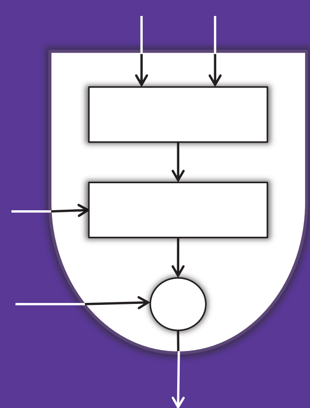
Department of Signal Processing and Acoustics

Nonlinear Abstract Sound Synthesis Algorithms

Jari Kleimola



Aalto University

DOCTORAL DISSERTATIONS

Aalto University publication series
DOCTORAL DISSERTATIONS 25/2013

Nonlinear Abstract Sound Synthesis Algorithms

Jari Kleimola

A doctoral dissertation completed for the degree of Doctor of Science in Technology to be defended, with the permission of the Aalto University School of Electrical Engineering, at a public examination held at the lecture hall S1 of the school on 22 February 2013 at 12 noon.

Aalto University
School of Electrical Engineering
Department of Signal Processing and Acoustics

Supervising professor
Prof. Vesa Välimäki

Thesis advisor
Prof. Vesa Välimäki

Preliminary examiners
Dr. Tamara Smyth, Simon Fraser University Surrey, Canada
Prof. Roger B. Dannenberg, Carnegie Mellon University, USA

Opponent
Prof. Philippe Depalle, McGill University, Canada

Aalto University publication series
DOCTORAL DISSERTATIONS 25/2013

ISBN 978-952-60-5015-7 (printed)
ISBN 978-952-60-5016-4 (pdf)
ISSN-L 1799-4934
ISSN 1799-4934 (printed)
ISSN 1799-4942 (pdf)
http://urn.fi/URN:ISBN:978-952-60-5016-4

Unigrafia Oy
Helsinki 2013

Finland

NORDIC ECOLABEL
441 697
Printed matter

Aalto University

Aalto University, P.O. Box 11000, FI-00076 Aalto www.aalto.fi

Abstract

Author
Jari Kleimola

Name of the doctoral dissertation
Nonlinear Abstract Sound Synthesis Algorithms

Publisher School of Electrical Engineering

Unit Department of Signal Processing and Acoustics

Series Aalto University publication series DOCTORAL DISSERTATIONS 25/2013

Field of research Acoustics and audio signal processing

Manuscript submitted 17 September 2012 **Date of the defence** 22 February 2013

Permission to publish granted (date) 10 December 2012 **Language** English

☐ **Monograph** ☒ **Article dissertation (summary + original articles)**

Abstract

This thesis explores abstract sound synthesis methods in digital musical instrument applications and proposes new algorithms for sound production, aliasing reduction, and efficient control of synthesis parameters. The specific focus is on nonlinear distortion and audio-rate modulation techniques, which are approached from two complementary perspectives.

First, the classic view, built on closed-form mathematical expressions and computer algorithms, was seen to converge into a compound model where different abstract synthesis methods both generalize and reinforce each other. In this view, the recent phaseshaping technique was investigated as pipelined phaseshaper expressions, with applications in efficient and novel oscillator effects algorithms discovered in the thesis, such as an efficient super sawtooth simulation. In addition, a two-dimensional multi-point extension of the phase distortion method called vector phaseshaping synthesis (VPS) was proposed and demonstrated as an intuitive parametrization of the complex phase modulation technique. The method is well suited for contemporary multi-touch interaction and planar control- and audio-rate modulation.

The second perspective into the nonlinear distortion and audio-rate modulation techniques, based on periodically linear time-varying filters, led to the discovery of a synthesis algorithm where the coefficients of an allpass filter chain are modulated at an audio rate. In addition, the filter approach enabled an alternative interpretation of the feedback amplitude modulation (FBAM) technique, whose first-order form was extended and a second-order form was introduced.

To complement the sound production stage of digital musical instrument applications, two aliasing reduction methods were introduced, one based on scaled sinusoids and another on polynomial transition regions (PTR). The latter is currently the most efficient method for implementing alias-suppressed virtual analog oscillators. Finally, a streamlined control protocol that dramatically reduces the bandwidth of control data streams was proposed.

The efficient and novel algorithms introduced in the thesis are useful for sound synthesis in resource constrained mobile platforms, web browsers, and in applications requiring a high polyphony.

Keywords Acoustics, digital signal processing, music, nonlinear filters, sound synthesis

ISBN (printed) 978-952-60-5015-7 **ISBN (pdf)** 978-952-60-5016-4

ISSN-L 1799-4934 **ISSN (printed)** 1799-4934 **ISSN (pdf)** 1799-4942

Location of publisher Espoo **Location of printing** Helsinki **Year** 2013

Pages 135 **urn** http://urn.fi/URN:ISBN:978-952-60-5016-4

A? Aalto-yliopisto

Aalto-yliopisto, PL 11000, 00076 Aalto www.aalto.fi

Tiivistelmä

Tekijä
Jari Kleimola

Väitöskirjan nimi
Epälineaariset abstraktit äänisynteesialgoritmit

Julkaisija Sähkötekniikan korkeakoulu

Yksikkö Signaalinkäsittelyn ja akustiikan laitos

Sarja Aalto University publication series DOCTORAL DISSERTATIONS 25/2013

Tutkimusala Akustiikka ja äänenkäsittelytekniikka

Käsikirjoituksen pvm 17.09.2012 **Väitöspäivä** 22.02.2013

Julkaisuluvan myöntämispäivä 10.12.2012 **Kieli** Englanti

☐ **Monografia** ☒ **Yhdistelmäväitöskirja (yhteenveto-osa + erillisartikkelit)**

Tiivistelmä

Tämä väitöskirja käsittelee digitaalisten soittimien abstrakteja äänisynteesimenetelmiä ja esittelee uusia äänen tuottamiseen, laskostumisen vähentämiseen ja synteesiparametrien ohjaukseen kehitettyjä algoritmeja. Tutkimuksen painopiste on epälineaarisissa äänitaajuisissa säröytys- ja modulaatiotekniikoissa, joita lähestytään kahdesta näkökulmasta.

Matemaattisiin yhtälöihin ja tietokonealgoritmeihin perustuvassa klassisessa lähestymistavassa abstraktit synteesimenetelmät konvergoituivat yhtenäismalliksi, jossa eri menetelmät sekä yleistävät että täydentävät toisiaan. Hiljakkoin esiteltyä vaihemuotoilutekniikkaa tutkittiin sarjamuotoisina vaihemuotoilijaketjujuina, joiden sovelluksina syntyi uusia ja tehokkaita oskillaattoriefektialgoritmeja. Lisäksi tutkimuksessa kehitettyä vaihesäröytysmenetelmän monipisteistä laajennosta (VPS) demonstroitiin vaihemodulaation intuitiivisena parametrointitekniikkana. Kehitetty metodi soveltuu erityisesti synteesiparametrien kaksiulotteiseen muokkaukseen esimerkiksi monikosketuspohjaisen ihmisen ja koneen välisen vuorovaikutusmekanismin välityksellä.

