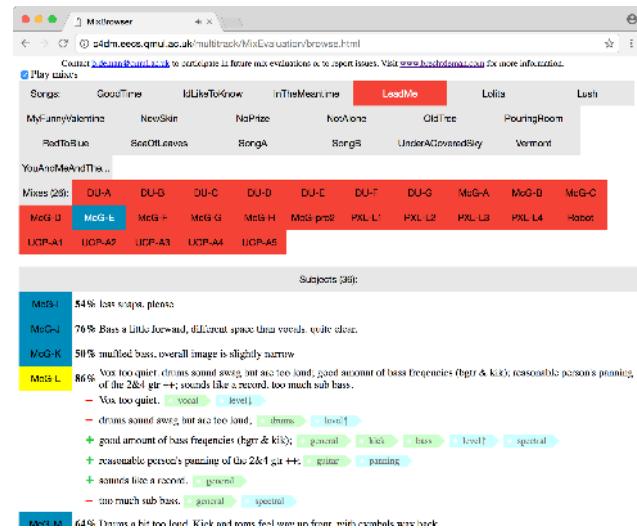


Figure 1: Online interface to browse the contents of the Mix Evaluation Dataset

the research and the designed systems. Even when some mixes are available, extracting data from mix sessions is laborious at best. For this reason, existing research typically employs lab-based mix simulations, which means that its relation to professional mixing practices is uncertain.

The dataset presented here is therefore based on a series of controlled experiments wherein realistic, ecologically valid mixes are produced — i.e. by experienced engineers, in their preferred environment and using professional tools — and evaluated. The sessions can be recreated so that any feature or parameter can be extracted for later analysis, and different mixes of the same songs are compared through listening tests to assess the importance and impact of their attributes. As such, both high-level information, including instrument labels and subjective assessments, and low-level measures can be taken into account. While some of the data presented here has been used in several previous studies, the dataset is now consolidated and opened up to the community, and can be browsed on c4dm.eecs.qmul.ac.uk/multitrack/MixEvaluation/, see Figure 1.

With close to 5000 mix evaluations, the dataset is by far the largest study of evaluated mixes known to the authors. MedleyDB, another resource shared with researchers on request, consists of raw tracks including pitch annotations, instrument activations, and metadata [7]. [8] analyses audio features extracted from a total of 1501 unevaluated mixes from 10 different songs. The same au-

thors examine audio features extracted from 101 mixes of the same song, evaluated by one person who classified the mixes in five preference categories [9]. In both cases, the mixes were created by anonymous visitors of the Mixing Secrets Free Multitrack Download Library [10], and principal component analysis preceded by outlier detection was employed to establish primary dimensions of variation. Parameter settings or individual processed stems were not available in these works.

This paper introduces the dataset and shows how it allows to further our understanding of sound engineering, and is structured as follows. Section 2 presents the series of acquisition experiments wherein mixes were created and evaluated. The resulting data is described in Section 3. Section 4 then demonstrates how this content can be used to efficiently obtain knowledge about music production practices, perception, and preferences. To this end, previous studies and key results based on part of this dataset are listed, and new findings about the influence of subject expertise based on the complete set are presented. Finally, Section 5 offers concluding remarks and suggestions for future research.

2. METHODOLOGY

2.1. Mix creation

Mix experiments and listening tests were conducted at seven institutions located in five different countries. The mix process was maximally preserved in the interest of ecological relevance, while information such as parameter settings was logged as much as possible. Perceptual evaluation further helped validate the content and investigate the perception and preference in relation to mix practices.

Students and staff members from sound engineering degrees at McGill University (McG), Dalarna University (DU), PXL University College (PXL), and Universidade Católica Portuguesa (UCP) created mixes and participated in listening tests. In addition, employees from a music production startup (MG) and researchers from Queen Mary University of London (QM) and students from the institution’s Sound and Music Computing master (SMC) took part in the perceptual evaluation stage as well.

Table 1 lists the songs and the corresponding number of mixes created from the source material, as well as the number of subjects evaluating (a number of) these mixes. Numbers between parentheses refer to additional mixes for which stems, Digital Audio Workstation (DAW) sessions, and parameter settings are not available. These correspond to the original release or analogue mixes, see Section 3.2. Songs with an asterisk (*) are copyrighted and not available online, whereas raw tracks to others can be found via the Open Multitrack Testbed¹ [11]. For two songs, permission to disclose artist and song title was not granted. Evaluations with an obelus (†) indicate that subjects included those who produced the mixes. Consistent anonymous identifiers of the participants (e.g. ‘McG-A’) allow exclusion of this segment or examination of the associated biases [12].

The participants produced these mixes in their preferred mixing location, so as to achieve a natural and representative spread of environments without a bias imposed by a specific acoustic space, reproduction system, or playback level. The toolset was restricted somewhat so that each mix could be faithfully recalled and analysed in depth later, with a limited number of software plugins available, typically consisting of those which come with the respective DAWs. All students used Avid Pro Tools 10, an in-

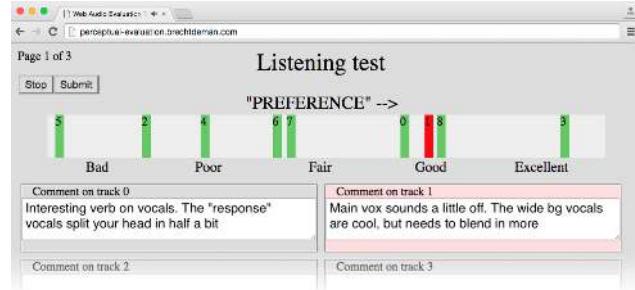


Figure 2: Example interface used to acquire subjective assessments of nine different mixes of the same song, created with the Web Audio Evaluation Tool

dustry standard DAW, except for the PXL group who used Apple Logic Pro X. Instructions explicitly forbade outboard processing, recording new audio, sample replacement, pitch and timing correction, rearranging sections, or manipulating audio in an external editor. Beyond this, any kind of processing was allowed, including automation, subgrouping, and multiling.

2.2. Perceptual evaluation

The different mixes were evaluated in a listening test using the interface presented in [13]. With the exception of groups McG and MG, the browser-based version of this interface from the Web Audio Evaluation Tool [14] was used, see Figure 2.

As dictated by common practices, this listening test was conducted in a double blind fashion [15], with randomised presentation order [16], minimal visual information [17], and free and immediate switching between time-aligned stimuli [18]. The interface presented multiple stimuli [19] on a single, ‘drag-and-drop’ rating axis [20], and without ticks to avoid build-up around these marks [21]. A ‘reference’ was not provided because it is not defined for this exceedingly subjective task. Indeed, even commercial mixes by renowned mix engineers prove not to be appropriate reference stimuli, as these are not necessarily rated more highly than mixes by students [12].

For the purpose of perceptual evaluation, a fragment consisting of the second verse and chorus was used. With an average length of one minute, this reduced the strain on the subjects’ attention, likely leading to more reliable listening test results. It also placed the focus on a region of the song where the most musical elements were active. In particular, the elements which all songs have in common (drums, lead vocal, and a bass instrument) were all active here. A fade-in and fade-out of one second were applied at the start and end of the fragment [1].

The headphones used were Beyerdynamic DT770 PRO for group MG and Audio Technica M50x for group SMC. In all other cases, the listening tests took place in dedicated, high quality listening rooms at the respective institutions, the room impulse responses of which are included in the dataset. This knowledge could be used to estimate the impact of the respective playback systems, although in these cases the groups differ significantly in other aspects as well.

Comments in other languages than English (DU, PXL, and UCP) were translated by native speakers of the respective languages, who are also proficient in English and have a good knowledge of audio engineering.

Table 1: Overview of mixed content, with number of mixes (left side) and number of subjects (right side) per song

ARTIST – SONG	GENRE	NUMBER OF MIXES				NUMBER OF SUBJECTS						
		McG	DU	PXL	UCP	McG	MG	QM	SMC	DU	PXL	UCP
The DoneFors – Lead Me	country	8 (2)	(7)	(4)	5	15	8	10	4	39†	6†	10†
Fredy V – In The Meantime	funk	8 (1)	(7)	(7)	5	22†		10	5	38†	8†	10†
Joshua Bell – My Funny Valentine*	jazz	8 (2)				14	7	10	5			
Artist X – Song A*	blues	8 (2)				14	8	10	9			
Artist Y – Song B*	blues	8 (2)				14		10	5			
Dawn Langstroth – No Prize*	jazz	8 (2)				14	8	10	5			
Fredy V – Not Alone	soul	8 (2)				13		10	5			
Broken Crank – Red To Blue	rock	8 (2)				13		10	4			
The DoneFors – Under A Covered Sky	pop	8 (2)				13		10	4			
The DoneFors – Pouring Room	indie	8 (1)				22†		9	6			
Torres – New Skin	indie		(7)				9	6	38†			
Filthybird – I'd Like To Know	pop rock			7			11	5		13†		
The Districts – Vermont	pop rock	2	5				11	5		13†		
Creepoid – Old Tree	indie rock				5						5	
Purling Hiss – Lolita	hard rock				5						5	
Louis Cressy Band – Good Time	rock				4						5	
Jokers, Jacks & Kings – Sea Of Leaves	pop rock				4						5	
Human Radio – You & Me & the Radio	pop rock				4						5	

Table 2 shows additional details of the perceptual evaluation experiments.

3. CONTENT

3.1. Raw tracks

Raw tracks of the mixes can be found via the Open Multitrack Testbed¹. The first two sections of Table 1 are newly presented here and were recorded by Grammy-winning engineers. Six of the ten songs are made available in their entirety under a Creative Commons BY 4.0 license. Raw tracks to songs from the last two sections of the table can be downloaded from Weathervane Music's Shaking Through² and Mike Senior's Mixing Secrets Multitrack Library [10], respectively. Many more mixes of these tracks are available on the forums of these websites, albeit without associated parameter settings or evaluations.

3.2. Mixes and stems

All stereo mixes are available in uncompressed, high resolution WAV format. Unique to this dataset is the availability of DAW session files, which includes all parameter setting of ‘in-the-box’ mixes. Where the mix and its constituent elements could be recreated, stems of the vocal, kick drum, snare drum, rest of the drums, and bass instrument are rendered. Similarly, the sum of all reverberation signals (‘wet’) and the rest of a mix (‘dry’), as in [22], are shared as well.

The dataset also contains mixes which were produced mostly through analogue processing. While this makes detailed analysis more difficult, it increases the diversity and allows a wider range of possible research questions the data could answer. To mitigate this relative lack of control, approximate parameter settings can be derived from recall sheets, pictures of the devices, the parsed recall files from the SSL AWS900 console (DU), and a recording of a fragment of each channel as the engineer sequentially solos each track (PXL).

¹multitrack.eecs.qmul.ac.uk

²weathervanemusic.org/shakingthrough

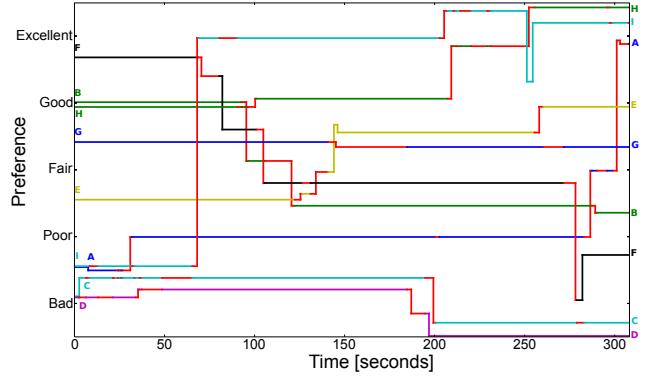


Figure 3: Built-in timeline visualisation of the Web Audio Evaluation Tool, showing playback (red) and movement of sliders of a single subject rating nine mixes of a single song

3.3. Preference ratings

Evaluation of audio involves a combination of hedonic and sensory judgements. Preference is an example of a hedonic judgement, while (basic audio) quality — ‘the physical nature of an entity with regards to its ability to fulfill predetermined and fixed requirements’ [23] — is a more sensory judgement [24]. Indeed, preference and perceived quality are not always concurrent [25]: a musical sample of lower perceived quality, e.g. having digital glitches or a ‘lo-fi’ sound, may still be preferred to other samples which are perceived as ‘clean’, but don’t have the same positive emotional impact. Especially when no reference is given, subjects sometimes prefer a ‘distorted’ version of a sound [26]. Personal preference was therefore deemed a more appropriate attribute than audio quality or fidelity. Such a single, hedonic rating can reveal which mixes are preferred over others, and therefore which parameter settings are more desirable, or which can be excluded from analysis. Where the Web Audio Evaluation Tool was used, the positions of the sliders over time was registered as well, see Figure 3.

Table 2: Overview of evaluation experiments

	McG	MG	QM	SMC	DU	PXL	UCP	TOTAL
Country	Canada		United Kingdom	Sweden	Belgium	Portugal		
#subjects	33	8	21	26	39	13	10	150
#songs	10	4	13	14	3	4	7	18
#mixes	98	40	111	116	21	23	42	181
#evaluations	1444	310	1129	639	805	236	310	4873
#statements	4227	585	2403	1190	2331	909	1051	12696
#words/comment	13.39	11.76	11.32	12.39	18.95	31.94	25.21	15.25
Male/female	28/5	7/1	18/3	14/12	33/6	13/0	9/1	122/28
Loudspeakers/headphones	LS	HP	LS	HP	LS	LS	LS	

3.4. Comments

A single numerical rating does not convey any detailed information about what aspects of a mix are (dis)liked. For instance, a higher score for mixes with a higher dynamic range [12] may relate to subtle use of dynamic range compression (e.g. preference for substantial level variations), but also to a relatively loud transient source (e.g. preference for prominent snare drums). In addition, the probability of spurious correlation increases as an ever larger number of features is considered. Furthermore, subjects tend to be frustrated when they do not have the ability to express their thoughts on a particular attribute, and isolated ratings do not provide any information about the difficulty, focus, or thought process associated with the evaluation task.

For this reason, free-form text response in the form of comment boxes is accommodated, facilitating in-depth analysis of the perception and preference with regard to various music production aspects. The results of this ‘free-choice profiling’ also allow learning how subjects used and misused the interface. An additional, practical reason for allowing subjects to write comments is that taking notes on shortcomings or strengths of the different mixes helps them to keep track of which fragment is which, facilitating the complex task at hand.

Extensive annotation of the comments is included in the form of tags associated with each atomic ‘statement’ of which the comment consists. For instance, a comment ‘Drums a little distant. Vox a little hot. Lower midrange feels a little hollow, otherwise pretty good.’ comprises four separate statements. Tags then indicate the instrument (‘drums’, ‘vocal’, ...), feature (‘level (high)’, ‘spectrum’, ...), and valence (‘negative’/‘positive’).

The XML structure of the Web Audio Evaluation Tool output files was adopted to share preference ratings, comments, and annotation data associated with the content.

3.5. Metadata

3.5.1. Track labels

Reliable track metadata can serve as a ground truth that is necessary for applications such as instrument identification, where the algorithm’s output needs to be compared to the actual instrument. Providing this data makes this dataset an attractive resource for training or testing such algorithms as it obviates the need for manual annotation of the audio, which can be particularly tedious if the number of files becomes large.

The available raw tracks and mixes are annotated on the Open Multitrack Testbed¹, including metadata describing for instance the respective instruments, microphones, and take numbers. This

metadata further allows tracks and mixes to be found through the Testbed’s search and browse interfaces.

3.5.2. Genre

The source material was selected in coordination with the programme’s teachers from the participating institutions, because they fit the educational goals, were considered ecologically valid and homogeneous with regard to production quality, and were deemed to represent an adequate spread of genre. Due to the subjective nature of musical genre, a group of subjects were asked to comment on the genres of the songs during the evaluation experiments, providing a post hoc confirmation of the musical diversity. Each song’s most often occurring genre label was added to Table 1 for reference.

3.6. Survey responses and subject demographics

The listening test included a survey to establish the subjects’ gender, age, experience with audio engineering and playing a musical instrument (in number of years and described in more detail), whether they had previously participated in (non-medical) listening tests, and whether they had a cold or condition which could negatively affect their hearing.

4. ANALYSIS

4.1. Prior work

The McG portion of this dataset has previously been used in studies on mix practices and perception, as detailed below.

The mild constraints on tools used and the availability of parameter settings allows one to compare signal features between different mixes, songs, or institutions, and identify trends. A detailed analysis of tendencies in a wide range of audio features — extracted from vocal, drum (kick drum, snare drum, and other), bass, and mix stems — appeared in [27]. As an example, Figure 4 shows the ITU-R BS.1770 loudness [28] of several processed stems for two songs, as mixed by engineers from two institutions (McG and UCP). No significant differences in balance choices are apparent here.

Correlation between preference ratings and audio features extracted from the total mixes have shown a higher preference for mixes with relative higher dynamic range, and mixes with a relatively strong phantom centre [12].

In [29], relative attention to each of these categories was quantified based on annotated comments. Figure 5 shows the relative proportion of statements referring to detailed feature categories for the complete dataset (all groups).

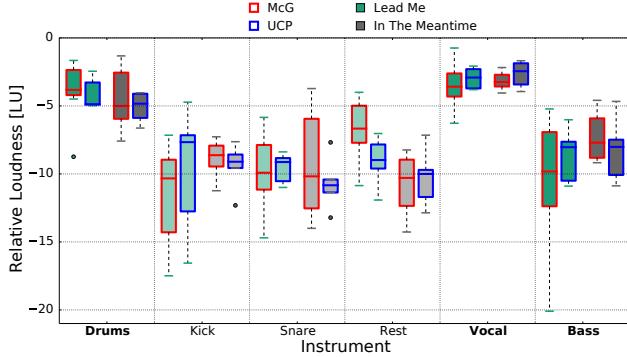


Figure 4: Stem loudness relative to the total mix of Fredy V’s In The Meantime and The DoneFors’ Lead Me, as mixed by students from the McG and UCP groups

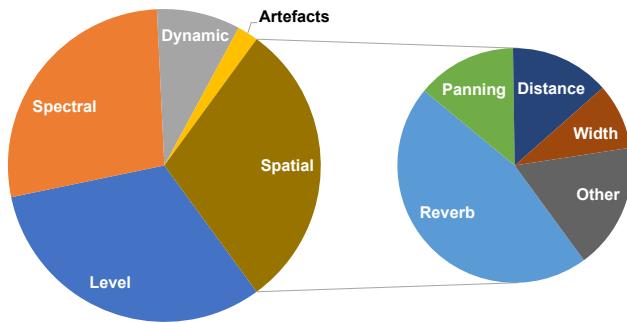


Figure 5: Relative proportion of statements describing mix aspects

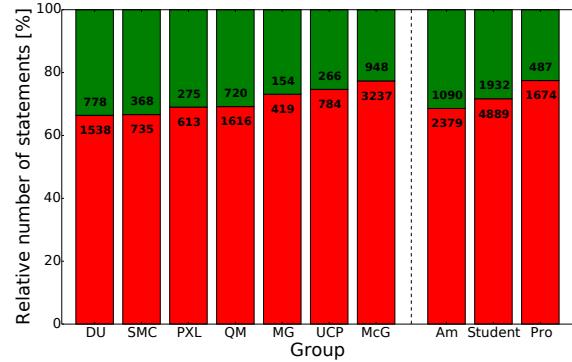
Finally, through combination of the comment annotations with preference ratings and extracted audio features, more focused research questions about music production can be answered. Proving this concept, [22] showed a notably lower preference rating for mixes tagged as overly reverberant than for those which have an alleged lack of reverberant energy, and determined that the optimal reverb loudness relative to the total mix loudness is close to -14 LU.

In addition to being able to render the entire mix or any part thereof, availability of DAW session files also presents a unique opportunity to study workflow and signal routing practices from working mix engineers in a realistic setting. As an example, the process of subgrouping has been studied in [30], where a strong correlation was shown between the number of raw tracks used and the number of subgroups that was created, as well as a medium correlation between the number of subgroups which were processed by EQ and the average preference rating for that mix.

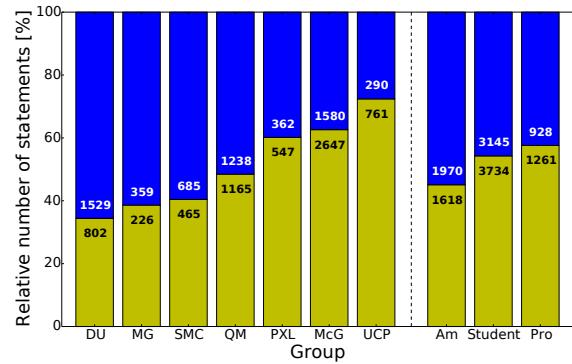
4.2. Effects of subject background

Access to the subject’s level of experience, institution, and demographics makes it possible to determine the influence of these factors on subjective preference and perception.

For different levels of expertise, the average rating from professionals (teaching and/or practising sound engineering professionally) is lower than from amateurs (no formal training in sound engineering) and students (currently training to be a sound engineer, and contributing mixes to the experiment), as expected [1].



(a) Proportion of negative (red) vs. positive (green) statements



(b) Proportion of instrument-specific (yellow) vs. general (blue) statements

Figure 6: Statement categories as a function of subject data

The proportion of negative statements among the comments is strongly influenced by the level of expertise of the subject as well: there is a significant tendency to criticise more, proportionally, with increasing experience, see Figure 6a. Independent of level of expertise, the proportion of negative statements is also significantly different per group.

Likewise, it is clear that amateurs tend to give more ‘general’ comments, not pertaining to any particular instrument, as shown in Figure 6b. This accounts for 55% of their statements. For students and professionals this proportion is 46% and 42%, respectively. The different groups also meaningfully differ with regard to the proportion of statements that discuss the mix as a whole, from 25% at UCP to 63% at DU. As these two groups consisted of bachelor students only, the level of expertise is presumably similar and other factors must be at play.

Finally, the agreement within as well as between the groups is quantified, showing the relative number of statements which are consistent with each other. In this context, a (dis)agreement is defined as a pair of statements related to the same instrument-processing pair and mix (e.g. each discussing ‘vocal, level’ for mix ‘McG-A’ of the song ‘Lead Me’), with one statement confirming or opposing the other, respectively, with regard to either valence (‘negative’ versus ‘positive’) or value (‘low’ versus ‘high’). Only the processing categories ‘level’, ‘reverb’, ‘distance’, and ‘width’ have been assigned a value attribute. The ratio of agreements r_{AB} between two groups A and B is given by

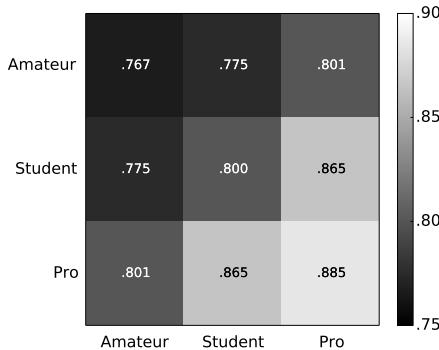


Figure 7: Level of agreement between groups of different expertise

$$r_{AB} = \frac{a_{AB}}{d_{AB} + a_{AB}} \quad (1)$$

where a_{AB} and d_{AB} are the total number of agreeing and disagreeing pairs of statements, respectively, where a pair of statements consists of a statement from group A and a statement from group B on the same topic.

Between and within the different levels of expertise, agreement increases consistently from amateurs over students to professionals, see Figure 7. In other words, skilled engineers are less likely to contradict each other when evaluating mixes (high within-group agreement), converging on a ‘common view’. This supports the notion that universal ‘best practices’ with regard to mix engineering do exist, and that perceptual evaluation is more reliable and efficient when the participants are skilled. Conversely, amateur listeners tend to make more statements which are challenged by other amateurs, as well as more experienced subjects (low within-group and between-group agreement). As the within-group agreement of amateurs is lower than any between-group agreement, this result does not indicate any consistent differences of opinion between two groups. For instance, there is no evidence that ‘amateurs want the vocal considerably louder than others’. Such distinctions may exist, but revealing them requires in-depth analysis of the individual statement categories. The type of agreement analysis proposed here can be instrumental in comparing the quality of (groups of) subjects, on the condition that the evaluated stimuli are largely the same.

Further analysis is necessary, for instance to establish which of these differences are significant and not spurious or under influence of other factors.

5. CONCLUSION

A dataset of mixes of multitrack music and their evaluations was constructed and released, based on contributions from five different countries. The mixes were created by skilled sound engineers, in a realistic setting, and using professional tools, to minimally disrupt the natural mixing process. By including parameter settings and limiting the set of software tools, mixes can be recreated and analysed in detail. Previous studies using this data were listed, and further statistical analysis of the data was presented.

In particular, it was shown that expert listeners are more likely to contribute negative and specific assessments, and to agree with others about various aspects of the mix. This is consistent with

the expectation that they are trained to spot and articulate problems with a mix. Conversely, one could suppose amateur subjects lack the vocabulary or previous experience to formulate detailed comments about unfavourable aspects, instead highlighting features that tastefully grab attention and stand out in a positive sense.

The dataset and potential extensions offer interesting opportunities for further cross-analysis, comparing the practices, perception, and preferences of different groups. At this point, however, the dataset is heavily skewed towards Western musical genres, engineers, and subjects, and experienced music producers. Extension of the acquisition experiments presented here, with an emphasis on content from countries outside of North America and Western Europe, can mitigate this bias and help answer new research questions. In addition, a substantially larger dataset can be useful for analysis which requires high volumes of data, such as machine learning of music production practices [31].

6. ACKNOWLEDGEMENTS

The authors would like to thank Frank Duchêne, Pedro Pestana, Henrik Karlsson, Johan Nordin, Mathieu Barthet, Brett Leonard, Matthew Boerum, Richard King, and George Massenburg, for facilitating and contributing to the experiments. Special thanks also go to all who participated in the mix creation sessions or listening tests.

7. REFERENCES

- [1] Alex Wilson and Bruno M. Fazenda, “Perception of audio quality in productions of popular music,” *Journal of the Audio Engineering Society*, vol. 64, no. 1/2, pp. 23–34, Jan/Feb 2016.
- [2] Enrique Perez Gonzalez and Joshua D. Reiss, “Automatic mixing: Live downmixing stereo panner,” *Proc. Digital Audio Effects (DAFx-07)*, Sep 2007.
- [3] Spyridon Stasis, Ryan Stables, and Jason Hockman, “A model for adaptive reduced-dimensionality equalisation,” *Proc. Digital Audio Effects (DAFx-15)*, Dec 2015.
- [4] David Ronan et al., “Automatic subgrouping of multitrack audio,” *Proc. Digital Audio Effects (DAFx-15)*, Nov 2015.
- [5] Justin Salamon, “Pitch analysis for active music discovery,” *33rd Int. Conf. on Machine Learning*, June 2016.
- [6] Chris Greenhalgh et al., “GeoTracks: Adaptive music for everyday journeys,” *ACM Int. Conf. on Multimedia*, Oct 2016.
- [7] Rachel Bittner et al., “MedleyDB: a multitrack dataset for annotation-intensive MIR research,” *15th International Society for Music Information Retrieval Conf. (ISMIR 2014)*, Oct 2014.
- [8] Alex Wilson and Bruno Fazenda, “Variation in multitrack mixes: Analysis of low-level audio signal features,” *Journal of the Audio Engineering Society*, vol. 64, no. 7/8, pp. 466–473, Jul/Aug 2016.
- [9] Alex Wilson and Bruno Fazenda, “101 mixes: A statistical analysis of mix-variation in a dataset of multi-track music mixes,” *Audio Engineering Society Convention 139*, Oct 2015.
- [10] Mike Senior, *Mixing Secrets*, Taylor & Francis, www.cambridge-mt.com/ms-mtk.htm, 2012.

- [11] Brecht De Man et al., “The Open Multitrack Testbed,” *Audio Engineering Society Convention 137*, Oct 2014.
- [12] Brecht De Man et al., “Perceptual evaluation of music mixing practices,” *Audio Engineering Society Convention 138*, May 2015.
- [13] Brecht De Man and Joshua D. Reiss, “APE: Audio Perceptual Evaluation toolbox for MATLAB,” *Audio Engineering Society Convention 136*, Apr 2014.
- [14] Nicholas Jillings et al., “Web Audio Evaluation Tool: A browser-based listening test environment,” *12th Sound and Music Computing Conf.*, July 2015.
- [15] Floyd E. Toole and Sean Olive, “Hearing is believing vs. believing is hearing: Blind vs. sighted listening tests, and other interesting things,” *Audio Engineering Society Convention 97*, Nov 1994.
- [16] S. Bech and N. Zacharov, *Perceptual Audio Evaluation - Theory, Method and Application*, John Wiley & Sons, 2007.
- [17] Jan Berg and Francis Rumsey, “Validity of selected spatial attributes in the evaluation of 5-channel microphone techniques,” *Audio Engineering Society Convention 112*, Apr 2002.
- [18] Neofytos Kaplanis et al., “A rapid sensory analysis method for perceptual assessment of automotive audio,” *Journal of the Audio Engineering Society*, vol. 65, no. 1/2, pp. 130–146, Jan/Feb 2017.
- [19] Sean Olive and Todd Welti, “The relationship between perception and measurement of headphone sound quality,” *Audio Engineering Society Convention 133*, Oct 2012.
- [20] Christoph Völker and Rainer Huber, “Adaptions for the MUlti Stimulus test with Hidden Reference and Anchor (MUSHRA) for elder and technical unexperienced participants,” *DAGA*, Mar 2015.
- [21] Jouni Paulus, Christian Uhle, and Jürgen Herre, “Perceived level of late reverberation in speech and music,” *Audio Engineering Society Convention 130*, May 2011.
- [22] Brecht De Man, Kirk McNally, and Joshua D. Reiss, “Perceptual evaluation and analysis of reverberation in multitrack music production,” *Journal of the Audio Engineering Society*, vol. 65, no. 1/2, pp. 108–116, Jan/Feb 2017.
- [23] Ute Jekosch, “Basic concepts and terms of ‘quality’, reconsidered in the context of product-sound quality,” *Acta Acustica united with Acustica*, vol. 90, no. 6, pp. 999–1006, November/December 2004.
- [24] Slawomir Zielinski, “On some biases encountered in modern listening tests,” *Spatial Audio & Sensory Evaluation Techniques*, Apr 2006.
- [25] Francis Rumsey, “New horizons in listening test design,” *Journal of the Audio Engineering Society*, vol. 52, pp. 65–73, Jan/Feb 2004.
- [26] Stanley P. Lipshitz and John Vanderkooy, “The Great Debate: Subjective evaluation,” *Journal of the Audio Engineering Society*, vol. 29, no. 7/8, pp. 482–491, Jul/Aug 1981.
- [27] Brecht De Man et al., “An analysis and evaluation of audio features for multitrack music mixtures,” *15th International Society for Music Information Retrieval Conf. (ISMIR 2014)*, Oct 2014.
- [28] Recommendation ITU-R BS.1770-4, “Algorithms to measure audio programme loudness and true-peak audio level,” Oct 2015.
- [29] Brecht De Man and Joshua D. Reiss, “Analysis of peer reviews in music production,” *Journal of the Art of Record Production*, vol. 10, July 2015.
- [30] David Ronan et al., “The impact of subgrouping practices on the perception of multitrack mixes,” *Audio Engineering Society Convention 139*, Oct 2015.
- [31] Stylianos Ioannis Mimalakis et al., “New sonorities for jazz recordings: Separation and mixing using deep neural networks,” *2nd AES Workshop on Intelligent Music Production*, Sep 2016.

THE SNAIL: A REAL-TIME SOFTWARE APPLICATION TO VISUALIZE SOUNDS

Thomas Hélie

S3AM team, STMS, IRCAM-CNRS-UPMC
1 place Igor Stravinsky, 75004 Paris, France
thomas.helie@ircam.fr

Charles Picasso

Analysis/Synthesis team, STMS, IRCAM-CNRS-UPMC
1 place Igor Stravinsky, 75004 Paris, France
Charles.Picasso@ircam.fr

ABSTRACT

The Snail is a real-time software application that offers possibilities for visualizing sounds and music, for tuning musical instruments, for working on pitch intonation, etc. It incorporates an original spectral analysis technology (patent-pending) combined with a display on a spiral representation: the center corresponds to the lowest frequencies, the outside to the highest frequencies, and each turn corresponds to one octave so that tones are organized with respect to angles. The spectrum magnitude is displayed according to perceptive features, in a redundant way: the loudness is mapped to both the line thickness and its brightness. However, because of the time-frequency uncertainty principle, using the Fourier spectrum (or also Q-transform, wavelets, etc) does not lead to a sufficient accuracy to be used in a musical context. The spectral analysis is completed by frequency precision enhancer based on a post-processing of the demodulated phase of the spectrum. This paper presents the scientific principles, some technical aspects of the software development and the main display modes with examples of use cases.

1. INTRODUCTION

Spiral representation of the audio spectrum allows the combination of scientific signal processing techniques and of a geometric organization of frequencies according to chroma and octaves. Compared to spectro-temporal representations, it offers a complementary or alternative solution that is natural for musical applications. For this reason, a piece of software or hardware applications have been developed (see [1, 2, 3] and [4] for a review). Scattering methods based on spiral geometries have also been proposed, with applications in audio classification, blind source separation, transcription or other processing tasks [5]. Other methods exploiting circular geometries through the use of chroma have been designed for several automatic musical analysis tasks (see e.g. [6, 7]). Such methods are efficient for music information retrieval and its applications. For the musicians and for musical or audio communities, visualizing raw data under such natural geometries (without any decision process) is also interesting: it allows humans to monitor their actions through a direct feedback, with potential human-learning issues if the perceptible feedback is accurate enough.

This paper presents a real-time application, *The Snail*, that gather several properties to provide an intuitive and accurate rendering:

(P1) Spiral abacus. One chroma is one angle and one round is one octave;

(P2) Perceptual simple¹ mapping. Twice louder is an audio stimulus, twice visible is the graphic symbol (with redundancy);

¹ Only the loudness of the spectrum is considered. Masking or dynamic loudness modeling are not taken in account in this study.

twice brighter, twice larger);

(P3) Frequency accuracy and stationarity. The frequency accuracy can be adjusted to enhance or select frequency components (partials) according to a targeted tolerancy, for: (a) instrument tuning tasks, or (b) musical interpretation (glissando, vibrato, orchestral mass effect, etc) and training.

For a tuning task, some "high accuracy" (that can still be controlled by musicians) can require about² a 2Hz-precision (voice and wind instruments), a 1Hz precision (string instruments) or much lower (0.1Hz) for analogue synthesizers in order to control slow beatings between several oscillators. To visualize musical interpretations, or in a musical training context, such an accuracy can be relaxed (typically from 4Hz to 10Hz) because of the vibrato, the mass effect (non synchronized signals when several instrumentists play the same notes), the pitch contour, etc. The software application has been designed to handle properties (P1-P3) and to propose solutions that cover such musical contexts.

The paper is organized as follows. Section 2 provides some recalls on the motivation and the problem statement. Section 3 presents the scientific principles used in the application. Section 4 addresses the software development, including the user interface design, the software structure and platforms. Finally, section 5 gives some conclusions and perspectives.

2. MOTIVATION AND PROBLEM STATEMENT

2.1. Motivation

Our very first motivation, at the basis of the Snail development, appeared in 2012: it simply consisted of representing the spectrum of sounds in a complete way (with magnitude and phase) on an abacus that organizes frequencies f (in Hertz) with respect to angles θ (in radians) according to chroma. This corresponds to the mapping³

$$\theta(f) = 2\pi \log_2(f/f^*), \quad (1)$$

on a typical audio frequency range $f \in [f^-, f^+]$ with $f^- = 20 > 0$ Hz and $f^+ = 20$ kHz, and where the tuning reference frequency (typically, $f^* = 440$ Hz) is mapped to angle 0. To have a bijective mapping between frequencies and such a chroma organization, angles θ must be complete by an octave information, under a 3D form (see [8, Fig. 8, p.105] and the so-called spiral array in [9, p.46]) or a spiral 2D form. A simple choice⁴ is to map the radius ρ such

²The musical values are usually measured in cents. Here, they are given in Hertz, typically for a reference note at frequency 440Hz.

³This formula provides a counter-clockwise winding, and its negative version a clockwise wending.

⁴Other conventions can be chosen. In particular, preserving the length of the frequency axis yields an analytic expression for ρ . However, for more than 2 octaves, this choice makes the low-frequency range too small for visualization in practice.

that it is increased by a unit value at each rising octave, as

$$\rho(f) = 1 + \log_2(f/f^-). \quad (2)$$

The goal was to bring together standard tools of signal processing (Fourier [10], or also constant-Q [11], wavelet analysis [12], etc) and a natural musical representation of frequencies, in order to explore its applicative interests in a real-time context. A first real-time tool were built, based on a Fourier analysis (see figure 1) and tested.

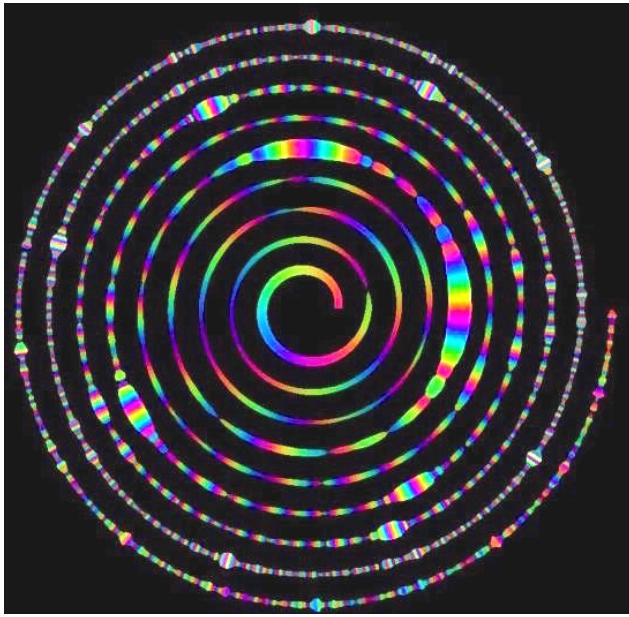


Figure 1: Basic representation of the spectrum on a spiral abacus (figure 3.18 extracted from [13]): the signal is composed of a collection of pure sines, analyzed with a Hanning window (duration 50 ms). The line thickness is proportional to the magnitude (dB scale on a 100dB range), the color corresponds to the phase (in its demodulated version to slow down the color time-variation, without information loss, and with a circular colormap to avoid jumps between 2π rd and 0 rd).

Its practical use on basic (monophonic or poor) musical signals proves to be attractive but the separation and the frequency location of partials are not accurate enough (also using constant-Q or wavelet transforms) to be used in a musical context.

2.2. Problem statement

To cope with this separation and accuracy difficulties, reassignment methods are available [14, 15] as well as methods based on signal derivatives [16, 17] (see also [18] for a comparison). These methods allows the estimation of frequencies from spectrum information, including for partials with locally time-varying frequencies.

To address property (P3), a basic method is proposed, that does not use frequency estimation. It consists of applying a contracting contrast factor to the spectrum magnitude (no reassignment). This factor is designed to weaken the energetic parts for which the phase time-evolution does not match with that of the bin frequency, according to a targeted tolerance (see § 3). In short, the method can

be illustrated by the following analogy: the idea is similar to applying a stroboscope to the rotating phase of each bin, at the bin frequency (phase demodulation), and to select the magnitude of the sufficiently slow rotations. This approach has some relevant interests for the visualizer application:

- the targeted accuracy is consistent with musical applications and can be adjusted independently from the analysis window duration;
- it is robust to noisy environment since the most unstationary is a component, the cleaner is the output;
- in particular, for tuning tasks, a sustained long note played by an instrumentist can be significantly enhanced compared to fast notes (played by some neighbors before a repetition), by using a very selective threshold (1 Hz), while the tuning accuracy is very high.

3. SCIENTIFIC PRINCIPLE

The Snail is composed of two modules [19]: (A) a sound analyzer, (B) a display.

3.1. Analyzer

The analyzer takes as input a monophonic sound, sampled at a frequency F_s . It delivers four outputs to be used in the visualizer: (1) a tuned frequency grid (Freq_v) and, for each frame n , (2) the associated spectrum magnitudes (Amp_v) and (3) demodulated phases (PhiDem_v), (4) a "phase constancy" index (PhiCstcy_v). Its structure is described in figure 2. It is decomposed into 7 basic blocks, labelled (A0) to (A6), see figure 2.

Blocks (A0-A1) are composed of a gain and a Short Time Fourier Transform with a standard window (Hanning, Blakman, etc) of duration T (typically, about 50 ms) and overlapped frames. The initial time of frame n is $t_n = n\tau$ ($n \geq 0$) where $\tau < T$ is the time step.

Blocks (A2-A3) process interpolations. A frequency grid (block A2) with frequencies

$$\text{Freq_v}(k) = f^* \cdot 2^{(m_k - m^*)/12}$$

is built according to a (rational) uniformly-spaced midi-code grid [20]

$$m_k = m^- + (k/K)(m^+ - m^-), \quad 0 \leq k \leq K,$$

where $m^* = 69$ is the midi code of the reference note (A4), m^- is that of the lowest note, m^+ that of the highest one, and where f^* denotes the reference tuning frequency (typically, $f^* = 440$ Hz). In the Snail application, K is chosen to have 50 points between each semi-tones, providing a resolution of 1 cent.

For each frame n , block (A3) builds the spectrum complex magnitude S_k associated with the frequency grid $\{f_k\}_{0 \leq k \leq K}$ based on an interpolation method (e.g. affine).

Block (A4) computes the modulus $\text{Amp_v}(k)$ and the phases φ_k from the complex magnitudes delivered by block (A3).

Block (A5) computes the demodulated phases

$$\text{PhiDem_v}(k) = \varphi_k - 2\pi \text{Freq_v}(k)t_n.$$

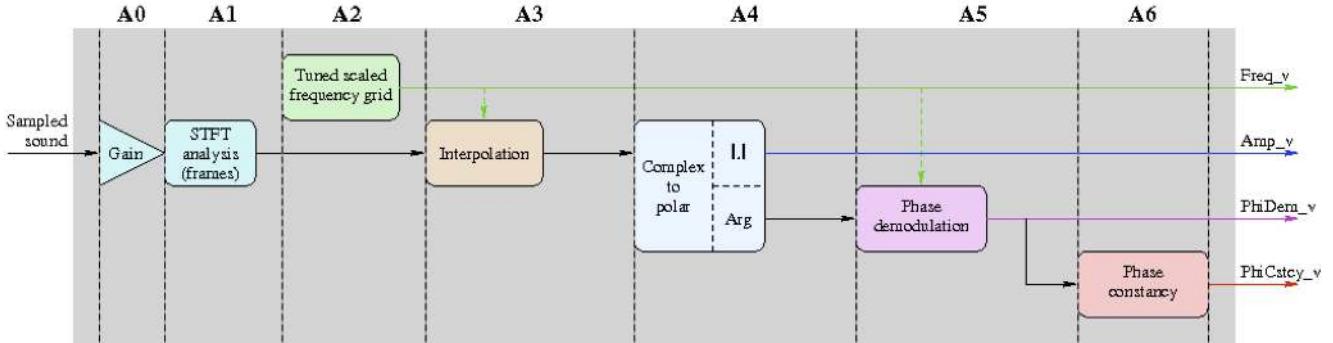


Figure 2: Block Diagram of the Analyzer

Block (A6) delivers a phase constancy index as follows. First, the demodulated phases $\text{PhiDem}_v(k)$ are converted into a complex value on the unit circle $z_k = \exp(i \text{PhiDem}_v(k))$. Second, for each k , independently, this complex value feeds (at each frame) a digital low pass filter (typically, a maximally flat Butterworth filter), with cutoff frequency F_c , at sampling frequency $1/\tau$. Third, the phase constancy index $\text{PhiCstcy}_v(k)$ is computed as the squared modulus of the filter output. Consequently, if the demodulated phase rotates less rapidly than F_c revolutions per second, the phase constancy index is close to 1. If the rotation speed is faster, the index is close to 0. In short, the phase constancy index provides a quasi-unit factor for the bins for which the spectrum phase is consistently synchronized with the bin frequency, up to a deviation of $\pm F_c$. It provides a quasi-zero factor outside this synchronization tolerance. Its effect is illustrated in figures 3-5, as detailed below.

3.2. Visualizer and illustration of some stages of the analyzer

The visualizer builds colored thick lines to represent the spectral activated zones on the spiral abacus. First, the magnitudes $\text{Amp}_v(k)$ are converted into loudness $L_v(k)$, according to the ISO226 norm [21]. Second, the line thickness is built as the product of the loudness $L_v(k)$ and the phase constancy index $\text{PhiCstcy}_v(k)$. Third, the color is built. The loudness $L_v(k)$ is mapped to the brightness. The hue and saturation are built according to several modes:

Magnitude: the hue and saturation are built as a function of the loudness $L_v(k)$;

Phase: they are built as a function of $\text{PhiDem}_v(k)$, according to a circular colormap (see figure 1);

Phase constancy: they are built as a function of $\text{PhiCstcy}_v(k)$ so that the color indicates the quality of the phase synchronization.

An illustration of the accuracy improvement that results from the analyzer is given in figures 3-5 for a C major chord (C-E-G) played on a Fender-Rhodes piano. These figures describe several intermediate stages of the analysis process. In figure 3, the color corresponds to the phase mode and the thickness corresponds to the loudness, without taking into account the correction by the phase constancy index PhiCstcy_v : the accuracy is the same as in figure 1. Figure 4 is the same as figure 3 except that (only for

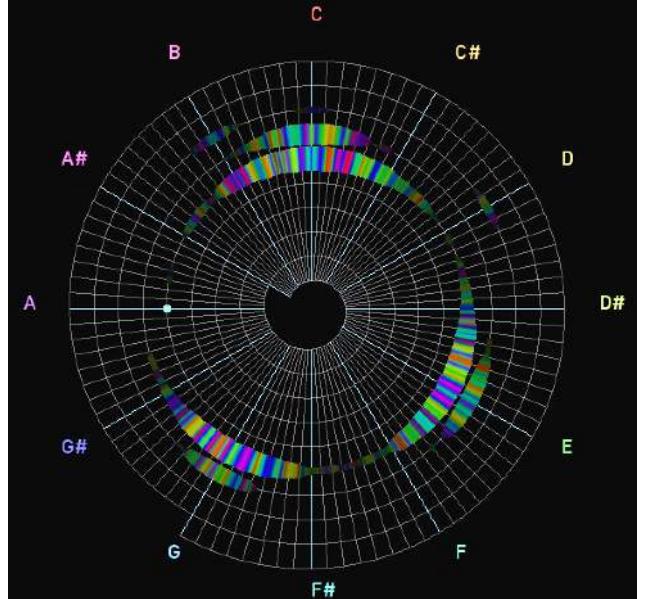


Figure 3: C major chord (Fender-Rhodes piano): the line thickness depends on the loudness but is not corrected by the phase constancy index; the color corresponds to the demodulated phases.

the illustration) the phase constancy index is mapped to the color saturation. The saturated parts correspond to the synchronized parts (here, up to $\pm F_c$ with $F_c = 2$ Hz) whereas the grey parts are the parts to reject. Finally, figure 5 provides the final result, in which the line thickness is multiplied by the phase constancy index: accordingly, the grey parts of figure 4 have disappeared.

This decomposition into stages show how the method transforms large lobes (Fourier standard analysis, in figure 3) into an sharp ones (extraction of the demodulated phases with slow evolution, in figure 5). We observe the marked presence of the fundamental components (harmonics 1) and the harmonics 2 of each note. The point in cyan indicates the tuning fork (here, 440Hz).

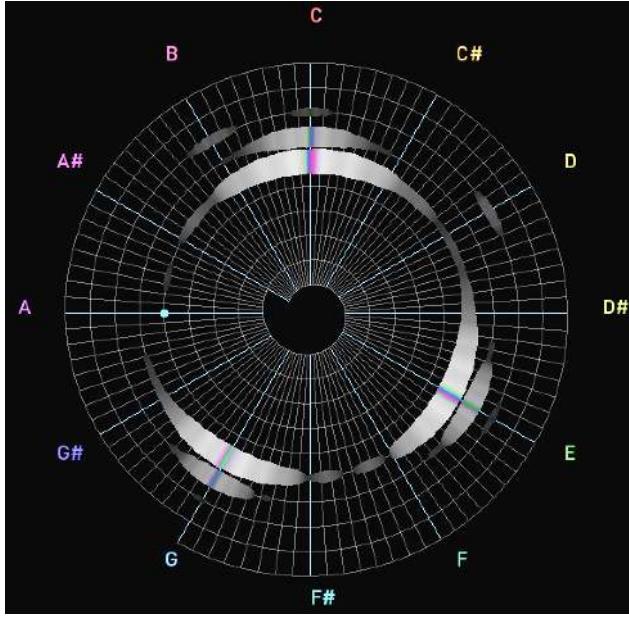


Figure 4: *C* major chord (Fender-Rhodes piano): the line thickness depends on the loudness but is not corrected by the phase constancy index; the color saturation corresponds to the phase constancy index. The grey parts are those to reject.

4. SOFTWARE DEVELOPMENT

4.1. User Interface

The Snail user interface (figure 6), designed by IRCAM and the Upmitt design studio (Paris, France, [22]), is composed of five parts:

1. The main view, which can be split into two views, that allows a secondary analysis display in the same space.
2. A global menu for the basic application operations and audio I/O configuration.
3. A side bar to access and change the most used display and engine properties.
4. An Advanced Settings (sliding) panel with more options to finely configure the analysis engine, the properties of the snail spiral abacus and the sonogram (other options are available but not detailed here).
5. (standalone only) A sound file player with a waveform display to feed the real-time audio engine.

Built around the main view, the interface is configured at startup to show the Snail spiral abacus. This abacus is built using the equal temperament but is not related to the engine. It is just used as a grid to help the eye and could be easily substituted by another grid type. Two additional representations of the analysis may also be displayed, either separately or simultaneously to the spiral abacus:

1. the real-time sonogram view, rendering the snail analysis over time. This sonogram may be used in standard (figure 7) or precise mode (figure 8), the latter exactly leading to the same accurate thickness as on the spiral abacus.

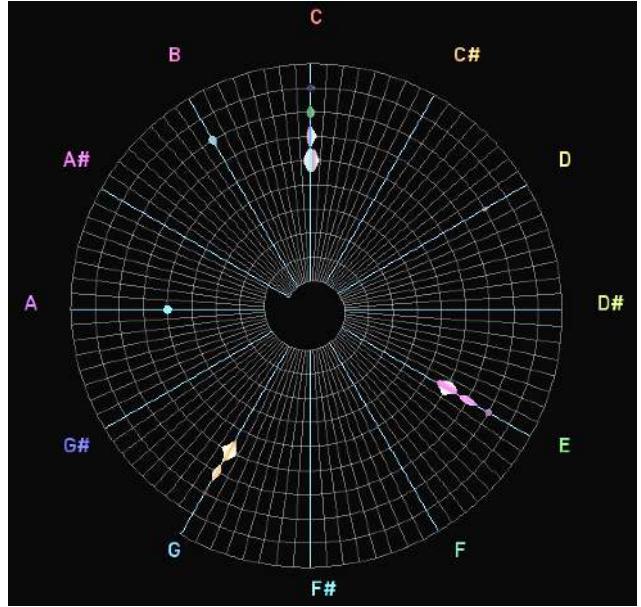


Figure 5: *C* major chord (Fender-Rhodes piano): the line thickness depends on the loudness and is corrected by the phase constancy index. The grey parts of figure 4 are rejected.

2. the tuner view, showing a rectified zoomed region of the spiral (figure 9) and aimed at accurately tuning an instrument when the Snail engine is setup in a Tuner Mode for high precision (figure 10).

Two modes are available in the side bar panel as convenient pre-configurations of the snail engine for two different purposes :

1. the Music mode, aimed at musical visualization, presents a relaxed analysis suitable for the visual tracking of more evolving sounds, like a polyphonic piece of music.
2. The Tuner Mode, a more refined analysis configuration, aimed at the precise visualization of stationary components and the tuning of instruments.

In addition, in order to reflect visually the demodulated phase, an hexagonal spinning shape is drawn above the Tuner View (figure 10) or at the center of the spiral (Snail view). Its angular speed and its color changing both indicate "how close" or "how far" the frequency is from the selected target frequency.

For the user convenience, a "F0 detection" activation switch is also available in the side bar. When set in a Tuner mode configuration, the interface centers the tuner view to the detected fundamental frequency.

Other properties available in the application are the Tuning Reference Frequency, Grid range for the abacus, Visual gain to enhance visually the input signal, and the various color modes, that allows the user to plot only specific parts of the analyzed signal, like the magnitude or the demodulated phase.

For the future version, we plan to integrate the sharable "scala" scale file format for users who want to create and use customized grids based on a tonality different than the equal temperament.



Figure 6: The Snail User Interface (IRCAM/Upmitt). The Settings panel (normally hidden at startup) is shown visible here (users can switch its visibility on/off).



Figure 8: Sonogram in its precise mode: compared to figure 7, only the (colored) refined part are displayed. This mode exactly mirrors the visual representation in the snail display.



Figure 7: Sonogram in its standard fft-mode: the brightness is related to the energy (Loudness scale). In this display mode, the central parts of the thick lines are colored according the frequency precision refinement based on the demodulated phases.

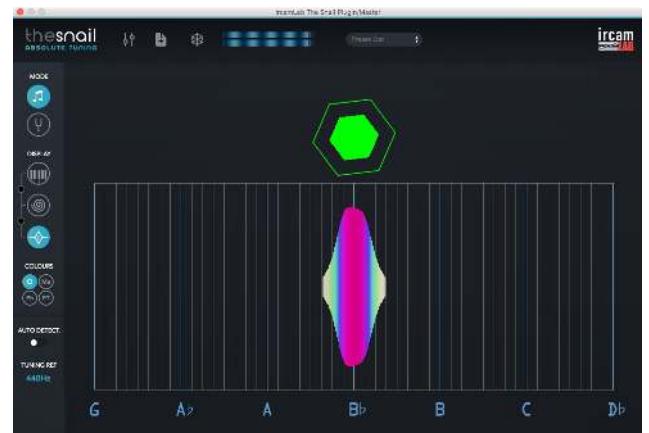


Figure 9: The Snail tuner view, showing the rectified analysis region with the hexagonal shape on top of it. The Music mode analysis is "on" so that the frequency lobe has not the sharpest size, indicating a more relaxed analysis.

4.2. Software structure

The Snail real-time application workload is decomposed into:

An Analysis Task performed most of the time directly in the real-time audio thread,

A Visualisation Task usually performed in the main application thread: it may be split, with a dedicated rendering thread for the OpenGL specific drawings depending on the development environment/framework (e.g. the JUCE Framework [23]). We leave it as an internal implementation detail.

The Analysis process (see figure 11①) is in charge of all the required treatments for the signal spectral analysis, including the production of the demodulated phase and the phase constancy index.

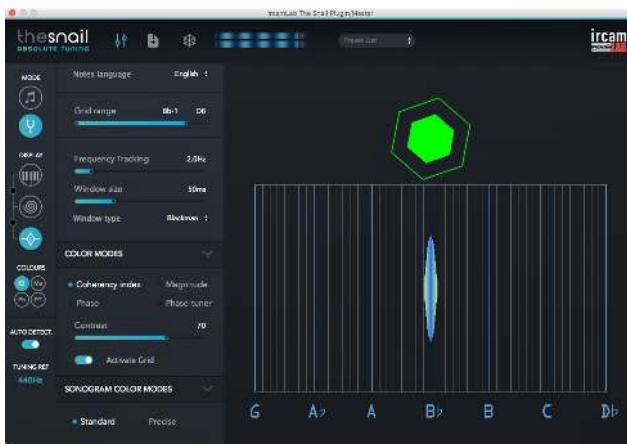


Figure 10: Same sound and component but with the Tuner mode "on" for a more refined analysis: the size the lobe is sharper and is more adapted to a tuning application). The green color of the spinner still indicates that the current frequency is "in tune".

The Display process (see figure 11⑤) is responsible for the conversion of the spectral output frames to the appropriate geometry and colors for the final displays (sonogram and snail), both rendered using OpenGL [24].

In order to communicate the analysis frames produced to the display task/thread in real-time, a shared FIFO queue (implemented as a Lock-Free FIFO, see figure 11⑥) is used and previously allocated with a sufficient amount of space to store the minimal amount of required frames expected for a fluid communication.

4.3. Platforms

The first prototype (research application) of The Snail was developed using the OpenFrameworks library [25] for both standalone application and a first mobile version (thanks to the OpenFrameworks iOS addon [26]).

The final application has been then converted and developed under the JUCE framework [23] in order to simplify the deployment in a standalone and several plugin formats. It is now released as a Standalone application and various plugin formats including VST [27], AudioUnit [28] and AAX [29] for both Mac and PC.

An iOS version [30] (iPhone only for now) is also available, which only offers the tuner display mode (the sonogram is not available as we restricted the mobile usage to a tuner only).

4.4. Examples of use cases

From the tuner perspective, the snail may serve as a high precision tool, with a configurable visual representation of the musical grid. The musician can tune his instrument as a usual tuner, but he may also use it on sounds without a clear pitch, like bells or inharmonic sounds. As the engine does not interpret the incoming sound and does not need the F0 information to adapt its grid on (although it is still possible to do it in the software), the user can focus on particular note (or frequency) and decide accordingly then how to "tune" its sound (let it be on the second harmonic if we wish too). No assumptions are made on how he should proceed, but everything relies on the interpretation of the precise visual feedback. That, by nature, will extend its usability and won't restrict "The Snail" to a specific set of instruments or sounds.

From a visualizer perspective, the analysis of the snail may also serve the musician, the sound engineer or even the *sound enthusiast* to see how the sound is structured on a musically-relevant abacus. The singer can visualize the harmonics, produced in real-time. The musician can see how his interpretation may clearly affect the produced timbre, still readable at a "note level" too. An interaction takes place as the user changes its "sound production" approach using the visual feedback given by the tool. As opposed to the spectrogram, which may be more relevant to see the global spectral balance of a sound, but which is not appropriate to spot a very specific note, "The Snail" is precise enough to make the user understand what is happening from a "note" perspective and so to spot them.

5. CONCLUSIONS

This paper has presented a real-time application that displays spectral information on a spiral representation such that tones are organized with respect to angles. To reach a frequency accuracy that is usable in a musical context (tuning tasks, work on intonation by instrumentists or singers, etc), the novelty is to complement the standard Fourier analysis by a process before displaying the spectral information. This process applies a contracting contrast factor on the magnitude, that only selects the bins for which the spectrum phase rotates at the bin frequency, in an adjustable tolerance range. Typically, if the maximal tolerance frequency deviation $\pm F_c$ is such that $F_c < 2 \text{ Hz}$, this results in a very precise tool for tuning tasks, that is robust in noisy environment (non stationary partials being rejected by the process). For $F_c \approx 6 \text{ Hz}$, the tool is adapted to work on intonation or music visualization. The real-time application has been conceived to run on several platforms (desktop and mobile) and operating systems.

In practice, the Snail has been presented and tested in several musical contexts: (1) with the Choir of Sorbonne Universités, (2) with violinists [31], (3) with piano tuners and manufacturers [32], etc.

Based on their reactions, future possible development is taken into account:

- the modification of the abacus (temperaments, micro-tones, etc.) by the user (e.g. scala format [33]);
- representing the first harmonics in the zoom mode (see figures 9 and 10);
- implementing a new index built on the demodulated phase (included in the patent but not yet implemented) allowing the handling of vibrato, glissando or frequency variations still with a high frequency precision.

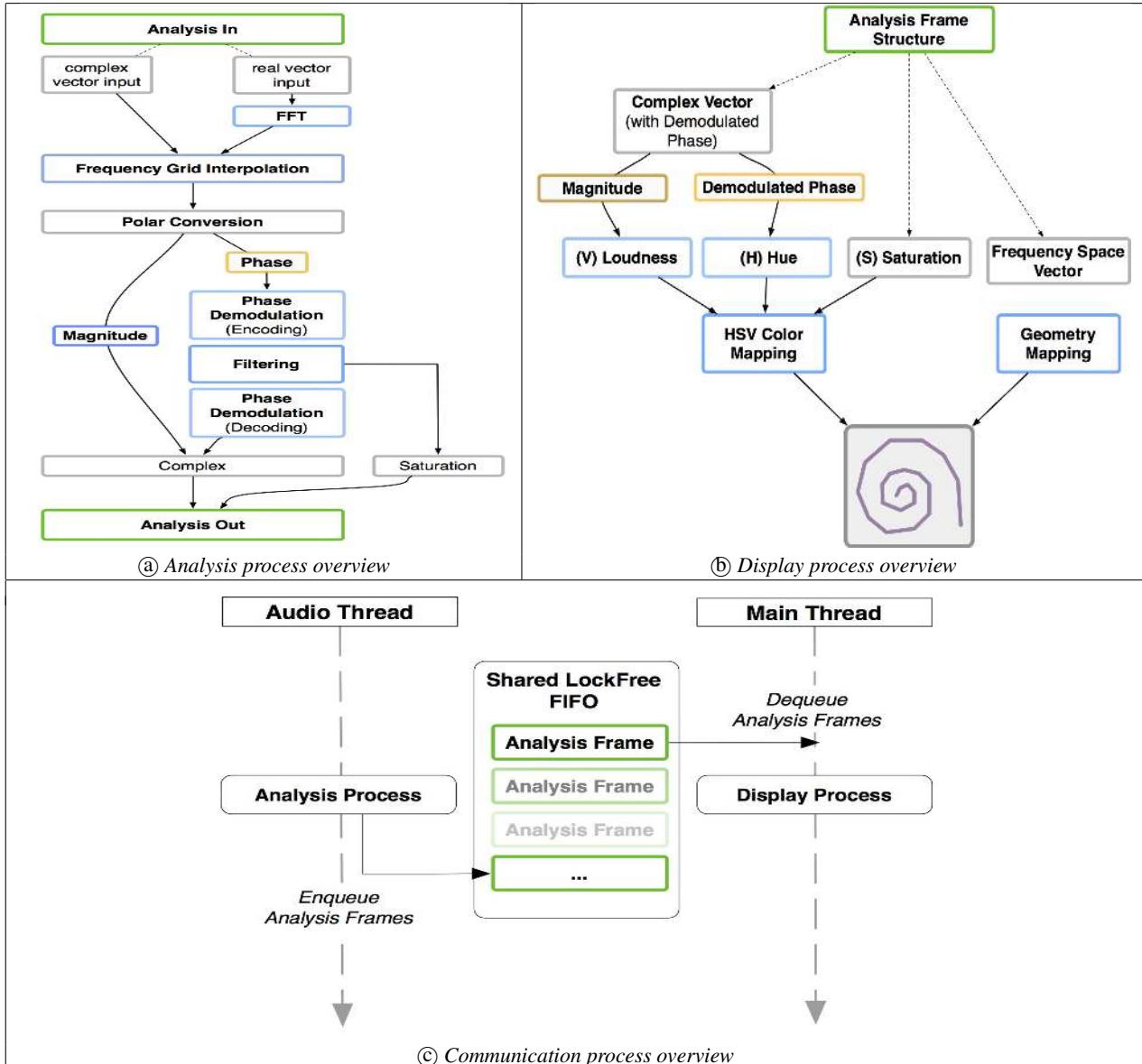


Figure 11: Overviews of the Snail processes : ① Analysis process, ② Display process, ③ Communication process.

6. REFERENCES

- [1] A.G. Storaasli, “Spiral audio spectrum display system,” 1992, US Patent 5,127,056.
- [2] Michel Rouzic, “Spiral Software Application,” <http://photosounder.com/spiral/>, 2013.
- [3] N. Spier, “SpectratunePlus: Music Spectrogram Software,” <http://nastechservices.com/Spectratune.php>, Access: 1 April 2016.
- [4] S. D. Freeman, *Exploring visual representation of sound in computer music software through programming and composition*, Ph.D. thesis, University of Huddersfield, 2013, <http://phd.samuelfreeman.me.uk>.

- [5] V. Lostanlen and S. Mallat, “Wavelet Scattering on the Pitch Spiral,” in *International Conference on Digital Audio Effects (DAFx)*, 2015, vol. 18, pp. 429–432.
- [6] G. Peeters, “Musical key estimation of audio signal based on hmm modeling of chroma vectors,” in *DAFx (International Conference on Digital Audio Effects)*, Montréal, Canada, 2006.
- [7] M. Mehnert, G. Gatzsche, D. Arndt, and T. Zhao, “Circular pitch space based chord analysis,” in *Music Information Retrieval Exchange*, 2008, <http://www.musicir.org/mirex/2008>.
- [8] R. N. Shepard, *Approximation to Uniform Gradients of Generalization by Monotone Transformations of Scale*, pp. 94–110, Stanford University Press, Stanford, 1965.

- [9] Elaine Chew, *Towards a Mathematical Model for Tonal-ity*, Ph.D. thesis, Massachusetts Institute of Technology, MA, USA, 2000.
- [10] L. Cohen, *Time-Frequency Analysis*, Prentice-Hall, New York, 1995, ISBN 978-0135945322.
- [11] J.C. Brown and M.S. Puckette, “An efficient algorithm for the calculation of a constant Q transform,” *JASA*, vol. 92, no. 5, pp. 2698–2701, 1992.
- [12] S. Mallat, *A Wavelet Tour of Signal Processing*, Academic Press, 3rd edition edition, 2008.
- [13] T. Hélie, *Modélisation physique d'instruments de musique et de la voix: systèmes dynamiques, problèmes directs et inverses*, Habilitation à diriger des recherches, Université Pierre et Marie Curie, 2013.
- [14] F. Auger and P. Flandrin, “Improving the readability of time-frequency and time-scale representations by the reassignment method,” *IEEE Transactions on Signal Processing*, vol. 43, no. 5, pp. 1068–1089, 1995, doi:10.1109/78.382394.
- [15] P. Flandrin, F. Auger, and E. Chassande-Mottin, *Time-frequency reassignment: From principles to algorithms*, chapter 5, pp. 179–203, Applications in Time-Frequency Signal Processing, CRC Press, 2003.
- [16] S. Marchand, “Improving Spectral Analysis Precision with an Enhanced Phase Vocoder using Signal Derivatives,” in *Digital Audio Effects (DAFx)*, Barcelona, Spain, 1998, pp. 114–118.
- [17] B. Hamilton and P. Depalle, “A Unified View of Non-Stationary Sinusoidal Parameter Estimation Methods Using Signal Derivatives,” in *IEEE ICASSP*, Kyoto, Japan, 2012, doi: 10.1109/ICASSP.2012.6287893.
- [18] B. Hamilton, P. Depalle, and S. Marchand, “Theoretical and Practical Comparisons of the Reassignment Method and the Derivative Method for the Estimation of the Frequency Slope,” in *IEEE WASPAA*, New Paltz, New York, USA, 2009, pp. 345–348.
- [19] T. Hélie (inventor) and CENTRE NATIONAL DE LA RECHERCHE SCIENTIFIQUE (assignee), “Procédé de traitement de données acoustiques correspondant à un signal enregistré,” French Patent App. FR 3 022 391 A1., 2015 Dec 18, (and Int. Patent App. WO 2015/189419 A1. 2015 Dec 17).
- [20] Website, “Midi association,” <https://www.midi.org/specifications/item/table-1-summary-of-midi-message>.
- [21] Technical Committee ISO/TC 43-Acoustics, “ISO 226:2003, Acoustics - Normal equal-loudness-level contours,” 2003-08.
- [22] Website, “Upmitt,” <https://www.behance.net/upmitt>.
- [23] Website, “JUCE,” <https://www.juce.com>.
- [24] Website, “OpenGL,” <https://www.opengl.org>.
- [25] Website, “OpenFrameworks,” <http://openframeworks.cc>.
- [26] Website, “iOSAddon,” <http://ofxaddons.com/categories/2-ios>.
- [27] Website, “VST (Steinberg),” <https://www.steinberg.net/>.
- [28] Website, “AudioUnit,” <https://developer.apple.com/reference/audiounit>.
- [29] Website, “AAX,” <http://apps.avid.com/aax-portal/>.
- [30] Website, “iOS10,” <http://www.apple.com/ios/ios-10/>.
- [31] T. Hélie and C. Picasso, “The snail: a new way to analyze and visualize sounds,” in *Training School on "Acoustics for violin makers"*, COST Action FP1302, ITEMm, Le Mans, France, 2017.
- [32] T. Hélie, C. Picasso, and André Calvet, “The snail : un nouveau procédé d'analyse et de visualisation du son,” *Pianistik, magazine d'Europiano France*, vol. 104, pp. 6–16, 2016.
- [33] Website, “Scala Home Page,” <http://www.huygens-fokker.org/scala/>.

BEAT-ALIGNING GUITAR LOOPER

Daniel Rudrich

Institute of Electronic Music and Acoustics,
University of Music and Performing Arts
Graz, Austria
rudrich@iem.at

Alois Sontacchi

Institute of Electronic Music and Acoustics,
University of Music and Performing Arts
Graz, Austria
sontacchi@iem.at

ABSTRACT

Loopers become more and more popular due to their growing features and capabilities, not only in live performances but also as a rehearsal tool. These effect units record a phrase and play it back in a loop. The start and stop positions of the recording are typically the player's start and stop taps on a foot switch. However, if these cues are not entered precisely in time, an annoying, audible gap may occur between the repetitions of the phrase. We propose an algorithm that analyzes the recorded phrase and aligns start and stop positions in order to remove audible gaps. Efficiency, accuracy and robustness are achieved by including the phase information of the onset detection function's STFT within the beat estimation process. Moreover, the proposed algorithm satisfies the response time required for the live application of beat alignment. We show that robustness is achieved for phrases of sparse rhythmic content for which there is still sufficient information to derive underlying beats.

1. INTRODUCTION

Music can never have enough of saying over again what has already been said, not once or twice, but dozens of times; hardly does a section, which consists largely of repetition, come to an end, before the whole story is happily told all over again.

— Victor Zuckerkandl [1]

Repetition is an essential characteristic of music on different time scales. The compositional technique *canon* is solely based on a repeating structure. It elaborates by combining several layers of musical phrases. The technique of looping proceeds the same principles and establishes new perspectives and approaches to create and combine musical ideas. Since the invention of digital recording, the looping approach has been technically available.¹ However, especially for beginners it is often difficult to handle the timing precisely enough when controlling the looper.

Fig. 1a visualizes a looped phrase being repeated seamlessly as start and stop cues are in-time. The musical beats are generally unknown by the system and are depicted for visualization of the problem only. When hitting the stop button too early also the repetition starts too early, leading to a gap of length ΔT between the actual cue and the intended cue (Fig. 1b). Alternatively, a gap also occurs when the stop cue comes too late (Fig. 1c). Both cases also happen if the timing of the start cue is off, or a combination of

¹With the analog predecessor of digital loop pedals, the *Time Lag Accumulator* [2], it was almost impossible to set the loop length while playing. First the use of digital memory instead of tape loops made it possible leading to a substantially different sound and use of live looping. [3]

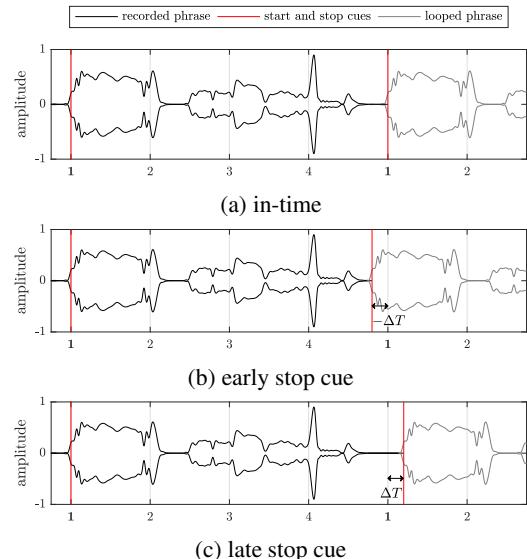


Figure 1: Examples of a one-bar-phrase getting looped with different stop-cue timings. The abscissa holds the musical beats, the black line represents the envelope of the recorded signal, red lines indicate the start and stop cues and the grey line represents the repeated phrase setting in directly after the stop cue.

both. Nevertheless, when both cues have the same offset, the loop gets repeated without any pauses or skips.

These artifacts can be audible and annoying, especially because of their continuous repetition. Some loopers solve this by quantizing the start and stop cues to a predefined click track. Here, the main objective was to develop a looper that estimates the underlying beats from the recorded phrase. In the proposed approach, the derivation of beats and tempo values is done in a final stage after gauging all possible candidates, in order to increase robustness and circumvent the general drawback of bottom-up approaches. As the recorded beat jitters with the accuracy and groove of the player, and the distinctness of transients depends on the style of play, this robustness is essential for a successful beat-alignment.

From the recorded musical phrase, we calculate a sparse energy distribution over time and compare it to given *tatum*² grids. In the following, the term *tatum* is used as a context-free measure of tempo, as it does not require a musical reference value as the musical measure *beats per minute* (bpm) does. The tatum τ embodies the time interval of the underlying grid and is given in seconds.

²Bilmes coined this term in his thesis *Timing is of the Essence* [4], which describes the lowest metrical level in a musical piece as a fine underlying grid to which every note and rest can be assigned to.

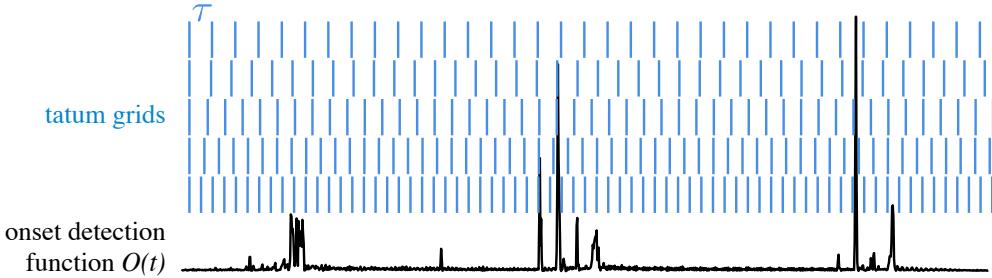


Figure 2: Qualitative illustration of the basic approach: Different tatum grids get placed over the onset detection function $O(t)$. The degree of matching provides information about the presence of different tatus in the signal.

The next section describes the structure of the proposed algorithm with its individual components. The subsequent sections then give a detailed description of these components followed by a performance evaluation of the algorithm including real time capability, perceptual evaluation, and performance examples.

2. OUTLINE OF THE BEAT-ALIGNMENT ALGORITHM

The beat-aligning looper should align the loop's start and stop cues so that no pauses or skips occur when the loop is repeated. The basic idea is to find the tatum of a recorded signal and align the start and stop cues to the corresponding beats. Tatum estimation is done by calculating an onset-detection-function (ODF) to which different tatum-grids are correlated to pick the one that fits best (see Fig. 2). Due to a possible variability of the tempo within the recorded phrase, the tatum estimation shouldn't be done for the entire sequence at once, but rather in blocks.

There is a variety of onset detectors. [5, 6] give a good overview of mainly energy based onset detectors. A more sophisticated approach based on recurrent neural networks is presented in [7]. However, as the audio signal from the guitar is expected to be clean without any effects (e.g. delay, reverb or noise) applied to it, we were able to choose a less complex onset detector. Therefore, the *spectral flux log filtered* approach proposed by Böck et al. [8] is used (Sec. 3). It is an advancement of the spectral flux algorithm by Masri [9]. The main differences are the processing in Mel frequency bands and the usage of the logarithm for the magnitudes to resolve loudness levels appropriately. The latter combined with calculating the difference between two consecutive frames (see Sec. 3.4) leads to an utilization of magnitude ratios instead of differences. This compensates for temporal variability (dynamic) of the audio signal as parts with high amplitudes do not get emphasized in comparison to parts with lower amplitudes.

Additionally, *Adaptive Whitening* suggested by Stowell and Plumley [10] is used to compensate for variability in both time and frequency domain. It normalizes the magnitude of each STFT bin with regards to a preceding maximum value and compensates spectral differences as higher frequencies often have lower energy (spectral roll-off). Without the spectral whitening, lower frequency content tends to dominate over higher frequency content that contains important onset information, as the surprisingly good results of the simple HFC (high frequency content) ODF demonstrate [9].

As a basis of beat estimation, the tempo estimation approach proposed by Wu et al. [11] was employed (Sec. 4). In their proposal, an STFT of the ODF is calculated for a subset of frequen-

cies, extending the tatum grids to complex exponential functions. This results in the tempogram, which holds a measure for the presence of a tatum in each ODF frame. Here, instead of picking the tatum with the highest magnitude for each block, a dynamic programming (DP) approach is used to find the optimum tempo path: a utility function is maximized with a gain for high magnitudes and a penalty for high tatum differences between two consecutive frames. This prevents paths with unlikely large variation in tempo. Different from existing DP approaches for beat estimation like [11, 12], we don't tie up beats to discrete onsets of the recorded signal as those onsets may be subject to the inaccuracy of human motor action. For this application it is advantageous to retrieve a beat grid without possible fluctuations. This is done by using phase information: the phase shift of the complex exponential function's periodicity (Fourier transformation) can be extracted by calculating the phase of the found optimum path. The phase directly measures the location of the beats as a beat occurs every time the phase strides a multiple of 2π . This approach is illustrated in Fig. 3. This leads to an increased precision similar to the instantaneous frequency tracking by [13]. As a result, without the necessity of a finely sampled set of tatum candidates efficiency is increased (see Sec. 5).

The following modifications were made regarding the enhancement of the tempogram and information extraction:

Modified tempogram The tempogram is enhanced by the information of how well the phase of a frame conforms with its expected value, calculated by the time difference between two consecutive frames and the tatum value.

Phase information In contrast to the tatum information of the optimum path, the phase of the path is used. It contains all the information needed to calculate the positions of the beats. Especially if the set of tatum grids is chosen to be coarse for low-cost implementation, the phase can be used to calculate the actual tatum more precisely.

Phase reliability A measure for the phase's reliability is used to spot frames, in which the phase information becomes meaningless. This happens when there are no significant onsets. For unreliable frames the beats will be discarded and replaced by an interpolating beat placement.

As a last step, the start and stop cues obtained by pressing a foot switch are aligned to the nearest estimated beat. There are no additional rules for the beat-alignment (e.g. loop length in beats has to be a multiple of 4) to keep enough responsibility for phrasing to the player.

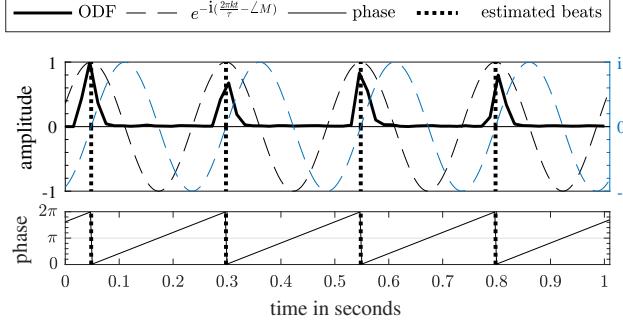


Figure 3: Visualization of the phase-based beat estimation. The cosine and sine components of the Fourier expansion for a tatum τ (dashed line) are shown in the graph on top. The Fourier transformation of the ODF yields a complex-valued tempogram value M , whose phase (bottom) carries the desired time-shift information of the corresponding tatum grid.

3. ODF: SPECTRAL FLUX LOG FILTERED

The *spectral flux log filtered* onset detection function consists of a block-wise transformation into the frequency domain followed by adaptive whitening. The signal then runs through a filter bank consisting of 50 Mel frequency bands. Last, after taking the logarithm, the temporal differences are summed up to obtain the ODF.

3.1. Short Time Fourier Transform (STFT)

Prior to the transformation into the frequency domain the signal $x(t)$ is blocked into N overlapping frames with a length of $K = 2048$ samples and the hopsize $h = 480$ samples, resulting in an overlap of roughly 77 %. With $f_s = 48\,000$ Hz subsequent frames are 10 ms apart (resulting in an ODF sampling frequency of $f_{s,O} = 100$ Hz). Each frame is windowed with a Hann window $w(t)$. The windowed signals $x_n(t)$ can be calculated with

$$x_n(t) = x(t + nh)w(t), \quad n \in [0, N - 1], \quad t \in [0, K - 1]. \quad (1)$$

Each frame is transformed into the frequency domain using the discrete Fourier transform:

$$X(n, k) = \sum_{t=0}^{K-1} x_n(t) \cdot e^{-i \frac{2\pi k t}{K}}, \quad k \in [0, K - 1] \quad (2)$$

with n denoting the frame, k the frequency bin, and t the discrete-time index.

3.2. Adaptive Whitening

The *peak spectral profile*

$$P(n, k) = \begin{cases} \max\{|X(n, k)|, r\} & \text{for } n=0, \\ \max\{|X(n, k)|, r, \mu P(n-1, k)\} & \text{otherwise} \end{cases} \quad (3)$$

determines the spectral whitening in the division

$$X(n, k) \leftarrow \frac{X(n, k)}{P(n, k)}, \quad (4)$$

in which μ is the forgetting factor and r the floor parameter [10]. The choice $\mu = 0.997$ and $r = 0.6$ has proved suitable for the application.

3.3. Logarithmic-Magnitude Mel Spectrogram

The main difference to the regular spectral flux onset detection is the analysis in sub-bands. Consequently, the magnitude bins of the spectrogram $|X(n, k)|$ are gathered by filter windows $F(k, b)$ with $B = 50$ overlapping filters with center frequencies from 94 Hz to 15 375 Hz. Each window has the same width on the Mel-scale. The windows are not normalized to constant energy, which yields an emphasis on higher frequencies. The Mel spectrogram $X_{\text{filt}}(n, b)$ is given by:

$$X_{\text{filt}}(n, b) = \sum_{k=0}^{K-1} |X(n, k)| \cdot F(k, b), \quad b \in [0, B - 1] \quad (5)$$

with b denoting the sub-band number. The logarithmic-magnitude Mel spectrogram is obtained by applying the logarithm to the Mel spectrogram

$$X_{\text{filt}}^{\log}(n, b) = \log(\lambda \cdot X_{\text{filt}}(n, b) + 1) \quad (6)$$

with the compression parameter λ . A value of $\lambda = 2$ yielded good results. The additive term $+1$ assures only positive values.

3.4. Difference

The final step to derive the onset detection function $O(n)$ is to calculate the difference between the frame n and its previous frame $n - 1$, with a subsequent summation of all spectral bins. The half-wave rectifier function $H(x) = \frac{x+|x|}{2}$ ensures that only onsets are considered. Altogether, we obtain the following equation:

$$O(n) = \sum_{b=0}^{B-1} H(X_{\text{filt}}^{\log}(n, b) - X_{\text{filt}}^{\log}(n - 1, b)). \quad (7)$$

Fig. 4 depicts the above-mentioned steps for the calculation of the onset detection function.

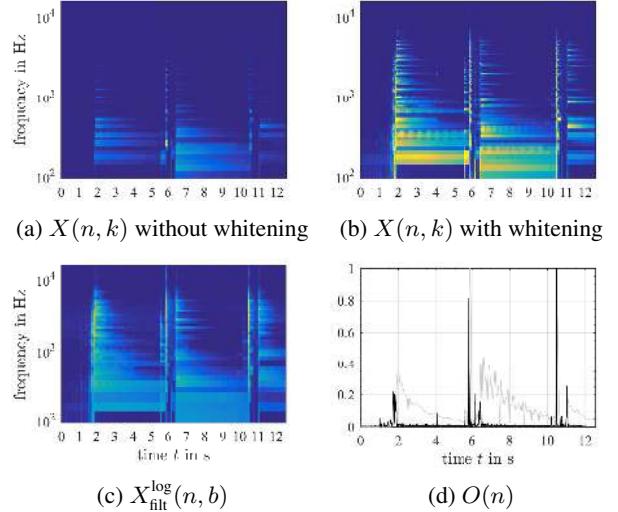


Figure 4: Steps of calculation the *spectral flux log filtered* ODF: a) spectrogram b) adaptive whitening c) log-magnitude Mel spectrogram d) final ODF (black) and the audio signal (gray). The data of all four figures is normalized to a maximum value of 1.

4. BEAT ESTIMATION AND START/STOP ALIGNMENT

Out of the derived onset detection function the tempogram is calculated. A dynamic programming approach computes the most likely tatum path within the tempogram. Afterwards, beats are estimated by extracting the optimum path's phase information and a possible subsequent interpolation of areas with non-reliable phase information.

4.1. Tempogram

As described, the tempogram $M(j, \tau)$ is obtained by an STFT of $O(n)$, evaluated for specific frequencies corresponding to a set of tatus. This can be expressed as:

$$M(j, \tau) = \sum_{n=0}^{L-1} O(n+j, \tau) w(n) e^{-i\frac{2\pi n}{\tau f_{s,O}}}, \quad j \in [0, J-1] \quad (8)$$

with j being the ODF frame index, L the ODF frame length, $w(n)$ the Hann window function, $f_{s,O}$ denoting the ODF sampling frequency (here 100 Hz) and $\tau \in T$ denoting the tatum out of a set T with different tatum values between 60 ms and 430 ms. An ODF frame length of $L = 150$ yielded good results, meaning that one tempogram value represents a 1.5 s frame and the beginning of one frame is $\frac{1}{f_{s,O}} = 10$ ms apart from the beginning of its predecessor. The total number of time-steps j can be expressed as $J = N - L + 1$.

To emphasize on phase continuity the phase difference between two consecutive frames $d\Phi(j, \tau)$ is calculated and compared to the expected value $\hat{d}\phi(\tau)$. The difference of both results into the phase deviation matrix:

$$\Delta\Phi(j, \tau) = d\Phi(j, \tau) - \hat{d}\phi(\tau) \quad (9)$$

with

$$d\Phi(j, \tau) = \angle M(j, \tau) - \angle M(j-1, \tau) \quad \text{and} \quad (10)$$

$$\hat{d}\phi(\tau) = \frac{2\pi}{\tau f_{s,O}}. \quad (11)$$

$$\Delta\Phi_{\text{mapped}} = ((\frac{\Delta\Phi}{\pi} + 1) \bmod 2) - 1 \quad (12)$$

maps the values to a range of $[-1, 1]$, with a value of 0 indicating a perfect phase deviation of 0. The modified tempogram $M'(j, \tau)$ gets calculated as follows:

$$M'(j, \tau) = M(j, \tau) \cdot (1 - |\Delta\Phi_{\text{mapped}}(j, \tau)|)^{\kappa} \quad (13)$$

whereas κ denotes the degree of factoring in the phase conformance. A value of $\kappa = 100$ was ascertained experimentally and suited well for this application. Fig. 5 shows the sharpening of the tempogram due to this modification. The used signal is a hi-hat sequence with leaps in tempo. With κ the amount of sharpening can be adjusted.³

As a last step, $M'(j, \tau)$ gets normalized to a maximum absolute value of 1:

$$M'(j, \tau) \leftarrow \frac{M'(j, \tau)}{\max_{j, \tau} |M'(j, \tau)|}. \quad (14)$$

³In general, for a coarsely sampled set of tatus a lower κ value should be chosen. Otherwise, the phase nonconformance as a consequence of a non-sampled tatum would lead to a falsification of the tempogram.

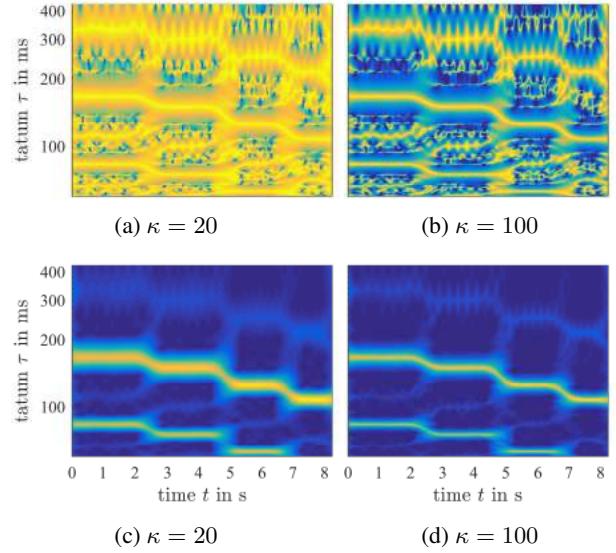


Figure 5: Effect of different κ values on the modified tempogram. The range of the depicted values is between 0 and 1, with dark blue representing 0, yellow representing a value of 1. a) and b) show $(1 - |\Delta\Phi_{\text{mapped}}(j, \tau)|)^{\kappa}$, c) and d) show the modified tempograms $M'(j, \tau)$.