Toinen lähestymistapa perustui jaksollisesti lineaaristen aikamuuttuvien suotimien teoriaan. Tämä johti uuden synteesialgoritmin löytämiseen, jossa kokopäästösuodinketjun kertoimia moduloidaan äänitaajuudella. Suodinlähestymistapa mahdollisti lisäksi takaisinkytketyn amplitudimodulaatiotekniikan (FBAM) vaihtoehtoisen tulkintamuodon, jonka avulla takaisinkytketyn amplitudimodulaatiotekniikan ensimmäisen ja toisen asteen muotoja laajennettiin ja esiteltiin.

Tutkimuksessa kehitettiin myös kaksi laskostumisen vähentämismenetelmää. Menetelmät pohjautuvat skaalattuihin siniaaltomuotoihin sekä polynomisiin transitioalueisiin (PTR), joka on tehokkain nykyisin tunnettu virtuaalianalogiaoskillaattoreiden toteutusmenetelmä. Tutkimuksessa lisäksi kehitetty tehostettu ohjausprotokolla vähentää ohjausvirtojen kaistanleveyttä merkittävästi.

Väitöskirjassa esitetyt uudet ja tehokkaat algoritmit soveltuvat erityisesti polyfonisesti rikkaiden ja resurssirajoitteisten mobiili- ja selainpohjaisten äänisynteesisovellusten implementointiin.

Avainsanat Akustiikka, digitaalinen signaalinkäsittely, epälineaariset suodattimet, musiikki, äänisynteesi

ISBN (painettu) 978-952-60-5015-7 **ISBN (pdf)** 978-952-60-5016-4

ISSN-L 1799-4934 **ISSN (painettu)** 1799-4934 **ISSN (pdf)** 1799-4942

Julkaisupaikka Espoo **Painopaikka** Helsinki **Vuosi** 2013

Sivumäärä 135 **urn** http://urn.fi/URN:ISBN:978-952-60-5016-4

Preface

When I bought my first synthesizer and started modestly tweaking the parameters of the preset patches, little did I know that I would be later writing this preface. However, as preset tweaking gradually morphed into a quest for musically expressive timbres that had never been in the air before, I began to wonder what was actually going on within the parameter space. That curiosity eventually transformed into a dream of researching the synthesis methods in more detail.

The dream became reality at the Department of Signal Processing and Acoustics at Aalto University School of Electrical Engineering, and this thesis is the result of the work and play carried out at the department during the years 2008 - 2012. That time has been truly inspiring, and I am greatly indebted to many sources of support for making it possible.

First of all, I want to express my deepest gratitude to my supervisor and instructor Prof. Vesa Välimäki. His wisdom, creativity, enthusiasm, and passionate approach to research guided my journey throughout the work. His influence is present all over this thesis. I'm extremely thankful of learning the concepts, methods, and tools that I could apply in this work, and moreover that I'm able to carry them into my future academic works. I would also like to thank Prof. Petri Vuorimaa for guiding my initial steps as a researcher, and for letting me continue the collaboration with his team.

I am immensely grateful to co-authors Dr. Victor Lazzarini, Dr. Joseph Timoney, Dr. Jonathan Abel, Dr. Henri Penttinen, Jussi Pekonen, and Patrick McGlynn. This research would not exist without their contribution. The contribution of Dr. Lazzarini and Dr. Timoney is especially acknowledged, since many of the concepts and methods explored in this thesis were discovered in close collaboration with them. I am a great admirer of your research. Thanks also for hosting me during my research visit, I really miss the springtime in Maynooth.

I wish to sincerely thank the pre-examiners of this thesis Dr. Tamara Smyth and Prof. Roger B. Dannenberg for their encouraging comments and criticism. I have learned so many things from your publications that I am truly honored you accepted me to become part of the research community. I am also greatly indebted to Luis Costa whose proofreading improved the quality of all publications and the introduction included in this thesis.

The work was financially supported by the European Union 7th Framework Programme (SAME project, ref. 215749), the Academy of Finland (project no. 122815), ERASMUS Exchange Programme, and the Department of Signal Processing and Acoustics. This support is gratefully acknowledged.

I would like to thank the members of the research projects and colleagues at the acoustics lab for their support. A special thanks to my room mate Henkka for hours of on and off topic discussions and for providing the Mopho playground. I thank my present and past band mates for reminding me that the synthesizer is really a musical instrument – not just an interesting research platform – and composer Otto Romanowski for opening my ears in the first place.

Most importantly, I want to thank my parents Arja and Veikko Kleimola, my brother Jyrki, my grandmother Aino, and Anja for their unconditional love and support in every aspect of life. I would also like to thank Oskar, Kristian, Aurora, Patrik, Nina, Rasmus, Kiira, Heidi and Juha for balancing my abstract endeavours with real life. Finally, I want to express my deepest appreciation and love to Tarja. With you I am whole.

Espoo, January 27, 2013,

Jari Kleimola

Contents

List of publications

This thesis consists of an overview and of the following publications.

P-1 Kleimola, J., Lazzarini, V., Timoney, J. and Välimäki, V., 2010. Phaseshaping oscillator algorithms for musical sound synthesis. In: *Proc. Sound and Music Computing Conf.*, Barcelona, Spain, July 2010, pp. 94–101.

P-2 Kleimola, J., Lazzarini, V., Timoney, J. and Välimäki, V., 2011. Vector phaseshaping synthesis. In: *Proc. Int. Conf. Digital Audio Effects*, Paris, France, Sept. 2011, pp. 233–240.

P-3 Kleimola, J., Pekonen, J., Penttinen, H., Välimäki, V. and Abel, J., 2009. Sound synthesis using an allpass filter chain with audio-rate coefficient modulation. In: *Proc. Int. Conf. Digital Audio Effects*, Como, Italy, Sept. 2009, pp. 305–312.

P-4 Kleimola, J., Lazzarini, V., Välimäki, V. and Timoney, J., 2011. Feedback amplitude modulation synthesis. *EURASIP J. Adv. Signal Process.*, March 2011, pp. 1–18.

P-5 Lazzarini, V., Kleimola, J., Timoney, J. and Välimäki, V., 2011. Aspects of second-order feedback AM synthesis. In: *Proc. Int. Computer Music Conf.*, Huddersfield, UK, July – Aug. 2011, pp. 92–95.

P-6 Kleimola, J. and Välimäki, V., 2012. Reducing aliasing from synthetic audio signals using polynomial transition regions. *IEEE Signal Process. Lett.*, 19(2), pp. 67–70.

P-7 Kleimola, J. and McGlynn, P., 2011. Improving the efficiency of Open Sound Control with compressed address strings. In: *Proc. Sound and Music Computing Conf.*, Padova, Italy, July 2011, pp. 479–485.

Author's contribution

P-1: Phaseshaping oscillator algorithms for musical sound synthesis

The author produced and presented the entire work, excluding Section 1.

P-2: Vector phaseshaping synthesis

The author discovered the synthesis technique and its modulation extensions together with the second author of the paper. He derived the synthesis formulas in Eqs. 5–6, analyzed their parameters, and developed the aliasing suppression method. He also co-produced the software, the sound examples, and the webpage. The author wrote Section 3.2, co-authored Sections 3 and 3.1, produced Figs. 1–5, and edited the paper.

P-3: Sound synthesis using an allpass filter chain with audio-rate coefficient modulation

The author discovered the synthesis technique. He designed and implemented the synthesis model and its transient-free formula, interpreted and contextualized the method in terms of abstract synthesis techniques, and analyzed the effects of synthesis parameters. He also produced software implementations, applications, sound examples and analysis material for Section 2.3, and the webpage. The author wrote and presented the paper, excluding Sections 2.3, 4.3, and Eqs. 4 and 15–17.

P-4: Feedback amplitude modulation synthesis

The author co-discovered the synthesis technique with the second author of the article. He designed and implemented the parametric synthesis model, performed stability and scaling analyses, contextualized and evaluated the method, introduced variation #6, and produced the example applications. The software implementation, sound examples, and the website were co-produced by him. The author wrote Sections 2.6 and 4–6, excluding 2.6.2, 4.4 and 6.4. He produced Figs. 6, 7, 18–26 and reproduced the others, co-authored Sections 2.4 and 2.5, and edited the article.

P-5: Aspects of second-order feedback AM synthesis

The author designed and implemented the synthesis formulas for the second-order variants of P-4 and the coefficient-modulated Chamberlin state-variable filter, and analyzed their spectral properties. He wrote Sections 3.1 and 4, and produced Figs. 1, 3, 4, and 7.

P-6: Reducing aliasing from synthetic audio signals using polynomial transition regions

The author analyzed the DPW algorithm and discovered, derived, and evaluated its generalized PTR interpretation. The software implementation, applications, sound examples, and the website were produced by him. The author wrote the article, excluding Section 1.

P-7: Improving the efficiency of Open Sound Control with compressed address strings

The author produced and presented the entire work, excluding Section 4.

List of abbreviations

2-D	two-dimensional
AdFM	adaptive frequency modulation
AM	amplitude modulation
AMI	acoustic musical instrument
CM	coefficient modulation
DMI	digital musical instrument
DSP	digital signal processing
EG	envelope generator
FBAM	feedback amplitude modulation
FBFM	feedback frequency modulation
FM	frequency modulation
HCI	human computer interaction
IIR	infinite impulse response
IR	impulse response
LFO	low-frequency oscillator
LTI	linear time-invariant
ModFM	modified frequency modulation
PC	personal computer
PD	phase distortion
PG	phase generator
PLTV	periodically linear time-variant
PM	phase modulation
PWM	pulse width modulation
RM	ring modulation
SDF	spectral delay filter
SpSB	split-sideband
SSB	single sideband
UG	unit generator
VPS	vector phaseshaping synthesis
WG	wave generator
XOR	exclusive-or

List of symbols

a	distortion index
A	peak amplitude
$A(n)$	amplitude scaling signal
a_1	scaling factor
A_c	carrier amplitude
A_m	modulator amplitude
d	inflection point (abscissa)
$f(n)$	frequency signal
$F(\cdot)$	waveform function
f_0	fundamental frequency
f_c	carrier frequency
f_m	modulator frequency
f_s	sampling frequency
f_x	input signal frequency
$h(m,n)$	impulse response
I	modulation index (FM and PM)
$J_n(\cdot)$	Bessel function of the first kind
k	index
m	sample number (moment of signal activation)
M	modulation index (CM)
$m(n)$	modulation signal
n	sample number
N	integer denoting a count
P_0	period
$P(\cdot)$	phaseshaper
x	variable
$x(n)$	input signal
$y(n)$	output signal
v	inflection point (ordinate)
$w(n)$	wave generator output signal
$W(\cdot)$	waveshaper
$w_{AM}(n)$	amplitude modulated signal
$w_{WS}(n)$	waveshaper output signal
β	modulation index (FBAM)
θ	angle
$\phi(n)$	normalized unipolar phase signal
ϕ_0	initial phase
ϕ_Δ	phase increment
$\phi_0(n)$	phase offset signal
$\phi_c(n)$	carrier phase signal
$\phi_m(n)$	modulator phase signal
$\phi_{PS}(n)$	shaped phase signal

1. Introduction

The first affordable digital musical instruments appeared on the market thirty years ago. Since then, the advances in computing have been gradually shifting the center of attraction from custom hardware instruments towards software synthesizers that run in commodity personal computers (PCs). The recent trend towards mobile computing has decreased the form factor of the PCs to the point where the audio synthesis engines of the pioneering hardware musical instruments fit now into the user's pocket (see Fig. 1).

Fig. 1. Mobile audio synthesis applications implemented by the author of this thesis. The photograph shows Helsinki Mobile Phone Orchestra performing a bell choir arrangement of a Christmas song, employing accelerometer sensors and early FM synthesizer emulation.

The reduction in size comes at the price of computing power, however, since the mobile smartphones and tablet computers require components that are energy conservative. The efficient algorithms of the early digital musical instruments (DMIs) are therefore still attractive in contemporary mobile music making. They are also relevant in software synthesizer plugin implementations and browser-based web applications.

The conceptual model of the classic DMI is shown in Fig. 2 [Wanderley and Depalle, 2004]. The performer interacts with an external musical keyboard or other input device (Fig. 2(a)) that transduces the performance of the user into digital control signals. These signals are mapped to the synthesis parameters of the digital sound production mechanism (Fig. 2(b)). The produced sound is post-processed with additional digital signal processing (DSP) algorithms to generate the sonic output of the DMI (Fig. 2(c)). The performer receives primary feedback in haptic (Fig. 2(d)) and secondary feedback in audio (Fig. 2(e)) modalities.