4.2. Optimum Tatum Path

The optimum tatum path can be extracted out of the modified tempogram by maximizing the utility function $U(\tau, \theta)$. This function is designed in order that high absolute tempogram values $|M'(j, \tau)|$ (tatum conformance) are advantaged and high leaps in tempo/tatum will result in a penalty (second term in equation (15)). The goal is to find a series of tatum values $\tau = [\tau_0, \dots, \tau_j, \dots, \tau_{J-1}]$, with τ_j the tatum value for ODF frame j , which maximizes the utility function

$$U(\tau, \theta) = \sum_{j=0}^{J-1} |M'(j, \tau_j)| - \theta \sum_{j=1}^{J-1} \left| \frac{1}{\tau_{j-1}} - \frac{1}{\tau_j} \right|, \quad (15)$$

with θ denoting the penalty factor for a tatum difference between two consecutive frames. With $\theta = 0$ the maximization could be replaced by picking the tatum with the highest absolute tempogram value. The higher θ the smoother the tempo path due to a higher penalty for tempo changes. A value of $\theta = 20$ suited well for this application.

The search for the maximum of the utility function can be done efficiently with dynamic programming. Therefore, the maximum can be written as:

$$\max_{\tau} U(\tau, \theta) = \max_{\tau} D(J-1, \tau) \quad (16)$$

with the recurrent equation

$$D(j, \tau) = \begin{cases} |M'(0, \tau)| & \text{if } j=0, \\ |M'(j, \tau)| + \max_{\tau_{j-1}} (D(j-1, \tau) - \theta \left| \frac{1}{\tau_{j-1}} - \frac{1}{\tau_j} \right|) & \text{otherwise} \end{cases} \quad (17)$$

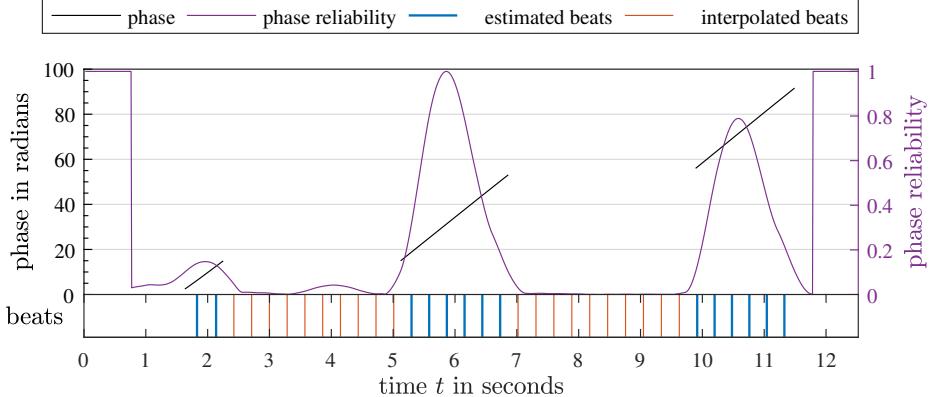


Figure 6: Example of filling gaps with low phase reliability. The black line represents the phase with gaps as the phase reliability (purple) drops below 0.1. The estimated beats (blue) are gathered with the phase information, the interpolated beats (orange) are filled in by interpolation. The impression of the phase keeping its value right before and after a gap is an artifact of phase-unwrapping.

Basically, after initialization the first frame $j = 0$, for every tatum τ_j of the frame j the algorithm looks for that tatum τ_{j-1} of the previous frame which yields the most rewarding transition $\tau_{j-1} \rightarrow \tau_j$. With memorizing $\tau_{j-1,\max}$ for every $D(j, \tau)$, the optimum path can be found by backtracking $\tau_{j-1,\max}$ starting with the tatum $\arg \max_{\tau} D(J-1, \tau)$ of the last frame.

The optimum path extracted for the previous shown tempogram is depicted in Fig. 7. The path (red line) follows the leaps in tempo.

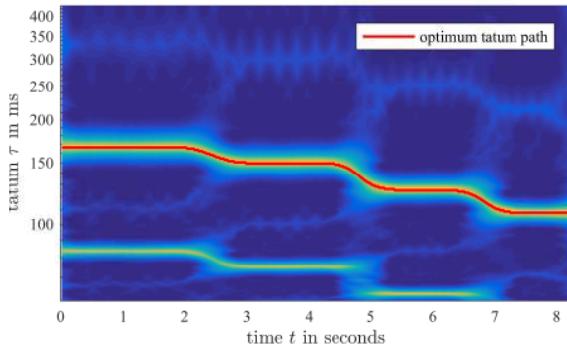


Figure 7: Resulting optimum tatum path for a hi-hat signal with leaps in tempo.

4.3. Beat Placement and Start/Stop Alignment

As described above, the beat placement uses the phase $\phi(n)$ of the optimum tatum path $\boldsymbol{\tau} = [\tau_0, \dots, \tau_j, \dots, \tau_{J-1}]$. The phase can be obtained from the tempogram $M(j, \tau_j)$ by calculating the angle of the complex values. Also the modified tempogram $M'(j, \tau)$ can be used here, as it holds the same phase information. To find a phase value for every time step n of the ODF, the tempogram time steps j have to be mapped to n :

$$j \rightarrow n = j + \frac{L}{2}. \quad (18)$$

The offset of $\frac{L}{2}$ issues from the phase information being calculated for a frame with length L (see equation (8)). So the center of each frame was chosen for the mapping. Nevertheless, the phase information is valid for the beginning of each frame, therefore, the phase itself also has to be adjusted to the frame center by the expected amount $\frac{L}{2}\hat{d}\phi(\tau)$. So the extraction of the phase can be formulated as follows:

$$\phi(n) = \phi(j + \frac{L}{2}) = \angle M(j, \tau_j) + \frac{L}{2}\hat{d}\phi(\tau_j), \text{ for } j \in [0, J-1]. \quad (19)$$

The remaining phase information for $n < \frac{L}{2}$ and $n \geq J + \frac{L}{2} = N - \frac{L}{2} + 1$ has to be derived by extrapolation. The phase then can be used to place beats to the ODF. A beat occurs every time the phase strides a multiple of 2π . This search is equal to the calculation of positive⁴ zero crossings of $\sin(\phi(n))$.

With the phase difference between two consecutive frames $d\phi(n)$ yielding the current phase step, the current tatum can be calculated analogous to equation (11):

$$\tau(n) = \frac{2\pi}{d\phi(n)f_{s,O}}. \quad (20)$$

The $\tau(n)$ values are not bound to those of the tatum set T and, as a consequence, are sampled with a higher precision.⁵ Averaging these values results into the mean tatum $\bar{\tau}$, which can be used for the phase extrapolation and interpolation of beat gaps (described hereafter).

Additionally to the angle of the tempogram, the magnitude is used as a measure of phase reliability. If the magnitude is lower than a predefined threshold, the phase information is considered meaningless. Low tempogram magnitudes can occur in frames with only few or no onsets. In that case, the phase information gets discarded and the resulting gaps get filled with equally spaced beats corresponding to the mean tatum. An example of interpolated beats is shown in Fig. 6.

The last step is to align the start and stop cues to the estimated beat positions. This is easily done by shifting each cue the closest beat.

⁴transition from negative to positive values

⁵see the example in Section 5.2 for demonstration

5. PERFORMANCE EVALUATION

This section shows exemplarily how the algorithm reacts to different signals. Also the performance concerning the real time capability is investigated. Additionally, the perceptibility of the remaining gaps is treated here.

5.1. Beat Interpolation of Low Magnitude Gaps

Fig. 8 shows ODF and tempogram of a recorded phrase with sustaining chords. Hence, only a few onsets exist at around $t = 5$ s and 10 s as both ODF and tempogram reveal. Even the start of the phrase does not hold any useful tatum information. Nevertheless, the algorithm found the optimum path, which fits the audio signal best. Due to the vanishing magnitude of the tempogram between $t = 2$ s and 4 s and between $t = 6$ s and 9 s, the phase reliability measure is not high enough to place reliable beats. As a consequence, two gaps emerge after discarding unreliable beats, which are filled with interpolated beats, as shown before in Fig. 6.

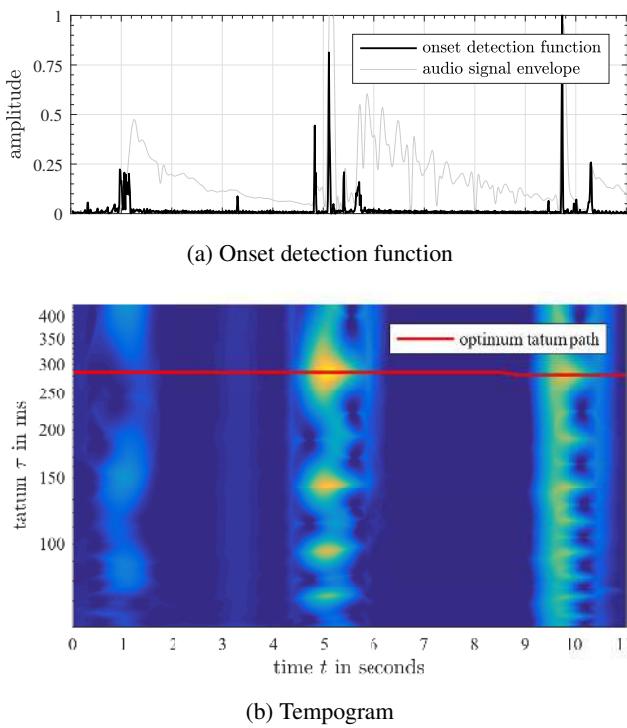


Figure 8: Onset detection function (a) and tempogram (b) of a recorded phrase with only a few onsets.

5.2. High Tatum Precision despite Coarse Sampling

To demonstrate the higher tatum precision than the sampled tatum grids due to factoring in phase information, a semiquaver hi-hat signal at tempo 93 bpm is used. The resulting tatum is $\tau_0 = 161.29$ ms. The used tatum grids of the tempogram stage are sampled with a coarse resolution: the examined tatum values next to τ_0 are 155.01 ms and 165.89 ms. The corresponding tempogram is depicted in Fig. 9.

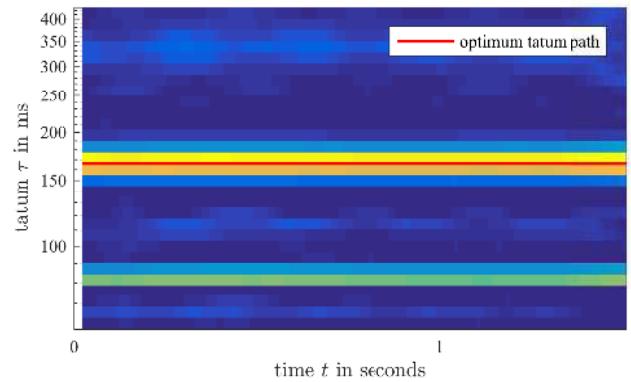


Figure 9: Tempogram with low tatum sampling.

The optimum path search yielded a constant tatum of $\tau = 165.89$ ms, as being closest to τ_0 . However, the phase information yielded an average tatum of $\bar{\tau} = 161.40$ ms, which is remarkably closer, with a difference of just 0.11 ms instead of 4.6 ms.

5.3. Greatest Common Divisor (GCD) of Existing Tatus

The algorithm tries to find that tatum which fits best to all occurring tatus. This mechanism is demonstrated with a signal consisting of alternating blocks of quaver notes and quaver triplets. At tempo 100 bpm, the time difference between two successive quaver notes is $\tau_1 = 300$ ms and for quaver triplets $\tau_2 = 200$ ms, respectively. As expected, these tatum values also occur in the tempogram with a high presence measure (see Fig. 10). However, the optimum tatum path was found for a tatum τ_3 , which does not occur explicitly, but is implied by the two tatus τ_1 and τ_2 by being the greatest common divisor $\tau_3 = \text{gcd}(\tau_1, \tau_2) = 100$ ms. All occurring events can be assigned to beats placed with that tatum.

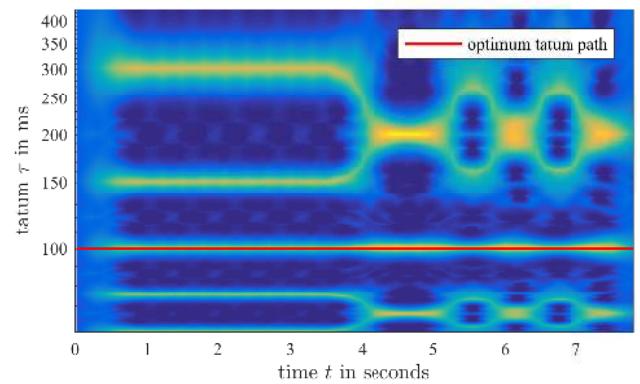


Figure 10: Effects of complex rhythms on the algorithm's performance. Optimum tatum path (red) does not follow the explicitly occurring tatus (150 ms and 300 ms), but their greatest common divisor (100 ms).

Table 1: Results of time profiling of two audio files with different durations and different number of tatum n_τ . The algorithm was implemented in Matlab and executed on a 2.3 GHz Intel Core i7 processor. ODF: onset detection function; TG: tempogram calculation; path: optimum path search; BE: beat estimation

duration (in s)	n_τ	computation time (in % of duration)				
		ODF	TG	path	BE	overall
9.3	60	1.76	0.08	2.51	0.10	4.45
9.3	120	1.83	0.16	5.93	0.14	8.06
9.3	240	1.76	0.26	16.48	0.14	18.64
22.2	60	1.02	0.09	2.85	0.07	4.03
22.2	120	1.02	0.17	6.80	0.06	8.05
22.2	240	1.00	0.23	17.02	0.07	18.32

5.4. Real Time Capability

The above described algorithm only embodies a useful tool for live looping if it is real time capable. At a first glance this means, that all the calculations have to be completed when the musician stops the recording and expects the phrase being repeated seamlessly. This actually is not possible as the back tracking process of the optimum tatum path estimation cannot start until the last frame, in which the stop cue occurs, was processed.

Fortunately, by accepting a possibly perceivable gap at the first iteration, real time capability can be easily achieved: now the algorithm has to be completed within the first unaltered iteration of the phrase, which usually has a duration between 3 s to 10 s.

Table 1 shows the needed time for computations in percentage of the duration of two different audio signals. The data shows that the algorithm’s performance in regards of computation time depends strongly on the number of evaluated tatum n_τ as the computing effort of the tempogram increases linearly and that of the optimum path search quadratically with n_τ . The rather constant relative overall time shows a linear relationship with the signal’s duration. Note that these results are gathered by a single execution of the algorithm for each combination of audio signal and number of tatum and, therefore, may be affected by different CPU loads. Also these results were gathered in an offline version of the algorithm. By computing the ODF and parts of the tempogram and the optimum path search during the recording phase, these values can even be lowered. It can therefore be concluded that the algorithm is real time capable and only needs a fraction of the recorded signal’s duration for computation.

5.5. Perceptibility of Remaining Gaps

Hibi [14] and Rudrich [15] investigated the perception of rhythmic perturbances. Latter found a dependency on musical training onto the audibility of rhythm perturbances in a monotonic, isochronous sequence. The found threshold of subjects with musical background was remarkably lower with 5.4 % of the inter onset interval than that of subjects without musical training (9.0 %). We used these thresholds to validate the algorithm’s performance regarding the reduction of perceivable gaps induced due bad timing of the start and stop cues. Actually, for a seamless performance these cues do not have to sit perfectly on the same musical measure. It is sufficient when both cues have the same offset to an arbitrary measure. As a consequence, only the difference of both offsets is used as a measure for the gap.

Table 2: Results of validation of the algorithm. Values are given in ms and show the resulting gaps introduced by the algorithm. Marked (*) $L_{16\text{th}}$ levels indicate the algorithm finding the tatum for quaver notes (300 ms), otherwise the tatum of semiquavers (150 ms) is found.

σ 1/f noise STD in samples / ms	$L_{16\text{th}}$ in dB				
	0	-10	-20	-30*	$-\infty^*$
0 / 0	0.23	0.29	0.44	1.92	1.88
100 / 2.08	0.17	0.15	0.42	0.10	1.54
200 / 4.17	-0.56	1.02	2.96	-0.75	1.23
300 / 6.25	-3.40	-1.73	-2.06	2.65	0.23
400 / 8.33	-6.31	1.21	0.85	-1.56	-4.35
500 / 10.42	5.02	4.25	1.00	0.73	-2.44

To validate the algorithm in an analytical and ecological valid way, a synthetic drum beat was used due to its advantages:

- easy reproducibility
- control of parameters (rhythm fluctuation and presence of the tatum)
- start and stop cues are easy to determine
- musical context.

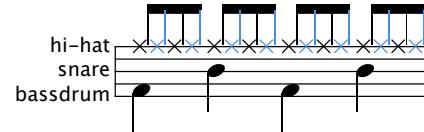


Figure 11: Score of the synthetic drum beat used for evaluation.

The drum beat’s score is depicted in Fig. 11. The *tatum presence* was adjusted with the $L_{16\text{th}}$ level, representing the level of every other hi-hat note (highlighted blue). *Rhythm Fluctuation* is realized with an inserted 1/f noise jitter with standard derivation values between 0 ms and 10 ms.

For different combinations of the above mentioned parameters, the algorithm processes audio files with pre-defined and perfectly timed start and stop cues. The alteration of these cues leads to a gap introduced by the algorithm, which serves as a measure for performance as it shows how well the beats can be aligned at best.

The validation’s results are depicted in Table 2. The data shows the gap introduced by the algorithm. The smaller the values, the better the algorithm’s performance. Negative values indicate a negative gap, meaning the interval was shortened (otherwise lengthened). For $L_{16\text{th}} = -30$ dB and $L_{16\text{th}} = -\infty$ dB the tatum found was 300 ms corresponding to the quaver notes in the drum-beat. As can be seen for no introduced jitter ($\sigma = 0$) the algorithm creates a gap, especially when the tatum doubles. Nevertheless, when comparing to the results of Hibi [14] and Rudrich [15] all gaps are smaller than the found thresholds of perception.

6. CONCLUSIONS AND OUTLOOK

We proposed a real-time capable beat-aligning guitar looper. It is mainly characterized by its ability to support the temporally accurate handling of the looper without any further prior information and additional constraints. Start and stop cue alignment is done automatically during the first repetition of the recorded phrase. Independent of tempo, this adjustment guarantees that the resulting gap stays within rhythmic inaudibility. For traceability, this was evaluated within an ecologically valid setup.

It is obvious that the computation time of the described algorithm is dominated by the optimum path search algorithm. Informal tests showed, that a divide and conquer approach reduces the complexity and computation time. In this approach the optimum path search combines the output of separately analyzed tatum octave sets, instead of the entire tatum set.

As the beat-aligning looper retrieves all beats of the recorded phrase, it is a promising basis for further development of an automatic accompaniment (e.g. rhythm section) for practice or live-performance purposes.

A detailed view on the conducted listening tests and the evaluation of the algorithm can be found in [15].

7. REFERENCES

- [1] V. Zuckerkandl, “Sound and Symbol,” Princeton University Press, 1969.
- [2] T. Baumgärtel, *Schleifen: zur Geschichte und Ästhetik des Loops*, Kulturverlag Kadmos, Berlin, 2015.
- [3] M. Grob, “Live Looping - Growth due to limitations,” http://www.livelooping.org/history_concepts/theory/growth-along-the-limitations-of-the-tools/, 2009, Last accessed on Jan 11, 2017.
- [4] J. Bilmes, *Timing is of the Essence: Perceptual and Computational Techniques for Representing, Learning, and Reproducing Expressive Timing in Percussive Rhythm*, Ph.D. thesis, Massachusetts Institute of Technology, Program in Media Arts & Sciences, 1993.
- [5] S. Dixon, “Onset detection revisited,” in *Proceedings of the 9th International Conference on Digital Audio Effects (DAFx-06)*, Montreal, Canada, September 18-20, 2006, pp. 133–137.
- [6] 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.
- [7] S. Böck, A. Arzt, F. Krebs, and M. Schedl, “Online real-time onset detection with recurrent neural networks,” in *Proceedings of the 15th International Conference on Digital Audio Effects (DAFx)*, York, UK, September 17-21, 2012.
- [8] S. Böck, F. Krebs, and M. Schedl, “Evaluating the Online Capabilities of Onset Detection Methods,” in *Proceedings of the 13th International Society for Music Information Retrieval Conference, ISMIR 2012*, Mosteiro S.Bento Da Vitória, Porto, Portugal, October 8-12, 2012, pp. 49–54.
- [9] P. Masri, *Computer modelling of sound for transformation and synthesis of musical signals*, Ph.D. thesis, University of Bristol, 1996.
- [10] D. Stowell and M. Plumley, “Adaptive whitening for improved real-time audio onset detection,” in *Proceedings of the International Computer Music Conference (ICMC’07)*, Copenhagen, Denmark, August 27-31, 2007, pp. 312–319.
- [11] F. H. F. Wu, T. C. Lee, J. S. R. Jang, K. K. Chang, C. H. Lu, and W. N. Wang, “A Two-Fold Dynamic Programming Approach to Beat Tracking for Audio Music with Time-Varying Tempo,” in *Proceedings of the 12th International Society for Music Information Retrieval Conference, ISMIR*, Miami, Florida, USA, October 24-28, 2011, pp. 191–196.
- [12] D. P. W. Ellis, “Beat Tracking by Dynamic Programming,” *Journal of New Music Research*, vol. 36, no. 1, pp. 51–60, Mar. 2007.
- [13] B. Boashash, “Estimating and interpreting the instantaneous frequency of a signal—Part I: Fundamentals,” *Proceedings of the IEEE*, vol. 80, no. 4, pp. 520–538, Apr. 1992.
- [14] S. Hibi, “Rhythm perception in repetitive sound sequence,” *Journal of the Acoustical Society of Japan (E)*, vol. 4, no. 2, pp. 83–95, 1983.
- [15] D. Rudrich, “Timing-improved Guitar Loop Pedal based on Beat Tracking,” M.S. thesis, Institute of Electronic Music and Acoustics, University of Music and Performing Arts Graz, Jan. 2017.

A METHOD FOR AUTOMATIC WHOOSH SOUND DESCRIPTION

Eugene Cherny

Embedded Systems Laboratory
Abo Akademi University
Turku, Finland
eugene.cherny@abo.fi

Johan Lilius

Embedded Systems Laboratory
Abo Akademi University
Turku, Finland
johan.lilius@abo.fi

Dmitry Mouromtsev

ISST Laboratory,
ITMO University
St. Petersburg, Russia
mouromtsev@mail.ifmo.ru

ABSTRACT

Usually, a sound designer achieves artistic goals by editing and processing the pre-recorded sound samples. To assist navigation in the vast amount of sounds, the sound metadata is used: it provides small free-form textual descriptions of the sound file content. One can search through the keywords or phrases in the metadata to find a group of sounds that can be suitable for a task. Unfortunately, the relativity of the sound design terms complicate the search, making the search process tedious, prone to errors and by no means supportive of the creative flow. Another way to approach the sound search problem is to use sound analysis. In this paper we present a simple method for analyzing the temporal evolution of the “whoosh” sound, based on the per-band piecewise linear function approximation of the sound envelope signal. The method uses spectral centroid and fuzzy membership functions to estimate a degree to which the sound energy moves upwards or downwards in the frequency domain along the audio file. We evaluated the method on a generated dataset, consisting of white noise recordings processed with different variations of modulated band-pass filters. The method was able to correctly identify the centroid movement directions in 77% sounds from a synthetic dataset.

1. INTRODUCTION

There are two different approaches to sound design according to how a sound is generated and used [1]. A sample-oriented approach is based on the processing of the recorded sounds, which are then arranged according to a moving picture content. A procedural approach implies building up signal synthesis and processing chains and making them reactive to happenings on the screen. But the complexity of implementing DSP algorithms for artistic tasks made this approach rarely used in the industry, hence the sample-oriented method is used most in sound design. In this case the design process starts with searching for sounds in libraries.

To facilitate search and decrease the time spent searching, library manufacturers provide textual annotations for sounds that could be written in filenames, in PDF files, or encoded in spreadsheet for integration with sound metadata management software. In either case, sound search is basically a keyword-search in a corpus of text. This format makes it almost impossible to annotate many aspects of sounds, as the annotations will grow too big to be manageable by a human. So, the creators of sound libraries provide annotations that are short and concise. The quality and richness of metadata varies from company to company, but mostly all provide keywords for common sound categories, representing purpose, materials, actions and other high-level sound-related concepts.

The keywords may have synonyms, change meaning depending on a context (polysemy), or even be interpreted differently by

different people (relativity) [2, 3]. These aspects lead to a poor user experience in sound search applications, when the search yields a lot of results, and a designer would need to listen through many sounds before even starting doing the creative work [4]. To narrow down the search, an experienced professional may want to exclude specific libraries from search or use boolean expressions to include synonyms, but all these actions are far from the creative work.

One way to approach this problem could be augmenting the keyword-based search with audio analysis algorithms to estimate some sound parameters that are pertinent to the workflow. There are a number of concepts in sound design which have more or less unambiguous meanings that many people agree on (like whooshes, risers, hits or pads to name a few), and we can model these sounds without doing a substantial amount of knowledge elicitation work. These models can be used in the search context to filter out the sounds which do not fit into the model parameters a user is interested in.

Of course, a comprehensive method to address the problem would include many of such models, providing a sound designer with a diverse set of content-based search filters. But in this paper we will only consider a model of a whoosh sound, as it has a relatively straightforward definition and is widely used in the production. The AudioSet ontology [5] defines whoosh as “a sibilant sound caused by a rigid object sweeping through the air.” In sound design this term is usually interpreted in a more generic way, for example as “movements of air sounds that are often used as transitions, or to create the aural illusion of movement” [6]. This movement is an essential quality of a whoosh sound and can be expressed as volume or spectral movements. The definitive characteristics of this movement are relatively slow attack and decay, a prominent climax point, and often a noticeable spectral change from the higher frequencies down to the lower ones or the opposite way around. Sound of this class are widely used, e.g. in game menu navigation, trailer scene transitions, for sounds of space ships passing by, and so forth.

In this paper we describe a whoosh model and a set of audio analysis algorithms to fit an arbitrary sound file into this model. The model is used to estimate the spectral movement descriptor quantifying the degree of the whoosh transition up or down along the spectrum. The model is based on splitting the sound into frequency bands, and approximating the volume envelopes in each band as piecewise linear function. This representation is similar to the multi-stage envelopes used in sound synthesizers¹, so we can easily extract attack and decay values from it. We test our method on a synthetic dataset consisting of different variations of white noise processed with a modulated band-pass filter.

The paper is organized as follows. Section 2 provides an overview of the related work. Section 3 describes the whoosh model

¹Also known as breakpoint function.

and its centroid metric. In Section 4 we describe procedures to fit an arbitrary sound to the model. Section 5 describes the evaluation procedures, and Section 6 reports the evaluation results. Lastly, in Section 7, we discuss the results, and then conclude the paper in Section 8 by summarizing contributions and possible applications.

2. RELATED WORK

A good introduction to the sound retrieval problem can be found in [7]. This work explored what words subjects used to describe sound, and grouped verbal descriptions into three major classes: *sound itself* (onomatopoeia, acoustic cues), *sounding situation* (what, how, where, etc.) and *sound impression* (adjectives, figurative words, etc.). These classes provide different conceptual tools to organize sound.

The *sound impression* class deals with the interpretation of how words connected to sound [8, 9, 10]. This class consists of descriptions that are usually used in informal communication, and they often do not have established interpretations. Thus, building a recognition system for them would require the elicitation of strict definitions (like the authors of [11] and [12] did).

The *sounding situation* class contains descriptions of setting in which a sound production occurred. They may include a sound source (a bird, a car), a sounding process (twittering, rumbling), a place (a forest, a garage), a time of production (morning), etc. Essentially, commercial sound libraries mostly annotated with the concepts of this class, but adding all word variations and synonyms to enrich the metadata would sacrifice readability and search simplicity. Moreover, treating linguistic relativity may require a more deep understanding of what is actually recorded in a sound. For example, a *long* hit sound in the *user interface library* could be much shorter than a *short* hit from the *trailer library*. One could address this issue by adding a *trailer* keyword to the annotation, but this is just the one example of many possible whoosh usages, and covering up them all may just be impossible to achieve with limited human resources. To some extent these problems can be addressed with natural language processing or knowledge elicitation [3, 13, 14], but treating the relativity problem with these methods would require a substantial amount of knowledge engineering to categorize and organize sound concepts. There are ontologies that provide some structured information about audio-related concepts [5, 15, 16, 17] and with some adaptations they can assist in the sound search. For example, in [12] authors describe the implementation of the timbre attribute prediction software that uses concepts from the ontology provided in [15].

The *sound itself* class describes acoustic features of the sound “as one hears it.” It includes sound representation models (music theory, spectromorphology, onomatopoeia, etc.). For example, in [18] authors explore the relevance of sound vocalizations compared to the actual sound. Interestingly, in this paper authors connected vocalizations with the morphological profiles of the sound, based on Schaeffer’s spectromorphology concepts. There is a substantial work done on automatic sound morphological profile recognition [19, 20, 21], but real sound designers rarely have to deal with the morphological concepts.

3. WHOOSH MODEL

According to the previous section, our work is related to the modelling sound properties (the *sound itself* class), as we want to quantify the parameters of some known class of sounds (whooshes). As

was discussed in the Introduction, the main property of a whoosh sound is the *movement* which can be temporal, spectral or both at the same time. This movement is characterized by the prominent peak somewhere in the middle of the sound: the intensity rises to this peak from silence and decays down after passing it. A spectral change can also happen together with the intensity, often with prominent transitions from higher to lower part of the spectrum or other way around. We build our model with the proposition that two movements together define the whoosh sound.

We continue with the following assumptions:

1. The sound intensity movement of the whoosh sound can be described using either an *attack-decay* (AD) or an *attack-hold-decay* (AHD) envelopes. This comes directly from the proposition we mentioned just above.
2. The model will be used in the search context, where a user already knows what sounds are whooshes, and only wants to filter out some of them according to the settings. Thus, a recognition between whoosh and non-whoosh sounds is not needed.

The purpose of the model is to describe the temporal envelope and the spectral movement of a whoosh sound. The model consists of a simplified multi-band representation of the sound intensity envelope and a movement metric, which quantifies a degree to which the frequency content moved up or down in the spectrum.

The *attack-decay* envelope is an ideal representation of a whoosh movement, but in practice, it is often hard to define one single peak, especially in the long sounds. In this case an *attack-hold-decay* envelope could be a better fit. The AHD envelope can be modeled as a piecewise linear function defined somewhere at a sound with length L :

$$E_i(x) = \begin{cases} a_1x + b_1, & \text{if } x \in [x_0^i, x_1^i) \\ a_2x + b_2, & \text{if } x \in [x_1^i, x_2^i) \\ a_3x + b_3, & \text{if } x \in [x_2^i, x_3^i] \\ 0, & \text{otherwise} \end{cases}, \quad x \in [0, L] \quad (1)$$

where x is an arbitrary time point in a sound file, x_0^i and x_3^i are the beginning and the end of the envelope, and x_1^i and x_2^i are the breaking points. The AD envelope model is a special case of AHD, where $x_1^i = x_2^i$. A sound can contain several non-overlapping sound envelopes, and i denotes an index of the envelope in the sound.

For each sound we define two fuzzy membership functions for the “beginning” and the “end” linguistic variables:

$$m_{beg}(x) = \frac{x}{L} \in [0, 1], \quad (2)$$

$$m_{end}(x) = 1 - m_{beg}(x) \in [0, 1], \quad (3)$$

which results can be interpreted as “to what extent the time point x is located in the beginning or in the end of a sound.” We use these functions to weigh an arbitrary envelope value:

$$V_l^i(x) = m_l(x)E_i(x), \quad (4)$$

where l denotes a linguistic variable, which can be either “beg” or “end”.

The breaking points x_1^i and x_2^i from the Eq. 1 are considered as representing points, where the most sound energy is located.

We are using them to calculate a *cumulative peak value* for the linguistic variable l :

$$p_l = \sum_i (V_l^i(x_1^i) + V_l^i(x_2^i)), \quad (5)$$

which value can be interpreted as “how much of sound energy lies in the beginning or in the end of a sound.”

We use the piecewise functions (Eq. 1) to compress envelope signals in the each band of a sound, passed though a crossover. With a set of models in each frequency band, we can estimate the centroid of the peak values for each linguistic variable, adapting the standard spectral centroid formula [22]:

$$C_l = \frac{\sum_b b p_l^b}{\sum_b p_l^b}, \quad (6)$$

where b is a band index, and p_l^b is a cumulative peak value of the band b for the linguistic variable l .

And finally, the centroid descriptor is a difference between centroids of two linguistic variables:

$$C_d = C_{end} - C_{beg} \quad (7)$$

C_d estimates how the centroid changes over time, with negative values suggesting a decrease, and positive values — an increase.

4. FITTING A SOUND TO THE MODEL

To construct the model we need to find values for all non-input variables in the Eq. 1. Our strategy here is to first find the boundaries of the envelope (points x_0^i and x_3^i), and then to find breaking points x_1^i and x_2^i . Knowing envelope values at these points, we can calculate the linear function coefficients a and b in each piece by solving a trivial linear equation system. This approach is similar to the one used in [19], where the maximum sound envelope values are being connected to the both, beginning and end of the sound, thus forming a two-segment AD model. In our method we construct the AHD model which can often be more realistic approximation. The rest of this section will explain, how exactly all these steps has been performed.

In sound libraries a single audio recording can contain several variations of the same sound separated by silence. This is a common practice, e.g. for the sounds of weapon shooting or footsteps, etc. So, we start with looking for the long silence segments and use them as separators to split recordings into multiple sounds.

The separated sounds are then split into frequency bands with a crossover filter bank, and the RMS envelope signals are estimated for each band. Based on these signals we split each band into regions separated by silence. A simple threshold processing is used to find the initial set of regions, that may get merged if they are close to each other.

The piecewise models are then built for each resulting region. The beginning and the end points (x_0^i and x_3^i) are defined by regions’ bounds, thus we need to find and connect climax points to create the linear representation. We do this by essentially the same algorithm as we used in the silence-split operation above, but with a higher threshold value relative to the maximum envelope value in the region. The new operation will yield a number of “high-energy” regions, and their maximum values are used as climax points for the multi-segment model. However, in many cases, the resulted model will contain a large number of segments due to

noise and irregularities in sound. To overcome this, we discard all points between the first and the last climax-points. The two peak points left will be used as x_1^i and x_2^i , thus creating the AHD model. If there is only one peak point found after performing this operation, then we can create the AD model, where ($x_1^i = x_2^i$). The AD model can then be simplified by removing the middle piece from the piecewise model in the Eq. 1.

Having all breakpoints of the piecewise model found, determining the a and b coefficients for each linear function is as trivial, as finding the equation of the line given two points.

5. EVALUATION

We use the following method parameters in the evaluation:

- The sample rate is set to 44100 Hz.
- The RMS window size for the envelope signal estimation is set to 20 ms.
- For the separating sounds in the sound file we set the non-silence split threshold T_{ss} to 0.001 (-60dB), and non-silence merge threshold T_{sm} to 500 ms. This means, that all non-silence regions with value higher than 0.001 will be merged together if the silence gap between them is shorter than half a second.
- The crossover is implemented using a bank of 4-order Butterworth filters. It can have a different number of frequency bands, but the split frequencies should to be spread evenly on the equivalent rectangular bandwidth scale (ERB) [23]. This provides a perceptual weighting for comparing signal energies between different frequency bands.
- To find the envelope bounds in frequency bands, the T_{ss} value is set to 75 ms.
- To find the high energy regions in envelopes, the peak split threshold (T_{ps}) is set to 80% of the maximum envelope signal value between the bounds. The peak merge threshold T_{pm} is set to 5 ms. I.e. the high energy regions with the envelope value more than 80% of the maximum will be merged together into one if the gap between them is shorter than 5 ms. The individual peaks will be identified as the maximum value in the each region.

First, we demonstrate the method for a simple signal by feeding in a 1-second sine wave with frequency sweeping from 50 to 10000 Hz into it (Fig. 1).

We also test the method on a synthetic dataset generated with Csound [24]. It consists of 1-second sound files with the variations of the white noise processed by a modulated band-pass filter. The modulation linearly changes the center frequency and the bandwidth of the filter from cf_1 to cf_2 , and from bw_1 to bw_2 respectively. We choose the set of center frequencies and bandwidths for modulation as follows:

- Create an list of tuples (f_{l1}, f_{h1}) with all possible pairwise combinations of the set $\{0, 100 \dots 20000\}$, so that $f_{l1} < f_{h1}$. f_{l1} and f_{h1} are the bottom and the top frequencies of the pass-band respectively.
- Combine the resulted tuples into a list of $((f_{l1}, f_{h1}), (f_{l2}, f_{h2}))$. Each element of the list represents a pair of initial and target modulation parameters. The parameters are transformed into the center frequency and bandwidth pairs as follows: $bw_i = f_{hi} - f_{li}$, $cf_i = f_{li} + 0.5bw_i$, where $i \in \{1, 2\}$.

- To reduce the number of samples, we first leave out the tuples where the absolute difference between the initial and the target center frequencies are more than 5000², and then sampling every 200th sample from what is left.

This procedure leaves us with 365330 variations of the initial and the target filter frequency-bandwidth pairs. We pass these values into Csound [24] to generate the sound files³. Our method is evaluated using different number of frequency bands. The goal of this experiment is to find, in which situations our method fails to correctly predict the spectral direction of whoosh sounds.

6. RESULTS

Fig. 1 shows how the method works on a trivial example: a sine wave with frequency rising up linearly from 1000 Hz to 10000 Hz. We see how the algorithm fits the linear model to the envelope signals in each band. The red vertical lines depict peak positions. Values at the peaks are weighted for each linguistic variable as follows (Eq. 4):

$$\begin{aligned} V_{beg}^1 &\approx 0.002 & V_{end}^1 &\approx 0 \\ V_{beg}^2 &\approx 0.354 & V_{end}^2 &\approx 0.0 \\ V_{beg}^3 &\approx 0.314 & V_{end}^3 &\approx 0.04 \\ V_{beg}^4 &\approx 0.007 & V_{end}^4 &\approx 0.347 \end{aligned}$$

Which yields the following centroids after applying Eq. 6:

$$C_{beg} \approx 1.445 \quad C_{end} \approx 2.893$$

The “end” centroid is higher, suggesting the sound’s frequency content is moving up in the spectrum.

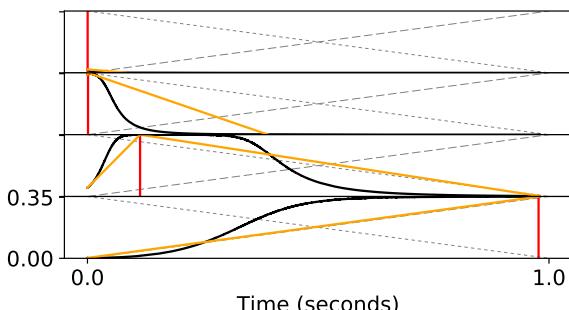


Figure 1: A sine sweep analyzed with the described method. The frequency goes up linearly from 1000 to 10000 Hz. The figure presents four graphs with signal envelopes and linear functions for each frequency band (the lowest band is at the top row). Split frequencies are 422, 1487 and 4596 Hz. Y-axis in each graph is the envelope amplitude. All graphs have identical scales, so only the bottom one’s scale is annotated. Black solid lines are the envelope signals. Yellow lines are the piecewise models (Eq. 1) fitted to the envelopes. Red horizontal lines depict the envelope’s peak value position. Dotted and dashed gray lines are fuzzy membership functions for the “beginning” and the “end” linguistic variables respectively. The signal in the first band is weak compared to the other bands, but it passed the threshold defined by T_{ss} .

²We decided not to include samples with such prominent spectral transition, as it may be easy for the algorithm to identify the whoosh direction in such cases.

³The second-order Butterworth filter was used, see the *butterbp* opcode, URL: <http://csound.github.io/docs/manual/butterbp.html>

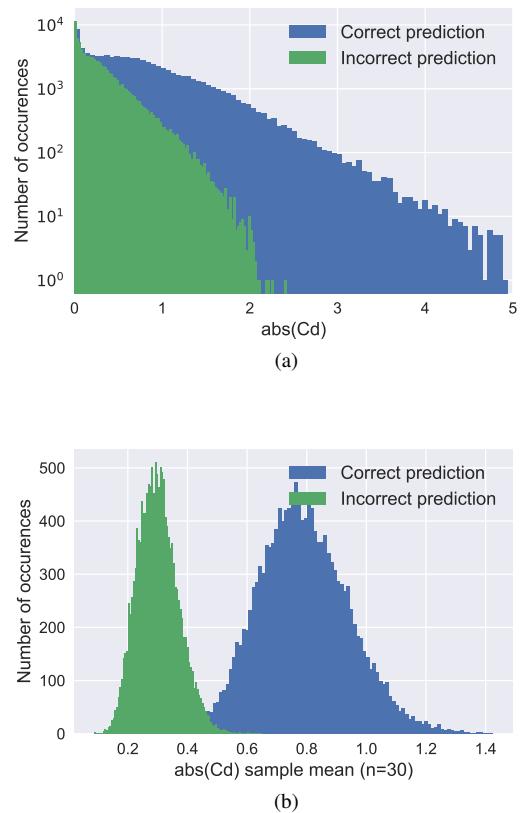
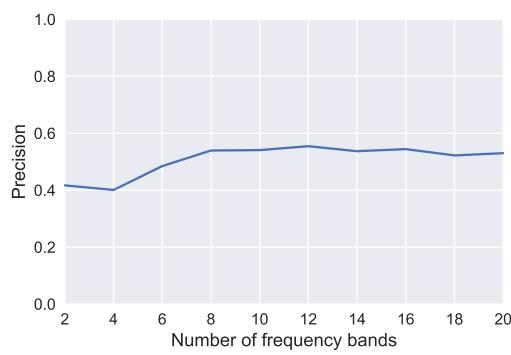


Figure 2: a) Distribution of the absolute centroid descriptor ($abs(C_d)$) for correct and incorrect predictions. For this histogram the correct predictions data has been sampled to be comparable with the amount of incorrect predictions data available. b) Sampling distribution of the mean of $abs(C_d)$ with sample size equal to 30.

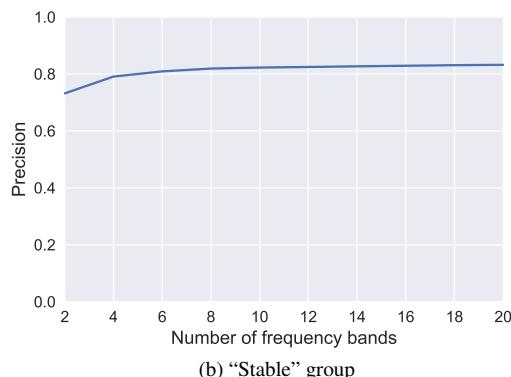
In the second experiment we test the method on a set of synthetically prepared sounds, as described in the previous section. From the 365330 variations of filter frequency sweeps of the white noise signal, the centroid movement direction of 282815 sounds (approx. 77%) were identified correctly.

Fig. 2-a shows the distribution of the absolute centroid descriptor for both correct and incorrect predictions. We see that on average, the prediction errors happen more often when the centroid movement is low (Fig. 2-b). There is a slight positive relationship between the absolute centroid descriptor and the prediction correctness ($r \approx 0.319$). Approximately the same correlation exists between the difference of the start and the end center frequencies of the band-bass filter modulation in generated sounds ($r \approx 0.317$).

We test how the algorithm performs when evaluated with different number of frequency bands. The obvious outcome of using less bands is lesser frequency domain precision and inability to recognize whoosh direction when spectral variations of a sound are completely covered by a single band. So, the ideal method may fail to recognize the direction of some whooshes when evaluated with a few bands, but it will eventually succeed after increasing the



(a) “Unstable” group



(b) “Stable” group

Figure 3: A precision of whoosh direction discrimination in two groups of sounds.

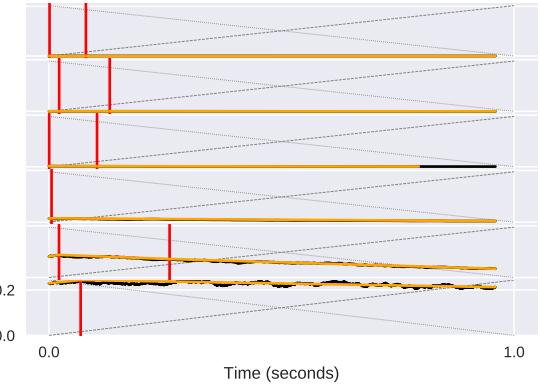
number of bands. But test results are not that simple. We observed three group of sounds:

1. Sounds where the estimated whoosh direction is the same for all numbers of frequency bands.
2. Sounds where the direction estimation is incorrect for a low number of bands, but it improves after increasing their number and does not regress anymore.
3. Sounds, where the direction estimation varies for different number of frequency bands without any apparent pattern (e.g. it estimates correctly for 4 and 6 bands, then regresses at 10 bands, then improves for 18 bands, etc.).