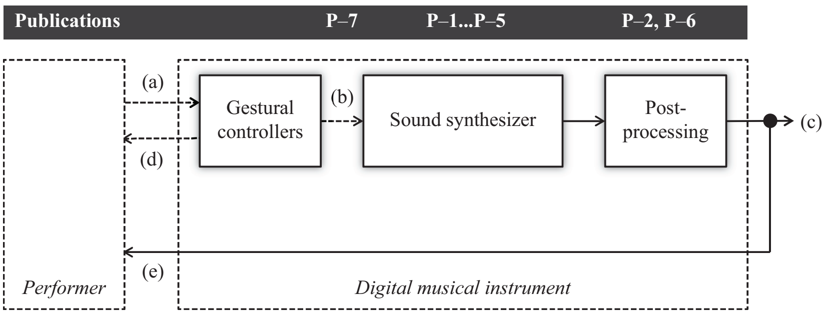


Fig. 2. Digital musical instrument (a) input gestures, (b) mapped control signals, (c) audio output, (d) primary feedback: haptic, (e) secondary feedback: audio.

The primary scope of this thesis is in the sound production and post-processing stages of the DMI pipeline, with specific focus on abstract sound synthesis algorithms. Abstract algorithms form one of the four main categories of digital sound synthesis techniques [Smith, 1991; Tolonen et al., 1998], complementing the processed recording, spectral, and physical modeling categories. The methods in the *processed recordings* category reproduce existing sounds accurately, but have large memory requirements even if looping is applied [Massie, 1998; Maher, 2006]. The *spectral* methods model the receiver of the sound using generic sinusoidal models that are very flexible, but require dense control streams [Serra and Smith, 1990; Rodet and Depalle, 1992; Depalle et al., 1993; Savioja et al., 2011]. The methods in the *physical modeling* category model the source of the sound. They produce realistic timbres with intuitive control properties, but pose high demands on computational power [Smith, 1992; Välimäki et al., 1996; Smyth and Abel, 2012].

Abstract sound synthesis techniques are computationally efficient, have low memory requirements, and are capable of achieving dynamic spectral control with just few parameters. These properties are ideal for resource constrained device platforms such as mobile phones and tablet computers and for software synthesizer plugin implementations running in digital audio workstation and sequencing environments where the demand for polyphony is high. Web applications running in browsers benefit from efficient client-side synthesis algorithms as well, since the download time for sampled sounds can be eliminated entirely. Abstract techniques use mathematical functions and computer algorithms to create sounds that do not have a direct physical interpretation or connection to existing natural sounds. This is both an asset and a limitation: the strength of abstract synthesis techniques is not in the emulation of acoustic instrument sounds, but in the ability to produce musical timbres that are impossible to achieve by physical means. In fact, these timbres contribute a large part of the characteristic sound of a modern musical instrument, the synthesizer.

The characteristic timbres of classic abstract sound synthesis methods are produced by two main subclasses. In the digital domain, virtual analog techniques form an active research area [Stilson, 2006; Pekonen and Välimäki, 2011a; Pakarinen et al., 2011]. The second subclass is formed by nonlinear distortion and audio-rate modulation techniques, which is the finer scope of this thesis.

The primary aim of the research was to explore established nonlinear abstract sound synthesis methods to gain better understanding of their parameters and mutual relationships, to extend them, and to discover entirely novel synthesis techniques. The secondary aim of the research focused on improving the sound quality of the produced timbres and the efficiency of the performance control streams.

Publications P-1 through P-5 introduce extended and novel synthesis techniques, P-2 and P-6 derive new efficient aliasing reduction algorithms, and P-7 proposes an improvement for a real-time sound synthesis control protocol (see Fig. 2).

Given the field of music technology and the constructive research method, practical applicability served as a parallel driving goal throughout the research. Each proposed technique was implemented in software and tested with musical and sensor-based controller devices. This process was iterative and led to adjustments in the initially proposed algorithms in terms of spectral diversity, parametrization, and efficiency. Online and offline analysis tools were used to investigate the relation between synthesis parameters and spectral properties and to derive the theoretical connections. Most publications were accompanied with a webpage containing open source software implementations and sound examples to increase reproducibility and to encourage practical implementation efforts in music-related applications.

The contents of the thesis consist of this introduction and seven peer-reviewed papers, which have been published in journals and international conferences. The rest of the introduction is structured as follows. Section 2 surveys abstract sound synthesis techniques and presents a compound model for their relationships. This provides a contextual framework for publications P-1 through P-6. Section 3 discusses performative control aspects of the presented compound sound synthesis model, providing context for P-7. The main results of the research are summarized in Section 4, and finally, Section 5 concludes with thoughts on future research directions.

2. Abstract sound synthesis techniques

This section surveys abstract sound synthesis techniques. The focus is on classic methods, i.e., nonlinear distortion and audio-rate modulation, since they provide a well-established frame of reference for publications P-1 through P-6 of this thesis. Recent research related to these methods was collected from conference proceedings, journals, and textbooks. A comprehensive tutorial review extending up to the year 1995 is found in [Roads, 1996].

This section also provides insight to the relationships and shared concepts between the classic methods. The information leading to such convergence was gathered from prior research, and aggregated here in the form of principal equations and spectral descriptions, and as a flow-based model of an augmented digital oscillator. This enables knowledge transfer between seemingly distinct sets of research results. For example, the Bessel expansion of the frequency modulation spectrum is usable in the analysis of the phaseshaping technique.

The relationships are presented in Section 2.5. Before that, Sections 2.1 and 2.2 provide a conceptual framework for this introduction in terms of unit generators, instrument graphs, and the classic digital oscillator. Section 2.3 describes nonlinear phase and amplitude distortion techniques (which are related to P-1 and P-2). Audio-rate modulation and feedback (P-3 through P-5) are explored in Sections 2.4 from the classic point of view, and in Section 2.6 from a more recent filter-based perspective. Alias-suppression methods are discussed in Section 2.7 (P-2, P-6).

2.1 Unit generator

The first computer music system was developed in the 1950s at the Bell Laboratories by Max Mathews [1963]. One of the fundamental concepts of Mathews' system was the unit generator (UG), which is a generic building block that abstracts the low-level computation of a DSP algorithm as a black box. The box is interfaced with input and output ports, which may be interconnected to form more complex composite instrument graphs.

The connections between the ports carry static or time-varying signals. The *static signals* provide initial one-shot parametrization, while the time-varying control-, audio-, and sensor-rate signals are used for modulation purposes. *Control-rate* signals operate at haptic rates [Verplank et al., 2000], which are typically lower than 15 Hz. They are created either externally by tracking the interactive expressions of the performer, e.g., using modulation wheels, knobs, joysticks, and touch sensors, or internally as the outputs of control-rate UGs, e.g., using the low frequency oscillator (LFO) and the envelope generator (EG). The perceptual effects of the control-rate signals depend on the parametrization of the unit generator. For example, an LFO-modulated amplitude port generates a tremolo effect, while an LFO-modulated frequency port produces vibrato.