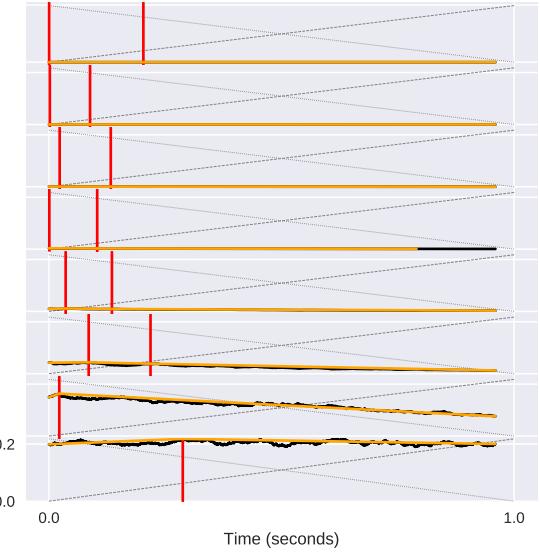
For convenience we call first two groups of sounds *stable* and the third one *unstable*. The *unstable* subset contains 9506 sounds (including both, correct and incorrect predictions), which is 26% of the test data. Fig. 3 compares the bandwise method precision in the two subsets. We see that the precisions are approximately 0.58 and 0.81 for the *unstable* and *stable* respectively. The precision grow as the number of frequency bands increase, and stabilize at approximately 8 bands in both groups.

7. DISCUSSION

In this section we will briefly discuss the method performance analysis results from the previous section.



(a) 6 bands, incorrect pred., $C_{beg} \approx 4.46$, $C_{end} \approx 4.37$



(b) 8 bands, correct pred., $C_{beg} \approx 6.06$, $C_{end} \approx 6.57$

Figure 4: An example of noisy peak detection. The graph axes are the same as in the Fig. 1. The noise sound ($cf_1 = 9292$, $bw_1 9697$, $cf_2 = 11413$, $bw_2 = 7071$) has been analyzed with the method two times: with 6 and 8 bands. Note the peak position in the bottom bands: although the envelope signal is essentially the same, the estimated peak position vary significantly due to noise in the signal. This lead to the incorrect direction prediction in the case (a).

First, there is an evident correlation between absolute C_d and the method precision. This descriptor is a difference between centroid values weighted for the two linguistic variables representing a temporal placement of an event. This suggests, that the method may have worse performance for the sounds with subtle spectral centroid movements. Currently, the implementation lacks notion of the “still” category, which contains sounds that does not perceptually go “up” nor “down”, and many incorrect predictions might go into this class. But the implementation of this class requires to conduct a qualitative research determining, what is a perceptually significant whoosh movement. This is an important question for the content-based sound search, but it is out of scope of this

particular paper.

Second, the data clearly shows that in general, we can increase the number of frequency bands to get a higher precision, but at some point, adding more bands will stop affecting it. This suggests that some bands can be redundant to represent a particular sound. Thus, the piecewise representation can be improved by introducing an adaptive analysis algorithm, that would change the number of bands according to the sound's spectral content.

Lastly, the inconsistent method performance for different number of frequency bands suggests the peak-detection sensitivity noise. The white noise has equal intensity at different frequencies and relatively steady volume envelope. Filtering the noise with a band-pass filter will increase irregularities in the envelope, thus providing noise for our peak-detection algorithm. Fig. 4 shows one sound processed by the method using 6 and 8 frequency bands. We see, that peak positions in the multi-segment representation vary significantly, which affects the prediction. We can tackle the noise in both, the envelope signal by increasing its smoothness with filtering, as well as making more stable piecewise linear models by using more advanced fitting algorithm [25, 26]. Another way to improve the method robustness could be to use the integration of the product of membership and piecewise functions instead of peak values in V_l^i estimation (Eq. 5). This way the method will not be dependent on the peak detection errors caused by the sound irregularities.

Another point of improvement could be testing the method on real sounds, but we have found extremely little sounds on Freesound [27] that contain tags for both, the “whoosh” keyword and the direction. E.g. there are only 10 whooshes annotated with the tag “down”. One option could be to annotate arbitrary whooshes by hand and validate the method, but many sounds are intended to be the “still” whooshes, so without proper differentiation of this class by the method the hand-annotation will be prone to errors. But manual annotation will could be prone to bias: as the method’s creators, we may be annotating to make it pass the test. Thus, a third party should be involved in annotation process.

8. CONCLUSIONS

We presented a descriptor for describing spectral centroid movement in audio files, based on fuzzy membership functions and piecewise linear model representation of sound envelopes in different frequency bands. The method has been evaluated on a dataset consisting of 365330 synthetic test sounds. It predicted spectral centroid movement correctly in 77% of sounds in a dataset. We identified the probable cause of prediction errors as noisiness in the peak detection algorithm, and suggested ways to improve it.

The described method can be used in the sound search context to filter or sort whoosh sounds according to the spectral movement. The centroid descriptor C_d can not only be used for up-down whoosh differentiation, but also to estimate the degree of spectral movement in sounds, and to sort according to it. Also, knowing the break points positions, we can filter sounds based on the relative length of the attack and decay lengths. These filters can be implemented, for example, as user interface widget in the sound metadata management software to assist the search.

The source code used for this paper is available on-line: <https://github.com/ech2/papers>.

9. ACKNOWLEDGMENTS

This work was partially financially supported by the Government of the Russian Federation, Grant #074-U01. The *librosa* [28] Python library for sound analysis was used in this project.

10. REFERENCES

- [1] Andy Farnell, *Designing sound*, MIT Press, 2010.
- [2] Fabiano M Belém, Jussara M Almeida, and Marcos A Gonçalves, “A survey on tag recommendation methods,” *Journal of the Association for Information Science and Technology*, vol. 68, no. 4, pp. 830–844, 2016.
- [3] Frederic Font and Xavier Serra, “Analysis of the Folksonomy of Freesound,” in *Proc. of the 2nd CompMusic Workshop*, Istanbul, Turkey, July 12, 2012, pp. 48–54.
- [4] Eugene Cherny, “Email interview with Axel Rohrbach (BOOM Library GbR) about problems with metadata management in sound design,” unpublished.
- [5] Jort F. Gemmeke, Daniel P. W. Ellis, Dylan Freedman, Aren Jansen, Wade Lawrence, R. Channing Moore, Manoj Plakal, and Marvin Ritter, “Audio set: An ontology and human-labeled dataset for audio events,” in *Proc. IEEE ICASSP 2017*, New Orleans, LA, USA, 2017.
- [6] Tim Prebble, “SD101: Wooshes,” Available at <http://www.musicofsound.co.nz/blog/wooshes-101>, 2007 (accessed on June 21, 2017).
- [7] Sanae Wake and Toshiyuki Asahi, “Sound retrieval with intuitive verbal expressions,” in *Proceedings of the 1998 International Conference on Auditory Display (ICAD'98)*, Swindon, UK, Nov. 01-04, 1998, pp. 30–30.
- [8] Michel Bernays and Caroline Traube, “Expression of piano timbre: gestural control, perception and verbalization,” in *Proceedings of CIM09: The 5th Conference on Interdisciplinary Musicology*, Paris, France, 2010.
- [9] Graham Darke, “Assessment of timbre using verbal attributes,” in *Proc. Conference on Interdisciplinary Musicalogy*, Montreal, Quebec, 2005.
- [10] Asterios Zacharakis, Konstantinos Pastiadis, and Joshua D. Reiss, “An interlanguage unification of musical timbre: Bridging semantic, perceptual, and acoustic dimensions,” *Music Perception: An Interdisciplinary Journal*, vol. 32, no. 4, pp. 394–412, 2015.
- [11] Thomas Grill, “Constructing high-level perceptual audio descriptors for textural sounds,” in *Proceedings of the 9th Sound and Music Computing Conference (SMC 2012)*, Copenhagen, Denmark, 2012, pp. 486–493.
- [12] Andy Pearce, Tim Brookes, and Russell Mason, “First prototype of timbral characterisation tool for semantically annotating non-musical,” Audio Commons project deliverable D5.2, Surrey, UK, 2017, available at <http://www.audiocommons.org/materials>.
- [13] Frederic Font, Joan Serrà, and Xavier Serra, “Audio clip classification using social tags and the effect of tag expansion,” in *AES 53rd International Conference on Semantic Audio*, London, UK, Jan. 26, 2014, pp. 157–165.

- [14] Eugene Cherny, Johan Lilius, Johannes Brusila, Dmitry Mouromtsev, and Gleb Rogozinsky, “An approach for structuring sound sample libraries using ontology,” in *Proc. International Conference on Knowledge Engineering and Semantic Web (KESW 2016)*, Prague, Czech Republic, Sept. 21-23, 2016, pp. 202–214.
- [15] Andy Pearce, Tim Brookes, and Russell Mason, “Hierarchical ontology of timbral semantic descriptors,” Audio Commons project deliverable D5.1, Surrey, UK, 2016, available at <http://www.audiocommons.org/materials>.
- [16] Tohomiro Nakatani and Hiroshi G. Okuno, “Sound ontology for computational auditory scene analysis,” in *Proceeding for the Fifteenth national Conference on Artificial Intelligence (AAAI-98)*, Madison, WI, USA, July 26-30, 1998, pp. 1004–1010.
- [17] Jens Lehmann, Robert Isele, Max Jakob, Anja Jentzsch, Dimitris Kontokostas, Pablo N Mendes, Sebastian Hellmann, Mohamed Morsey, Patrick Van Kleef, Sören Auer, et al., “Dbpedia—a large-scale, multilingual knowledge base extracted from wikipedia,” *Semantic Web*, vol. 6, no. 2, pp. 167–195, 2015.
- [18] Guillaume Lemaitre, Olivier Houix, Frédéric Voisin, Nicolas Misdariis, and Patrick Susini, “Vocal imitations of non-vocal sounds,” *PLoS ONE*, vol. 11, no. 12, pp. 1–28, Dec., 2016.
- [19] Geoffroy G. Peeters and Emmanuel Deruty, “Automatic morphological description of sounds,” *The Journal of the Acoustical Society of America*, vol. 123, no. 5, pp. 3801–3801, 2008.
- [20] Geoffroy Peeters and Emmanuel Deruty, “Sound indexing using morphological description,” *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 18, no. 3, pp. 675–687, 2010.
- [21] Lasse Thoresen and Andreas Hedman, “Spectromorphological analysis of sound objects: an adaptation of Pierre Schaeffer’s typomorphology,” *Organised Sound*, vol. 12, no. 2, pp. 129, 2007.
- [22] Wikipedia, “Spectral centroid,” accessed June 21, 2017, Available at https://en.wikipedia.org/wiki/Spectral_centroid.
- [23] 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.
- [24] Victor Lazzarini, Steven Yi, John ffitch, Joachim Heintz, Øyvind Brandtsegg, and Iain McCurdy, *Csound - A Sound and Music Computing System*, Springer, 2016.
- [25] Eamonn Keogh, Selina Chu, David Hart, and Michael Pazzani, “An online algorithm for segmenting time series,” in *Proc. IEEE International Conference on Data Mining (ICDM 2001)*, 2001, pp. 289–296.
- [26] Vito MR Muggeo, “Segmented: an R package to fit regression models with broken-line relationships,” *R news*, vol. 8, no. 1, pp. 20–25, 2008.
- [27] Frederic Font, Gerard Roma, and Xavier Serra, “Freesound technical demo,” in *Proceedings of the 21st ACM International Conference on Multimedia*, New York, NY, USA, 2013, MM ’13, pp. 411–412, ACM.
- [28] Brian McFee, Matt McVicar, Oriol Nieto, Stefan Balke, Carl Thome, Dawen Liang, Eric Battenberg, Josh Moore, Rachel Bittner, Ryuichi Yamamoto, Dan Ellis, Fabian-Robert Stoter, Douglas Repetto, Simon Waloschek, CJ Carr, Seth Kranzler, Keunwoo Choi, Petr Viktorin, Joao Felipe Santos, Adrian Holovaty, Waldir Pimenta, and Hojin Lee, “librosa 0.5.0,” Feburary 2017, Available at <https://doi.org/10.5281/zenodo.293021>.

ANALYSIS AND SYNTHESIS OF THE VIOLIN PLAYING STYLE OF HEIFETZ AND OISTRAKH

Chi-Ching Shih, Pei-Ching Li*, Yi-Ju Lin,
Yu-Lin Wang, Alvin W. Y. Su

SCREAM Lab., Department of CSIE,
National Cheng-Kung University
Tainan, Taiwan
P78021015@mail.ncku.edu.tw

Li Su

Music and Culture Technology Lab.,
Institute of Information Science,
Academia Sinica
Taipei, Taiwan
lisu@iis.sinica.edu.tw

Yi-Hsuan Yang

Music and Audio Computing Lab.
Research Center for IT Innovation,
Academia Sinica
Taipei, Taiwan
yang@citi.sinica.edu.tw

ABSTRACT

The same music composition can be performed in different ways, and the differences in performance aspects can strongly change the expression and character of the music. Experienced musicians tend to have their own performance style, which reflects their personality, attitudes and beliefs. In this paper, we present a data-driven analysis of the performance style of two master violinists, *Jascha Heifetz* and *David Fyodorovich Oistrakh* to find out their differences. Specifically, from 26 gramophone recordings of each of these two violinists, we compute features characterizing performance aspects including articulation, energy, and vibrato, and then compare their style in terms of the accents and legato groups of the music. Based on our findings, we propose algorithms to synthesize violin audio solo recordings of these two masters from scores, for music compositions that we either have or have not observed in the analysis stage. To our best knowledge, this study represents the first attempt that computationally analyzes and synthesizes the playing style of master violinists.

1. INTRODUCTION

Music performance is composed of two essential parts: the compositions of the composers (scores) and the actual interpretation and performance of the performers (sounds) [1]. A piece of music can be interpreted differently depending on the style of the performers, by means such as extending the duration of accents or manipulating the energy of individual notes in a note sequence, etc.

Performance aspects that characterize the style of a violinist may include interpretational factors such as articulation and energy, and playing techniques such as shifting, vibrato, pizzicato and bowing. For example, for the articulation aspect of music, the time duration of individual note (DR) and the overlap in time between successive notes (or key overlap time, KOT) in the audio performance have been found important in music analysis and synthesis [2]. These two features characterize the speed of the music and the link between successive notes in a note sequence. Maestre *et al.* [3] used them to develop a system for generating an expressive audio from a real saxophone signal, whereas Bresin *et al.* [4] used them to analyze the articulation strategies of pianists.

Research has also been done concerning the techniques of violin playing. For instance, Matsutani [5] analyzed the relationship between performed tones and the force of bow holding. Ho *et al.* [6] proposed that we can achieve good intonation while performing vibrato by maintaining temporal coincidence between the intensity peak and the target frequency. Baader *et al.* [7] presented a quantitative analysis of the coordination between bowing and fingering in violin playing. These studies all collected scientific data, such

as the location of fingerprint on bow, the finger shifting data, pitch values, and velocity of finger, that may help amateur violinists improve their performing skills.

The analysis of musician style is also used in sound synthesis to improve the quality of synthetic music pieces. For example, Mantaras *et al.* [8] studied compositional, improvisational, and performance AI systems to characterize the expressiveness of human-generated music. Kirke *et al.* [9] studied factors including testing status, expressive representation, polyphonic ability, and performance creativity, to highlight the possible future directions for media synthesis. Li *et al.* [10] analyzed how expressive musical terms of violin are realized in audio performances and proposed a number of features to imitate human-generated music.

The style differences among violinists have been studied for a long time [11, 12, 13, 14]. For instance, Jung [15] presented an analysis of the playing style of three violinists, *Jascha Heifetz*, *David Fyodorovich Oistrakh* and *Joseph Szigeti*. He argued that the precision in the performance of Heifetz gives listeners the feeling that Heifetz is “cold” and “unemotional.” In contrast, Oistrakh was described as “warm” and “capable of communicating emotional feelings.” To some extent, the style of Heifetz and Oistrakh can be considered as two extremes of the spectrum, making the comparison between them interesting.

In this paper, we present a step forward into expressive performance analysis and synthesis of violin music by learning from the music of these two masters, Heifetz and Oistrakh. To this end, we first compile a new dataset consisting of some manual annotations of 26 excerpts from famous violin concertos that these two masters have both played (cf. Section 2). As perceptually Heifetz and Oistrakh are quite different in terms of the *velocity* and *accent* of their music, we choose to focus on the analysis of the articulation, energy, and vibrato aspects of the music in this dataset and propose a few features (cf. Section 3.1). In particular, we find that DR and KOT are indeed useful in characterizing the differences in the accents of their music (cf. Section 3.2). Then, we combine the vibrato synthesis procedure proposed by Yang *et al.* [16] and the proposed features to obtain synthetic sounds that imitate the style of these two violinists (cf. Section 4). We hope that this endeavor can make us closer to the dream of reproducing the work of the two masters via expressive synthesis.

2. DATASET

To compare the style of Heifetz and Oistrakh, we choose the music pieces that both of them have played before. In particular, we focus on the work recorded in the prime of their lives, for they are considered to be representative of their style. In addition, we pick only

Table 1: Dataset information.

Composer	Music Name	Movement	Bars
L. V. Beethoven	<i>Violin concerto in D major, Op. 61</i>	I	96, 97-98, 100-101, 128-130, 130-131, 134-136, 137-139
		II	77, 81-82, 83
J. Brahms	<i>Violin concerto in D major, Op. 77</i>	I	115-118, 124-127, 129-132
P. I. Tchaikovsky	<i>Violin concerto in D major, Op. 35</i>	I	28-29, 31-33, 39-41, 105-106
M. C. F. Bruch	<i>Scottish Fantasy, Op.46</i>	II	97-101, 107-109
E. V. A. Lalo	<i>Symphonie espagnole in D minor, Op. 21</i>	I	41-51, 66-69
J. Sibelius	<i>Violin concerto, Op. 47</i>	I	26-31
		III	124-127, 128-130, 133-135, 146-152

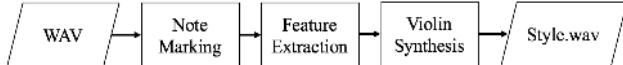


Figure 1: Flowchart of playing style analysis and synthesis system.

the violin solo parts to keep the influence from the background accompaniment minimal. Table 1 provides information about the resulting 26 excerpts we select from six violin concertos. We note that, because we have the performance of both the two masters for each of the excerpts, we have in total 52 audio recordings.

The original sources of these music pieces are gramophone recordings.¹ For analog to digital conversion, we use the Technics SL-1200 turntable, the phono amplifier ASR Basis phono preamplifier, and the Audio-Technica AT440MLa Dual Moving Magnet Cartridge audio interface. The resulting wave files are recorded by Adobe Audition, with 32-bit depth and 48 kHz sampling rate. The average length of these excerpts is around 6 seconds.

To investigate the performance difference between Heifetz and Oistrakh, we annotate the excerpts by hand with detailed information about the onset, offset, legato, accent, bar, note type and tempo. The process of annotation takes over four months to complete. This dataset represents to our best knowledge the first manually annotated dataset on the masterpieces of Heifetz and Oistrakh.

Annotating the accent and note type gives us more information on the note level. An accent is an emphasis on a particular note using louder sound or longer duration. As people may have different ways emphasizing notes, characterizing the accents are important. Except for the accent marks on scores, people may also emphasize a non-accent in their performance. Therefore, the criterion of regarding a note as an accent is not based on the audio data, but is according to the wedge-shaped markings on the score, the stressed beats such as one and three in common time (quadruple), and our judgment while listening to the excerpts.

The note type, on the other hand, helps us differentiate notes of different lengths in our analysis of articulation and energy. For simplicity, we consider the following six types of notes: whole note, half note, quarter note, eighth note, sixteenth note, and thirty-second note. For any note of other types, we assign its note type

¹For information regarding label and released year of the gramophone recordings, see <http://screamlab-ncku-2008.blogspot.tw/2017/03/synthesis-result-of-analysis-and.html>. Although we use this dataset to synthesize music, it can also be used for different purposes, such as for education. For example, students can understand what kinds of interpretations these two masters use by listening and analyzing the same music pieces. Besides, students can try to play the violin while listening to one specific excerpt, to feel the diverse tempo between the two masters and to imitate their distinct playing style.

to be the one with the next longest duration. For instance, a dotted eighth note will be considered as a quarter note in our analysis.

3. METHOD

Figure 1 shows the diagram of the proposed system, whose goal is to extract style-related features and use them for expressive synthesis. Given the audio waveform of an excerpt converted from a gramophone recording, we first mark and segment the notes of the excerpt by hand, for automatic methods for score-to-audio alignment may suffer from the influence of the white noise of gramophone recordings, and the interference of background accompaniments such as cello or the second violin. After getting the onset and offset information from the manual annotations, we calculate note-level features such as duration, energy contour and vibrato extent of every individual note (cf. Section 3.1). Then, according to the scores, we compare the features extracted from the performances of Heifetz and Oistrakh for the same excerpt, with respect to musical notations such as accent, legato (slur) and dynamics (cf. Section 3.2). As a result, we have two sets of statistical parameters that respectively characterize the style of Heifetz and Oistrakh. Depending on which set of statistical parameters we choose in the final synthesis stage, we hope to synthesize audio performance in the style of either Heifetz or Oistrakh (cf. Section 4).

3.1. Feature Extraction

We consider three types of note-level features: articulation, energy, and vibrato. These features are computed for each note. These note-level features are aggregated (pooled) into excerpt-level ones by taking the average across 12 types of note groups: accents and non-accents for each of the six types of notes (i.e. 2×6). We also refer to the excerpt-level features as the statistical parameters.

3.1.1. Articulation

Articulation features, such as DR and KOT, are related to the music tempo and the performers' expressive intentions. Therefore, they are important in playing style analysis [2]. For example, a sequence of short-duration notes without silent intervals between them might make audiences stressful or excited; a sequence of lively music with fixed and short intervals gives the impression of being delightful and lovely, and a long-duration note sequence creates a feeling of peace.

As illustrated in Figure 2, DR represents the duration of a note and we calculate the DR for the n -th note by:

$$DR_n = Offset_n - Onset_n, \quad (1)$$

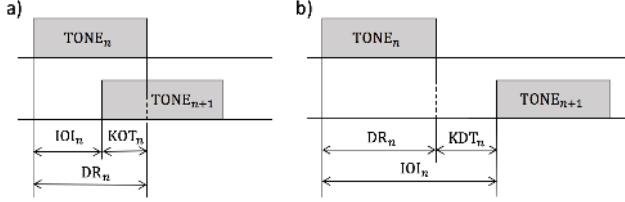


Figure 2: Illustration of duration (DR), key overlap time (KOT), key detached time (KDT), and inter onset interval (IOI) for (a) overlapping and (b) non-overlapping successive notes $TONE_n$ and $TONE_{n+1}$. For (b), we set $KOT_n = -KDT_n$.

where $Offset_n$ and $Onset_n$ are the offset and onset times of this note. The KOT, on the other hand, represents the overlap between two successive notes. Higher KOT indicates faster transition between notes. We calculate KOT for the n -th note by:

$$KOT_n = Offset_n - Onset_{n+1}. \quad (2)$$

If there is no overlap in time between the n -th note and the next note, we can calculate the key detached time (KDT):

$$KDT_n = Onset_{n+1} - Offset_n. \quad (3)$$

In such cases, we consider KOT as the negative KDT.

The excerpt-level features of DR and KOT for an excerpt are composed of the mean value of DR and KOT for accents and non-accents for each type of notes that presents in the excerpt. For example, the excerpt shown in Figure 3 only contains eighth and sixteenth notes. Therefore, the articulation parameters of this excerpt are computed by averaging the note-level DR and KOT values of the eighth note accents, eighth note non-accents, sixteenth note accents, and sixteenth note non-accents, respectively.

Please note that we do not normalize DR and KOT by the tempo of each excerpt, because we consider the tempo also as an indicator of the expression style of a performer.

3.1.2. Energy

Energy is an essential characteristic to distinguish playing style. Performers might follow the dynamic notations of music score in their interpretation. However, oftentimes they would also choose to emphasize a particular note, phrase, or chord by playing it louder, according to their own opinion, to convey different expressions. In this paper, the note-level energy features are the mean energy of each individual note and the mean energy contour for each of the six types of notes. Besides, an excerpt-level feature, accent energy ratio, is also considered.

Before calculating the energy features, we have to carry out the energy normalization due to the different recording environments and equipment of the recordings. The normalization is performed by dividing the energy of an excerpt by its maximum value, making the range of the values [0, 1]. After that, we use the short-time Fourier transform with Hanning window to calculate the energy contour of each note, based on its corresponding fundamental frequency F_0 , expressing in dB scale. To be specific, given the spectrogram $M(t, k)$ of a note for each time frame t and frequency bin k , the *energy contour* of the note with respect to F_0 is calculated by:

$$EC_{F_0}(t) = 20 \log(M(t, F_0)). \quad (4)$$

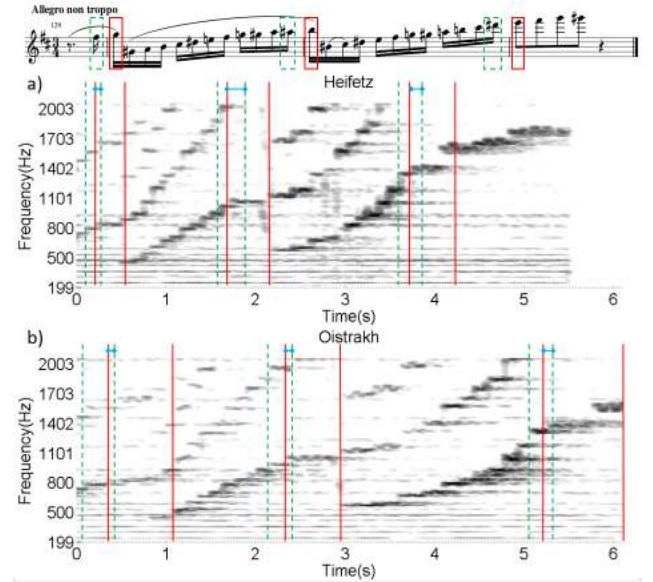


Figure 3: The spectrograms of Brahms' Violin Concerto, Mov. I, bars 129-132 performed by (a) Heifetz and (b) Oistrakh, with red solid vertical lines and green dashed lines indicating the onset and offset times of accents (red) and their preceding notes, which are typically non-accents (green), respectively. We can see that the KOTs (blue) of (a) Heifetz are larger than those of (b) Oistrakh.

Then, the energy normalization is applied again for the $EC_{F_0}(t)$ of each note, to keep the maximal peak value equal to one. Lastly, the mean energy contour for a note type is computed by averaging the energy contours of all the notes belonging to that note type.

Moreover, the mean *energy* of an individual note E_n is calculated by summing the amplitudes of the spectrogram across all the frequency bins, expressed in dB scale, divided by the note length:

$$E_n = \frac{20 \log(\sum_k M(t, k))}{\text{note length}}. \quad (5)$$

Finally, an excerpt-level feature, *accent energy ratio* (AER), representing the energy ratio of accents to non-accents within an excerpt, is calculated by:

$$AER = \frac{(\sum_{i=1}^p E_{A_i}) / p}{(\sum_{j=1}^q E_{NA_j}) / q}, \quad (6)$$

where E_A is the mean energy of an accent, E_{NA} is the mean energy of a non-accent, p is the number of accents, and q is the number of non-accents. For calculating $EC_{F_0}(t)$ and E_n , we use a frame size of 1,024 samples and a hop size of 64 samples.

3.1.3. Vibrato

Vibrato, the frequency modulation of fundamental frequency, is another essential element in violin performance. We consider the two common note-level features, *vibrato rate* (VR) and *vibrato extent* (VE). VR represents the rate of vibrato, i.e. the number of periods in one second. VE represents the frequency deviation between a peak and its nearby valley. We adopt the vibrato extraction process presented by Li *et al.* [10]. To determine whether a

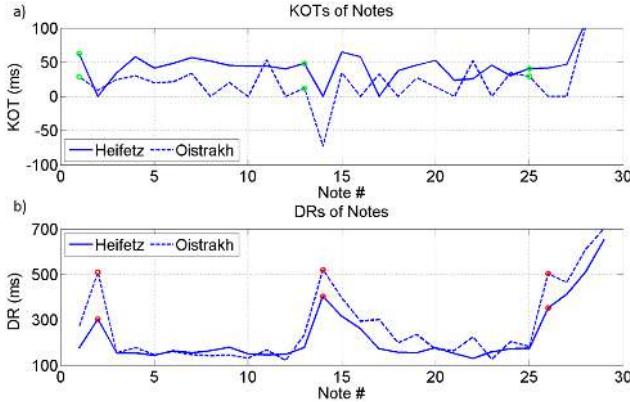


Figure 4: (a) KOT and (b) DR values of each individual note of the same excerpt used in Figure 3. The circles in (a) indicate the notes right before the accents, while in (b) the circles mark the accents.

note has vibrato, we require that its VR is between 3 and 10 Hz, VE is between 5 and 100 cents, and note length is over 125ms. The parameters of VR and VE of each excerpt is calculated as the mean value over all the vibrato notes. Moreover, we concern the excerpt-level feature, *vibrato ratio* (vibRatio), also adopted by Li *et al.* [10], to measure the percentage of vibrato notes within an excerpt. The frame size and hop size are the same as those used in computing energy-related features.

3.2. Difference between Heifetz and Oistrakh

The musical notations, accent and legato (slur), are considered in the comparison of articulation and energy features between Heifetz and Oistrakh. Specifically, we focus on the comparison of the KOTs of the notes right before accents, the DRs of accents, and the average energy curve of legato groups. Two excerpts with obvious characteristics are taken as running examples below.

3.2.1. KOTs of the notes right before accents

We observe that the KOTs of the notes right before the accents, which are typically non-accents, are important factors to distinguish Heifetz from Oistrakh. Let's take Brahms' *Violin Concerto*, Mov. I, bars 124-127 as an example. As shown in Figure 3, we mark three accents and the notes right before them by red solid lines and green dashed lines, respectively. According to equation (2), we mark the values of KOTs in blue horizontal lines. It can be seen that the KOTs of Oistrakh are smaller than those of Heifetz. This observation is actually in line with our subjective listening experience: Oistrakh often halts for a moment before playing the accents, while Heifetz, on the contrary, plays without a stop. Figure 4(a) shows the KOT of each note and the circles represent the KOT values but also the mean KOT of Heifetz are higher than those of Oistrakh, implying that Heifetz performs in faster tempo.

3.2.2. DRs of accents

From Figure 3, we can also observe for accents, the performance by Oistrakh has larger DRs than that by Heifetz. Besides, according to the legato notation of bar 130 on the score and its corre-

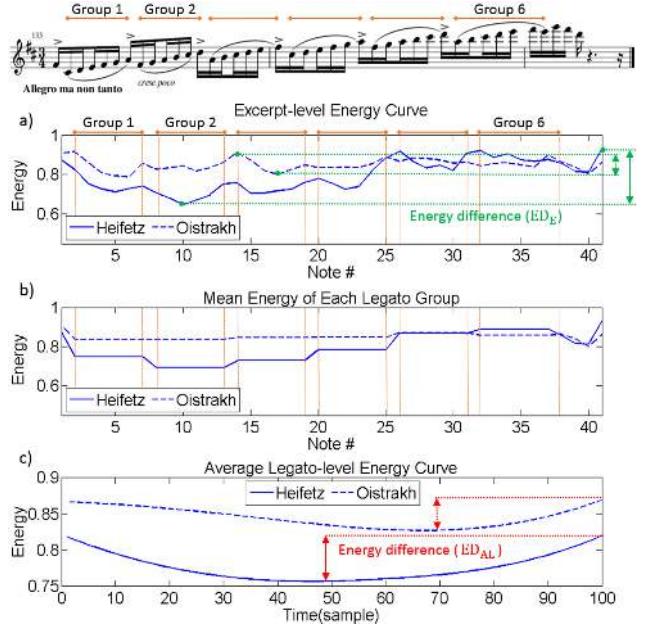


Figure 5: Energy features for Sibelius' *Violin Concerto*, Mov. III, bars 133-135: (a) excerpt-level energy curve, (b) mean energy of each legato group, and (c) average legato-level energy curve of the six legato groups.

sponding spectrogram, we can see that the DR of the accent takes a large proportion of the legato group, compared with those of the non-accents. In addition, from Figure 4(b), where we use circles to represent the DRs of accents, we have similar finding. This indicates that Oistrakh tends to prolong the duration of accents in comparison to Heifetz. As for the non-accents, both Heifetz and Oistrakh use similar speed to perform them.

3.2.3. Average energy curve of legato groups

In addition to the energy contour of an individual note, the excerpt-level energy curve is also an important factor in playing style analysis. Let's take Sibelius' *Violin Concerto*, Mov. III, bars 133-135 as an example this time. As shown in Figure 5(a), the average energy of each individual note constitutes quite different energy curves for the performances of Heifetz and Oistrakh. We mark the maximal energy differences in green according to the maximal peak value and the minimal valley value within each of these two excerpt-level energy curves. We can see that Heifetz demonstrates an obvious contrast between the energy of the first two bars, whereas Oistrakh uses nearly the same energy for the two bars.

To analyze the energy variation during a set of notes, we use the legato notations on the score and partition all the notes of this excerpt into six legato groups and one non-legato group, which includes the first note and the last three notes. Figure 5(b) shows the mean energy of the six legato groups, from which we also see the gradual tendency of energy increases from the performance of Heifetz, which corresponds to the *crescendo* notation.

It is hard to say anything from the resulting legato-level energy curves, for they seem to differ legato by legato. To simplify our comparison between the style of the two masters, we take the average of the energy curves for all the legato groups and show

the resulting average legato-level energy curve in Figure 5(c). After doing so, we can see that the average legato-level energy curve of Heifetz is more convex than that of Oistrakh, implying higher variation of the legato-level energy curves of Heifetz.

Furthermore, we come up with an interesting observation by examining every legato-level energy curve of all the ten excerpts in our dataset that have legato notations on the scores. We find that, when the number of notes of a single legato group is *less than eight*, the legato-level energy curve tends to be a *convex* one. In contrast, when the number of notes of a single legato group is *more than eight*, the legato-level energy curve is *concave*. We will use this finding as a *convex-concave style rule* in our synthesis of the performance of the two masters.

In short, the other two energy-related features that are useful in style-specific expressive synthesis are the maximal *energy difference of an excerpt* (ED_E) and the *energy difference of average legato-level energy curve* (ED_{AL}). The range of the ED_E and ED_{AL} values is $[0, 1]$ due to the energy normalization.

4. SYNTHESIS AND RESULTS

Our experiment on expressive synthesis contains two tasks. The first task is to synthesize excerpts that are part of our dataset (i.e. excerpts that have been observed in the analysis stage), in order to test the feasibility of our proposed features and parameters to simulate the fine structure of the original sound. Here, we take Brahms' *Violin Concerto*, Mov. I, bars 129–132, and Sibelius' *Violin Concerto*, Mov. III, bars 133–135, both of which are the two excerpts taken as examples in the previous section. To receive a synthetic sound that looks like the original, the statistical parameters are from the excerpt itself. In other words, we calculate the parameters of each of the two excerpts independently, and apply them to synthesize the playing style of Heifetz or Oistrakh.

The second task is to synthesize an excerpt that is outside of the dataset, Beethoven's *Violin Sonata No.5*, Op.24, Mov. I, bars 1–10, which is also known as *Spring*. The parameters, however, are from the entire dataset, because we do not have any prior knowledge on the expressive parameters of *Spring*. In other words, we want to test via this task whether a computer is able to imitate the two violinists' playing style on an excerpt, which is not collected in the dataset.

The sources are obtained from the part of violin in the RWC database [17], and we particularly select the notes which have no vibrato. Therefore, the synthetic versions can be considered as if the two violinists use another violin to perform the same excerpts.

4.1. Synthesis method

The synthesis process is as follows:

- STEP 1 Find the DRs and the KOTs for every accent and every non-accent for each type of notes.
- STEP 2 Decide the energy contour of each individual note based on note type.
- STEP 3 Decide the mean energy for every accent and every non-accent based on the specific synthetic energy curve of the synthetic target.
- STEP 4 Decide which notes are vibrato and which notes are not.

In the first task, the first two steps are straightforward, because the synthetic parameters, the DR, KOT, and energy contour of each

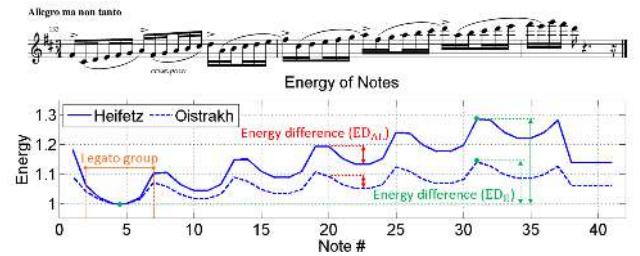


Figure 6: Synthetic energy curve of Sibelius' Violin Concerto, Mov. III, bars 133–135.

individual note, are from the original excerpt itself. In particular, to simplify the procedure, we cut the length of source, according to the specific DR for every accent and non-accent for every note type. As for STEP 3, the fine structure synthetic energy curve is somewhat complicated from the computed parameters. Based on the analysis of excerpt-level and legato-level energy curves, we know the parameters of ED_E and ED_{AL} as well as the convex-concave style rule of that excerpt. However, we need to determine the energy difference between successive legato groups and the energy of non-legato groups.

We have an interesting finding by investigating the excerpts, which are composed of several legato groups with increasing pitch height; that is, both violinists tend to play with increasing energy level among the legato groups for such excerpts. In other words, the mean energy of a legato group increases if its mean pitch is higher than its previous legato group, and vice versa. Therefore, we simply utilize this finding to create an energy curve that is either descending or ascending. Besides, to avoid a sudden change of synthetic energy curve between legato and non-legato groups within an excerpt, we assign the energy of every note of non-legato groups to be the mean value over all the legato groups.

In our implementation, the *synthetic energy curve* of an excerpt is computed as follows:

1. Set the curve as a flat one, implying that the energy of all the notes is the same at the beginning.
2. Decide the tendency of the curve according to the mean pitch of each legato group, and the maximal energy difference of the curve based on the ED_E value.
3. Decide the trend for every legato group according to the convex-concave style rule, and the energy difference of each legato group based on the ED_{AL} value.
4. Decide the energy for every note of non-legato groups by taking the average across all the notes of legato groups.
5. Modify the energy for every accent via multiplied by AER value.

Moreover, we set the minimal valley value of each synthetic energy curve to be one, making all the synthetic sounds at the same baseline. Figure 6 shows the synthetic energy curve of Sibelius' *Violin Concerto*, Mov. III, bars 133–135.

Next, in STEP 4, following Yang *et al.* [16], we decide whether a note is a vibrato note based on the vibRatio value. That is, all the notes within an excerpt are sorted in descending order of duration, and the top longest notes are assigned to have vibrato. The exact number of vibrato notes within an excerpt is calculated by the multiplication of the vibRatio value and the total number of

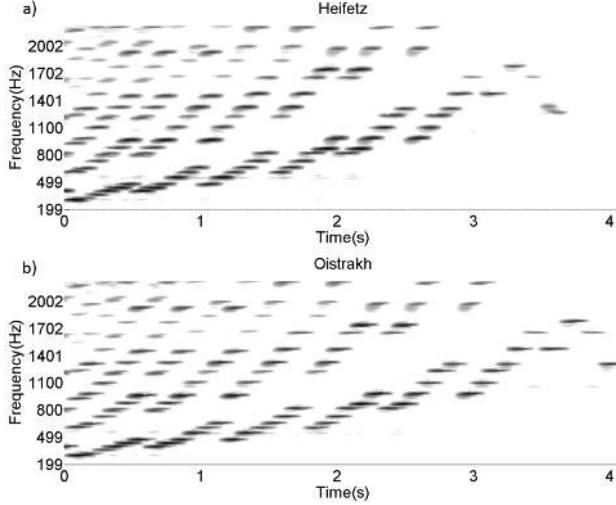


Figure 7: Synthesis result of Sibelius’ Violin Concerto, Mov. III, bars 133-135.

notes in such excerpt. We adopt the vibrato manipulation procedure proposed by Yang *et al.* [16] to synthesize a vibrato note. It is implemented by partitioning a non-vibrato note into a sequence of short fragments, shifting the pitch of each fragment via the phase vocoder to fit the specific vibrato contour, which is constructed by the VR and VE values, and overlapping and adding the fragments to obtain a smooth and natural vibrato sound. Finally, we obtain a synthesis result based on above four steps.

In the second task, there is no prior knowledge on the expressive parameters of *Spring*, which is outside of the dataset, so we take the average of each parameter across all the excerpts at the beginning. However, we find that the synthetic version does not resemble the two performers at all, because the characteristics of some excerpts are diverse, affecting the statistical parameters. To solve this problem, we narrow down the scope of excerpts to those, whose beat per minute (BPM) is around 120, according to the *Allegro* notation on the score of *Spring*. Specifically, we choose 18 excerpts from the dataset, take the mean value of each parameter over such ones, and then apply them to the synthesis system. The synthesis process is the same as the first task.

4.2. Synthesis result

Figure 7 shows the synthesis result of Sibelius’ Violin Concerto, Mov. III, bars 133-135, whose corresponding synthetic energy curve is shown in Figure 6. The characteristic of accents of both the masters are not obvious; however, Heifetz has clear properties of the KOT, EDE, and ED_{AL}, implying that Heifetz uses faster tempo and stronger strength to perform this excerpt than Oistrakh.

The score and synthetic energy curve of Brahms’ Violin Concerto, Mov. I, bars 129-132, are shown in Figure 8. Although the excerpt only has two legato notations on the score, we consider the remainder notes, except for the last four notes, as one legato group, resulting in three legato groups and one non-legato group. There are three concave trends on the synthetic energy curve, because the number of notes of all the three legato groups is more than eight. Besides, Oistrakh has higher energy in his performance, based on the larger EDE value. As illustrated in Figure 9, Heifetz performs

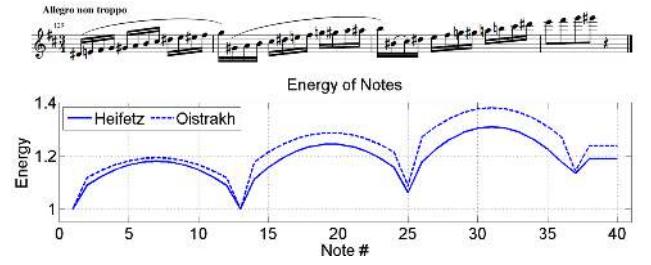


Figure 8: Synthetic energy curve of Brahms’ Violin Concerto, Mov. I, bars 129-132.

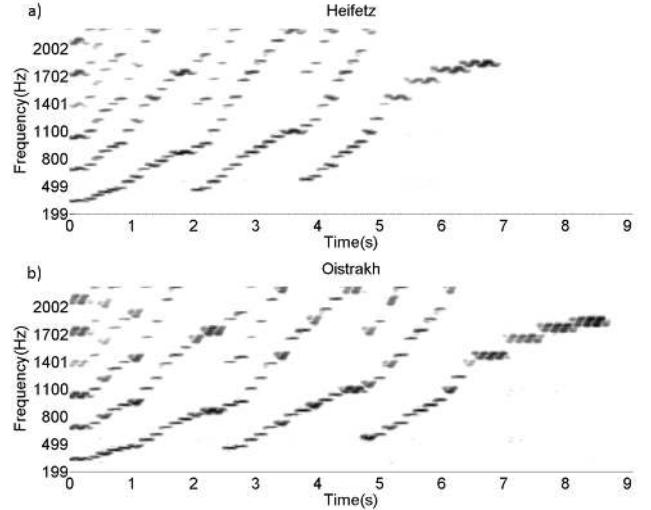


Figure 9: Synthesis result of Brahms’ Violin Concerto, Mov. I, bars 129-132.

in faster tempo, especially in the three sequences of short-duration notes, due to his larger KOT value. On the other hand, Oistrakh has strong properties of accents and vibrato notes, especially the last four ones. The former are due to his larger DR and smaller KOT of the notes before accents, and the latter are due to his larger VR, VE, and vibRatio.