Audio-rate signals operate at frequencies from 20 Hz upwards, creating modulation effects that have a more profound effect on the produced timbre. For example, the amplitude modulated sinusoidal oscillator, when modulated with another audio-rate sinusoid, produces an output spectrum with two or three partials. A similar two-oscillator setup arranged in a frequency modulation constellation is able to generate more complex spectra. *Sensor-rate* signals operate as control- or audio-rate modulators. The scan rate of a typical sensor is 30–100 Hz.

The unit generator concept supports encapsulation and polymorphism and therefore works well with object-oriented and flow-based programming paradigms. It continues to form the basis of contemporary digital audio synthesis frameworks. The Bell Laboratories system transformed eventually into Music V [Mathews et al., 1969] and its descendants such as Csound [Boulanger, 2000; Boulanger and Lazzarini, 2010] and Cmusic [Moore, 1990], but the concept is utilized likewise in SuperCollider [Wilson et al., 2011], Pd [Puckette, 2007], Nyquist [Dannenberg, 1997], STK [Cook, 2002], Faust [Orlarey et al., 2009], and Max/MSP [Zicarelli, 2002], to name a few. Even commercial hardware synthesizers — starting from the modular devices by Moog and Buchla [Pinch and Trocco, 2004] and continuing to the most recent product releases — present or implement their architectures using interconnected modules.

2.2 Classic digital oscillator

A periodic signal source is modeled using a special unit generator called the digital oscillator. Its classic prototypic Music V form is parametrized with four inputs. These are the fundamental frequency f_0, initial phase ϕ_0, waveform function $F(\cdot)$, and peak amplitude A [Dodge and Jerse, 1997, pp. 75–76], as shown in Fig. 3(a). The fundamental frequency f_0 contributes to the perceived pitch, the waveform function $F(\cdot)$, which describes the shape of the produced signal, to the timbre, and the peak amplitude A to the loudness of the generated sound. The initial phase

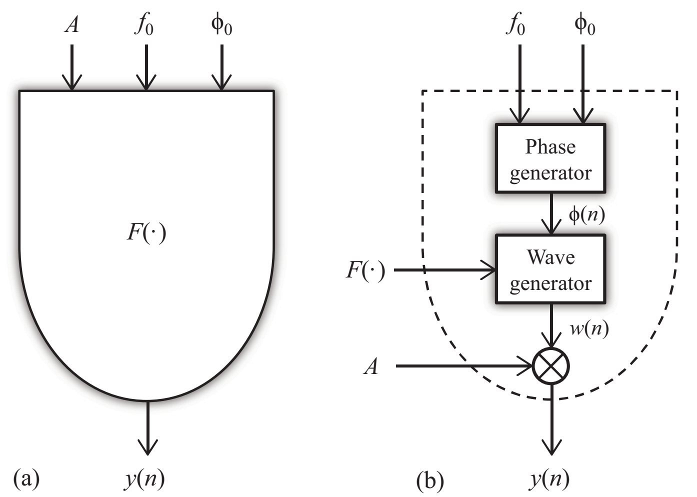


Fig. 3. (a) Classic digital oscillator, and (b) its internal structure.

ϕ_0 has only a minor perceived contribution for a single isolated oscillator. The output $y(n)$ of the oscillator is an audio- or control-rate signal, which may be subsequently routed to the inputs of other unit generators or to the system output. In this section, the output of the oscillator is an audio-rate signal.

Although the classic digital oscillator is usually considered as an atomic entity, it has an internal structure that encapsulates two cascaded UGs as shown in Fig. 3(b) [Moore, 1977a, p. 30]. The phase generator (PG), which takes f_0 and ϕ_0 as inputs, provides the periodic driving source of the oscillator. Its period in samples $P_0 = f_s/f_0$ and the reciprocal phase increment $\phi_\Delta = f_0/f_s$ are related to the sampling rate f_s of the system, thereby defining the pitch of the oscillator. The output signal of the PG is a unipolar ramp, i.e., the normalized phase signal $\phi(n)$ which is generated using a unipolar modulo counter (see for example [Moore, 1977a, pp. 61-63])

$$\phi(n) = [\phi(n-1) + \phi_\Delta] \bmod 1, \quad \phi(0) = \phi_0, \tag{1}$$

where n is the sample number, and 'mod 1' is the modulo-1 operation x – floor(x) that wraps the PG output within the range [0,1).

The wave generator (WG) takes the PG output signal $\phi(n)$ and transforms it according to the waveform function $F(\cdot)$ to produce the WG output signal

$$w(n) = F[a_1\phi(n)], \tag{2}$$

where a_1 is the scaling factor defining, for example, the length of a lookup table or the full radian cycle. The waveform function $F(\cdot)$ is usually evaluated either directly from the instantaneous $\phi(n)$ values on a sample-by-sample basis or indirectly using the scaled $\phi(n)$ as an index into a stored lookup table. The former is well suited for pipelined and memory constrained architectures [Välimäki, 2005]. Direct evaluation also allows waveform function parametrization as will be discussed in Section 2.4. The indirect approach is preferred in the computationally intensive waveform function evaluation [Kim and Park, 2010] and sample-based wavetable oscillator implementations [Massie, 1998], where fractional indices are interpolated [Moore, 1977b; Dannenberg, 1998]. The more infrequently used breakpoint interpolation methods fall between direct and indirect evaluation [Mitsuhashi, 1982; Collins, 1999]. Dynamic wavetables are utilized in pre-waveguide [Karplus and Strong, 1983], adaptive digital audio effects [Verfaille et al., 2006, Pakarinen et al., 2011], and other delayline-based algorithms [Nam et al., 2009; Lowenfels, 2003].

The oscillator output signal $y(n)$ is finally formed by scaling the WG output with the peak amplitude A:

$$y(n) = Aw(n) = AF\left[a_1\phi(n)\right]. \tag{3}$$

2.3 Nonlinear shaping

Deforming $\phi(n)$ of Eq. (1) and $w(n)$ of Eq. (2) with nonlinear shaping functions $P(\cdot)$ and $W(\cdot)$, respectively, introduce timbral changes in the oscillator output $y(n)$. The shaping operations are performed in the phaseshaper [Timoney, 2009] and waveshaper [Le Brun, 1979; Arfib, 1979] unit generators as shown in Fig. 4(a).