Figure 10 shows the score and synthetic energy curve of the *Spring*. To simplify the synthesis procedure, we make an exception to the definition of legato notations of the bar 9; that is, all the notes of this bar are regarded as one legato group. Therefore, except for the first two bars and the last one, the others have convex trends on the synthetic energy curve, according to the convex-concave style rule. Moreover, Heifetz plays louder than Oistrakh based on the larger EDE value. The synthesis result of the first four bars is shown in Figure 11, in order to clearly observe the spectrogram. Apparently, Heifetz performs in faster tempo based on the larger KOT. Although the properties of vibrato between the two masters are similar during these four bars, Oistrakh actually has larger vibRatio, indicating that Oistrakh has more vibrato notes within this excerpt than Heifetz does.

In addition, we adopt some post-processings for the original recordings and the synthesis results, to compare them. The frequency band which is under the lowest note of an excerpt, of each original recording is removed, to avoid the influence of the back-

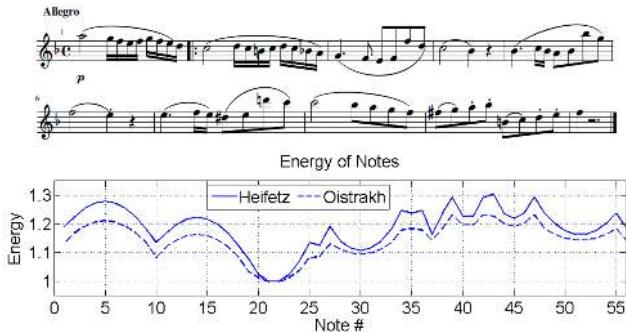


Figure 10: Synthetic energy curve of *Spring*.

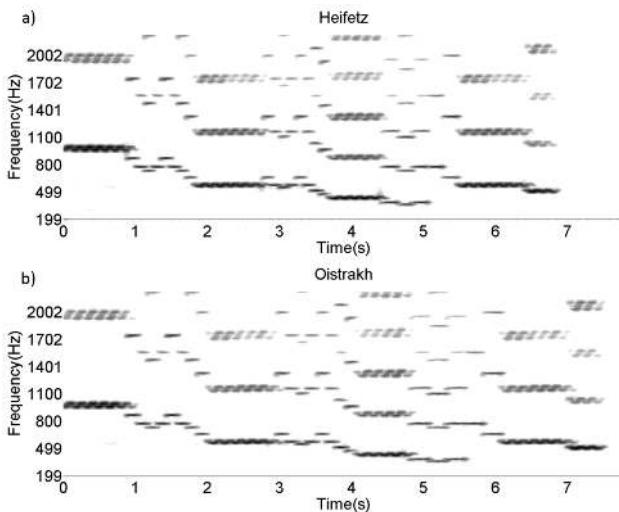


Figure 11: Synthesis result of *Spring*. To show the detail of spectrogram, it only contains bars 1-4.

ground accompaniment. Besides, the energy of each synthesis result is modified to be the same as the original version, whose lower frequency band has also been removed. This way, the audience can focus on the comparison between the original and synthetic sounds in terms of the playing style of the two masters. The sound samples can be found online.²

5. DISCUSSION

In this study, we give a quantitative analysis method on the performance of Heifetz and Oistrakh, using the style-related features. Then, we apply these features in expressive synthesis of the playing style of the two performers. As a first attempt to use signal-level features, score information as well as detailed annotations in analyzing this problem, we list some technical limitations of this work which could be left as future work.

The first limitation is the size of the dataset. Since there are only the 26 excerpts from 6 violin concertos in the dataset, it is not enough to cover all kinds of musical style, playing techniques and expressions. Besides, only 10 excerpts can be used in the legato

²<http://screamlab-ncku-2008.blogspot.tw/2017/03/synthesis-result-of-analysis-and.html>

analysis, and some of the energy changes in these excerpts are even constricted because of the symbol on score, which prevents us from observing the rule of energy change. Also, limited size of accent data reduces the reliability of accent modeling between the two violinists, because only the KOT and DR have such statistics. This problem also occurs in the statistics of note type. Furthermore, the dataset only contains 2 excerpts with staccato notes, making it hard to properly model the staccato notes in *Spring*; the parameters have to be calculated from the non-staccato notes.

The second limitation is the capability of vibrato-related features in the synthesis stage. Since we have not fully analyzed the vibrato features of the two masters, the vibrato features are not mentioned in Section 3.2. Although vibrato is an important factor in violin performance, the choice of vibrato notes, the onset time, the vibrato amplitude, etc., have not been considered in this paper. Instead, we compute only the mean vibrato extent and the vibrato rate, and choose the vibrato notes based on the vibrato ratio in the original excerpts. How to better design useful vibrato features and to assign them in the synthesis stage is left as future work.

We point out two further issues in the synthesis step: energy curve construction and accent selection. In the energy analysis, we do not have enough data to classify the energy curves with various shapes. We can only describe the tendency with rules by number of notes and pitch, and synthesize the curve by the difference of maximal and minimal values, instead of the average energy curve models. This way, in turn, causes a stiff energy curve, so that it needs to be improved. For the accent selection, our criterion, which is the first and third beats of a bar (it will only be the first beat when in a triple meter), actually does not yield a realistic synthetic version, especially for *Spring*. The reason is that the energy of the last three bars is not smooth due to the interruption of the energy of accents. Therefore, we would like to find a better selection criterion of accent in the future.

In short, for future work, we will expand the dataset and reconstruct it with no background noise by inviting violinists to imitate Heifetz and Oistrakh, and incorporate more information of energy, vibrato, and accent into expressive synthesis. It is also hoped that the same methodology can be applied to other violinists other than Heifetz and Oistrakh.

6. CONCLUSION

In this study, to our best knowledge, we compile the first manually annotated dataset on the masterpieces of Heifetz and Oistrakh. The annotation contains note-level information such as onset, offset, accent, legato, note type, and so on. We have presented the features distinguishing Heifetz and Oistrakh, and a method to synthesize music pieces based on the analyzed data. The process of expressive synthesis makes use of three signal-level features, including articulation, energy and vibrato, as well as including information from score like accent and legato group. Our result demonstrates the distinct difference between the two performers with respect to the duration, key overlap time, and accent.

7. ACKNOWLEDGMENTS

The authors would like to thank the Ministry of Science and Technology of Taiwan for its financial support of this work, under contract MOST 103-2221-E-006-140-MY3.

8. REFERENCES

- [1] Sandrine Vieillard, Mathieu Roy, and Isabelle Peretz, “Expressiveness in musical emotions,” *Psychological research*, vol. 76, no. 5, pp. 641–653, 2012.
- [2] Roberto Bresin, “Articulation rules for automatic music performance,” in *Proc. of the International Computer Music Conference 2001 (ICMC)*, 2001.
- [3] Esteban Maestre, Rafael Ramírez, Stefan Kersten, and Xavier Serra, “Expressive concatenative synthesis by reusing samples from real performance recordings,” *Computer Music Journal*, vol. 33, no. 4, pp. 23–42, 2009.
- [4] Roberto Bresin and Giovanni Umberto Battel, “Articulation strategies in expressive piano performance analysis of legato, staccato, and repeated notes in performances of the andante movement of Mozart’s sonata in G Major (K 545),” *Journal of New Music Research*, vol. 29, no. 3, pp. 211–224, 2000.
- [5] Akihiro Matsutani, “Study of relationship between performed tones and force of bow holding using silent violin,” *Japanese journal of applied physics*, vol. 42, no. 6R, pp. 3711, 2003.
- [6] Tracy Kwei-Liang Ho, Huann-shyang Lin, Ching-Kong Chen, and Jih-Long Tsai, “Development of a computer-based visualised quantitative learning system for playing violin vibrato,” *British Journal of Educational Technology*, vol. 46, no. 1, pp. 71–81, 2015.
- [7] Andreas P Baader, Oleg Kazennikov, and Mario Wiesendanger, “Coordination of bowing and fingering in violin playing,” *Cognitive brain research*, vol. 23, no. 2, pp. 436–443, 2005.
- [8] Ramon Lopez De Mantaras and Josep Lluis Arcos, “AI and music: From composition to expressive performance,” *AI magazine*, vol. 23, no. 3, pp. 43, 2002.
- [9] Alexis Kirke and Eduardo Reck Miranda, “A survey of computer systems for expressive music performance,” *ACM Computing Surveys (CSUR)*, vol. 42, no. 1, pp. 3, 2009.
- [10] Pei-Ching Li, Li Su, Yi-Hsuan Yang, and Alvin WY Su, “Analysis of expressive musical terms in violin using score-informed and expression-based audio features.,” in *Proc. of the 16th International Society for Music Information Retrieval Conference (ISMIR)*, 2015, pp. 809–815.
- [11] Rafael Ramirez, Esteban Maestre, Alfonso Perez, and Xavier Serra, “Automatic performer identification in celtic violin audio recordings,” *Journal of New Music Research*, vol. 40, no. 2, pp. 165–174, 2011.
- [12] Eitan Ornoy, “Recording analysis of JS Bach’s G Minor Adagio for solo violin (excerpt): A case study,” *Journal of Music and Meaning*, vol. 6, pp. 2–47, 2008.
- [13] Heejung Lee, *Violin portamento: An analysis of its use by master violinists in selected nineteenth-century concerti*, VDM Publishing, 2009.
- [14] Fadi Joseph Bejjani, Lawrence Ferrara, and Lazaros Pavlidis, “A comparative electromyographic and acoustic analysis of violin vibrato in healthy professional violinists,” 1990.
- [15] Jae Won Noella Jung, *Jascha Heifetz, David Oistrakh, Joseph Szigeti: Their contributions to the violin repertoire of the twentieth century*, Ph.D. thesis, Florida State University, 2007.
- [16] Chih-Hong Yang, Pei-Ching Li, Alvin WY Su, Li Su, and Yi-Hsuan Yang, “Automatic violin synthesis using expressive musical term features,” in *Proc. of the 19th International Conference on Digital Audio Effects (DAFx)*, 2016.
- [17] Masataka Goto, Hiroki Hashiguchi, Takuichi Nishimura, and Ryuichi Oka, “RWC music database: Database of copyright-cleared musical pieces and instrument sounds for research purposes,” *Transactions of Information Processing Society of Japan(IPSJ)*, vol. 45, no. 3, pp. 728–738, 2004.

A NONLINEAR METHOD FOR MANIPULATING WARMTH AND BRIGHTNESS

Sean Enderby

Digital Media Technology Lab
Birmingham City University
Birmingham, UK
sean.enderby@mail.bcu.ac.uk

Ryan Stables

Digital Media Technology Lab
Birmingham City University
Birmingham, UK
ryan.stables@bcu.ac.uk

ABSTRACT

In musical timbre, two of the most commonly used perceptual dimensions are *warmth* and *brightness*. In this study, we develop a model capable of accurately controlling the *warmth* and *brightness* of an audio source using a single parameter. To do this, we first identify the most salient audio features associated with the chosen descriptors by applying dimensionality reduction to a dataset of annotated timbral transformations. Here, strong positive correlations are found between the centroid of various spectral representations and the most salient principal components. From this, we build a system designed to manipulate the audio features directly using a combination of linear and nonlinear processing modules. To validate the model, we conduct a series of subjective listening tests, and show that up to 80% of participants are able to allocate the correct term, or synonyms thereof, to a set of processed audio samples. Objectively, we show low Mahalanobis distances between the processed samples and clusters of the same timbral adjective in the low-dimensional timbre space.

1. INTRODUCTION

The perception and manipulation of musical timbre is a widely studied aspect of sound production. This is because timbre, unlike pitch and loudness, is difficult to measure linearly along a single intuitive dimension. This means the focus of timbral research is often on the use of dimensionality reduction [1, 2] as a method of interpreting some complex representation in timbre space. In the study of musical timbre, natural language is often used to define perceptual dimensions [3]. This is a method of quantifying descriptions of sound, often through the use of correlation with audio features [4]. In music production, this descriptive vocabulary can also be used to define sound transformations [5]. This means we are able to control audio processing functions using parameters that are intuitive to the user as they represent high-level perceptual aspects of sound, as opposed to low-level algorithm parameters. For example, SocialEQ [6] allows participants to select a descriptive term, then to derive its spectral representation by rating a series of examples. Similarly, the SAFE Project [7] allows users to describe audio effects transformations directly within a Digital Audio workstation (DAW), which can then be processed and recalled to crowd-source processing parameters.

One of the most common perceptual dimensions in timbral research is the warmth / brightness dimension [8, 9]. This is because participants often tend to agree with confidence on the statistical representation of the two descriptive terms [4], and they are often considered to be orthogonal [10]. Because of this, a number of studies have focussed specifically on manipulating this dimension for musical purposes. For example, Stasis et al. [11, 10] provide a 2-dimensional interface, derived using machine learning,

Zacharakis et al. [12] present a model using FM-Synthesis, and Williams and Brookes [13, 14] provide a timbre morphing system, capable of independently modifying warmth and brightness.

In this study, we identify the key audio features associated with warmth and brightness from a dataset of annotated musical transforms. We then propose an audio effect that combines linear and nonlinear modules in a way that allows us to manipulate the audio features directly associated with the perception of warmth and brightness in a sound source. To validate the performance of the effect, we then provide both objective and subjective validation.

2. PERCEPTUAL DIMENSIONS

To identify salient audio features associated with warmth and brightness, we compile a matrix of timbral data collected via the SAFE Project¹. Here, annotated audio effect transformations are collected from within a DAW environment and uploaded anonymously. Each instance contains a set of plug-in processing parameters, an extensive feature set taken before and after processing has been applied, a string of descriptive terms, and a table of user metadata. As shown in [4], warmth and brightness tend to be related to changes in the spectral envelope associated with equalisation and distortion, so we therefore discard entries from compression and reverb effects. This leaves us with 1,781 semantically annotated transforms. As the tests are distributed over a wide network we do not have extensive data about the test participants, however it is assumed that each of the users of the system have a reasonable level of production experience. To capture the timbral modification applied by each of the transforms, we analyse the feature differences over a range of temporal, spectro-temporal and abstract statistical audio features.

2.1. Features

To identify the most salient features associated with the *warmth* / *brightness* dimension, we apply dimensionality reduction to the dataset using Principal Component Analysis (PCA) and identify the most highly correlated features with the Principal Components (PCs) that explain the highest variance. This is demonstrated in Figure 1, in which the first two PCs of the timbre space describing the feature manipulations are shown. Here, the audio feature differences of each described transform is projected into two dimensional space, and the centroid for each term is found. The size of the term indicates its relative confidence (a measure which is inversely proportional to the variance within a cluster). These confidence scores for entries in the dataset are given in Table 1. Additionally, Figure 2 shows the isolated transforms described as either

¹Data and plug-ins available at: www.semanticaudio.co.uk

warm or bright. This shows the distribution of points from each class are separable along PC 2. In the other PCs, these descriptors occupy very similar ranges, suggesting that the distinction between warm and bright is heavily loaded onto the second PC. To identify the salient features associated with each dimension, we correlate each feature vector with the first two PCs. The audio features with correlations which satisfy $|r| > .7$ and $p < .001$ are shown below.

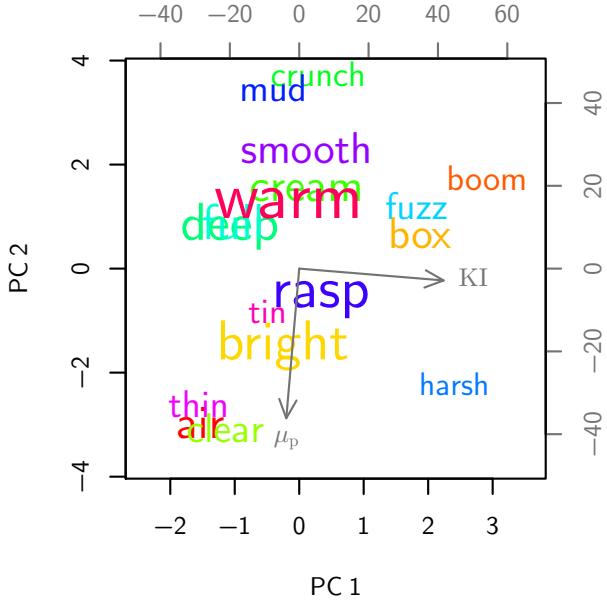


Figure 1: A biplot of the feature difference timbre space, where μ_p and KI represent the peak spectral centroid and spectral irregularity projected into low-dimensional space.

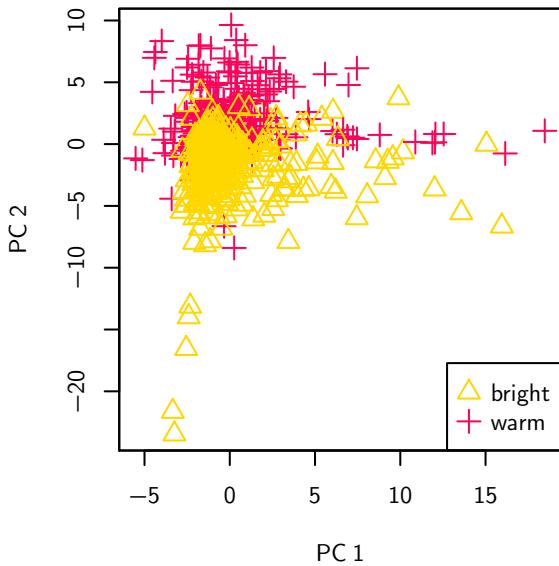


Figure 2: Transforms labelled with warm and bright in the feature difference timbre space.

Term	C	Term	C	Term	C
warm	1.5	box	0.7	boom	0.4
deep	1.4	clear	0.5	tin	0.4
bright	1.1	thin	0.5	crunch	0.3
full	0.9	mud	0.4	harsh	0.2
air	0.8	fuzz	0.4	smooth	0.1
cream	0.8	rasp	0.4		

Table 1: The confidence scores (C) for terms in the feature difference timbre space.

PC 1: Irregularity ($r = 0.985$), Irregularity_P ($r = 0.964$), Kurtosis_P ($r = 0.935$), Skew_S ($r = 0.929$), Irregularity_H ($r = 0.927$), Kurtosis_H ($r = 0.919$), Skew_P ($r = 0.890$), Std ($r = 0.873$), RMS ($r = 0.873$), Skew_H ($r = 0.865$), Kurtosis_S ($r = 0.835$), Var ($r = 0.812$).

PC 2: Centroid_P ($r = -0.855$), Centroid_H ($r = -0.853$), Rolloff_S ($r = -0.852$), Std_H ($r = -0.845$), Std_P ($r = -0.834$), Centroids_S ($r = -0.817$), Slope_S ($r = -0.771$).

Where, the subscript s denotes features extracted from a magnitude spectrum, p denotes features taken from a peak spectrum, and h denotes features taken from a harmonic peak spectrum. Features with no subscript are either temporal or abstract spectral features. Spectral irregularity in this study is calculated using the method described by Krimpoff et al. [15].

These results indicate that the dimension along which warmth and brightness can be considered separable (PC 2), is highly correlated with spectral centroid, spectral standard deviation and spectral roll off. Negative values of PC 2 correspond to an increase in these features and positive values to a decrease. Signals can therefore be made warmer by reducing the spectral centroid and ‘brighter’ by increasing it. This is most often done by introducing more energy at low or higher frequencies respectively. These findings are aligned with similar studies of musical timbre such as [8, 2], in which the spectral centroid of a sound source is demonstrated to be a salient low-level descriptor in the perception of both warmth and brightness.

2.2. Synonyms

Given that the dataset has a large number of descriptive terms, we apply agglomerative clustering to the feature space in order to identify potentially synonymous terms in the dataset. This allows us to judge the relative distances between data points in this space, thus providing a method of evaluating dissimilarities during subjective evaluation. Terms with less than 4 entries are omitted for readability and the distances between data points are calculated using Ward criterion [16]. The clusters are illustrated in Figure 3, where the prefix E : represents transforms taken from the equaliser and D : represents transforms taken from the distortion. The position, $\mu_{d,k}$ of a term, d , in the k^{th} dimension of the audio feature space is calculated as the mean value of feature k across all N_d transforms labelled with that descriptor, given in Eq. 1.

$$\mu_{d,k} = \frac{1}{N_d} \sum_{n=1}^{N_d} \bar{x}_{d,n,k} \quad (1)$$

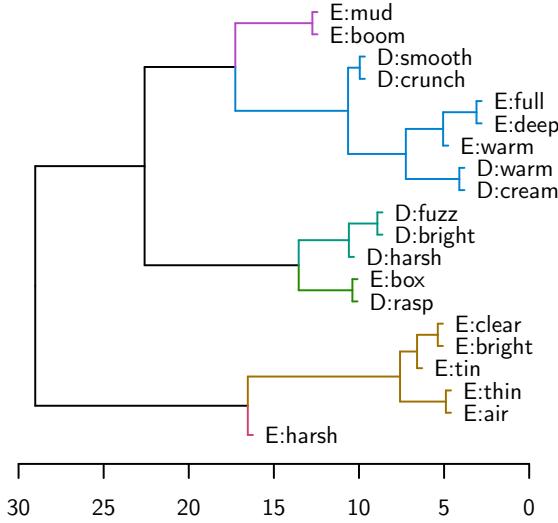


Figure 3: Clustering of descriptors from both the distortion and equaliser.

The agglomerative clustering process demonstrates that the data points tend to cluster based on relative spectral energy. For example, terms that typically describe signals with increased high frequency energy such as *bright*, *thin*, *tin* and *air* all have low cophenetic distances to each other. Similarly, terms typically associated with increased low frequency energy such as *boom*, *warm*, *deep* and *full* fall within the same cluster. In this case, the cluster containing *warm* and the cluster containing *bright* are clearly separated, with a minimum cophenetic distance of 22.6.

3. PARAMETERISATION OF SPECTRAL CENTROID

Given the correlation between the *warmth / brightness* dimension (PC 2) and spectral centroid, we investigate methods for manipulating the feature directly, thus being able to increase *brightness* by increasing the centroid, or increase *warmth* by lowering the spectral centroid. A primitive method for moving the centroid towards a given frequency (μ) is to increase the spectral energy at that frequency, either by introducing a sinusoidal component to the signal or to use a resonant filter, centered around μ . Whilst this works conceptually, it is destructive to the original structure of the spectrum. As the centroid is moved towards the desired frequency the spectrum is dominated by a sinusoid or resonance at μ .

Less destructive methods include that used by Zacharakis et al [12], where the spectrum is split into two bands, one above and one below the spectral centroid. The relative amplitudes of these bands can then be altered to manipulate the frequency of the spectral centroid. This more accurately preserves the original signal's structure, as no additional spectral components are introduced and the relative levels of partials within each band remain the same. Using this method the new spectral centroid will lie somewhere between the respective centroids of the two bands. The relative gains of the two bands required to reproduce a given spectral centroid μ , can be calculated using Equation 2. To facilitate precise control of the spectral centroid these bands should not share any frequency components.

$$\frac{\sum_{n=c+1}^N a_n}{\sum_{n=1}^c a_n} = \frac{\mu_l - \mu}{\mu - \mu_u}, \quad \mu_l \leq \mu < \mu_u \quad \text{or} \quad \mu_u < \mu \leq \mu_l \quad (2)$$

Where μ_l and μ_u are the spectral centroid of the lower and upper bands, and c is the highest frequency spectral component in the lower band.

Alternatively, Brookes et al. [13] employ a spectral tilt to modify spectral centroid, applying gain to the partials of a signal as a linear function of their frequency. This allows the spectral centroid to be altered in frequency, whilst still retaining the frequency content of the signal. A disadvantage of this method is that the change in centroid cannot be easily parameterised as it depends on the content of the signal being processed.

3.1. Proposed Model

We propose a more flexible method for directly manipulating the spectral centroid of the input signal using a nonlinear device (NLD). The effects of an NLD are more easily predicted for sinusoidal inputs, generating a series of harmonics of the input frequency. To ensure a sinusoidal input to the NLD, the system is restricted to processing only tonal signals with a single pitch, which can be represented by their fundamental frequency (f_0). A low-pass filter is applied to isolate the f_0 which is then processed with an NLD generating a series of harmonics relevant to the signal. This is then high-pass filtered, leaving a band that consists solely of generated harmonics. A second band is generated by low-pass filtering the signal at the spectral centroid and the relative levels are then adjusted in order to manipulate the frequency of the centroid μ_s . Separating the bands at the spectral centroid in this way ensures that their respective spectral centroids sit either side of the input's centroid after processing has been applied. In this instance, we generate harmonics in the high frequency band by applying Full Wave Rectification (FWR) to the isolated fundamental, this ensures the system is positive homogeneous. A schematic overview of the system is presented in Figure 4.

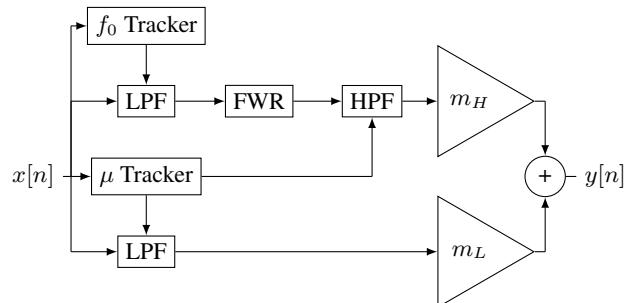


Figure 4: The system employed in the warmth / harshness effect.

This is conceptually similar to the two band method proposed by [12], but allows the second band to contain frequency content which was not in the original signal. This method has advantages over manipulating the amplitudes of two bands using only linear filters. For example, the nonlinear device can still be used to reconstruct the high frequency portion of the signal and the relative gains adjusted similarly to if two filters were used. Alternatively, the properties of the NLD can be altered to change the upper band's centroid. This provides more flexibility allowing the centroid to

be changed independently of some other features. For example, changing the gains of two bands will change the spectral slope of the signal. If instead additional partials are introduced to the upper band, with amplitudes which are determined by the signal's current slope, the centroid can be changed, whilst the slope is unaltered.

The effect is controlled using a single parameter p , which ranges from 0 to 1 and is used to calculate the respective gains m_L and m_H , applied to the low and high frequency bands using Equation 3.

$$\begin{aligned} m_H &= p^3 \\ m_L &= 1 - m_H \end{aligned} \quad (3)$$

When $p = 0$ the output is a low pass filtered version of the input signal resulting in a lower spectral centroid than the input. This corresponds to transforms described as *warm* in the SAFE dataset. When $p = 0.5$ additional harmonic energy is introduced into the signal, meaning the transform should be perceived as *bright*. To achieve this, the exponent in p^n was set experimentally, so that the Mahalanobis distance between the *bright* cluster and the transform's feature differences is minimised. When $p = 1$ the output signal consists primarily of high order harmonics resulting in an extreme increase in spectral centroid. This is perceived as Harshness, which in the SAFE dataset is defined as an increase in spectral energy at high frequencies, as shown in Figure 5.

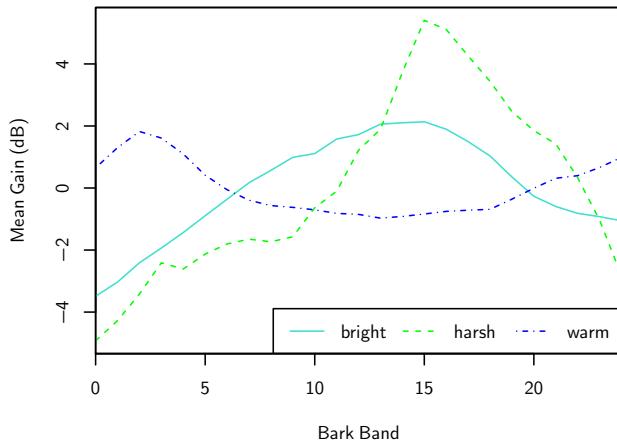


Figure 5: Mean Bark spectra of transforms labelled *warm*, *bright* and *harsh* in the SAFE dataset. Here, Bark spectra are used to represent the spectral envelopes of the transforms as these are the only spectral representations collected in-full by the SAFE plug-ins due to data protection.

When transforms are applied using a distortion, *bright* and *harsh* are considered to be very similar, with a low cophenetic distance of 10.6 (see Figure 3), however they exhibit some subtle differences in transforms taken from the equaliser, with a cophenetic distance of 16.5. This is demonstrated in Figure 1, where *harsh* sits below *bright* in PC 2.

4. MODEL VALIDATION

The performance of the effect is evaluated using a set of ten test signals, comprising two electric bass guitars (B1 and B2), a flute

(F), two electric guitars (G1 and G2), a marimba (M), an oboe (O), a saxophone (S), a trumpet (T) and a violin (V). The signals were adjusted to have equal loudness prior to experimentation. Firstly the effects are evaluated objectively by comparing them to the analysis performed in section 2. Secondly the effects are evaluated subjectively through a series of perceptual listening tests.

4.1. Objective Evaluation

The performance of the effect is evaluated objectively by examining how it manipulates the features of the test signals. The effect is used to process each of the signals with its parameter p set to 0 (*warm*), 0.5 (*bright*) and 1 (*harsh*). The audio features of the unprocessed and processed signals in each of these applications are calculated in the same manner as in the SAFE plug-ins. These audio features are then compared to those taken from the SAFE dataset.

Each combination of descriptor and test signal is measured to find the distance between changes in the feature space caused by the effect, and points labelled with the descriptor in the feature difference timbre space. The performance of the effect, with a particular parameter setting on a particular test signal is measured by projecting the extracted audio features to a point in the timbre space. The Mahalanobis distance, $M(x, d)$, between this point x , and the distribution of transforms labelled with the relevant term d , is taken using Equation 4

$$M(x, d) = \sqrt{(x - \mu_d)^T \Sigma_d^{-1} (x - \mu_d)} \quad (4)$$

Where x is a column vector containing the coordinates of the point in the timbre space, μ_d is a column vector containing the mean coordinates of all transforms in the timbre space labelled with descriptor d and Σ_d is the covariance matrix of those transforms' coordinates in the timbre space. Where there are more than five transforms in the distribution, the coordinates in the first five PCs of the timbre space are used. Where the number of points in the distribution, N_d , is lower, only the first $N_d - 1$ coordinates can be used in order to avoid Σ_d being singular. Where the descriptor, d , is represented by two distributions of transforms, one from the distortion and one from the equaliser, the Mahalanobis distance from both distributions is taken and the minimum distance is considered the measure of performance.

4.2. Subjective Evaluation

To assess the performance of the effect subjectively, a series of perceptual listening tests were undertaken. For the purposes of testing, the *warmth / brightness* effect was implemented a DAW plug-in. Participants were presented with a DAW session containing a track for each of the test signals. The plug-in was used on each track with a simple interface labelled “Plug-In 1”, shown in Figure 6. To mitigate influence of the plug-in’s interface layout on the result of the experiment, the direction of the parameter slider was randomised for each participant. The order of the tracks in the DAW session was also randomised to mitigate any effect the order of tracks may have on results.

For each signal, participants were asked to first audition the effect to become accustomed to how changing the parameter value affects that particular input signal. Once they had investigated the operation of the effect they were asked to label the parameter slider at three positions (p is equal to 0, 0.5 and 1) with a term they



Figure 6: The interface used for assessing the performance of the warmth / brightness effect.

felt best described the timbre of the effect at that parameter setting. A list of available terms was provided in a drop down list at each of the 3 parameter values to be labelled, pictured in Figure 6. The available terms were *airy*, *boomy*, *boxy*, *bright*, *clear*, *creamy*, *crunchy*, *deep*, *full*, *fuzzy*, *harsh*, *muddy*, *raspy*, *smooth*, *thin*, *tinny* and *warm*. These were chosen for their confidence scores and number of entries in the SAFE dataset. For each combination of signal and parameter positions, there is an intended descriptor (those the effects were designed to elicit) and a descriptor provided by the participant.

We compare the participant responses against the hierarchical clustering performed in Section 2.2. The dendrogram shown in Figure 3 provides information about how similar the transforms described by certain terms are. This information can be used as a metric describing the proximity of the users' responses to the intended response. The proximity of two descriptors is measured as the cophenetic distance between the clusters in which the two descriptors lie. Where a descriptor appears twice in the dendrogram (from both the distortion and equaliser) the combination of points which yield the lowest cophenetic distance is used. All listening tests were undertaken using circumaural headphones in a quiet listening environment. In total 22 participants took part in the listening tests, all of whom reported no known hearing problems. On average participants took 25 minutes to complete the test.

5. RESULTS / DISCUSSION

5.1. Objective Evaluation

The Mahalanobis distances between the test signals after being processed by the effect and the distributions of corresponding transforms in timbre space are shown in Figure 7. Here, each of the instrument samples is processed using $p = 0$ (*warm*), $p = 0.5$ (*bright*) and $p = 1$ (*harsh*).

The results show that overall, the *warmth* setting is timbrally very similar to the corresponding entries into the SAFE dataset, with a mean Mahalanobis distance of 1.03 ($\sigma = 0.53$). *Bright* samples are also very similar, with a mean distance of 1.36 ($\sigma = 0.36$). *Harshness* however is less similar to the distribution of terms in the dataset $\mu = 2.41$, $\sigma = 1.71$. This is potentially due to the term's ambiguity, and relatively low-confidence. *Harshness* for distortion and *harshness* for equalisation, for example, fall into different groups when hierarchical clustering is applied (see Figure 3). Also, due to the relative number of dataset entries for each term (*warm* = 464 entries, *bright* = 427 entries, *harsh* = 8 entries), smaller distances from the *harsh* distribution are deemed

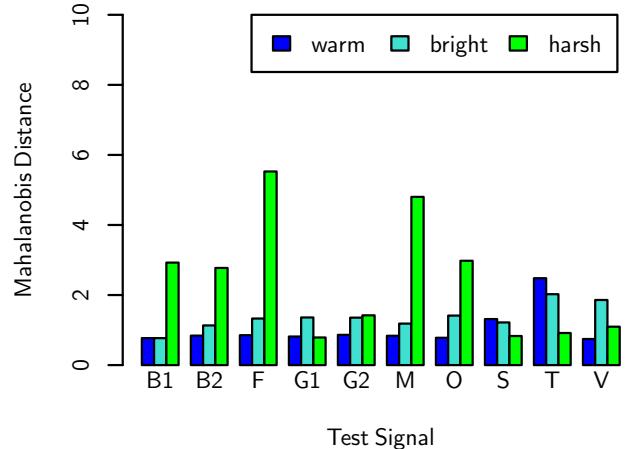


Figure 7: Mahalanobis distances for the warmth / brightness effect.

to be more statistically significant.

5.2. Listening test Results

The mean cophenetic distances between the participants' annotations of the effect's parameters and the descriptors *warm*, *bright* and *harsh* taken from the SAFE dataset are shown in Figure 8. Here the error bars represent the 95% confidence intervals for each mean. To show the performance of each instrument sample, markers on the y-axis indicate the cophenetic distances that correspond to the cluster heights for the groups containing *bright* from the distortion, *bright* from the EQ, and *warm* from both plug-ins, as per Figure 3.

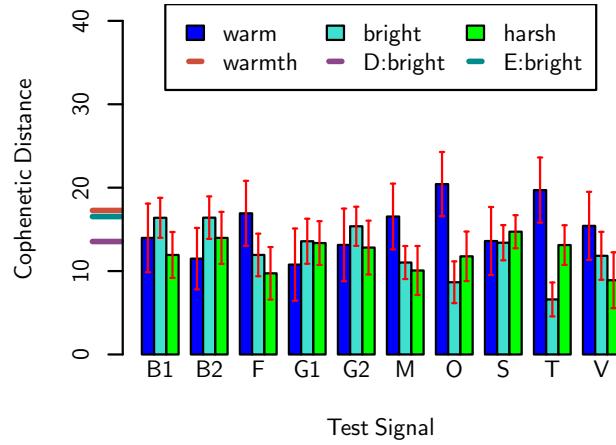


Figure 8: Cophenetic distances for the warmth / brightness effect.

The results show that almost all of the instrument samples have mean cophenetic distances that fall within the same cluster when the effect applies a *bright* or *harsh* transform. This means participants label parameter states with terms synonymous to the intended term. Samples processed to be *warm* also have similar sub-

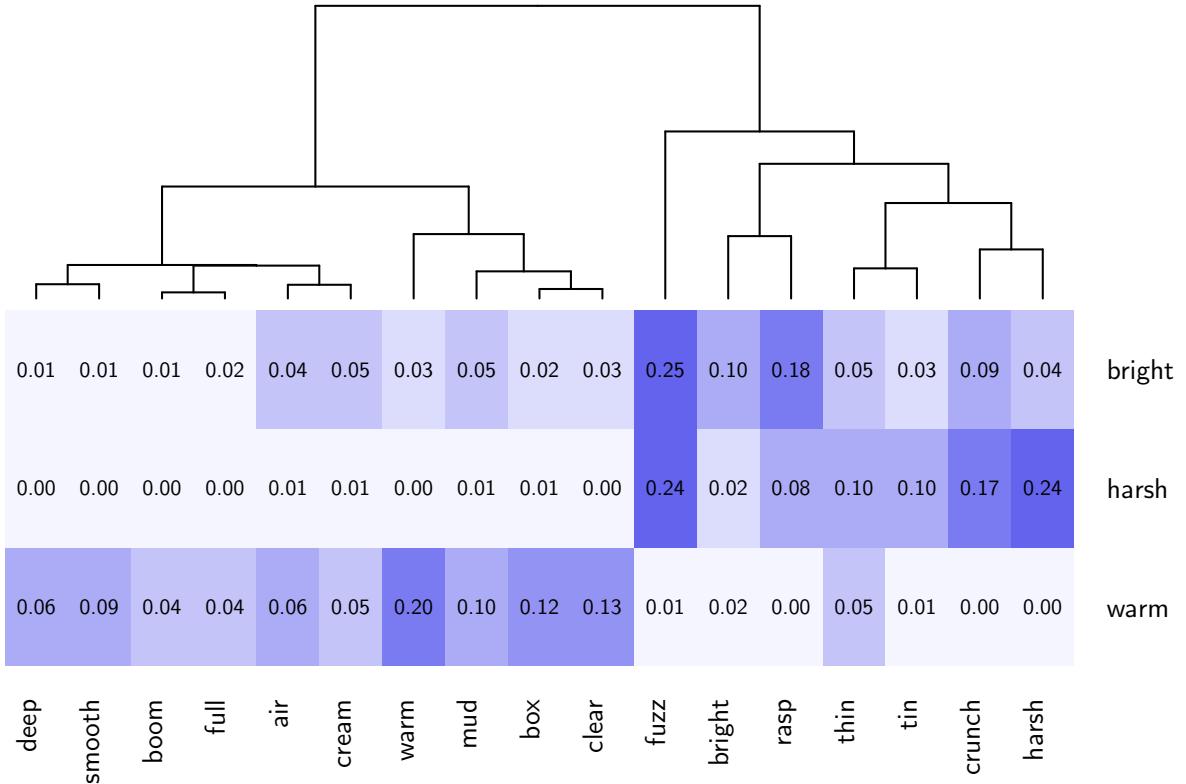


Figure 9: A matrix showing subjective mis-classifications from the warmth / brightness effect, organised by the frequency of the mis-classifications across the effect’s intended terms.

jective accuracies, however two of the instrument samples (oboe and trumpet) have mean distances which are larger than the cluster heights, suggesting there is more ambiguity in their definition.

Figure 9, shows a mis-classification matrix comparing the usage of terms by test participants and the terms that the effects were designed to produce. Each cell in the matrix represents the frequency of which each of the available descriptors (bottom) was used to describe the corresponding timbral effect at a given parameter position. Above the figure is a dendrogram representing the clustering of terms based on their frequency of usage.

From the figure, it is clear that *warm* is often correctly assigned to the intended transform, but summing the cells of the row shows that participants only used descriptors in the same cluster as *warm* 54% of the time. This suggests that the addition of low frequency energy to the signal does not necessarily invoke the use of synonyms of *warm*. When describing the effect, participants used a term related to the intended descriptor 74% of the time for *bright* and 80% of the time for *harsh*, suggesting that these transforms were perceptually more similar to the transforms in the dataset.

6. CONCLUSION

We first present empirical findings from our analysis of a dataset of semantically annotated audio effect transforms, and show that the *warmth / brightness* dimension can be loaded heavily onto a single PC. The variance of this PC is explained predominantly by the centroid of the peak, harmonic peak, and magnitude spectra. We

then present a model for the manipulation of timbral features using a combination of filters and nonlinear excitation, and show that the model is able to manipulate the respective *warmth* and *brightness* of an audio signal. We verify this by performing objective and subjective evaluation on a set of test signals, and show that subjects describe the transforms with synonymous descriptors 54%, 74% and 80% of the time for *warmth*, *brightness* and *harshness* respectively.

By using a NLD component in the feature manipulation process, we are able to increase the flexibility of the timbral modifier. This is because other audio features can be preserved, whilst the spectral centroid is modified independently. Conversely, the algorithm is currently limited to pitched monophonic sound sources due to its reliance on tracking the f_0 of the input signal. This issue will be addressed in future work.

7. REFERENCES

- [1] J. M. Grey, “Multidimensional perceptual scaling of musical timbres,” *the Journal of the Acoustical Society of America*, vol. 61, no. 5, pp. 1270–1277, 1977.
- [2] A. Zacharakis, K. Pasiadis, J. D. Reiss, and G. Papadelis, “Analysis of musical timbre semantics through metric and non-metric data reduction techniques,” in *Proceedings of the 12th International Conference on Music Perception and Cognition (ICMPC12)*, 2012, pp. 1177–1182.
- [3] M. Sarkar, B. Vercoe, and Y. Yang, “Words that describe

- timbre: A study of auditory perception through language,” in *Proceedings of the 2007 Language and Music as Cognitive Systems Conference*, 2007, pp. 11–13.
- [4] R. Stables, B. De Man, S. Enderby, J. D. Reiss, G. Fazekas, and T. Wilmering, “Semantic description of timbral transformations in music production,” in *Proceedings of the 2016 ACM on Multimedia Conference*. ACM, 2016, pp. 337–341.
 - [5] T. Zheng, P. Seetharaman, and B. Pardo, “Socialfx: Studying a crowdsourced folksonomy of audio effects terms,” in *Proceedings of the 2016 ACM on Multimedia Conference*. ACM, 2016, pp. 182–186.
 - [6] M. Cartwright and B. Pardo, “Social-eq: Crowdsourcing an equalization descriptor map,” in *ISMIR*, 2013, pp. 395–400.
 - [7] R. Stables, S. Enderby, B. De Man, G. Fazekas, and J. D. Reiss, “Safe: A system for the extraction and retrieval of semantic audio descriptors,” 2014.
 - [8] E. Schubert and J. Wolfe, “Does timbral brightness scale with frequency and spectral centroid?”, *Acta Acustica united with Acustica*, vol. 92, no. 5, pp. 820–825, 2006.
 - [9] J. Marozeau and A. de Cheveigné, “The effect of fundamental frequency on the brightness dimension of timbre,” *Journal of the Acoustical Society of America*, vol. 121, no. 1, pp. 383–387, 2007.
 - [10] S. Stasis, R. Stables, and J. Hockman, “A model for adaptive reduced-dimensionality equalisation,” in *Proceedings of the 18th International Conference on Digital Audio Effects, Trondheim, Norway*, 2015, vol. 30.
 - [11] S. Stasis, R. Stables, and J. Hockman, “Semantically controlled adaptive equalisation in reduced dimensionality parameter space,” *Applied Sciences*, vol. 6, no. 4, pp. 116, 2016.
 - [12] A. Zacharakis and J. Reiss, “An additive synthesis technique for independent modification of the auditory perceptions of brightness and warmth,” in *Audio Engineering Society Convention 130*, May 2011.
 - [13] T. Brookes and D. Williams, “Perceptually-motivated audio morphing: Brightness,” in *122nd Audio Engineering Society Convention*. Audio Engineering Society, 2007.
 - [14] D. Williams and T. Brookes, “Perceptually motivated audio morphing: warmth,” in *128th Audio Engineering Society Convention*, 2010.
 - [15] J. Krimphoff, S. McAdams, and S. Winsberg, “Caractérisation du timbre des sons complexes. ii. analyses acoustiques et quantification psychophysique,” *Le Journal de Physique IV*, vol. 4, no. C5, pp. C5–625, 1994.
 - [16] J. H. Ward Jr, “Hierarchical grouping to optimize an objective function,” *Journal of the American statistical association*, vol. 58, no. 301, pp. 236–244, 1963.