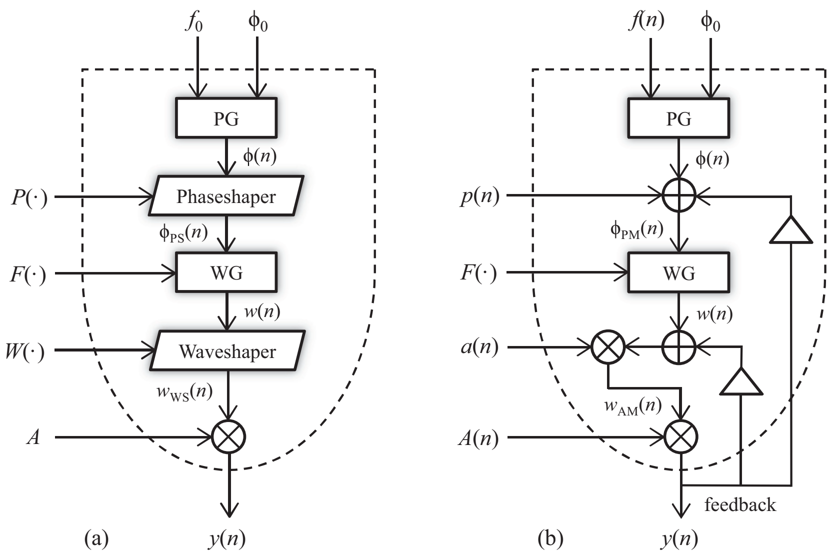


Fig. 4. Classic digital oscillator augmented with (a) nonlinear shaping, and (b) audio-rate modulation and feedback techniques.

2.3.1 Phaseshaping

The phaseshaper transforms the uniform phase signal $\phi(n)$ into the shaped phase $\phi_{PS}(n)$ using function mapping

$$\phi_{PS}(n) = P[\phi(n)]. \tag{4}$$

The shaped phase signal is routed to the wave generator WG, and therefore WG is in fact a terminal phaseshaper (or initial waveshaper) acting on $\phi_{PS}(n)$. The shaping operation $P(\cdot)$ can be implemented using direct evaluation, conditional branching, or more seldom, using table lookup.

Publication P-1 explored direct evaluation in terms of function composition, i.e., identifying a set of reusable phaseshapers and nesting them into more and more complex phaseshaping pipelines. Simple operations, such as the modulo of Eq. (1), the linear transformation of Eq. (2), and the absolute value function were first identified and pre-parametrized into a set of atomic semantic entities. The entities were then cascaded into elementary piecewise linear phaseshapers, which were reused in turn to construct more elaborate shaper pipelines. Essl [2010] describes a dataflow engine that predefines a similar set of semantically fixed signal transformations. He applies the transformations at various frame rates to unify the manipulation of control-, sensor-, and audio-rate signals.

P-1 evaluated the proposed set of reusable phaseshapers by engaging them in several classic and novel oscillator effects algorithms. The novel algorithms comprised single-oscillator hard and soft sync, triangle modulation, efficient supersaw simulation, and sinusoidal waveshape modulation effects (see Fig. 5). The last eventually led to the discovery of the Vector Phaseshaping Synthesis technique introduced in Publication P-2.

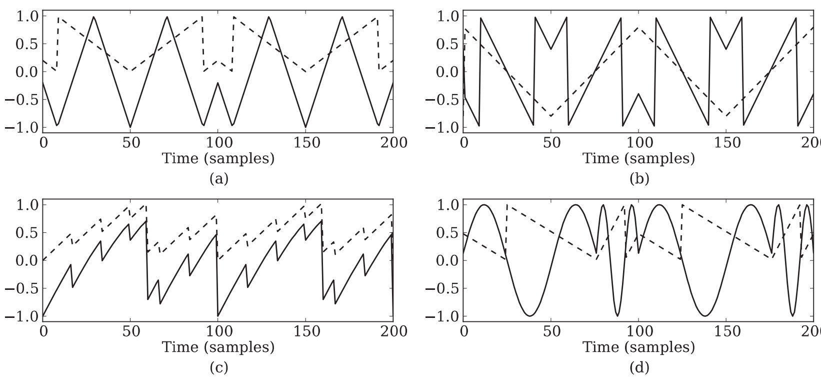


Fig. 5. Phaseshaper pipeline (dashed) and WG output (solid) signals, f_0 = 441 Hz. (a) Single oscillator soft sync, a_1 = 1.2, (b) triangle modulation, a_{TM} = 0.8, (c) supersaw simulation, m_1 = 0.25, m_2 = 0.8, a_1 = 1.5, and (d) sinusoidal waveshape modulation, w = 0.75, a_1 = 1.5.

The direct evaluation approach of P-1 is especially efficient in serial shaping operations, where the result does not depend on a particular state of the input. An alternative implementation strategy employs conditional branching to evaluate mutually exclusive choices based on the instantaneous value of the phase signal. This approach was utilized in the classic Phase Distortion (PD) synthesis technique [Ishibashi, 1987], which introduced an inflection point parameter d to bend the uniformly sloped phase signal $\phi(n)$ into two or more piecewise linear segments. The first embodiment of the PD technique produces a smoothed sawtooth waveform using a phaseshaper [Lazzarini, et al., 2009a]

$$P_{\mathrm{PD}}(x) = \begin{cases} 0.5\dfrac{x}{d}, & 0 \le x \le d \\ 0.5\dfrac{x-d}{1-d} + 0.5, & d < x < 1, \end{cases} \tag{5}$$

where $x = \phi(n)$ and $0 < d < 1$. Figure 6(a) shows the shaped ($\phi_{\mathrm{PD}}(n) = P_{\mathrm{PD}}[\phi(n)]$, solid) and the uniform ($\phi(n)$, dashed) phase signals for $d = 0.05$. During the first segment (i.e., $0 \le x \le d$), the slope of the phaseshaped signal is steeper than the uniform phase increment ϕ_Δ and more gradual during the rest of the cycle. The shaped phase signal is converted into a WG cosine lookup table index. Larger phase increments increase the scanning speed of the lookup, producing the tilted waveshape of Fig. 6(b), and the complex harmonic spectrum shown in Fig. 6(c). The bandwidth of the spectrum increases with the deviation abs(0.5 – d) from the undistorted state $d = 0.5$, affording dynamic brightness control. The loose physical interpretation of this is the "brassification" effect discussed in [Cooper and Abel, 2010]: at high sound pressure levels peaks travel faster in air than the troughs, which distorts the propagating pressure waveform.