SOUNDSCAPE CATEGORISATION AND THE SELF-ASSESSMENT MANIKIN

Francis Stevens

AudioLab

Department of Electronic Engineering
University of York
York, UK
fs598@york.ac.uk

Damian T Murphy

AudioLab

Department of Electronic Engineering
University of York
York, UK
damian.murphy@york.ac.uk

Stephen L Smith

Intelligent Systems Group
Department of Electronic Engineering
University of York
York, UK
stephen.smith@york.ac.uk

ABSTRACT

This paper contains the results of a study making use of a set of B-format soundscape recordings, presented in stereo UHJ format as part of an online listening test, in order to investigate the relationship between soundscape categorisation and subjective evaluation. Test participants were presented with a set of soundscapes and asked to rate them using the Self-Assessment Manikin (SAM) and in terms of three soundscape categories: natural, human, and mechanical. They were also asked to identify the important sound sources present in each soundscape. Results show significant relationships between soundscape categorisation and the SAM, and indicate particularly significant sound sources that can affect these ratings.

1. INTRODUCTION

This paper presents the results of study investigating the relationship between soundscape preferences ratings made using the self-assessment manikin (SAM) [1] and three category ratings (natural, human, and mechanical). Following previous studies the SAM has now been established as an appropriate and powerful tool for the evaluation of soundscapes: for example in [2] where results from the SAM were compared with the use of semantic differential (SD) pairs; or in [3], where it was used to establish the ecological validity of stereo UHJ conversions of B-format soundscape recordings.

This paper presents additional results from the study presented in [3] exploring further the utility of the SAM in soundscape evaluation and research. Here the purpose was to investigate the relationship between soundscape preference ratings, and soundscape sound source identification and categorisation.

A soundscape can be considered as the aural equivalent of a landscape [4], and soundscape research is a field where environmental recordings are used and analysed to give an insight into a location's acoustic characteristics beyond noise level measurement alone. A soundscape can therefore be considered as being comprised of a set of individual sounds sources coming together in a particular context to form an environmental experience.

These sound sources can broadly be divided between three categories: natural, human, and mechanical. This test was designed in order to see how the presence of particular sounds belonging to each of these categories can affect the subjective rating of a soundscape in terms of emotional experience, and in terms of how the soundscape as a whole is perceived as belonging to each of these three categories.

It is hypothesised firstly that the soundscapes that are rated as being more mechanical will exhibit low valence and high arousal in the SAM results, and that highly natural soundscapes will exhibit high valence and low arousal. Soundscapes highly rated in the

human category are expected to exhibit high arousal with valence determined by contextual cues from the other sound sources present.

In terms of the sound sources identified it is hypothesised that birdsong and traffic will be the most commonly identified sources, and that traffic noise and human activity (e.g. conversation or footsteps), when present, will have a significant effect on the category rating.

This paper starts with a presentation of the methods used in the study, including the collection of the soundscape data used in the test, the stereo UHJ conversion process, and the development of the test procedure including the SAM and category questions.

The results of the test are then presented, starting with a brief overview of the demographics of the test participants. The SAM results are then examined, after which a summary of the category ratings for each presented soundscape is presented alongside the sound sources identified. Correlational analysis is then used to indicate the relationships between the different metrics: firstly comparing the SAM results with the category ratings; then comparing category ratings with the percentages of the sound sources identified in each case that belong to each of the three categories. The paper concludes with a consideration of avenues for further research.

2. METHODS

This section begins with an examination of the soundscape data collected for this study, including the motivation behind the choice of locations where the recordings were made and the contents of those recordings. The conversion of the recorded B-format soundscapes to stereo UHJ format is considered before a presentation of the assessment methods used in the listening test: the SAM, and the soundscape categorisation and sound source identification questions, and the overall test procedure.

2.1. Data Collection

The data used in this study were collected from various locations around Yorkshire in 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 B-format using a Soundfield STM 450 microphone [5].

Table 1 gives details of the sound sources present in each of the 16 clips used in the listening test. Each of these clips were 30 seconds long and extracted from the 10 minutes of soundscape recording made at each location. These clips were used in their original B-format in a previous study [2].

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.

2.2. Stereo UHJ Conversion

In order to present the recorded soundscape material online, the B-format signals had to be converted to a suitable two-channel format. It was decided to make use UHJ stereo format, where the **W**, **X**, and **Y** channels of a B-format recording are used to translate the horizontal plane of the soundfield into two-channel UHJ format [6].

This is a two channel format that can be easily shared online and reproduced over headphones, allowing the B-format recordings presented in a previous study to be used here with the spatial content of the **W**, **X**, and **Y** channels preserved in reproduction.

The following equations are used to convert from the **W**, **X**, and **Y** channels of the B-format 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 [7]. Note that the Cartesian reference for B-format signals is given by ISO standard 2631 [8], and the **Z** channel of the B-format recording is not used.

2.3. Assessment

This section will cover the assessment methods used in this study, starting with the SAM, followed by the category rating and sound source identification questions. This section concludes with a summary of the overall test procedure.

2.3.1. The Self-Assessment Manikin

A previous study [2] 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 [9]. 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 [10–17]. Whilst useful in certain scenarios, they can be time-consuming and unintuitive [2]. 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, 20] and visual stimuli [1] (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.

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 [2].

2.3.2. Category Rating

The soundscape recordings used in this test (as detailed in Table 1) were selected in order to cover as wide a range of soundscapes 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 [17, 21–25] three main groups of sounds are identified:

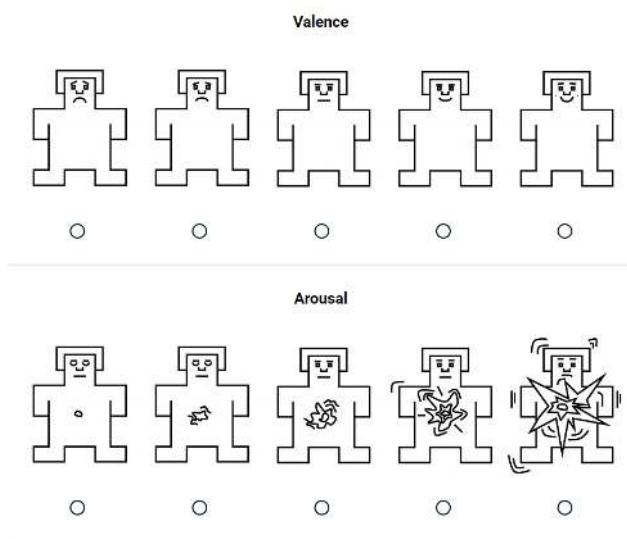


Figure 1: The Self-Assessment Manikin (SAM) as used in this study, after [18].

- **Natural sounds:** These include animal sounds (bird song is an oft-cited example), and other environmental sounds such as wind, rustling leaves, and flowing water.
- **Human sounds:** 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:** Sounds such as traffic noise, industrial and construction sounds, and aeroplane noise.

The aim of this test is to see how subjective assessments of soundscapes made using the SAM relate to the categorisation of those soundscapes into these three groups. Fig. 2 shows the category ratings question as presented to the test participants. They were asked to give a rating in each category for each of the sixteen soundscapes. They were also given a free text entry field alongside each one where they were asked to identify the sound sources present. Participants were required to enter at least one sound source into this field in order to progress, but no further requirements were in place. This was done in order to identify the sound sources that were most noticeable to test participants and dominated their perception of the aural scene.

2.4. Test Procedure

The listening test in this study was presented online using Qualtrics [26], and the overall procedure was as follows:

1. An introductory demographics questionnaire including questions regarding age, gender, nationality, and profession. From the results of previous studies [2, 3] it is not expected that any of these demographic factors will have an effect on the results.
2. An introduction to the questions accompanying each soundscape, covering the SAM, the categorisation question, and the sound source identification entry field. Two example

To what extent does the soundscape belong in each of the following categories?

	Not at all	Somewhat	Very much	
Natural/animal	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Human	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Industrial/mechanical	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

Figure 2: The category ratings question as presented to test participants.

questions (using soundscape clips 1A and 7B) are presented in order to get baseline results for the participant. After listening to the first clips the participant was instructed not to change the playback volume.

3. The 16 soundscape recordings were then presented to the individual in a random order. After listening to each one they were asked to rate their experience of the soundscape using the SAM, to categorise it, and to identify the sound sources present. They were also given the opportunity to add any comments they had in an optional text entry field.

3. RESULTS

This section presents the results of this study, with a discussion following in Section 4. Following a summary of the SAM results, the category ratings are examined. An overview of the identified sound sources is then shown, after which the sound sources identified for each soundscape clip are categorised themselves and the percentage of these sources that belong to each of the three categories is presented. Correlational analysis is then used to examine the relationships between these metrics.

3.1. SAM Results

Fig. 3 shows a summary of the SAM results gathered in this test. As can be seen in this figure, the clips presented in this study cover a wide range of arousal and valence ratings. A full analysis of these results can be seen in [3]. A comparison of these results with the categorisation results can be seen in Section 3.4.1.

3.2. Category Ratings

Fig. 4 shows a summary of the category ratings results. As with the SAM results it is clear from this figure that the selected soundscapes cover a wide range of ratings in each category.

3.3. Sound Source Identification

Fig. 5 shows a pie chart of all of the sound sources identified by test participants. In total there were 1369 sound source instances identified which contained 24 unique sound sources (as listed in Fig. 5). The overall breakdown of these instances between the three categories is as follows: 38.9% natural, 26.9% human, and 34.2% mechanical.

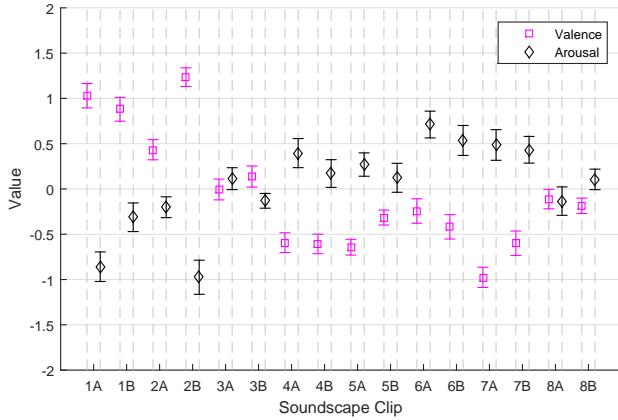


Figure 3: Summary of SAM results from this listening test, showing the mean valence and arousal ratings, and the standard error associated with each mean, for each of the 16 clips.

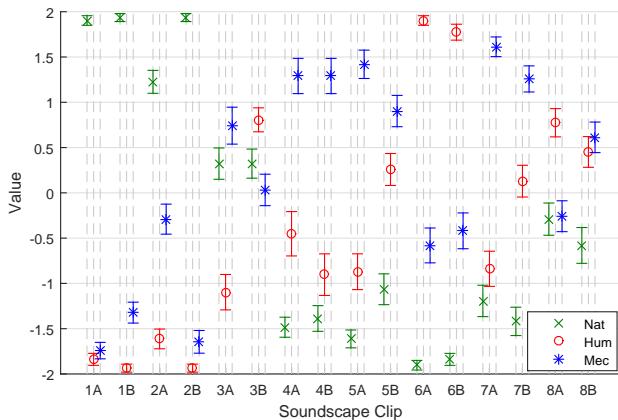


Figure 4: Summary of category ratings results from this listening test, showing the mean ratings for each category, and the standard error associated with each mean, for each of the 16 clips.

3.3.1. Percentages

In order to make a comparison between the sound sources identified and the categorisation of each clip, it was decided that the sound sources identified for each clip should be grouped by category and then the number of sound sources instances in each of these categories should be expressed as a percentage of the total number of sound source instances in each case. Fig. 6 shows the sound source category percentage breakdown for each soundscape recordings.

3.4. Correlational Analysis

Having now presented a brief summary of the SAM and categorisation results, and having made use of the sound sources identified to create a data set suitable for comparison with those result, correlational analysis can be used to compare these three sets of variables to identify relationships between them. First the SAM and categorisation results will be compared, after which the category ratings will be compared with the percentage breakdown of the sound sources identified.

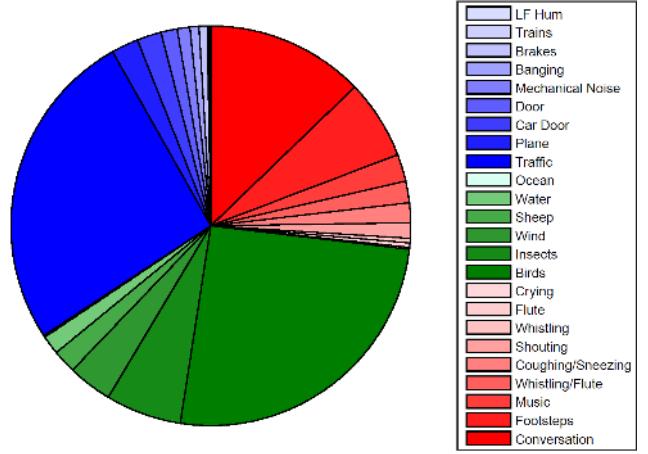


Figure 5: Pie chart indicating all of the unique sound sources identified, shown colour-coded into categories by colour where blue is mechanical, green is natural, and red is human.

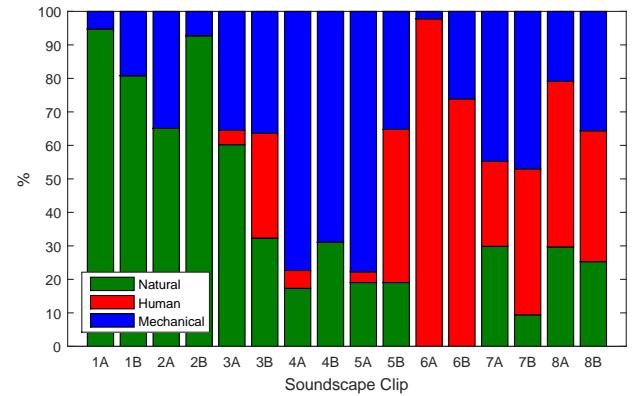


Figure 6: Sound source identification instances for each clip split into categories and expressed as a percentage.

3.4.1. SAM and category ratings

Fig. 7 shows scatter plots of the mean category ratings against the mean valence and arousal ratings. The result of a Pearson's correlation analysis [27] of the data presented in these plots can be seen in Table 2.

Table 2: Correlation coefficient and *p*-values comparing the valence results with category ratings.

	Natural	Human	Mechanical
	<i>R</i> -values		
Valence	0.93	-0.51	-0.90
	-0.91	0.65	0.70
<i>p</i> -values			
Valence	2×10^{-7}	0.04	2×10^{-7}
	9.4×10^{-7}	0.006	0.002

The *R*-values indicate the degree of correlation between two variables, and the *p*-values indicate the statistical significance of this

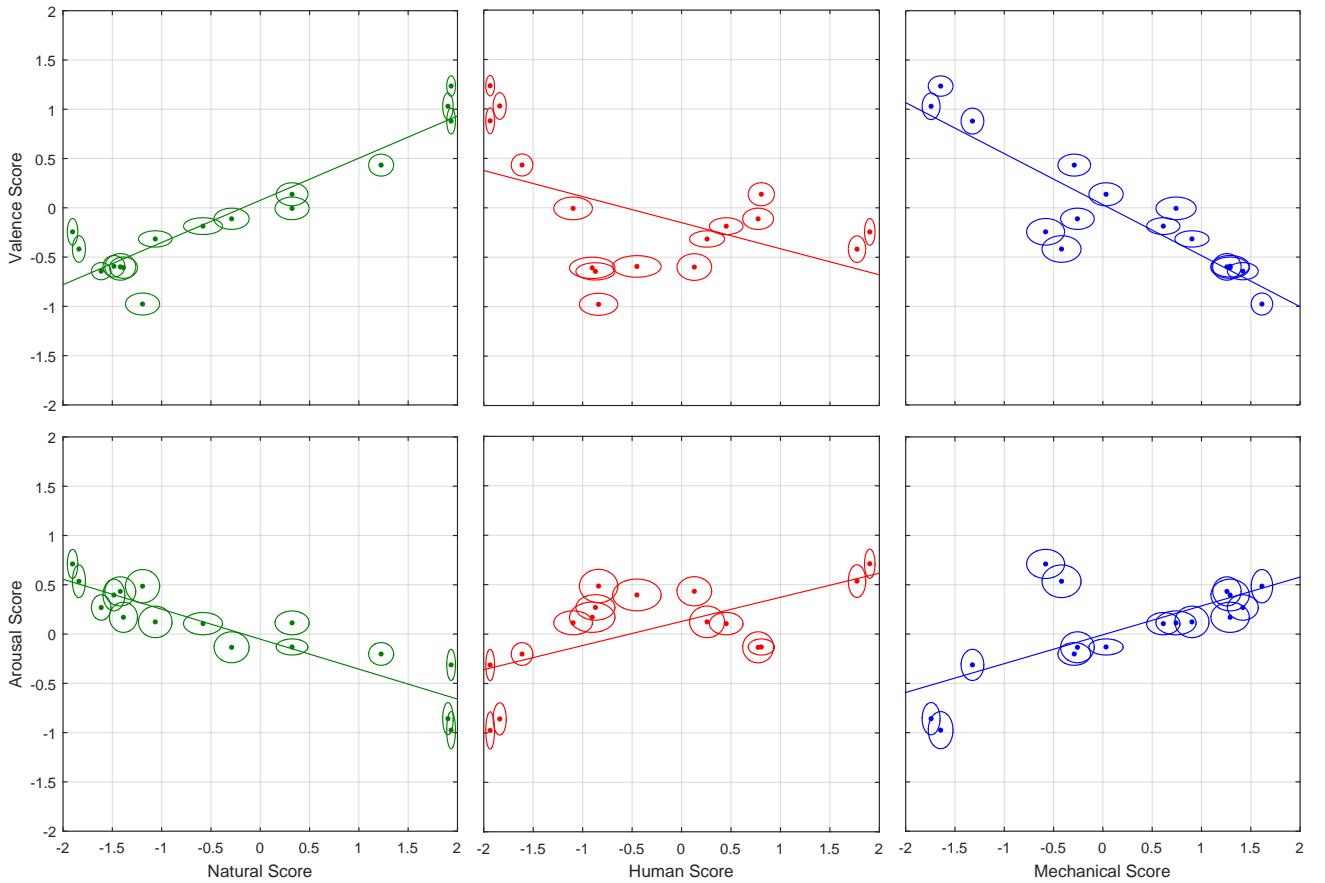


Figure 7: Scatter plots showing the mean category and SAM ratings for each clip. The ellipses surrounding each data point indicate the standard error associated with each mean value. Trend lines have been included to reflect the correlation results shown in Table 2.

relationship where $p \leq 0.05$ it indicates a statistically significant correlation [27].

The natural and mechanical category ratings show a very strong set of correlations with the SAM results, where the Natural category is shown to have a highly significant positive correlation with the valence dimension, and a highly significant negative one with the arousal dimension. The mechanical category exhibits the opposite relationship with each dimension. From this it can be said that the locations rated as highly natural are typically rated as pleasant and relaxing, and locations that are rated as highly mechanical are unpleasant and invoke feelings of stress and anxiety. These are the emotions indicated by these SAM results when considering the circumplex model of affect [28].

The human category rating results also show a (just-significant) negative correlation with valence, and a positive correlation with arousal. This is perhaps due to the fact that most of the soundscapes that include human sounds also contain mechanical sounds, particularly traffic noise. Examining the top middle plot in Fig. 7 gives an indication of why the relationship between the human category ratings and valence is only just significant. Whilst the lowest human category ratings corresponding to high valence, as the human category rating increases there is a ‘dip’ where the valence values for human ratings around 1 are in fact lower than those for higher human ratings. This is due to a disproportionate number of the

soundscapes containing very few or no human sounds (i.e. clips 1A–2B, as seen in Fig. 6).

The results shown in Table 2 support the hypotheses laid out in Section 1 detailing the expected relationships between the three category ratings and the SAM.

3.4.2. Category ratings and categorised sounds source percentages

Table 3 shows the correlation results indicating the relationships between the category ratings and the sound source percentage results. The following section includes a discussion of the results and the other results presented in this section.

4. DISCUSSION

This section contains a discussion of the correlation results presented in Table 3, considering the following three areas of the correlation results in turn:

- Comparison of categorisation results.
- Comparison of the categorised sound sources.
- Comparison of categorisation results with the categorised sounds sources.

Table 3: Correlation results for the category ratings and sound source percentage results. Indicated here are the R values. The numbers in boldface indicate correlation results where $p \leq 0.05$ (a significant result); and the presence of an asterisk indicates $p \leq 0.01$ (a highly significant result).

	Natural	Human	Mechanical	Nat%	Hum%	Mech%
Natural	-	-	-	-	-	-
Human	-0.69*	-	-	-	-	-
Mechanical	-0.73*	0.19	-	-	-	-
Nat%	0.95*	-0.83*	-0.64*	-	-	-
Hum%	-0.58	0.92*	0.01	-0.71*	-	-
Mech%	-0.52	-0.08	0.84*	-0.41	-0.35	-

Following this discussion, the key sound sources and soundscapes that explain these results are identified and examined in further detail.

4.1. Category Ratings

There is significant negative correlation between natural and human, and natural and mechanical ratings, indicating that where soundscapes were rated as more natural they were also rated as less human and less mechanical. Perhaps surprisingly there is not a significant relationship between the human and mechanical category ratings.

4.2. Sound source percentage results

The correlation results comparing the sound source category percentage results in Table 3 show that the only significant relationship is the negative correlation between the percentage of sound sources identified that are in the natural category, and those belonging to the human category. This is indicated by the sound source categorisation results for clips 6A and 6B as shown in Fig. 6.

For both of these clips the overwhelming majority of sound sources identified ($> 70\%$) were in the human category, with the remaining sound sources belonging to the mechanical category and no identified sounds in the natural category. Likewise the results for clips 1A-2B contain very high percentages of natural sound sources, some mechanical sources, and no human ones.

4.3. Categories and percentages

There is a significant correlation between the category ratings and percentage of identified sound sources for each of the three categories. This gives some indication that the percentage breakdown of the identified sound sources into each category is a valuable metric that in some way reflects the overall categorisation of a soundscape.

The natural category rating shows significant negative correlation with the human and mechanical sound source percentage metric, as does the natural sound sources percentage metric with the human and mechanical category ratings. The two pairs of variables that show no significant correlation in this group are human category rating and mechanical percentage metric, and the mechanical category rating and the human percentage metric.

4.4. Key Sound Sources

As identified in Fig. 5, the three most commonly identified sound sources were traffic noise, birdsong, and conversation; sound sources belonging to the mechanical, natural, and human categories respectively. This section will consider each one in turn and see how the

presence of these sound sources in the soundscapes has impacted on the SAM and category ratings of those soundscapes.

4.4.1. Birdsong

Table 1 indicates that the soundscape clips recorded in Dalby Forest contain many instances of birdsong. As can be seen in Fig. 4, these soundscapes (clips 1A-2B) were rated as most belonging to the natural category, and, as shown in Fig. 3, received the highest valence and lowest arousal ratings of all the soundscape clips. This result is in accord with the correlation results presented in Table 2 that indicate the same relationship between higher natural category ratings and the SAM results.

It is also interesting to note the difference in category rating between clips 7B and 8A. These clips were recorded at two nearby locations in Leeds: clip 7B was recorded next to a busy road, and clip 8A was recorded in the middle of a small park. These two clips are fairly similar, but the presence of birdsong in the latter is partly responsible for a significant difference in the natural category rating, which contributes to the relative increase in valence rating (and decrease in arousal rating) as seen in Fig. 3.

4.4.2. Traffic

The correlation between the mechanical category ratings and the SAM results (negative for valence, positive for arousal), as indicated by Table 2, is shown clearly in the difference between SAM results and category ratings for clips 2A and 2B. These two clips were recorded in the same location next to a lake in a forest: Clip 2A contains the recording a single car driving on a nearby road where clip 2B does not. As shown in Fig. 4 the presence of this car is responsible for big increase in the mechanical category ratings, and a big decrease in the natural category rating, for clip 2A relative to clip 2B.

Fig. 3 shows the effect of this car on the SAM ratings for this clip. It is interesting to note how big a difference the presence of a single car can make. This same pattern can be seen to a lesser extent in the results for clips 1A and 1B. These were both recorded at different positions in the forest next to a footpath away from any roads. In this case it is the presence of an aeroplane passing overhead in clip 1B that is not present in clip 1A that results in a similar pattern of differences between the two clips.

It is also worth noting that the clips recorded at roadside location were rated as having the lowest valence levels (clips 4A-5A and 7A-7B), and that these clips were also rated as most belonging to the mechanical environment category.

4.4.3. Conversation

The most significant sound source identified as belonging to the human category was conversation. Its presence in clip 3B and absence in clip 3A, whilst not responsible for any appreciable difference in SAM results, produced a big change in category ratings resulting in a much higher human category rating for clip 3B.

Fig. 6 also indicates the perceptual dominance of conversation when present in a soundscape recording - clips 6A and 6B are the only soundscapes for which no natural sound sources were identified by test participants, despite some birdsong and wind noise being present in those recordings.

5. SUMMARY

This paper has shown the results of an online listening test presenting stereo UHJ renderings of B-format soundscape recordings. Test participants were asked to describe the emotional state evoked by those soundscapes using the SAM, rate the extent to which each soundscape belonged in three environmental categories, and identify the key sound sources present in each one.

The purpose of this test was to identify any relationships between the subjective assessment of a soundscape and the extent to which that soundscape is perceived to be natural, human, and mechanical. A secondary aim was to use the sound sources identified by test participants to sound sources that are the most important to soundscape perception, both in terms of subjective experience and category rating.

Correlational analysis indicated strong relationships between the natural and mechanical category ratings and the valence and arousal dimensions of the SAM. The natural category ratings were shown to be positively correlated with valence, and negatively correlated with arousal. The mechanical category ratings showed the opposite correlations in each case. The human category ratings showed similar but less strong correlation results to the mechanical category ratings with some ambiguity in the relationship between the human category ratings and the valence dimension of the SAM. These results support the study's hypotheses which predicted a negative emotional response to highly mechanical soundscapes, and a positive emotional response to highly natural soundscapes. The hypothesis regarding emotional response to soundscape rated as highly human was also suggested with a clear correlation between human rating and arousal and a more ambiguous relationship with valence.

These results indicate that more natural soundscapes invoke a relaxation response (low arousal, high valence), and that more mechanical soundscapes invoke a stress response (high arousal, low valence). The human category rating result's relative lack of significance indicate a contextual dependence where human sounds are not necessarily disliked or stressful, but are mainly present in soundscapes where traffic noise (i.e. mechanical sounds) are also present.

An analysis of the sound sources identified by test participants has presented, which indicated a key sound source for each category: conversation, birdsong, and traffic. The identification of these sources supports the hypothesis predicting which sound sources would be most commonly identified, and as such would be the most useful predictor of soundscape categorisation. The presence and absence of these sources in the soundscapes presented was examined by studying particular examples. This examination demonstrated where these sound sources in particular produced differences in

the SAM and category ratings that exhibited with correlational relationships previously identified.

Future work will include a listening test presenting the same soundscapes used in this study alongside spherical panoramic images of the recordings sites. Test participants will again be asked rate their experience of the environment using the SAM, to categorise the environment, and to identify the key aural and visual features. This test should produce results that answer the following questions:

- Will the presence of different visual features change the SAM results, indicating a change in the evoked emotional states?
- Will the key visual feature identified be the same as the aural features?
- Will the presence of visual features change which aural features are identified by test participants?
- Will the presence of visual elements change the category ratings for the presented environments?

This results presented in this paper certainly confirm the SAM as being a powerful tool for soundscape analysis, with soundscape categorisation and sound source identification offer suitable methods to pair with the SAM to offer further insight into which aural features can most dramatically affect emotional responses to environmental sound.

6. ACKNOWLEDGMENTS

This project is supported by EPSRC doctoral training studentship: reference number 1509136.

7. REFERENCES

- [1] 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.
- [2] 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*, 2016.
- [3] ——, "Ecological validity of stereo uhj soundscape reproduction," in *In Proceedings of the 142nd Audio Engineering Society (AES) Convention*, 2017.
- [4] 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>
- [5] E. Benjamin and T. Chen, "The native b-format microphone," in *Audio Engineering Society Convention 119*. Audio Engineering Society, 2005.
- [6] R. Elen, "Ambisonics: The surround alternative," in *Proceedings of the 3rd Annual Surround Conference and Technology Showcase*, 2001, pp. 1–4.
- [7] M. A. Gerzon, "Ambisonics in multichannel broadcasting and video," *Journal of the Audio Engineering Society*, vol. 33, no. 11, pp. 859–871, 1985.

- [8] 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.
- [9] C. Osgood, “The nature and measurement of meaning.” *Psychological bulletin*, vol. 49, no. 3, p. 197, 1952.
- [10] 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.
- [11] W. Davies, N. Bruce, and J. Murphy, “Soundscape reproduction and synthesis,” *Acta Acustica United with Acustica*, vol. 100, no. 2, pp. 285–292, 2014.
- [12] 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.
- [13] A. Zeitler and J. Hellbrück, “Semantic attributes of environmental sounds and their correlations with psychoacoustic magnitude,” in *Proc. of the 17th International Congress on Acoustics [CDROM]*, Rome, Italy, vol. 28, 2001.
- [14] T. Hashimoto and S. Hatano, “Effects of factors other than sound to the perception of sound quality,” *17th ICA Rome, CD-ROM*, 2001.
- [15] B. Schulte-Fortkamp, “The quality of acoustic environments and the meaning of soundscapes,” in *Proc. of the 17th international conference on acoustics*, 2001.
- [16] M. Raimbault, “Qualitative judgements of urban soundscapes: Questioning questionnaires and semantic scales,” *Acta acustica united with acustica*, vol. 92, no. 6, pp. 929–937, 2006.
- [17] S. Harriet, “Application of auralisation and soundscape methodologies to environmental noise,” Ph.D. dissertation, University of York, 2013.
- [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 and P. Lang, “Affective reactions to acoustic stimuli,” *Psychophysiology*, vol. 37, no. 02, pp. 204–215, 2000.
- [21] A. Léobon, “La qualification des ambiances sonores urbaines,” *Natures-Sciences-Sociétés*, vol. 3, no. 1, pp. 26–41, 1995.
- [22] 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>
- [23] 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.
- [24] L. Anderson, B. Mulligan, L. Goodman, and H. Regen, “Effects of sounds on preferences for outdoor settings,” *Environment and Behavior*, vol. 15, no. 5, pp. 539–566, 1983.
- [25] 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.
- [26] J. Snow and M. Mann, “Qualtrics survey software: handbook for research professionals,” 2013.
- [27] K. Pearson, “Note on regression and inheritance in the case of two parents,” *Proceedings of the Royal Society of London*, vol. 58, pp. 240–242, 1895.
- [28] J. A. Russell, “A circumplex model of affect,” *Journal of Personality and Social Psychology*, vol. 39, no. 6, pp. 1161–1178, 1980.

DEVELOPMENT OF A QUALITY ASSURANCE AUTOMATIC LISTENING MACHINE (QUAALM)

Daniel J. Gillespie *

Newfangled Audio
New York, NY
DGillespie@newfangledaudio.com

Woody Herman

Eventide Inc.
Little Ferry, NJ
W.Herman@eventide.com

Russell Wedelich

Eventide Inc.
Little Ferry, NJ
R.Wedelich@eventide.com

ABSTRACT

This paper describes the development and application of a machine listening system for the automated testing of implementation equivalence in music signal processing effects which contain a high level of randomized time variation. We describe a mathematical model of generalized randomization in audio effects and explore different representations of the effect's data. We then propose a set of classifiers to reliably determine if two implementations of the same randomized audio effect are functionally equivalent. After testing these classifiers against each other and against a set of human listeners we find the best implementation and determine that it agrees with the judgment of human listeners with an F1-Score of 0.8696.

1. INTRODUCTION

When Eventide began development of their new H9000 effects processor a primary task was the porting of a large signal processing code base from the Motorola 56000 assembly language to C++ targeted to several different embedded processors. While industry standard cancellation testing was possible for deterministic effects, most of the effects involved some amount of random time variation causing test results to vary significantly from the judgment of human listeners. This created a problem when ascertaining that the thousands of presets used in the H8000 were working correctly on the H9000. To solve this problem we developed a machine listening system to replace the first stage of human listening to determine if two different implementations sound equivalent.

Comparing the similarity of two audio signals is done in many contexts. The field of audio watermarking aims to embed a unique code in an audio stream such that the code is imperceptible to the human ear and is recoverable after the application of several different auditorily transparent transformations such as time scale modification [1][2]. Several different techniques are used to quantify how perceptible the added code is to the listener. These include the Perceptual Audio Quality Measure (PAQM), the Noise to Mask Ratio (NRM), and the Perceptual Evaluation of Audio Quality (PEAQ) [3]. However, while these methods aim to make a perceptual comparison between two recordings, the comparison is primarily interested in determining if two signals sound exactly the same. Unfortunately, these comparisons are not useful to us as we need to determine if two sets of unique signals are generated by the same random process, even when these sets of signals vary significantly inside each set.

There have also been several attempts to reverse engineer the specific settings of audio effects [4][5][6], however these systems are often aimed at specific effects types and rely on a specific

model. At the most general, Barchiesi and Reiss assume that the effects are linear and time-invariant[4], which we cannot assume here.

The fields of audio query by example and music recommendation systems have a looser definition of audio similarity which might have some applications to our problem. A comparison of some systems which aim to define this more general similarity is given by Pampalk et al in [7]. A particularly interesting implementation from Helén et al [8] uses a perceptual encoder followed by a compression algorithm to determine similarity by the compression ratio of the signals separately vs combined. Another interesting probabilistic approach is given by Virtanen [9]. These models may have the capacity to model the variation we expect between and within the sets of signals we are evaluating, however, their implementations are very resource intensive and the nature of these problems is that there is no ground truth by which to compare the algorithms to each other. We suspect that we might be able to solve our problem with a more compact model.

In Section 2 we give a deeper description of the problem. Section 3 describes the theory behind our solution. Section 4 mentions some practical considerations of our system. Section 5 describes the validation experiments we ran, and Section 6 gives results of these experiments and some analysis. Section 7 gives a brief conclusion, Section 8 some acknowledgments, and Section 9 includes references.

2. PROBLEM DESCRIPTION

When testing whether two audio signal processing implementations are functionally equivalent it is common to send a test signal $x(t)$ through each of them and compare the output signals $y_1(t)$ and $y_2(t)$ [10]. If the magnitude difference $z(t)$ between these two outputs is less than 2η , for some very small η , the two implementations can be considered equivalent.

This can be written as

$$y_n(t) = f_n(x(t)) \quad (1)$$

$$z(t) = |y_1(t) - y_2(t)|, \quad (2)$$

$$z(t) < 2\eta; \forall t \in T, \quad (3)$$

where $f_n(x)$ represents the n^{th} distinct implementation being tested.

This test works well for musical effects that can be characterized by a deterministic system plus a small residual, as long as the system memory is less than the measurement time scale T . This is true because this test relies on the implicit signal model

$$f_n(x(t)) = f(x(t)) + \eta N_n(t) \quad (4)$$

* For Eventide Inc.

where $f(x(t))$ is the theoretically perfect implementation and $N_n(t)$ is a continuous random process on the range $[-1, 1]$ representing the implementation-specific variation from $f(x(t))$. The test in Equation (3) places no constraints on $N_n(t)$ other than that it is strictly bounded to the range $[-1, 1]$, nor do we assume any. It is likely that some realizations of $N_n(t)$ produce more audible effects than others, but in practice the cancellation tests works by keeping η very small.

A very simple example of a deterministic effect that uses this signal model is a gain effect with a constant gain of g . This system can be represented as

$$f(x(t)) = gx(t) \quad (5)$$

When implemented digitally it becomes

$$f_n(x(t)) = g[x(t) + q_n(t)] + t_n(t) \quad (6)$$

Where $q_n(t)$ represents the quantization noise and $t_n(t)$ represents the truncation error of the multiply, each in the n^{th} implementation. Both of these types of error can be modeled by additive noise, and with a slight re-arrangement we can show that it fits the signal model of Equation (4)

$$f_n(x(t)) = gx(t) + gq_n(t) + t_n(t) \quad (7)$$

$$f_n(x(t)) = f(x(t)) + \eta N_n(t) \quad (8)$$

$$|f_1(x(t)) - f_2(x(t))| = |\eta N_1(t) - \eta N_2(t)| \quad (9)$$

and as long as $|gq_n(t) + t_n(t)| < \eta$ is strictly true for two different implementations, we can call them equivalent using the cancellation test in Equation (3).

A slightly more complicated deterministic process which still passes the test in Equation (3) is the simple tremolo effect. To create a tremolo effect we define g as a function of time, now the system can be described by

$$g(t) = \sin(w_0 t) \quad (10)$$

$$f(x(t)) = \sin(w_0 t)x(t) \quad (11)$$

By adding an additional $s_n(t)$ term to represent the error in generating the sine wave, we can write an implementation-specific version.

$$f_n(x(t)) = (\sin(w_0 t) + s_n(t))(x(t) + q_n(t)) + t_n(t) \quad (12)$$

After grouping the additive components we can still factor the implementation specific components into a simple additive term.

$$f_n(x(t)) = \sin(w_0 t)x(t) + \eta N_n(t), \quad (13)$$

$$\eta N_n(t) = \sin(w_0 t)q_n(t) + s_n(t)(x(t) + q_n(t)) + t_n(t). \quad (14)$$

As long as the implementation errors $q_n(t)$, $s_n(t)$, and $t_n(t)$ are kept small enough the two implementations can be considered equivalent, as is demonstrated by

$$|f_1(x(t)) - f_2(x(t))| = \eta |N_1(t) - N_2(t)|. \quad (15)$$

When considering the addition of intentional randomization to these types of effects it might be tempting to consider them to be additive processes using the existing signal model. For instance, if we consider the implementation error of the sine wave $s_n(t)$ in equation (12) to instead represent an intentional random process integral to the sound of the effect, we might consider trying to reduce its contribution to the measurement error by averaging over

several recordings of the effect. If all sources of randomization can be considered purely additive, we could take several measurements of each implementation and compare the mean and variance of each to make a determination which may prove sufficient to call them functionally equivalent.

Unfortunately, many effects implement processing whose randomization cannot be treated as simple additive noise. For these effects even a relaxation of the cancellation test in Equation (3) is not appropriate because the signal model used does not have the capacity to discriminate between equivalent implementations and ones that differ. We can show an example of this by adding a random initial phase θ_n to our tremolo example and deriving the equation for the cancellation test. If we treat θ_n as a uniformly distributed random variable between $-\pi$ and π the system can be described by

$$f_n(x(t)) = (\sin(w_0 t + \theta_n) + s_n(t))(x(t) + q_n(t)) + t_n(t) \quad (16)$$

After grouping the additive components we are not able to factor the theoretically perfect $f(x(t))$ out of the model.

$$f_n(x(t)) = \sin(w_0 t + \theta_n)x(t) + \eta N_n(t), \quad (17)$$

and therefore the difference will depend on θ_n

$$\begin{aligned} &|f_1(x(t)) - f_2(x(t))| = \\ &|(\sin(wt + \theta_1)x(t) - \sin(wt + \theta_2)x(t)) + (\eta N_1(t) - \eta N_2(t))| \end{aligned} \quad (18)$$

Even if we make the additive sources of error very small, if we cannot control the distribution of θ_n there will still be a large source of error determined by this initial condition. This might seem to be a contrived comparison, but consider that the two implementations might be running on different hardware, or even implemented in an analog circuit. In these situations we may not be able to synchronize θ_n . More complicated effects often have several processes with randomized initial conditions which we will store in the vector Θ_n and may even be the result of one or more random processes $\Psi_n(t)$. We can write this as the more general formulation

$$y_n(t) = f_n(x(t), \Psi_n(t), \Theta_n) \quad (19)$$

In these instances, $\Psi_n(t)$ and Θ_n may not always be able to be controlled between various implementations, and when they are not the same we can expect the test in Equation (3) to fail.

However, in practice we found that both in-house testers and end users are able to reliably detect similarity or difference in these categories of randomized time-variant effects, even when sources of randomization are not controlled. Therefore we hoped to design a machine listening algorithm to make these implementation equivalence determinations in much the same way that humans do. The goal was to create an algorithm that would reliably detect functional equivalence between implementations of both the easily tested deterministic, as well as the highly randomized audio effects without any prior knowledge about the effect under test.

3. THEORY

To extend the simple model in Equation (3) to systems with undefined random variations we will form a probabilistic interpretation

and use several samples of each implementation $f_n(\cdot)$ to learn a model of its behavior. Once we learn models for each $f_n(\cdot)$ we can compare these models to make a decision about the similarity of the implementations.

3.1. Representing the Data

As is done in the classical problem description above, we chose to represent each $f_n(\cdot)$ by testing the appropriate response $y_n(t)$ to the common test signal $x(t)$. Therefore the learning problem happens on vectors representing the $y_n(t)$ signals rather than the $f_n(\cdot)$ processes directly. Because of this, the result of the test depends both on the test method as well as the forcing signal $x(t)$. Specifically, in the case of linear time-invariant systems, $y_n(t)$ must be long enough to capture the entire expected impulse response of $f_n(\cdot)$, and it must have sufficient bandwidth to measure the expected bandwidth of $f_n(\cdot)$. When the system is expected to be nonlinear it must have sufficient changes in amplitude to capture these effects. Finally, when the system is expected to be time-variant, $y_n(t)$ must be long enough to capture this time-variance, or there must be enough samples of $y_n(t)$ to sufficiently span the expected range of variance.

So given the test signal $x(t)$, we will now learn a model Y_n for each implementation $f_n(\cdot)$ representing its response $y_n(t)$.

To learn the Y_n we must represent the signal $y_n(t)$ as a fixed length feature vector \mathbf{y}_n , where each discrete sample in \mathbf{y}_n will be a feature in our probabilistic model. For these representations we consider several choices. We can choose to simply use the discretized samples $y_n[i]$ directly. This has the benefit of being similar to the tried and true method in Equation (3) and does not lose any data. Additionally, when using this representation, any purely additive or multiplicative differences between implementations of $f_n(\cdot)$ will result in purely additive or multiplicative differences in \mathbf{y}_n . Therefore, even if we assume the features of Y_n are strictly independent, Y_n will still be able to capture these differences. However, if $f_n(\cdot)$ has memory, errors in the part of the system with memory will be spread across several different features in \mathbf{y}_n and we will have to include some dependency structure in Y_n to accurately model these differences, which can make the model more complicated.

We can also choose to take the magnitude FFT of $y_n[i]$, which has the added benefit of keeping most of the data when the FFT size is large. However, if we do so, convolutions in the $f_n(\cdot)$ process will become simple multiplications. Therefore a model that assumes independence will be more robust to small convolutive variations, like a small change in a delay time parameter, at the expense of being less robust to small multiplicative variations, like a small change in a gain coefficient.

A compromise solution might be to take the magnitude STFT of y_n and vectorize it. This would combine the benefits of both the sample-wise and FFT representations, while maintaining the locality of differences in both time and frequency. This might put a limit on the amount of dependency the model needs to assume, or the amount of error we'd expect to see by assuming independence.

Finally, considering we're looking to replace human listeners we might take a cue from speech recognition research [11], and choose to represent the audio as a vectorized set of MFCC coefficients. While this representation does throw away data, depending on the length of the test signal $x(t)$, losing this data may help us with regard to the "curse of dimensionality" [12], which describes how the modeling capacity of a given model can suffer as the di-

mensionality of the training data increases. In Experiment 1 we will try all four of these data representations to determine which gives the best classification accuracy across a number of effects.

3.2. Forming a Probabilistic Representation

When dealing with sources of variation it is often useful to build the model in terms of a probabilistic formulation. If we believe that M successive applications of $f_n(\cdot)$ to $x(t)$ will result in M distinct results $y_{mn}(t)$ and \mathbf{y}_{mn} then we might choose to model the \mathbf{y}_{mn} vectors as a random process from which we can draw samples. A commonly used generative model for high dimensional continuous data is the multivariate normal distribution with the assumption of independence between the dimensions [13]. This model has the benefit of fitting many naturally occurring processes, while also having a tractable calculation of the log likelihood that a particular sample has come from the underlying model. This is true even when used with very high dimensional feature vectors, like we expect to have here.