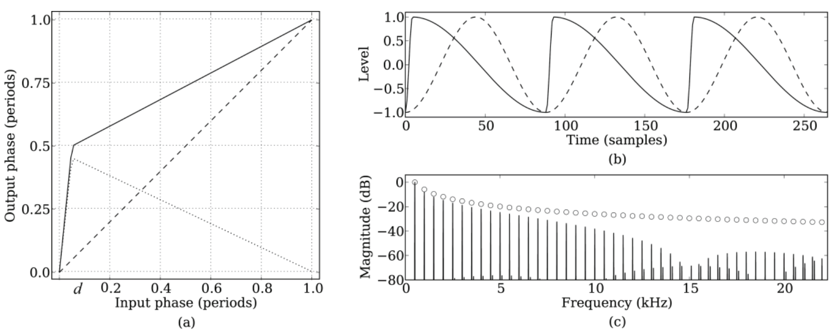


Fig. 6. Sawtooth-like PD waveform, $f_0 = 500$ Hz, $d = 0.05$: (a) shaped phase (solid), uniform phase (dashed), and their difference (dotted), (b) distorted (solid) and undistorted (dashed) inverted cosine signal, and (c) spectrum of the distorted signal. The harmonic magnitudes of an ideal sawtooth signal are indicated with circles.

The second and third embodiments of the PD technique use symmetry properties to approximate square and impulse waveforms, while the fourth and fifth ones generate resonance peaks by allowing $\phi_{\mathrm{PD}}(n)$ to exceed unity. In the latter case, non-integral $\phi_{\mathrm{PD}}(n)$ periods produce incomplete cosine cycles, which generate discontinuities in the waveshape. Abrupt jumps in the time-domain signal increase the high end spectral content but, unfortunately, lead to unwanted aliasing artefacts as well. The fifth embodiment of the technique suppresses aliasing using audio-rate amplitude modulation to fade out the trailing end, i.e., the discontinuity of the generated waveshape.

PD was utilized commercially in the Casio CZ [1984] series synthesizers, whose patch programming interfaces were modeled after the classic subtractive synthesizer architecture, i.e., a cascaded source — filter — amplifier pipeline. However, the strength of the classic PD technique is not in virtual analog modeling as can be seen from Fig. 6(c). The spectrum differs considerably from that of an ideal sawtooth (indicated with circular markers), because the amount of distortion (here d = 0.05) needs to be controlled in order to reduce excessive aliasing. Lazzarini and Timoney [2010a] reduce aliasing of sawtooth and pulse waveforms by employing bandlimited phaseshapers. Timoney [2009a] suggests the phaseshaper $P(x) = x^{1/a}$, which produces a morph between a smooth asymmetric impulse ($a < 1$), a sinusoid ($a = 1$), and a sawtooth-like ($a > 1$) waveshape.

Each PD embodiment discussed above requires a dedicated phaseshaper. The Vector Phaseshaping Synthesis (VPS) method introduced in Publication P-2 proposes a single phaseshaper that implements the functionality of all PD embodiments and extends the original method in four additional ways to increase its expressive power and timbral space. First, instead of a single value d, the inflection point is expressed as a two-dimensional (2-D) vector $\boldsymbol{p} = (d, v)$, where $v > 0$ gives the ordinate value of the inflection. A single-point VPS waveshaper is given by

$$P_{\mathrm{VPS}}(x) = \begin{cases} v\dfrac{x}{d}, & 0 \le x < d \\ (1-v)\dfrac{x-d}{1-d} + v, & d \le x < 1, \end{cases} \tag{6}$$

which reduces to Eq. (5) when $v = 0.5$.

Second, VPS allows multiple inflection points per cycle, which enables more complex waveshapes than the original technique (see Fig. 7). The VPS waveshaper with N inflection points is expressed as

$$P_{\mathrm{VPS}}(x) = \begin{cases} v_0 \dfrac{x}{d_0}, & 0 \le x < d_0 \\ (v_k - v_{k-1})\dfrac{x - d_{k-1}}{d_k - d_{k-1}} + v_{k-1}, & d_{k-1} \le x < d_k \\ \ldots & \\ (1 - v_{N-1})\dfrac{x - d_{N-1}}{1 - d_{N-1}} + v_{N-1}, & d_{N-1} \le x < 1, \end{cases} \tag{7}$$

which reduces to Eq. (6) when $N = 1$.

Third, VPS proposes an aliasing-suppression mechanism that preserves the high end spectral content better than the amplitude modulation-based solution of PD (see Section 2.7). Finally, VPS extends inflection point modulation to audio rates, which enables, among other effects, the synthesis of inharmonic spectra.

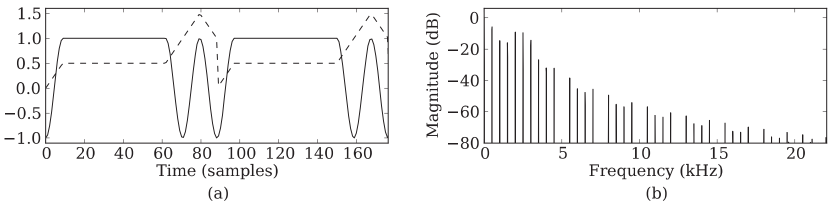


Fig. 7. (a) VPS phase (dashed) and waveform (solid), and (b) its spectra using three inflection points (0.1,0.5), (0.7,0.5), and (0.9,1.5) with $f_0 = 500$ Hz.

Despite these generalizations, VPS manages to retain the conceptual simplicity of the original method. For example, the added dimension and the increased number of inflection points afford expressive timbral manipulation using intuitive 2-D multi-touch controllers. Furthermore, since each cycle of the waveshape is independent, changing the entire set of inflection points at once produces more dramatically evolving timbres. Such sequences have applications as abstract forms of wave sequencing [Phillips, 1994, pp. 92–100] and concatenative synthesis [Schwarz, 2007].

Another dramatic phase distortion effect is achieved with the oscillator hard sync [Brandt, 2001], where the phase of the slave oscillator is reset when the phase of the master wraps around. This classic oscillator effect is used in early analog synthesizers and in their virtual analog reincarnations. An efficient single-operator hard sync algorithm is described in Publication P-1, whose alias-suppressed form is described in P-6 (see Section 2.7).

2.3.2 Waveshaping

The waveshaper unit generator distorts the amplitude of the WG output signal $w(n)$ using function mapping

$$w_{\mathrm{WS}}(n) = W\big[aw(n)\big], \tag{8}$$

where $a \in [0,1]$ is the distortion index, and $W(\cdot)$ is the transfer function describing the nonlinearity of the system. In classic waveshaping, $w(n)$ is sinusoidal and $W(\cdot)$ is time-invariant [Le Brun, 1979; Arfib, 1979].