To do this, we define

$$Y_n \sim \mathcal{N}(\mu_n, \sigma_n^2). \quad (20)$$

$$\mu_n = \frac{1}{M} \sum_{m=1}^M \mathbf{y}_{nm} \quad (21)$$

$$\sigma_n = \frac{1}{M-1} \sum_{m=1}^M (\mathbf{y}_{nm} - \mu_n)(\mathbf{y}_{nm} - \mu_n)^T. \quad (22)$$

where \mathcal{N} is the multivariate normal distribution, μ_n is the mean of the M samples of \mathbf{y}_n , and σ_n is the variance vector. In this instance the covariance matrix reduces to a vector because independent features imply that all non-diagonal elements of the covariance matrix are zero.

3.3. Making a Decision

Once we have a model Y_n representing the effect $f_n(\cdot)$ for each implementation n , we must compare them to measure their similarity. A common method of measuring the similarity between two probability distributions is the Kullback-Leibler divergence $D_{KL}(P||Q)$ [14]. The Kullback-Leibler divergence from Q to P measures the information gained when one modifies the prior distribution Q to the posterior distribution P .

When P and Q are distributions of a continuous random variable the Kullback-Leibler divergence is defined by the probability densities of P and Q , respectively $p(x)$ and $q(x)$, as

$$D_{KL}(P||Q) = \int_{-\infty}^{\infty} p(x) \log \frac{p(x)}{q(x)} dx, \quad (23)$$

To determine the similarity of two implementation models Y_n we can choose the target implementation to represent the true distribution P and choose the implementation under test to represent the prior distribution Q . The smaller the information gained by modifying Q to match P , the more similar the distributions Q and P are considered. Therefore, by finding the Kullback-Leibler divergence $D_{KL}(Y_{target}||Y_{test})$ we can gain a measure of similarity between the known good implementation $f_{target}(\cdot)$ and the one being tested $f_{test}(\cdot)$.

When P and Q are both multivariate normal distributions \mathcal{N}_P and \mathcal{N}_Q with independent components the KL divergence can be simplified to:

$$D_{\text{KL}}(\mathcal{N}_P \parallel \mathcal{N}_Q) = \frac{1}{2} \sum_D \ln |\sigma_Q| - \sum_D \ln |\sigma_P| - D + \sum_D \frac{\sigma_P}{\sigma_Q} + (\mu_Q - \mu_P)' \frac{1}{\sigma_Q} (\mu_Q - \mu_P) \quad (24)$$

This can be thresholded to make a hard decision, or be simply reported as an error measure.

Another option is to treat this decision as a classification problem by determining if it is more likely that a given sample \mathbf{y}_n came from a model representing one specific implementation, or a joint model representing both representations. For a multivariate normal distribution with independent dimensions, the log likelihood can be calculated by

$$\ln(L) = -\frac{1}{2} \sum_{d=1}^D \frac{(x_d - \mu_d)^2}{\sigma_d^2} - \frac{D}{2} \ln 2\pi - \frac{1}{2} \sum_{d=1}^D \ln \sigma_d^2 \quad (25)$$

We then pull out one sample $\mathbf{y}_{n,m}$ and form two sets from the remaining samples. The set S_{test} represents the remaining samples from the implementation being tested, while the set S_{all} is formed from the union of S_{test} and S_{target} , which is the set of all samples from the implementations being tested against.

$$S_{\text{test}} = \mathbf{y}_{n,l}; \forall l \neq m \quad (26)$$

$$S_{\text{all}} = \mathbf{y}_{k,l}; \forall k \neq n \cup l \neq m \quad (27)$$

Now we build models Y_{test} from set S_{test} and Y_{all} from set S_{all} and calculate the log likelihoods, $\ln(L_{\text{test}})$ and $\ln(L_{\text{all}})$. If $\ln(L_{\text{test}}) > \ln(L_{\text{all}})$, then the sample being tested fits the samples that came from the test implementation better than it fits a collection of the samples from all implementations. This is an indication that the implementation being tested differs from the target. However, if $\ln(L_{\text{all}}) > \ln(L_{\text{test}})$, then the two implementations are similar enough that the held-out sample cannot be grouped specifically with the test implementation. This is an indication that the two implementations are functionally equivalent.

To reduce the effect of outliers in this decision making, we use a method similar to leave-one-out cross validation, and repeat this process for each m in M , then use a voting method [15] to determine if the implementations should be considered equivalent.

4. PRACTICAL CONSIDERATIONS

In practice the Quality Assurance Automatic Listening Machine must run relatively quickly on a large variety of presets with as little human intervention as possible. For the purposes of the following experiments, the test signal $x(t)$ was chosen to be 3 seconds of full bandwidth noise, followed by 3 seconds of silence, followed by a 3 second long logarithmic chirp, followed by a final 3 seconds of silence. This specific signal was chosen to reflect the particulars of the expected effects. While we believe that there are some effects for which this signal will be insufficiently long, a necessary trade off must be made to keep the dimensionality reasonable for the classifiers sake and to keep the recordings short for the sake of the test duration and for human validation. We didn't experiment with different test signals, though we believe that this might be a good route for further improvement. Additionally, we chose

to build our models based on a sample of 10 recordings from each preset being tested.

In operation, we will also have a preference for false failures over false passes, because a false negative that allows a human listener to double check is preferable to a false positive that allows a potential implementation problem to go into production. For this reason, we will score the classification experiments using both the standard F Score as well as the Precision and Recall scores. Preference is given to the Precision score which penalize the algorithm only for false positives.

From [16] Precision is defined as

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}, \quad (28)$$

and Recall is defined as

$$\text{Recall} = \frac{\text{TN}}{\text{TN} + \text{FP}}, \quad (29)$$

where TP is the number of true positives, or instances where both the human listeners and the QuAALM decided that the implementations were the same, TN is the number of true negatives, or instances where both the human listeners and the QuAALM decided that the implementations were different, FP is the number of false positives, or instances where the QuAALM declared that the two implementations were the same, but the human listeners did not, and FN is the number of false negatives, or instances where the QuAALM declared that the two implementations were different, but the human listeners decided that they were the same.

Then, the F-measure is defined as the average of Precision and Recall:

$$F_1 = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} \quad (30)$$

$$F_1 = \frac{2\text{TP}}{2\text{TP} + \text{FP} + \text{FN}} \quad (31)$$

5. EXPERIMENTAL DESIGN

We ran two different experiments. The first is meant to find the best set of features and the best decision algorithm. The second is meant to validate the QuAALM machine by comparing it with human listeners.

5.1. Experiment 1: Determination of Best Signal Representation and Decision Algorithm

The first experiment intends to determine the best representation of the data between the options of the naive basis (samples), full magnitude FFT, magnitude STFT, and MFCC features. For each data representation the experiment finds the result using the the voting classifier method, as well as the thresholded Kullback-Leibler divergence. This test is run using 10 recordings from each of 19 individual effects which were hand picked to represent the diversity of expected effects. The ground truth was determined by the consensus of two expert listeners who made their determinations independently. To determine the ground truth, each listener was given 10 recordings of the test signal from the target implementation, and 10 recordings from the implementation under test. They were asked to make a determination regarding whether these recordings were made using implementations that were functionally equivalent, or whether there was a detectable difference. Each listener

Table 1: Evaluation of feature representations and decision methods: Are the implementations equivalent?
Bold indicates agreement.

Preset Number	Human Listeners	Samples Vote %	Samples log KL	FFT Vote %	FFT log KL	STFT Vote %	STFT log KL	MFCC Vote %	MFCC log KL
3524	<i>no</i>	10%	16.65	10%	25.95	10%	24.32	10%	16.25
3526	<i>no</i>	10%	16.94	10%	26.66	10%	24.62	10%	16.35
615	<i>no</i>	30%	13.44	10%	26.90	10%	22.27	0%	16.78
7613	<i>no</i>	100%	13.71	0%	18.78	10%	18.93	0%	10.62
5012	<i>no</i>	100%	12.10	100%	13.65	100%	12.93	0%	9.47
4311	<i>no</i>	50%	12.97	0%	15.39	0%	15.10	0%	11.65
1411	<i>no</i>	0%	15.60	0%	28.39	0%	22.82	0%	17.61
5430	<i>no</i>	100%	12.63	0%	13.63	0%	23.62	0%	14.67
2211	<i>no</i>	0%	15.24	0%	27.47	0%	22.02	0%	17.90
221	<i>no</i>	100%	12.13	100%	14.31	90%	13.45	100%	9.97
225	<i>no</i>	100%	11.93	0%	17.89	100%	13.04	100%	9.58
4720	<i>yes</i>	100%	14.75	0%	13.78	0%	13.95	0%	12.72
5809	<i>yes</i>	80%	14.79	0%	13.41	10%	16.51	90%	9.64
224	<i>yes</i>	50%	11.93	40%	17.23	60%	13.99	100%	11.80
1413	<i>yes</i>	50%	11.24	60%	12.56	100%	11.34	80%	9.59
815	<i>yes</i>	100%	11.87	0%	12.52	100%	11.76	100%	8.21
1116	<i>yes</i>	0%	16.97	0%	19.61	0%	18.39	100%	11.63
4510	<i>yes</i>	100%	14.14	0%	12.64	100%	11.80	100%	8.97
12	<i>yes</i>	100%	-4.79	100%	7.82	100%	2.26	100%	7.41
Correct %		63.2%	73.7%	57.9%	78.9%	68.4%	73.7%	84.2%	68.4%

has been trained as a musician and recording engineer and should be considered an expert in the field of listening to musical effects. Neither listener was aware of the results from the other listener, or of the results of the QuAALM listening machine at the time of their determination.

The QuAALM listening machine was then run on the same set of recordings from each of the 19 effects and reported a Kullback-Leibler divergence information gain, and Voting Classifier percentage for each feature type. The results of this test are reported in Section six.

5.2. Experiment 2: Validation and Comparison against Human Listeners

Based on the results of Experiment 1, six listeners meeting the same qualifications as Experiment 1 were asked to make independent determinations of an additional 18 randomly chosen effects. These determinations were analyzed for consistency across the human listeners, and compared to the QuAALM machine results running using the Voting Classifier test and MFCC features. This analysis and results are included in Section six.

6. RESULTS

6.1. Experiment 1: Determination of Best Signal Representation and Decision Algorithm

The results of Experiment 1 are shown in Table 1. For each preset tested, the Vote % and log KL divergence are reported for each of the four feature types. The natural logarithm of the KL divergence is reported because the range of the KL divergence for such high dimensional data makes these values hard to display. After seeing the results, a threshold for the log KL divergence was chosen to optimize the results for each feature type. In a final implementation, the threshold would be considered a user parameter, and for

this test an oracle threshold was chosen to evaluate this test on its optimal possible performance. A reported log KL divergence below this threshold is scored as an equivalent implementation and a value above this threshold is scored as not equivalent. These thresholds are reported in Table 2.

Table 2: Log KL divergence thresholds

Samples	FFT	STFT	MFCC
12	14	13	10

Similarly, the Vote % represents the average of the votes for which the implementations were equivalent in the 10 runs of the voting classifier. A Vote % greater than or equal to 50% is scored as an equivalent implementation, below 50% is not. In Table 1, scores which agree with the judgment of the human listeners are in boldface while those that disagree are not. Finally, a percentage of presets for which QuAALM agrees with the human listeners is calculated for each test and reported at the bottom of Table 1.

From these results we can see that both the KL divergence and voting classifier methods have some success depending on the features used. However, the clear standout is the voting classifier method when used with MFCC features. Not only does it show the highest percentage of correct classifications, but for the instances where it passed, the classification margins are wider. This implies that there might be a higher robustness to noise using this test. Because of these results, we chose the voting classifier method operating on MFCC features for our deployment and run the validation tests in Experiment 2 using this method.

Table 3: Validation and comparison against human listeners: Are the implementations equivalent?

PresetNumber	Listener1	Listener2	Listener3	Listener4	Listener5	Listener6	Consensus	QuAALM	Verdict
234	yes	yes	pass						
322	yes	yes	pass						
536	yes	no	fail						
1333	yes	yes	yes	no	yes	yes	yes	yes	pass
1910	yes	yes	pass						
2317	no	no	pass						
3211	yes	yes	pass						
4034	yes	yes	yes	yes	yes	no	yes	yes	pass
4138	yes	yes	pass						
4727	no	yes	no	yes	no	no	no	no	pass
4814	no	yes	no	yes	yes	yes	yes	no	fail
5729	yes	yes	yes	no	yes	yes	yes	yes	pass
5911	yes	yes	yes	yes	no	yes	yes	yes	pass
6663	yes	yes	pass						
6910	no	no	pass						
7211	yes	no	fail						
7419	yes	no	no	no	no	yes	no	no	pass
8214	no	no	pass						

Table 4: Validation scores

Precision	Recall	F ₁
1.0	0.625	0.8696

6.2. Experiment 2: Validation and Comparison against Human Listeners

For Experiment 2, we randomly chose 18 presets from the H8000, making sure that no two were from the same bank, and tested them on both the H8000 and H9000 hardware. As in Experiment 1, we used a digital audio connection and tested at a 48 kHz sampling rate, the test signal consisted of a 3 second long Gaussian White Noise burst, 3 seconds of silence, a 3 second exponential sine chirp, and a final 3 seconds of silence. Ten independent recordings were made of both the H8000 and the H9000 for each preset.

The 20 recordings were each listened to by the six listeners, independently and without sharing their results, and a determination was made as to whether the two implementations sounded equivalent. For 11 of the 18 presets, all listeners agreed on their conclusions; 8 times that the implementations were the same, and 3 times that they were different. On a further 4, a single listener was in the minority, while the final 3 presets were a vote of 4 to 2. No presets came to a split decision among the 6 listeners. We believe that this shows the listeners have a high degree of consistency when determining if implementation differences were perceptually equivalent. These preset numbers and results are tabulated in Table 3.

The final ground truth consensus was reached by a vote of the human listeners and is also tabulated in Table 3.

For each preset, the same 20 recordings were fed into the QuAALM machine running the voting classifier method on MFCC features, and its judgment was recorded in Table 3, along with a comparison of its judgment and the ground truth consensus.

We can see that the QuAALM machine disagreed with the human listeners for only 3 of the 18 presets, and in each of these instances it reported a false negative rather than a false positive.

This leads to a Precision score of 1, a Recall score of 0.625, and an F₁ score of 0.8696. These values are shown in Table 4.

As mentioned earlier, as a practical consideration, the perfect Precision score and relatively high F₁ score satisfied our goals and allowed us to put QuAALM into service testing real implementations.

7. CONCLUSIONS

In this paper, we developed an automated listening machine which meets the qualifications to serve as the first line of defense for quality assurance and automated testing of our effects hardware. In doing so we also developed a more robust framework for analyzing the similarity of effects utilizing some amount of random time variation. This framework may have applications to other listening intensive work like the automatic labeling of effect type.

The specific implementation described here was sufficient to serve our purposes, however, many improvements could be considered. Some potential improvements might come from exploring different test signals, or relaxing the strict independence criteria in the probability model, as we expect that there is likely some dependence between these samples. Additionally, given the limited test data reported here, we expect that the QuAALM machine might be failing for some specific types of effects. A larger data set will allow further investigation into these effect type and may lead us to new ways to improve the system.

Additionally, the comparisons made here were between two implementations of the same digital effects. However, the underlying technique should be useful to in determining the quality of analog models for certain types of randomized effects like chorus, flanger, phaser, or rotary speaker emulations. It would be interesting to see if this were the case.

8. ACKNOWLEDGMENTS

Thank you to Jeff Schumacher, Pete Bischoff, Patrick Flores, and Nick Solem for performing listening tests.

9. REFERENCES

- [1] Shijun Xiang, Jiwu Huang, and Rui Yang, “Time-scale invariant audio watermarking based on the statistical features in time domain,” in *International Workshop on Information Hiding*. Springer, 2006, pp. 93–108.
- [2] Michael Arnold, “Audio watermarking: Features, applications and algorithms,” in *Multimedia and Expo, 2000. ICME 2000. 2000 IEEE International Conference on*. IEEE, 2000, vol. 2, pp. 1013–1016.
- [3] Christian Neubauer and Jārgen Herre, “Digital watermarking and its influence on audio quality,” in *Audio Engineering Society Convention 105*, Sep 1998.
- [4] Daniele Barchiesi and Joshua Reiss, “Reverse engineering of a mix,” *J. Audio Eng. Soc*, vol. 58, no. 7/8, pp. 563–576, 2010.
- [5] Stanislaw Gorlow and Joshua D Reiss, “Model-based inversion of dynamic range compression,” *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 21, no. 7, pp. 1434–1444, 2013.
- [6] Luwei Yang, Khalid Z Rajab, and Elaine Chew, “The filter diagonalisation method for music signal analysis: frame-wise vibrato detection and estimation,” *Journal of Mathematics and Music*, pp. 1–19, 2017.
- [7] Elias Pampalk, Simon Dixon, and Gerhard Widmer, “On the evaluation of perceptual similarity measures for music,” in *of: Proceedings of the Sixth International Conference on Digital Audio Effects (DAFx-03)*, 2003, pp. 7–12.
- [8] Marko Helén and Tuomas Virtanen, “A similarity measure for audio query by example based on perceptual coding and compression,” in *Proc. 10th Int. Conf. Digital Audio Effects (DAFX)*, 2007.
- [9] Tuomas Virtanen and Marko Helén, “Probabilistic model based similarity measures for audio query-by-example,” in *Applications of Signal Processing to Audio and Acoustics, 2007 IEEE Workshop on*. IEEE, 2007, pp. 82–85.
- [10] Mike Elliott, “How to Null Test Your Gear: Part 1,” Available at <https://music.tutsplus.com/tutorials/how-to-null-test-your-gear-part-1--cms-22425>, accessed April 09, 2017.
- [11] Md Sahidullah and Goutam Saha, “Design, analysis and experimental evaluation of block based transformation in mfcc computation for speaker recognition,” *Speech Communication*, vol. 54, no. 4, pp. 543–565, 2012.
- [12] Gordon Hughes, “On the mean accuracy of statistical pattern recognizers,” *IEEE Transactions on Information Theory*, vol. 14, no. 1, pp. 55–63, 1968.
- [13] Yuichiro Anzai, *Pattern recognition and machine learning*, chapter Linear Models for Classification, pp. 200–203, Elsevier, 2012.
- [14] Yuichiro Anzai, *Pattern recognition and machine learning*, chapter Mixture Models and EM, pp. 450–451, Elsevier, 2012.
- [15] Giovanni Seni and John F Elder, “Ensemble methods in data mining: improving accuracy through combining predictions,” *Synthesis Lectures on Data Mining and Knowledge Discovery*, vol. 2, no. 1, pp. 1–126, 2010.
- [16] D. M. W. Powers, “Evaluation: From precision, recall and f-measure to roc., informedness, markedness & correlation,” *Journal of Machine Learning Technologies*, vol. 2, no. 1, pp. 37–63, 2011.

BLIND UPMIX FOR APPLAUSE-LIKE SIGNALS BASED ON PERCEPTUAL PLAUSIBILITY CRITERIA

Alexander Adami

International Audio Laboratories*,
Friedrich-Alexander Universität Erlangen
Erlangen, Germany
alexander.adami@audiolabs-erlangen.de

Lukas Brand

International Audio Laboratories*,
Friedrich-Alexander Universität Erlangen
Erlangen, Germany
lukas.brand@fau.de

Sascha Disch

Fraunhofer IIS,
Erlangen, Germany
sascha.disch@iis.fraunhofer.de

Jürgen Herre

International Audio Laboratories*,
Friedrich-Alexander Universität Erlangen
Erlangen, Germany
juergen.herre@audiolabs-erlangen.de

ABSTRACT

Applause is the result of many individuals rhythmically clapping their hands. Applause recordings exhibit a certain temporal, timbral and spatial structure: claps originating from a distinct direction (i.e., from a particular person) usually have a similar timbre and occur in a quasi-periodic repetition. Traditional upmix approaches for blind mono-to-stereo upmix do not consider these properties and may therefore produce an output with suboptimal perceptual quality to be attributed to a lack of plausibility. In this paper, we propose a blind upmixing approach of applause-like signals which aims at preserving the natural structure of applause signals by incorporating periodicity and timbral similarity of claps into the upmix process and therefore supporting plausibility of the artificially generated spatial scene. The proposed upmix approach is evaluated by means of a subjective preference listening test.

1. INTRODUCTION

Applause is a sound texture composed of many individual hand claps produced by a crowd of people [1]. It can be imagined as a superposition of discrete and individually perceivable transient foreground clap events and additional noise-like background originating from dense far-off claps as well as reverberation [2]. Due to the high number of transient events, applause-like signals form a special signal class which often needs a dedicated processing [3–5]. This is possible since applause sounds are well detectable among mixtures of other signal classes [6].

At first sight, the nature of applause sound textures may seem totally random [7]. However, previous publications and listening experiments indicate that a fully randomized applause texture, consisting of temporally and spatially randomized events with random timbre, is perceived as unnatural and non-plausible by listeners [8].

It was shown in [9] for sparse to medium dense applauses, or applause containing at least distinct dominant foreground claps, that listeners do expect a quasi-periodic occurrence of clapping events of a certain stable timbre [1, 10] originating from selected spatial locations in order to perceive a plausible spatial impression of being exposed to real-world applause. Phrased differently,

listeners seem to be able to distinguish the clapping sounds from single individual persons, each having a distinct clap timbre, in the foreground signal from the much more dense background signal and perceive these as repeated events of similar timbre originating from the same source.

If these special properties are disturbed, listeners report perceptual quality degradations in the stability of the perceived spatial image and also a lack of plausibility of the spatial scene.

Building upon the findings in [9], we propose a blind (or non-guided) spatial upmix from mono to stereo of applause signals that preserves the important properties of quasi-periodicity and timbral similarity of foreground claps to be attributed to a distinct source. Thereby, the plausibility of the artificially generated spatial scene is strengthened.

The blind upmix proposed in this paper relies on a separation of transient and individually perceivable foreground claps and the noise-like background, being reminiscent of the guided applause upmix published in [3]. While the noise-like background is subjected to decorrelation, the separated foreground claps are distributed by panning to arbitrarily chosen positions in the stereo sound stage. Though operated in an unguided manner, as a novel contribution, our algorithm most importantly ensures that each position will be populated by a clap event of suitable timbre in a quasi-periodic fashion, thus supporting the notion of plausibility of the artificially generated spatial scene.

2. APPLAUSE SEPARATION

Before the actual upmix process can take place, the monophonic input applause signal $A(k, m)$ has to be decomposed into a signal part corresponding to distinctly and individually perceivable *foreground* claps $C(k, m)$, and a signal part corresponding to the noise-like *background* signal $N(k, m)$ [9], where k and m denote the discrete frequency and block index in short-time Fourier transform domain. The frequency transformation is done with high temporal resolution, i.e., a block size of 128 samples with 64 samples overlap is used. The corresponding signal model is given by Eq. 1:

$$A(k, m) = C(k, m) + N(k, m). \quad (1)$$

* A joint institution of the Friedrich-Alexander-Universität Erlangen-Nürnberg (FAU) and Fraunhofer IIS, Germany

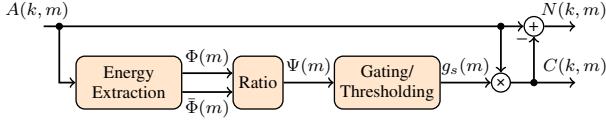


Figure 1: Block diagram of applause separation into distinctive individually perceivable foreground claps $C(k, m)$ and noise-like background $N(k, m)$.

The applause separation proposed in this paper is a modified version based on the approaches used in [9, 11]. Figure 1 depicts a block diagram describing the basic structure of the applause separation processing. Within the energy extraction stage, an instantaneous energy estimate $\Phi(m)$ as well as an average energy estimate $\bar{\Phi}(m)$ is derived from the input applause signal. The instantaneous energy is given by $\Phi(m) = \|A(k, m)\|_2$, where $\|\cdot\|_2$ denotes the L2-norm. The average energy is determined by a weighted sum of the instantaneous energies around the current block and given by

$$\bar{\Phi}_A(m) = \frac{\sum_{\mu=-M}^M \Phi_A(m-\mu) \cdot w(\mu+M)}{\sum_{\mu=-M}^M w(\mu+M)}, \quad (2)$$

where $w(\mu)$ denotes a weighting window (squared sine-window) with window index μ and length $L_w = 2M + 1$. In the next stage, the ratio of instantaneous and average energy $\Psi(m)$ is computed. It serves as an indicator whether a discrete clap event is present and is given by

$$\Psi(m) = \frac{\Phi_A(m)}{\bar{\Phi}_A(m)}. \quad (3)$$

If the current block contains a transient, the instantaneous energy is large compared to the average energy and the ratio is significantly greater than one. On the other hand, if there is only noise-like background at the current block, instantaneous and average energy are almost similar and consequently, the ratio is approximately one. The basic separation gain $g_s(m)$ can be computed according to

$$\hat{g}_s(m) = \sqrt{\max\left(1 - \frac{g_N}{\Psi(m)}, 0\right)}, \quad (4)$$

which is a signal adaptive gain $0 \leq \hat{g}_s(m) \leq 1$, solely dependent on the energy ratio $\Psi(m)$. To prevent dropouts in the noise-like background signal, the constant g_N is introduced. It determines the amount of the input signal's energy remaining within the noise-like background signal during a clap. The constant was set to $g_N = 1$, which corresponds to the average energy remaining within the noise-like background signal. Since for the upmixing we are mainly interested in the directional sound component of a clap (i.e., the attack phase), thresholding or gating is applied to $\Psi(m)$ according to

$$g_s(m) = \begin{cases} 0 & \text{if } \Psi(m) < \tau_{\text{attack}} \\ \hat{g}_s(m) & \text{if } \Psi(m) \geq \tau_{\text{attack}} \quad \text{if } g_s(m-1) = 0 \\ 0 & \text{if } \Psi(m) < \tau_{\text{release}} \\ \hat{g}_s(m) & \text{if } \Psi(m) \geq \tau_{\text{release}} \quad \text{if } g_s(m-1) \neq 0. \end{cases} \quad (5)$$

This means, the separation gain $g_s(m)$ is only different from zero after the energy ratio surpassed an attack threshold τ_{attack} and only as long as it is above a release threshold τ_{release} . When $\Psi(m)$ falls below τ_{release} , the separation gain is set back to zero. For the separation attack and release threshold $\tau_{\text{attack}} = 2.5$ and $\tau_{\text{release}} = 1$ were used. The final separated signals are obtained according to

$$C(k, m) = g_s(m) \cdot A(k, m) \quad (6)$$

$$N(k, m) = A(k, m) - C(k, m). \quad (7)$$

Figure 2 depicts waveforms and spectrograms of an exemplary applause signal on the left and the corresponding separated clap signal in the middle, as well as the noise-like background signal on the right. Sound examples are available at <https://www.audiolabs-erlangen.de/resources/2017-DAFx-ApplauseUpmix>.

3. APPLAUSE UPMIX

After having the input applause signal separated into individual claps and noise-like background, the signal parts are upmixed separately. Upmixing the noise-like background was realized by using the original background signal as the left channel and a decorrelated version $\tilde{N}(k, m)$ as the right channel of the upmix. Decorrelation was achieved by scrambling the temporal order of the original noise-like background signal. This method was originally proposed in [4], where the time signal is divided into segments which are themselves divided into overlapping subsegments. Sub-segments are windowed and their temporal order is scrambled. Applying overlap-add yields the decorrelated output signal. The processing for the noise-like background signal was modified in the sense that it operates in short-time Fourier transform (STFT)-domain and with a small segment size. A segment size of 10 blocks corresponding to 13 ms was used, where each block represented a subsegment.

Upmixing of foreground claps makes use of inter-clap relations. This means, the upmixing process incorporates the assumptions that claps originating from a certain direction should sound similar, as well as that claps originating from a certain direction should exhibit some sort of periodicity.

In the beginning, arbitrary discrete directions ϕ_d within the stereo panorama are chosen, where $d = 0 \dots D - 1$ denotes the index of a distinctive direction within the direction vector $\Phi = [\phi_0, \phi_1, \dots, \phi_{D-1}]$. The total number of directions is given by D . Furthermore, the mean clap spectrum for each detected clap is computed, whereby a clap is considered as a set of consecutive blocks, each of which having non-zero energy and framed by at least one block to each side containing zero energy. With the start and stop block index $\gamma_s(c)$ and $\gamma_e(c)$ of clap c , the claps' mean spectra are given by

$$\hat{C}_c(k) = \frac{1}{\gamma_e(c) - \gamma_s(c) + 1} \sum_{m=\gamma_s(c)}^{\gamma_e(c)} |C(k, m)|^2. \quad (8)$$

The upmix process operates on a per clap basis rather than on individual blocks. Considering the first clap to be upmixed, the target direction $\Theta(c)$ is chosen randomly from the vector of available directions $\Phi(d)$. The spectrum of the current clap is stored in the matrix $S(k, d)$ which holds the mean spectra of the last clap assigned to a distinctive direction:

$$S(k, d) = \begin{cases} \hat{C}_c(k) & \text{if } S(k, d) = 0 \\ 0.5(S(k, d) + \hat{C}_c(k)), & \text{else} \end{cases} \quad (9)$$

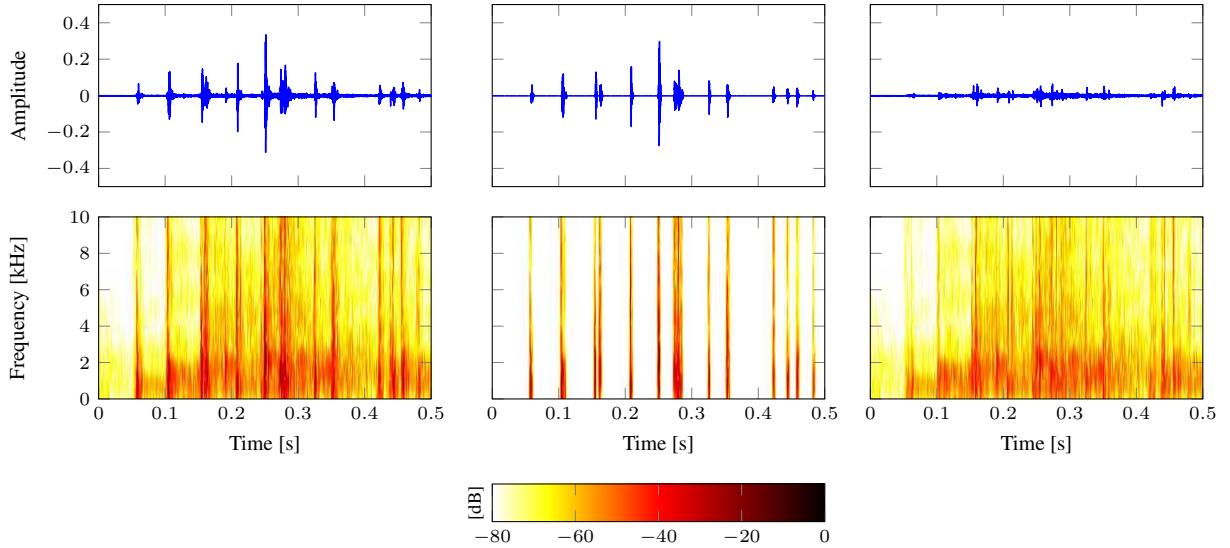


Figure 2: Waveforms and spectrograms of an applause signal (left) and the respective separated clap signal (middle) and residual noise-like background (right). In the Figure the spectrogram is only plotted in the range of 0 to 10 kHz.

Additionally, the start block number of the current clap is stored in a vector $T(d)$ holding the respective start block number of the last clap assigned to a direction:

$$T(d) = \gamma_s(c). \quad (10)$$

For all further claps, it is determined how well the respective current clap fits to the claps distributed previously in each direction. This is done with respect to timbral similarity as well as with respect to temporal periodicity. As a measure for timbral similarity, the log spectral distances of the current clap to the previously stored mean spectra of the claps attributed to a direction is computed according to

$$\text{LSD}(d) = \sqrt{\frac{1}{K_u - K_l + 1} \sum_{k=K_l}^{K_u} \left[10 \log_{10} \left(\frac{S_d(k)}{\widehat{C}_c(k)} \right) \right]^2}, \quad (11)$$

where K_l and K_u denote the lower and upper bin of the frequency region relevant for similarity. Similarity is mostly influenced by the spectral peaks/resonances resulting from the air cavity between hands as a consequence of the positioning of hands relative to each other while clapping; these peaks are in the region up to 2 kHz [1, 7, 10]. Based on this observation and including some additional headroom, frequency bins corresponding to the frequency region between 200 Hz and 4 kHz were considered in the log spectral distance measurement.

To determine how well the current clap fits into a periodicity scheme within a certain direction, the time difference (i.e., in blocks) of the current clap to the last distributed clap in every direction is computed:

$$\Delta(d) = \gamma_s(c) - T(d). \quad (12)$$

The resulting time differences are compared to a target clap frequency. In an experiment with more than 70 participants, Neda

et. al [12] found clap rates of 2 to 6 Hz with a peak at around 4 Hz. Based on this experimental data and some internal test runs, we chose a target clap rate of $\delta_t = 3$ Hz, corresponding to a target block difference of $\delta_m = 250$ blocks, with an additional tolerance scheme of ± 1 Hz. This means for the time difference, it has to be within the range of $\delta_m - 62$ and $\delta_m + 125$, i.e., 188 and 375 blocks to be considered as a periodic continuation of claps in a direction. In the case that two or more claps occur simultaneously the applause separator detects only one single clap which leads to a gap in the periodicity pattern of the directions the other (masked) claps would have belonged to. To compensate for this effect the search for periodicity is extended to multiples of the expected target block difference, specifically to $3 \cdot \delta_m$. The actually used time difference value is the one with smallest absolute difference to the considered multiples of the target frequency. If the current raw time difference is outside of the tolerance scheme, it is biased with a penalty.

In the next step, log spectral differences as well as time difference are normalized to have zero mean and a variance of one:

$$\widehat{\text{LSD}}(d) = \frac{\text{LSD}(d) - \mu_{\text{LSD}}}{\sigma_{\text{LSD}}} \quad (13)$$

$$\widehat{\Delta}(d) = \frac{\Delta(d) - \mu_{\Delta}}{\sigma_{\Delta}}. \quad (14)$$

Means μ and standard deviations σ are computed using the respective raw time differences and log spectral distances corresponding to the last 25 assigned claps. Finding the most suitable direction for the current clap can be considered as the problem of finding the direction d where the length of a vector $\begin{bmatrix} \widehat{\text{LSD}}(d) \\ \widehat{\Delta}(d) \end{bmatrix}$ is minimal. The vector norm $\Lambda(d)$ for every direction is computed by

$$\Lambda(d) = \sqrt{\widehat{\text{LSD}}(d)^2 + \widehat{\Delta}(d)^2}. \quad (15)$$

The direction to which the current clap fits best is determined as

the index d_{new} where $\Lambda(d)$ is minimal:

$$d_{\text{new}} = \arg \min_d \Lambda(d). \quad (16)$$

The current clap is then assigned to the direction $\Phi(d_{\text{new}})$ and the buffer for the respective mean and standard deviation computations as well as $S(k, d_{\text{new}})$ and $T(d_{\text{new}})$ are updated.

As long as each available direction was not assigned with at least one clap, it is additionally checked whether the vector norm

$\Lambda(d_{\text{new}})$ is larger than a threshold $\tau_\Lambda = \sqrt{\left(\frac{\tau_{\text{LSD}}}{\sigma_{\text{LSD}}}\right)^2 + \left(\frac{\tau_\Delta}{\sigma_\Delta}\right)^2}$ with $\tau_{\text{LSD}} = 1.9$ and $\tau_\Delta = 5.5$. If so, the current clap is considered as too different from the claps in the so far assigned directions and, consequently, is assigned to a new or ‘free’ direction. The new direction is chosen randomly from the available free directions.

Finally, the blocks corresponding to the current clap $\gamma_s(c) \leq m \leq \gamma_e(c)$ are scaled with the corresponding panning coefficient $g(\phi)$ [13] to appear under the determined direction $\phi_{d_{\text{new}}}$. Left and right clap signal are obtained according to

$$C_L(k, m) = g(\phi_{d_{\text{new}}}) \cdot C(k, m) \quad (17)$$

$$C_R(k, m) = \sqrt{1 - g(\phi_{d_{\text{new}}})^2} \cdot C(k, m). \quad (18)$$

The final upmixed left and right signals are obtained by superposing the respective left and right upmixed clap and noise signals:

$$L(k, m) = C_L(k, m) + \frac{1}{\sqrt{2}} N(k, m) \quad (19)$$

$$R(k, m) = C_R(k, m) + \frac{1}{\sqrt{2}} \hat{N}(k, m). \quad (20)$$

4. SUBJECTIVE EVALUATION

To subjectively evaluate the performance of the proposed blind upmix method, a listening test was performed, where the plausibility criteria-driven upmix was compared to a context-agnostic upmix.

4.1. Stimuli

Two sets of stimuli were used for the listening test: one set consisted of synthetically generated applause signals with controlled parameters, whereas the other set consisted of naturally recorded applause signals. All signals were sampled at a sampling rate of 48 kHz. Seven synthetic signals were generated based on the method proposed in [2] and with respective number of virtual people $\hat{P}_\Sigma = [2, 4, 8, 16, 32, 64, 128]$. The signals had a uniform length of 5 seconds.

The set of naturally recorded signals was a subset of the stimuli used in [11]. For reasons of comparability the same stimuli numbering was applied in this paper. In particular stimuli with numbers 1, 7, 10, 11, 13, 14, and 18 were used, each of which having a uniform length of 4 seconds. Both stimulus sets covered an applause density range from very sparse to medium-high.

Stimuli of both sets were blindly upmixed using different processing schemes. In the first scheme, upmixing was done according to the above proposed processing and will be denoted as *proposedUpmix*. In the second scheme, the applause signals were also separated into claps and noise-like background but the claps were distributed in a context-agnostic manner, i.e., claps were randomly assigned to the available directions within the stereo panorama.

This type of upmix is denoted as *randomUpmix*. For both upmixing schemes, 13 discrete directions were available; these were in particular $\Phi = [\pm 30, \pm 25, \pm 20, \pm 15, \pm 10, \pm 5, 0]$. To ensure equal loudness, all stimuli were loudness-normalized to -27 LUFS (loudness unit relative to full scale). Exemplary stimuli are available at <https://www.audiolabs-erlangen.de/resources/2017-DAFx-ApplauseUpmix>.

4.2. Procedure

The listening test procedure followed a forced-choice preference test methodology. This means, in each trial subjects are presented with two stimuli in randomized order as hidden conditions. Subjects have to listen to both versions and were asked which stimulus sounds more plausible. The order of trials was randomized, as well.

Before the test, subjects were instructed that applause can be considered as a superposition of distinctive and individually perceptible foreground claps and more noise-like background. It was furthermore stated that claps usually exhibit certain temporal, timbral, and spatial structures, e.g., claps originating from the same person do not vary considerably in spatial position and clap frequency, etc. As listening task, subjects were asked to focus on plausibility of foreground claps.

After the instructions, there was a training to firstly familiarize subjects with the concept of foreground claps and noise-like background and secondly with plausibility of foreground claps. In the first case, the same procedure as in [9, 11] was also used here: four exemplary stimuli of varying density and accompanied with additional supplementary explanations regarding foreground clap density were provided. For the second, two synthetically generated stereo stimuli with $\hat{P}_\Sigma = 6$ were presented whereby in one of which timbre and time intervals between consecutive claps were modified to decrease plausibility. Subjects were provided with supplementary information regarding the stimuli and their expected plausibility.

It should be noted that instructing subjects on such a detailed level appears to bear a risk of biasing them into a certain direction. However, this came as a result of a pre-test where subjects were simply asked to rate naturalness of applause sounds without providing any further information. In this pre-test, it was found in interviews that subjects based their ratings of naturalness on quite different aspects of the stimuli. For example, for some subjects, naturalness was predominantly influenced by the room/ambient sound, others focused more on imperfections of the applause synthesis and did not take spatial aspects into account. The plurality of influencing factors made it impossible to obtain a reasonably consistent rating between the subjects. Thus, it was decided to focus the listening test on the notion of foreground claps. Furthermore, it emerged from the subject interviews that asking for *plausibility* potentially puts more focus on properties of the clap sounds themselves than using the more broadly defined term *naturalness*.

The listening test was conducted in an acoustically optimized sound laboratory at the International Audio Laboratories Erlangen. Stimuli were presented via KSdigital ADM 25 loudspeakers.

4.3. Subjects

A total number of 17 subjects among which 3 female and 14 male took part in the listening test. Subjects’ average age was 33.1 (SD = 8) years ranging from 23 to 53 years. All subjects were stu-

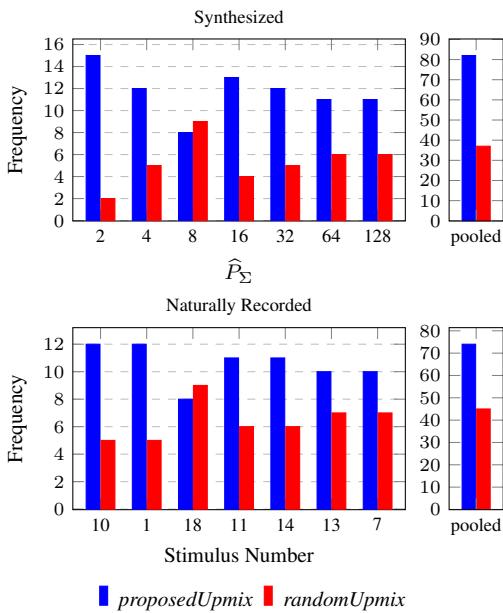


Figure 3: Histogram of subjects' preference for synthetically generated (top plane) and naturally recorded (bottom plane) stimuli. Additionally, the pooled data across stimuli is provided.

		Synthesized							
\hat{P}_Σ	p-value	2	4	8	16	32	64	128	pooled
2	.001								<.001
		Recorded							
Stimulus	p-value	10	1	18	11	14	13	7	pooled
10	.072								.005

Table 1: P-values of the statistical analysis by means of binomial testing of preference data.

dents or employees either at Fraunhofer IIS or at the International Audio Laboratories Erlangen with varying degree of experience. In a questionnaire after the listening test, subjects reported in how many listening test they have participated in so far; a number of up to 5 was considered as low-experienced (2 subjects), a number between 6 and 14 was considered as medium-experienced (4 subjects), and anything above was considered as expert listener (11 subjects).

4.4. Results

Figure 3 depicts subjects' preferences for each stimulus where in the top plane responses for the synthetically generated and in the bottom plane responses for the naturally recorded stimuli are depicted. Stimuli are ordered according to increasing applause density. On the respective right hand sides, the pooled data across stimuli are provided.

For the synthesized signals and except for $\hat{P}_\Sigma = 8$, subjects in total preferred *proposedUpmix*. There is even a trend of the subjects' preference recognizable: at (very) low density the *proposedUpmix* is clearly preferred over *randomUpmix* and with increasing density, subjects' preference for *proposedUpmix* decreased.

An exception is $\hat{P}_\Sigma = 8$ where both upmix methods were about equally frequently chosen. The pooled results support a general preference for *proposedUpmix*.

Also regarding the naturally recorded stimuli, there is a general preference towards *proposedUpmix* visible in the pooled results. Considering the individual stimuli, the data suggests that except for stimulus number 18, where preference for both methods was about similar, *proposedUpmix* was preferred. There is also a weak trend in the data visible indicating that preference for *proposedUpmix* decreases with increasing density. In no case of both stimulus sets, the *randomUpmix* was clearly preferred over *proposedUpmix*.

The indicated trend of preferences in both stimulus sets makes sense given that at high densities clap events occur in a more chaotic and pseudo-random fashion and the event rate gets too dense to be evaluated by the human auditory system on a per clap basis. Instead, the denser the clapping becomes, the more it can be considered as a sound texture and, in further consequence, general signal statistics gain more relevance for perception than properties of individual clap events [14].

Additionally to the visual evaluation, statistical test results are provided in Table 1. For every stimulus as well as the respective pooled data, a one-tailed binomial test was carried out which tested against the hypothesis that the upmix methods were chosen by chance, i.e., an expected relative frequency of 0.5. Considering the individual stimuli only results for $\hat{P}_\Sigma \in [2, 16]$ of the synthesized and none of the recorded stimulus set are significant, where a significance level of $\alpha = 0.05$ was used. However, the overall results show that for both stimulus sets, *proposedUpmix* was clearly and statistically significantly preferred over *randomUpmix*.

5. CONCLUSION

A blind upmix approach for applause-like signals incorporating quasi-periodicity and timbral similarity of consecutive claps from individual spatial directions was proposed and evaluated. The input signal was firstly decomposed into distinctive and individually perceivable foreground claps and more noise-like background. While the background signal was simply decorrelated, the foreground claps were distributed amongst random positions in the stereo panorama based on timbral similarity and temporal periodicity of claps. The proposed upmix was evaluated by means of a preference test and based on synthetically generated as well as naturally recorded applause stimuli. Results showed that the perceptual quality with respect to plausibility of the spatial scene produced by the proposed upmix was clearly preferred over the one of an upmix where foreground claps were distributed in a context-agnostic manner. Statistical analysis proved subjects' overall preference to be significant.

6. REFERENCES

- [1] B. H. Repp, "The Sound of two hands clapping: An exploratory study," *Journal of the Acoustical Society of America*, vol. 81, no. 4, pp. 1100–1109, 1987.
- [2] A. Adami, S. Disch, G. Steba, and J. Herre, "Assessing Applause Density Perception Using Synthesized Layered Applause Signals," in *Proceedings of the 19th International Conference on Digital Audio Effects (DAFx-16)*, Brno, Czech Republic, 2016, pp. 183–189.

- [3] S. Disch and A. Kuntz, “A Dedicated Decorrelator for Parametric Spatial Coding of Applause-like Audio Signals,” in *Microelectronic Systems*, A. Heuberger, G. Elst, and R. Hanke, Eds. Springer-Verlag Berlin Heidelberg, 2011, pp. 363–371.
- [4] G. Hotho, S. van de Par, and J. Breebaart, “Multichannel Coding of Applause Signals,” *EURASIP Journal on Advances in Signal Processing*, vol. 2008, 2008.
- [5] F. Ghido, S. Disch, J. Herre, F. Reutelhuber, and A. Adami, “Coding of Fine Granular Audio Signals Using High Resolution Envelope Processing (HREP),” in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, New Orleans, USA, 2017.
- [6] C. Uhle, “Applause Sound Detection,” *J. Audio Eng. Soc*, vol. 59, no. 4, pp. 213–224, 2011.
- [7] L. Peltola, C. Erkut, P. R. Cook, and V. Välimäki, “Synthesis of Hand Clapping Sounds,” *Audio, Speech, and Language Processing, IEEE Transactions on*, vol. 15, no. 3, pp. 1021–1029, 2007.
- [8] W. Ahmad and A. M. Kondoz, “Analysis and Synthesis of Hand Clapping Sounds Based on Adaptive Dictionary,” in *Proceedings of the International Computer Music Conference*, vol. 2011, Huddersfield, UK, 2011, pp. 257–263.
- [9] A. Adami, L. Brand, and J. Herre, “Investigations Towards Plausible Blind Upmixing of Applause Signals,” in *142nd International Convention of the AES*, Berlin, Germany, 2017.
- [10] A. Jylhä and C. Erkut, “Inferring the Hand Configuration from Hand Clapping Sounds,” in *Proceedings of the 11th International Conference on Digital Audio Effects (DAFx-08)*, Espoo, Finland, 2008.
- [11] A. Adami and J. Herre, “Perception and Measurement of Applause Characteristics: Wahrnehmung und Messung von Applauseigenschaften,” in *Proceedings of the 29th Tonmeisterstagung (TMT29)*. Cologne, Germany: Verband Deutscher Tonmeister e.V., 2016, pp. 199–206.
- [12] Z. Néda, E. Ravasz, T. Vicsek, Y. Brechet, and A.-L. Barabási, “Physics of the rhythmic applause,” *Phys. Rev. E*, vol. 61, no. 6, pp. 6987–6992, 2000.
- [13] V. Pulkki, “Virtual Sound Source Positioning Using Vector Base Amplitude Panning,” *J. Audio Eng. Soc*, vol. 45, no. 6, pp. 456–466, 1997.
- [14] J. H. McDermott and E. P. Simoncelli, “Sound Texture Perception via Statistics of the Auditory Periphery: Evidence from Sound Synthesis,” *Neuron*, vol. 71, no. 5, pp. 926–940, 2011.

CO-AUTHORSHIP AND COMMUNITY STRUCTURE IN THE DAFX CONFERENCE PROCEEDINGS: 1998–2016

Alex Wilson

Acoustics Research Centre,

University of Salford

Greater Manchester, UK

a.wilson1@edu.salford.ac.uk

ABSTRACT

This paper presents the co-authorship network of the DAFX conference series, from its inception in 1998 to the present, along with subsequent analysis. In total 1,281 unique authors have contributed 1,175 unique submissions to this conference series. The co-authorship network is revealed to contain a large weakly connected component containing 667 authors ($\approx 52\%$ of the total network). The size of this component compares well to previous studies of other conference series of similar age and scope. Within this connected component, 24 communities were detected using the Louvain method. While some communities have formed based on geographic proximity, links between communities are observed. This shows a high level of collaboration in the network, possibly due to the speciality of the conference and the movement of academics throughout Europe.

1. INTRODUCTION

The DAFX conference series, which began in 1998 as a workshop series, evolved from the Action-G6 of the European COST programme, named “digital audio effects”. Its objectives were originally described as follows¹.

1. To compare the different methods of the European teams in terms of algorithms, implementations and musical use.
2. To bring together the knowledge of different European teams in the domain of digital sound processing, here designated as “digital audio effects”, in a form which can be made available inside and outside of the teams themselves.

This second objective refers to the development of research networks, within Europe and also beyond. This paper aims to examine the breadth and structure of the resultant network after a period of two decades. To this end a co-authorship network was created, indicating which authors of DAFX submissions have authored papers together. From this, the nature of collaboration within the DAFX community was examined. More clearly, the goals of this study were the following:

1. Collect bibliographic data from the entire DAFX conference series, 1998–2016
2. Create co-authorship network from this data
3. Identify connected components
4. Identify communities within the largest connected component

¹http://www.consilium.europa.eu/uedocs/cms_data/docs/dynadoc/out/cost/en/cost_at_g6.pdf

2. BACKGROUND AND LITERATURE REVIEW

A graph is a structure which describes the relationships between a set of nodes. The links between nodes are called edges. Graphs are often categorised as being undirected or directed: a graph can be called undirected when the edges between pairs of nodes have no directional information, or directed when the edge describes a one-way connection between nodes. This paper will consider co-authorship graphs exclusively.

2.1. Co-authorship networks

Co-authorship is one of the more frequently-investigated forms of scientific collaboration. It has been noted that the proportion of single-authored papers, across various scientific disciplines, has decreased, from 25% in 1980 to 11% in 2000 [1]. Multi-author papers also gain more citations, including self-citations [2].

Scientific collaboration networks are typically represented as *undirected, unweighted* graphs, i.e. what is represented is a simple binary classification of whether or not two individuals have collaborated on a paper. This means that a lot of important information is usually absent, such as the number of co-authored works by the pair, and the assumption is made that the partnership is completely equal. In real collaboration networks one can often see that a pair will co-author numerous works or that the share of work is not equal. When the number of co-authored works is included it is typically represented by the weight of an edge between two nodes. It has been shown that *weighted, undirected* co-authorship networks have a high correlation with social networks, themselves influenced by geographic proximity [3, 4].

A number of strategies exist for creating *directed* co-authorship networks, which encode information about the partnership between the authors — which author should be given priority. These include “first author takes all” or “last author takes all” strategies but these both assume no relationships between intermediate authors in the list of authors. More nuanced approaches have been developed, which include interactions between intermediate authors in the author list [5].

The connected components of a graph G are the set of largest subgraphs of G that are each connected. A co-author network may consist of many individual connected components. A *bridge* is a node whose removal would cause the number of components to increase. In a co-authorship network, this is an author who has co-authored works with member of otherwise disparate and unconnected communities. These bridges can be formed as a result of the movement of researchers from one research group to another.

These concepts are often best understood in the context of ones own discipline [6]. Hence, this paper is concerned with the total

co-authorship network of the DAFx proceedings and pays particular attention to the largest connected component.

2.2. Bibliometrics of DAFx proceedings 1998–2009

A previous submission to the DAFx conference series examined the bibliometrics of the series, for the first twelve years [7]. The following is a brief summary of that work.

- Background to the DAFx conference series — list of locations and organisers
- There had been 722 submissions by 767 unique authors.
- Number of submissions per year, individual and cumulative
- Authorship distribution — number of authors per paper (the modal value was 2)
- A list of the most cited papers
- A list of the 20 most frequent authors
- Confirmation that the conference submissions followed Lotka's law, shown by a log-log plot of publications against number of authors.

The current submission attempts not to repeat any of these contributions, save for necessary updates to data after an additional seven years of submissions.

2.3. Bibliometrics & scientometrics of other conference series

Another paper by the author of the initial DAFx study focussed on the IEEE Transactions on Software Engineering [8]. This included further types of analysis not reported in the study of DAFx, such as collaboration between countries. Of course patterns of international collaboration can change with time. When considering the period of 1990-2000, the growth of the European Union and its funding for science was credited as a significant change in the scientific environment [9]. It was in this climate that the DAFx series began.

After its first nine years, a bibliometric analysis of the ISMIR (International Society for Music Information Retrieval) conference series was published [10]. This included a limited co-author network, in which only 22 authors were labelled. This paper suggested that the European research labs were tightly interconnected. The ISMIR community does contain some overlap with DAFx, in terms of authors. A recent follow-up examined the role of female authors in the ISMIR conference proceedings [11]. At this time, after 16 years, the total number of unique authors was 1,910. This paper included a brief co-author analysis. Nine clusters of authors are shown — clusters containing female authors with at least five co-authors. This showed that the two most prolific female authors were members of the same cluster. However, it is not clear whether these clusters can be connected to one another, i.e. whether they were drawn from a large connected component or numerous connected components. It can be inferred from [10] that some connections can be made (as a component cannot become smaller over time), yet insufficient data is presented.

It has been reported that the main component in the co-authorship network of the PACIS (Pacific Asia Conference on Information Systems) reached a size of 663 authors after a period of 15 years, which accounted for 33% of the total network [12]. This compared well to a value of 29% in a study of the ECIS (European Conference on Information Systems) after a period of twelve years [13]. Within ECIS, the second largest component contained only

37 authors, indicating how the main component grows by merging with smaller components, through the process of collaboration.

Herein, comparative data is presented for the DAFx conference series, after 19 years. The number of authors in the largest connected component was of particular interest, as was the detection of communities within that component.

3. DATA COLLECTION AND PROCESSING

Bibtex files from each of the first 12 conferences were already available [7]. Unfortunately, the file for DAFx-98 only listed one author per paper — this was manually corrected. JabRef² was used for Bibtex editing. Bibtex files for the seven subsequent conferences were created manually for this paper, from the conference proceedings listed on each conference website. Several editions of the DAFx conference proceedings (from 2005 to 2012) are indexed on Scopus and DAFx-03 is indexed on Web Of Science. These entries were checked against the manually created entries.

From the combined proceedings it was possible to construct a co-authorship network, showing which authors had directly collaborated in the writing of a paper. The following is a description of the process by which the network was created from the Bibtex data.

- All individual Bibtex files were merged and this file was imported into Network Workbench (NWB), a tool for network analysis and scientometrics [14]. The co-author network was extracted from the Bibtex file using the supplied routine.
- The names of authors will not always be consistent throughout all of their publications. This can be due to use of initials in place of full names, deliberate changes in name, reversal of first-name/family-name conventions, or simply human error in transcription. The merging of duplicate nodes is crucial for the accurate determination of graph metrics. Possible duplicates were highlighted by applying the Jaro distance metric [15, 16]. Through trial and error, authors were merged at 0.92 similarity providing the first two letters were in common, and it was noted when two entries were above 0.85 similarity. The data was then manually inspected. Where node labels only contained a first initial and not the full name, Google Scholar was used to identify whether this node matched any others. If 'both' authors had similar co-authors and wrote about similar topics, then the likelihood of them being duplicates was considered great enough to make a correction.
- The network was updated by merging nodes that were flagged as duplicates.
- The graph was split into separate, comma-separated files: one for nodes and one for edges.
- These node and edge tables were imported into Gephi [17]. Visual inspection of the graph was able to identify further duplicate nodes. An iterative approach was taken to the correction of duplicate nodes, by repeating these steps until all had been accounted for. The need for an iterative approach to data cleaning has also been described previously [12]. Of course, these methods were not able to detect any deliberate changes in name, such as by marriage.

²<http://www.jabref.org/>

Table 1: Summary statistics of entire co-author network

Number of nodes	1281
Number of edges	2058
Average degree	3.213
Average weighted degree	3.911
Connected components	269
...which are isolates	132
...which are dyads	58
...which are triads	37
The largest connected component consists of	667 nodes.

With the final network represented as a list of nodes and a list of the edges between them, this data was loaded in Gephi, for further processing to determine the connected components and perform community detection. Connected components were found using a depth-first search method [18]. Community detection was performed using the Louvain method [19]. With this additional data added as node attributes, the full set of nodes and edges was exported to .CSV files. Further processing and visualisation was performed using Matlab R2016a. In all calculations, the network was assumed to be undirected, i.e. the order of co-authors in a paper was not considered at this time.

4. RESULTS

After 19 years of conference proceedings, the number of unique authors has reached 1,281, from 1,175 submissions authored. Table 1 displays a summary of the total network. For all graph plots (Figures 1 and 2), node positioning was achieved using a force-directed layout [20]. As shown in Table 1, there are 269 connected components but this number includes 132 isolates (authors with degree = 0), 58 dyads (a pair of authors, each with degree = 1), 37 triads (three authors all connected to one another, with degree = 2) and other such small, highly-connected groups. Such a connected component can represent an individual paper, such as one paper submitted in 2012 which had 8 authors. Where a component represents a single paper, the component will be a complete graph, as each node connects to each other. The largest connected component (shown in Figure 1) contains 667 nodes, over half of the total nodes, making it roughly 44 times larger than the next largest connected component. This suggests that...

- a) new contributors to the conference proceedings are authors who are known to other authors in the network, such as their new students or colleagues.
- b) when smaller components begin to form it is not long before they merge with the main component. This forming of new collaborative bonds would be a natural consequence of authors meeting at conferences. The next largest connected components contain less than 15 nodes — there may be a critical mass a component reaches before it joins with another.

As shown in Table 1, there are 132 isolates in the network. An isolate is a node with a degree of 0, i.e. an author with no co-authors. In this conference, isolates make up roughly 10% of all authors. This number includes many of the keynote and tutorial submissions, which are usually credited to one author, frequently a local author from a related field not directly involved in DAFx. With a mean value of 62 papers per year, three keynote speakers

Table 2: Isolates (authors without co-authors) who have made more than one submission.

Name	N_{works}
Sinan Böksoy	3
Niels Bogaards	2
Christian Müller-Tomfelde	2
David Kim-Boyle	2
Richard Hoffmann-Burchardi	2
Angelo Farina	2
Tor Halmrast	2

and three tutorial presenters would ensure a figure of 10% isolates, were these speakers to be unique each year. Of course, the creation of research networks takes time, and so authors whose first DAFx submissions were single-authored and took place recently may remain isolates for a number of years. Additionally, some authors prefer to work without co-authors. Overall, the importance of these contributions should not be discounted. Only seven authors have made more than one contribution without having had a co-author. These are listed in Table 2.

4.1. Community detection

Within the main component, 24 communities were uncovered using the Louvain method [19], having between 4 and 88 members. Qualitative analysis of these communities reveals a clear geographic influence on collaborative patterns.

- Figure 2a displays the largest community, formed by the collaborations between some of the most frequently-contributing authors. Many USA-based authors are members of this community.
- Figure 2b appears to show pre-predominately researchers based in France and Canada. As shown in Fig. 1, this community could be broken down into smaller sub-communities (or ‘cliques’) in each continent..
- Figure 2c contains many individuals who were affiliated with Queen Mary University of London at the time of submission.
- Figure 2d describes a community of predominately French researchers and individuals with whom they have collaborated while based in France. In contrast to Fig. 2b, this community is focussed on IRCAM in Paris.

Each of these communities has hosted a DAFx conference, to which their large number of nodes can be at least partly-attributed (or *vice-versa*). Naturally, while the centre of each community may show a strong geographic influence, less-frequent collaborators in other regions are located further from the centre. Geographic proximity does facilitate academic collaboration (as described in section 2.1) but it is one of a number of factors.

Concerning the origins of the conference series as a means of disseminating knowledge within Europe, it can be seen in Figure 1 that a number of the 24 communities detected in the main component are of researchers beyond the continent. As mentioned above, the largest communities contain many North American-based authors. Additionally, the 17th largest community comprises of authors based in Taiwan, and is connected to the rest of the main

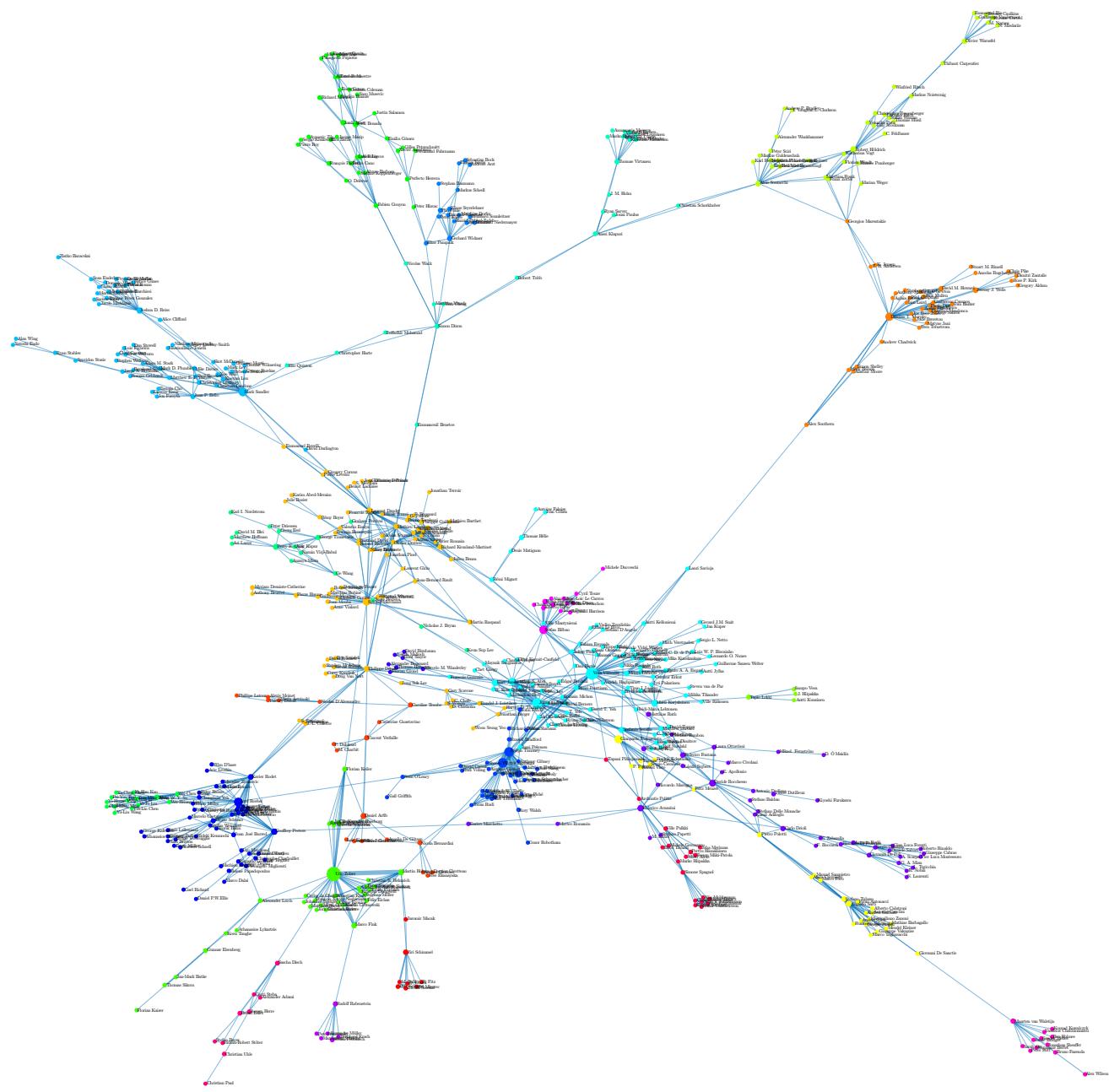


Figure 1: Largest connected component, consisting of 667 nodes. Node size is proportional to number of works created by that author. Edge thickness is proportional to the number of co-authored works between nodes. Colour represents the communities detected using the Louvain algorithm [19]. Node positioning was achieved using a force-directed layout [20].

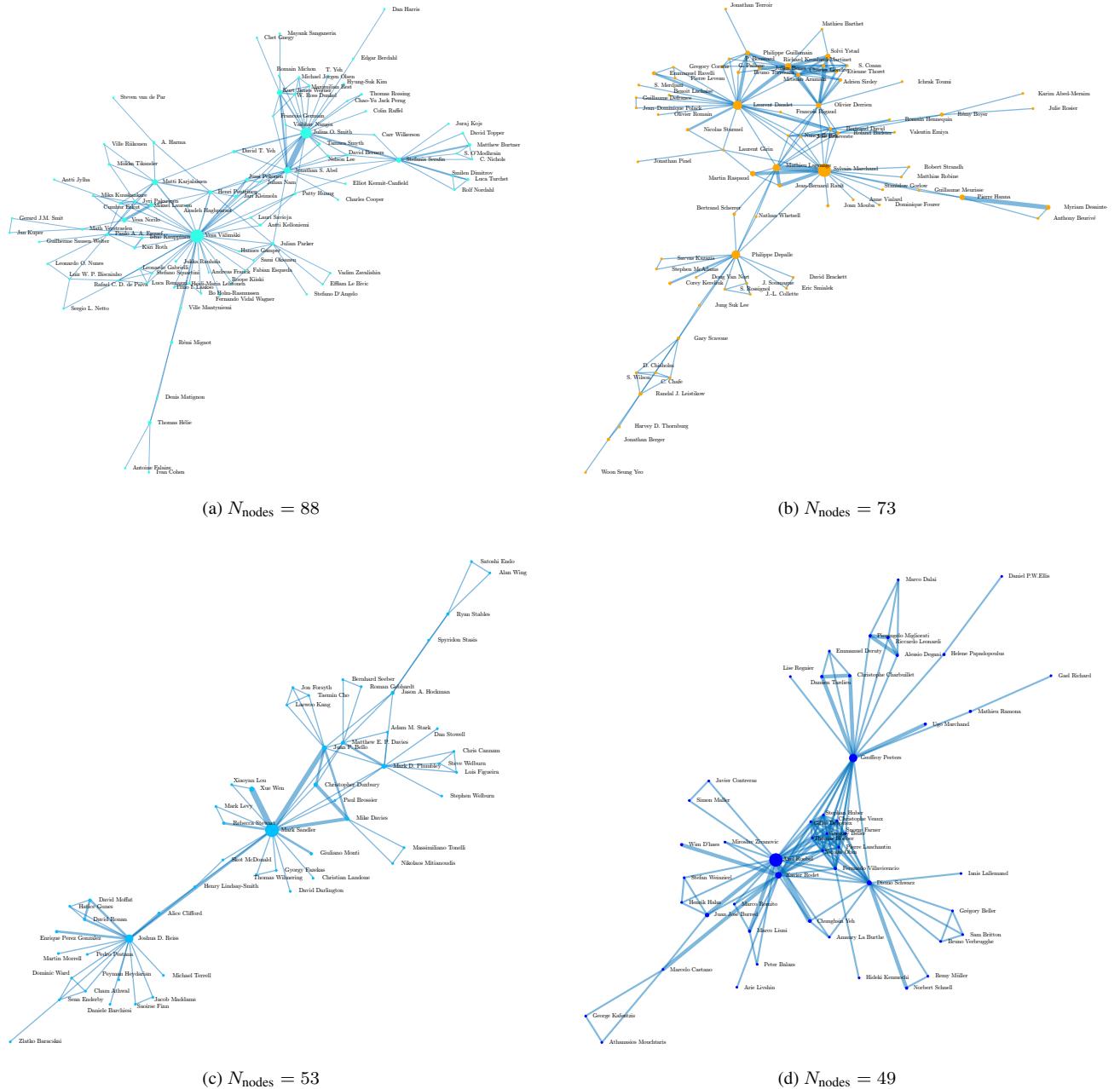


Figure 2: The four largest communities detected within the main component. Note that most nodes shown here have many more edges beyond these communities. Colours used are as in Figure 1. Node positioning was achieved using a force-directed layout [20], run separately for each community.

Table 3: Top 10 authors (within the main component) by weighted degree

Name	N_{works}	Degree	Weighted degree
Vesa Välimäki	36	40	62
Julius O. Smith	28	29	58
Joseph Timoney	24	22	56
Udo Zölzer	44	29	55
Victor Lazzarini	24	28	54
Jonathan S. Abel	15	22	42
Damian T. Murphy	18	27	36
Laurent Daudet	11	28	35
Mark Sandler	21	22	34
Axel Röbel	20	27	34

component via connections to the community in Fig. 2d. Of course, the main component shown in Figure 1 contains only 52% of the total number of authors. While the other connected components are relatively small, each containing less than 15 nodes, these include active research communities in Japan and China.

4.2. Node degree

In an undirected graph, the *degree* of a node refers to the number of other nodes to which it is connected. In a co-authorship network this is simply the number of co-authors. When ranking co-authors by degree this gives (perhaps) undue preference to authors who have managed to author many works but perhaps contributing little to each. If the edge weights are considered, then the *weighted degree* takes into account the number of times a pair of co-authors have worked together. Table 3 shows the ten authors with the greatest weighted degree.

4.3. Centrality

In attempting to measure the influence of nodes within a network a number of centrality measures have been developed [21]. This section will report on three of these: *closeness*, *betweenness* and *eigenvector centrality*. These three scores were calculated using Matlab R2016a.

4.3.1. Closeness

Closeness centrality uses the inverse sum of the distance from a node to all other nodes in the graph. Assuming that not all nodes can be reached (which is true for this network), the centrality of node i is:

$$\text{closeness}(i) = \left(\frac{A_i}{N - 1} \right)^2 \frac{1}{D_i} \quad (1)$$

Here, A_i is the number of nodes that can be reached from node i (not counting i itself), N is the number of nodes in the graph G , and D_i is the sum of distances from node i to all reachable nodes. If no nodes are reachable from node i , then $\text{closeness}(i)$ is zero. This expression assumes that all edge weights are equal to 1. The reciprocal of the actual edge weights (the number of co-authored works) were introduced as the ‘cost’ used in the centrality calculations. This is suitable because one can deduce that co-authors with many co-authored works exchange information more readily

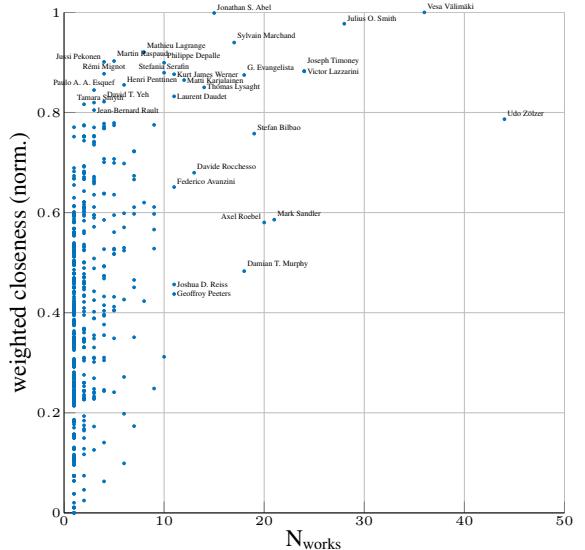


Figure 3: Scatterplot showing number of submissions vs. weighted closeness, which is normalised to the range [0 1]. The greater the value of closeness, the fewer steps (on average) are required to reach another author in the network.

than those with fewer co-authored works, and so the ‘cost’ associated with traversing this edge is lower. The authors with the highest weighted closeness centrality values are displayed in Fig. 3. Closeness generally increases with the number of submissions but there are notable exceptions, as highlighted.

4.3.2. Betweenness

The betweenness measure illustrates the importance of an author by means of assessing the flow of “traffic” that passes through that node. This is achieved by measuring the number of shortest paths from all nodes to all other nodes which involve passing through the node in question.

$$\text{betweenness}(i) = \sum_{s,t \neq i} \frac{n_{st}(i)}{N_{st}} \quad (2)$$

$n_{st}(i)$ is the number of shortest paths from source s to target t that pass through node i , and N_{st} is the total number of shortest paths from s to t . As with closeness, the reciprocal of the edge weights was used as a cost in calculating weighted betweenness. The values of weighted betweenness centrality are displayed in Fig. 4.

4.3.3. Eigenvector centrality

The eigenvector centrality measure assumes that when a node is connected to other high-scoring nodes that this counts for more than a connection to a lower-scoring node.

When eigenvector centrality was computed without weighting, the top 10 authors were not just all members of the same community (shown in Fig. 2d) but all co-authors of the same publication. For many of these authors, this has been their sole contribution to DAFX. When eigenvector centrality is calculated with edge weights taken into account, the results are plotted in Fig. 5.

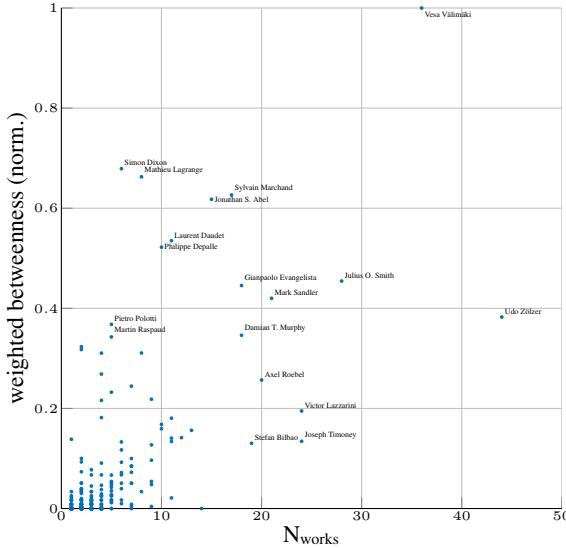


Figure 4: Scatterplot showing number of submissions vs. weighted betweenness, which is normalised to the range [0 1]. High betweenness scores indicate an author who co-authors submissions with a large number of communities.

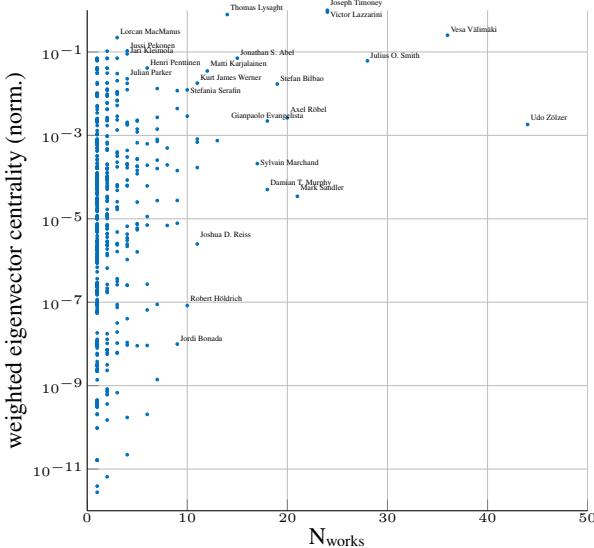


Figure 5: Scatterplot showing number of submissions vs. weighted eigenvector centrality, normalised to the scale [0 1]. Note the use of logarithmic scaling.

The three authors with the highest score according to this metric are all from Maynooth University, and have frequently collaborated. There are many infrequent collaborators with relatively high scores, while some frequently contributing authors in more isolated communities (according to this metric) are also highlighted.

5. DISCUSSION

The presentation in this paper of the DAFx co-authorship network allows for a number of possibilities: an author may use the network to assess their own contributions to the conference series and the contributions of their research group(s). In doing so, new collaborative opportunities can be located.

There is still an open question as to whether the density of co-authorship networks is increased by the formation of long-term collaborative bonds or if it is occasional and short-term collaborative efforts that causes the network to expand. At this stage, a series of questions is posed to the community, for discussion at conferences and beyond.

1. How could the network be used to identify potential collaborators?
2. Could more be done to integrate isolated communities?
3. There are very few frequently-contributing authors who lack co-authors. Could more be done to support authors who do not collaborate?
4. Would the size of the network be increased by the use of a double-blind review process, in which submissions are evaluated without knowing the names of the authors?

As shown by Table 3 and Figs. 3, 4 and 5, the relationship between these measures of node importance and more straightforward measures, such as the number of works, is not clear. It is possible for an author with few works to be considered highly important to the network as a whole. Each centrality measure has strengths and weaknesses and provides different insights into the topology of the network. Closeness centrality rewards authors who are prominent in communities which are themselves well connected to others, often achieved by high degree scores — the top three authors are members of the community in Fig. 2a. Betweenness centrality highlighted the efforts of a number of authors who have worked with a number of communities and authors who act as bridges — six communities are represented in the top 10 authors. In contrast, eigenvector centrality rewards groups of authors who frequently collaborate with one another.

Figures 1 and 2d show that large communities are not always so well-connected to the rest of the network. Does this indicate that these communities are large enough to be self-sustaining and in less need of outward collaboration?

One limitation in this study is that author order was not taken into account. Doing so would result in a directed network, allowing for a more sophisticated analysis particularly regarding centrality measures. However, establishing the relative contributions of each author in a paper is not a trivial task. As described in section 2.1, there are a variety of strategies which can be employed, but this task remains a focus of on-going study.

6. CONCLUSIONS

The aim of the study was to examine the nature of community and collaboration within the DAFx conference proceedings, after two decades. By collecting Bibtex archives for each conference a co-authorship network was created. This revealed a large connected component — 52% of all authors are connected to one another by a number of intermediate co-authors. Twenty-four communities were detected within this component, heavily influenced by geographical proximity. Communities are connected by the movement

of researchers between research groups and by the interactions and discussions at conferences. This network could be displayed online, allowing authors interactive access to the data. This would also facilitate the updating of the data with each new set of conference proceedings.

There have now been two papers explicitly describing the DAFx conference proceedings: one on basic bibliometrics and one describing the details of the network. There is scope for further work, in the context of DAFx and also more generally. The Bib-text entries for the DAFx conference could contain information on submission type: keynote, tutorial, oral presentation or poster presentation. It would be interesting to examine whether the choice of poster or oral presentation has an impact on the formation of collaborative links. Of course, DAFx does not exist in isolation, and its contributors also make submissions to other conference series. Larger co-authorship networks can be constructed by the merger of related conference proceedings. This could reveal the extent of the overlap and the interdisciplinary nature of the research groups involved.

7. REFERENCES

- [1] Wolfgang Glänzel and András Schubert, “Analysing scientific networks through co-authorship,” in *Handbook of quantitative science and technology research*, pp. 257–276. Springer, 2004.
- [2] Wolfgang Glänzel and Bart Thijs, “Does co-authorship inflate the share of self-citations?,” *Scientometrics*, vol. 61, no. 3, pp. 395–404, 2004.
- [3] Katy Börner, Shashikant Penumarthy, Mark Meiss, and Weimao Ke, “Mapping the diffusion of scholarly knowledge among major US research institutions,” *Scientometrics*, vol. 68, no. 3, pp. 415–426, 2006.
- [4] Howard D White, Barry Wellman, and Nancy Nazer, “Does citation reflect social structure?: Longitudinal evidence from the globenet interdisciplinary research group,” *Journal of the American Society for Information Science and Technology*, vol. 55, no. 2, pp. 111–126, 2004.
- [5] Jinseok Kim and Jana Diesner, “A network-based approach to coauthorship credit allocation,” *Scientometrics*, vol. 101, no. 1, pp. 587–602, 2014.
- [6] Howard D White and Katherine W McCain, “Visualizing a discipline: An author co-citation analysis of information science, 1972–1995,” *Journal of the American society for information science*, vol. 49, no. 4, pp. 327–355, 1998.
- [7] Brahim Hamadicharef, “Bibliometric study of the DAFx proceedings 1998–2009,” in *Proceedings of International Conference on Digital Audio Effects*, 2010, pp. 427–430.
- [8] Brahim Hamadicharef, “Scientometric study of the IEEE transactions on software engineering 1980–2010,” in *Proceedings of the 2nd International Congress on Computer Applications and Computational Science*. Springer, 2012, pp. 101–106.
- [9] Caroline S Wagner and Loet Leydesdorff, “Mapping the network of global science: comparing international co-authorships from 1990 to 2000,” *International journal of Technology and Globalisation*, vol. 1, no. 2, pp. 185–208, 2005.
- [10] Jin Ha Lee, M Cameron Jones, and J Stephen Downie, “An analysis of ISMIR proceedings: Patterns of authorship, topic, and citation.,” in *Proceedings of the 10th International Conference on Music Information Retrieval*, 2009, pp. 57–62.
- [11] Xiao Hu, Kahyun Choi, Jin Ha Lee, Audrey Laplante, Yun Hao, Sally Jo Cunningham, and J Stephen Downie, “WiMIR: An informetric study on women authors in ISMIR,” in *Proceedings of the 17th International Conference on Music Information Retrieval (ISMIR)*, New York, USA, 2016.
- [12] France Cheong and Brian J Corbitt, “A social network analysis of the co-authorship network of the pacific asia conference on information systems from 1993 to 2008,” *PACIS Proceedings*, 2009.
- [13] Richard Vidgen, Stephan Henneberg, and Peter Naudé, “What sort of community is the european conference on information systems? a social network analysis 1993–2005,” *European Journal of Information Systems*, vol. 16, no. 1, pp. 5–19, 2007.
- [14] NWB Team et al., “Network workbench tool. indiana university, northeastern university, and university of michigan,” 2006.
- [15] Matthew A Jaro, “Advances in record-linkage methodology as applied to matching the 1985 census of Tampa, Florida,” *Journal of the American Statistical Association*, vol. 84, no. 406, pp. 414–420, 1989.
- [16] Matthew A Jaro, “Probabilistic linkage of large public health data files,” *Statistics in medicine*, vol. 14, no. 5-7, pp. 491–498, 1995.
- [17] Mathieu Bastian, Sébastien Heymann, Mathieu Jacomy, et al., “Gephi: an open source software for exploring and manipulating networks.,” *ICWSM*, vol. 8, pp. 361–362, 2009.
- [18] Robert Tarjan, “Depth-first search and linear graph algorithms,” *SIAM journal on computing*, vol. 1, no. 2, pp. 146–160, 1972.
- [19] Vincent D Blondel, Jean-Loup Guillaume, Renaud Lambiotte, and Etienne Lefebvre, “Fast unfolding of communities in large networks,” *Journal of statistical mechanics: theory and experiment*, , no. 10, 2008.
- [20] Thomas MJ Fruchterman and Edward M Reingold, “Graph drawing by force-directed placement,” *Software: Practice and experience*, vol. 21, no. 11, pp. 1129–1164, 1991.
- [21] Tore Opsahl, Filip Agneessens, and John Skvoretz, “Node centrality in weighted networks: Generalizing degree and shortest paths,” *Social networks*, vol. 32, no. 3, pp. 245–251, 2010.

Author Index

- Abel, Jonathan S. 375, 381
Adami, Alexander 496
Aires-de-Sousa, João 360
Alary, Benoit 405
Arend, Johannes M. 345
Auclair, Théo 353
Avanzini, Federico 397
Avital, Eldad J. 307

Berners, David 2
Bilbao, Stefan 72, 367
Binev, Yuri 360
Bogason, Ólafur 130
Bonifácio, Vasco D. B. 360
Bouvier, Damien 3
Brand, Lukas 496
Brandtsegg, Øyvind 110, 239
Briani, Matteo 222
Bridges, Jamie 299
Buch, Michael 215

Canadas-Quesada, Francisco 276
Canfield-Dafilou, Elliot K. 247, 375
Caracalla, Hugo 234
Carvalho, Hugo T. 228
Carvalho, Luís F. V. 228
Cavaco, Sofia 360
Chafe, Chris 118
Chatzioannou, Vasileios 72
Cherny, Eugene 459
Chomette, Baptiste 64
Cuyt, Annie 222

Das, Orchisama 118
De Man, Brecht 436
Delikaris-Manias, Symeon 412
Depalle, Philippe 208, 268
Disch, Sascha 496
Dixon, Simon 420
Doclo, Simon 283
Dubach, Christophe 367
Ducceschi, Michele 80

Eichas, Felix 184
Enderby, Sean 103, 474
Esqueda, Fabián 192
Esterer, Nicholas 208
Evangelista, Gianpaolo 260

Fazekas, György 160
Fitzgerald, Derry 276
Gabrielli, Leonardo 11
Germain, François G. 168, 200
Geronazzo, Michele 397

Ghorbal, Slim 353
Gillespie, Daniel J. 489
Gray, Alan 367

Habets, Emanuël A. P. 337
Hamilton, Brian 2
Harte, Christopher 420
Hélie, Thomas 3, 48, 64, 87, 443
Herman, Woody 489
Herre, Jürgen 496
Hohnerlein, Christoph 125
Holmes, Ben 152
Holters, Martin 138, 152

Jillings, Nicholas 103
Jossic, Marguerite 64
Jot, Jean-Marc 2

Kartofelev, Dmitri 40
Katz, Brian F.G. 323
Kearney, Gavin 389
Keenan, Fiona 25
Kraft, Sebastian 255

Lebrun, Tristan 48
Lee, Wen-shin 222
Li, Pei-Ching 466
Lilius, Johan 459
Lin, Yi-Ju 466
Liski, Juho 95
Lobo, Ana M. 360

Maestre, Esteban 381
Mamou-Mani, Adrien 64
McCormack, Leo 412
McKenzie, Thomas 389
Mehes, Sandor 291
Moffat, David 428
Mohamad, Zulfadhli 420
Möller, Stephan 184
Morishima, Shigeo 32
Mouromtsev, Dmitry 459
Muller, Rémy 87
Murphy, Damian 329, 389, 481

Nakatsuka, Takayuki 32
Neri, Julian 268
Noisternig, Markus 323

Olsen, Michael J. 176, 200

Parker, Julian D. 2, 125, 145, 176, 192
Pauletto, Sandra 25
Peixoto, Daniela 360
Pereira, Florbela 360
Peruch, Enrico 397
Pfeifle, Florian 17

- Picasso, Charles 443
Poirier-Quinot, David 323
Politis, Archontis 405
Pöntynen, Henri 192
Pörschmann, Christoph 345
Prandoni, Fabio 397
Puckette, Miller 1
Pulkki, Ville 412
Quinton, Elio 215
Rabenstein, Rudolf 315
Rees-Jones, Joe 329
Reiss, Joshua D. 56, 110, 307, 428, 436
Rest, Maximilian 125, 145, 176
Rodrigues, Ian 360
Roebel, Axel 234
Ronan, David 428
Roze, David 3, 64
Rudrich, Daniel 451
Ruiz-Reyes, Nicolas 276

Sarkar, Saurja 110
Saue, Sigurd 239
Scavone, Gary P. 381
Schäfer, Maximilian 315
Schlecht, Sebastian J. 337
Schmutzhard, Sebastian 72
Séguier, Renaud 353
Selfridge, Rod 307
Sheng, Di 160
Shih, Chi-Ching 466
Siedenburg, Kai 283

Smith III, Julius O. 1, 118, 381
Smith, Stephen 481
Soladié, Catherine 353
Sontacchi, Alois 451
Squartini, Stefano 11
Stables, Ryan 103, 474
Stade, Philipp 345
Stasis, Spyridon 103
Stevens, Francis 481
Stoltzfus, Larisa 367
Stowell, Dan 56
Sturm, Bob L. 215
Su, Alvin W. Y. 466
Su, Li 466

Teixeira, Ricardo 360
Tomassetti, Stefano 11

Välimäki, Vesa 95, 192, 405
van Walstijn, Maarten 152, 291, 299
Vera-Candeas, Pedro 276

Wang, Avery 1
Wang, Yu-Lin 466
Wedelich, Russell 489
Werner, Kurt J. 130, 145, 176, 200, 247
Wilkinson, William J. 56
Wilson, Alex 502

Yang, Yi-Hsuan 466

Zinato, Carlo 11
Zölzer, Udo 138, 184, 255



is proudly sponsored by



Ableton

audiokinetic

K R O T O S