Waveshaping debuted as a synthesis method already in 1969 [Risset, 1969, recipe #150.1], which used a piecewise linear transfer function to distort the output of a sinusoidal oscillator into clarinet-like timbres. Schaefer [1970] employed Chebyshev polynomials to derive the relationship between the Fourier coefficients of the produced spectrum and the power series coefficients of the transfer function, using a normalized cosine input. Suen [1970] noted that the amplitude of the input has a profound influence on the harmonic amplitudes and extended Schaefer's equations to cover arbitrary-range cosine inputs, thereby enabling dynamic spectra.

The exploration of the method continued in two independent and partly complementary landmark papers by Le Brun [1979] and Arfib [1979]. Both works used sinusoidal inputs and the prior results of Schaefer and Suen in the transfer function definition. This enables generation of any steady-state harmonic and optionally bandlimited spectra, as long as the input sinusoid is kept at a constant level. Dynamic spectra are achieved by making the distortion index time variant, but although the bandwidth of the spectra is directly related to the distortion index, there may be large individual variations in the harmonic levels throughout the spectrum. Furthermore, the evolution of even and odd harmonics is unrelated. Amplitude variations in the input do not introduce extra harmonics higher than the order of the polynomial transfer function, however. Inharmonic spectra can be generated by multiplying the waveshaped signal with a carrier sinusoid whose frequency is in irrational relation to the waveshaper input.

Arfib [1979] proposes a double modulation technique, which multiplies the outputs of two waveshaper oscillators with a sinusoid. If the fundamental frequencies of all three oscillators have an integral relationship, the resultant spectrum is harmonic but more manageable since there is now separate control over two transfer functions. Irrational relations produce rich inharmonic timbres. Beauchamp and Horner [1992] used multiple parallel waveshaping oscillators to improve the results of parameter matching.

Le Brun's [1979] elegant phase quadrature form allows exact definition of the phase of each harmonic. The phases are controlled with an extra signal that is generated using heterodyned waveshaping based on Chebyshev polynomials of the second kind. Le Brun also describes a peak normalization algorithm, which uses a look-up table whose elements are computed from the maximum values of the transfer function. However, the normalization of the phase quadrature form is more involved and is not addressed in the work.

De Poli [1984] extended the previous works by decomposing a rational transfer function into a quotient polynomial (producing classic waveshaping) and a fractional part. The latter can be expanded using partial fractions into several additive terms, which contribute to the overall spectrum separately. Thus, it is possible to construct the desired spectral structure piece by piece and then consolidate the quotient and the additive terms into a single rational transfer function.

More recently, Timoney et al. [2009a] and Lazzarini and Timoney [2010a] produced triangular, square, and pulse waveforms using transcendental arccos, hyperbolic tangent, and exponential waveshapers. Although the Taylor series of these functions are infinite, frequency-dependent distortion index mapping can effectively control the amount of aliasing. Essl [2010] uses a set of piecewise linear, polynomial, exponential, logarithmic, and trigonometric waveshapers as semantically fixed atomic signal manipulators. Waveshaping was used commercially in the Korg 01/W synthesizer of 1991 and is employed in the contemporary Korg Kronos [2011] hardware and Roland Z3TA[1] software synthesizers.

2.4 Audio-rate modulation and feedback

The classic digital oscillator of Fig. 3 is parametrized with frequency, phase, amplitude, and waveshape inputs. The manipulation of these and other synthesis parameters, such as the distortion index discussed in the previous section, is called modulation. At control-rate, the time-variant amplitude $A(n)$ generates cresecendo, diminuendo, and tremolo effects, while the frequency input signal $f(n)$ produces glissando, portamento, and vibrato. The phase offset signal $\phi_0(n)$ does not compose perceivable effects at control rates, whereas the waveshape parameter modulation produces timbral variation also at low frequencies.

However, more profound timbral effects emerge when the modulation is performed at audio-rate. Figure 4(b) shows the audio-rate modulation signals $f(n)$, $p(n)$, $a(n)$, and $F(\cdot)$ producing frequency, phase, amplitude and waveshape parameter modulation, respectively, of the carrier oscillator. The feedback paths route the oscillator output back into the phase and amplitude modulation inputs of the carrier. The following subsections discuss each of these modulation targets in detail.

2.4.1 Frequency and phase modulation

The phase angle of a carrier oscillator may be manipulated using two closely related modulation techniques. Frequency modulation (FM) operates on the phase increment, while phase modulation (PM) manipulates the instantaneous phase of the carrier [Holm, 1992]. Using Eq. (1) as a reference, the phase signals for FM and PM are given by

$$\phi_{\mathrm{FM}}(n) = [\phi_\Delta + f(n)] + \phi(n-1)\,, \quad \phi(0) = \phi_0 \tag{9a}$$

$$\phi_{\mathrm{PM}}(n) = \phi(n) + p(n). \tag{9b}$$

Because the difference between the two techniques is minor, FM and PM produce perceptually identical timbres. PM has the slight advantage of keeping the modulation index independent of the carrier and modulator frequencies and is therefore considered the preferred implementation of angle modulation. The original FM formulation of Chowning's [1973] seminal paper applies PM, and it is given in discrete time by

$$y(n) = A\sin\{a_1\phi(n) + I\sin[a_1\phi_{\mathrm{m}}(n)]\} = A\sin[a_1\phi(n) + p(n)], \tag{10}$$

[1] http://www.cakewalk.com/Products/Z3TA/

where $a_1 = 2\pi$, $\phi(n)$ and $\phi_m(n)$ are the carrier and modulator phase signals, and I is the modulation index that provides control over dynamic spectra. When $I = 0$, there is no modulation and the output is the carrier sinusoid at frequency f_0. When I increases, the energy of the carrier is spread symmetrically to two sidebands, producing spectral components at $f_0 \pm kf_m$, where $k > 0$ and integer, and f_m is the modulation frequency. The magnitudes of the components are given by the Bessel functions of the first kind $J_n(I)$, which give rise to the characteristic fluctuating nature of the FM spectral evolution. The bandwidth of the spectrum is defined by the modulation index and the modulator frequency f_m, and is roughly equal to $2f_mI$. Negative frequencies, which are produced when $f_0 < f_mI$, reflect back from 0 Hz with phase inversion. The produced spectrum is harmonic if f_m / f_0 is integral or rational, otherwise it is inharmonic.

Numerous variations of the simple FM equation of Eq. (10) exist in the literature. Chowning [1973, 1989] employed complex carrier compositions, where a single modulator was used to manipulate multiple separately parametrized carriers. The resultant spectrum superimposes the contribution of each carrier, which enables, for example, the generation of multiple formant regions. Complex modulating signals were explored in parallel and cascaded multi-modulator setups in [Le Brun, 1977; Schottstaedt, 1977; Tan et al., 1994; Timoney and Lazzarini, 2012]. Tomisawa [1981] describes a feedback configuration (FBFM), where the output of the oscillator is connected back to its phase input. A self-modulating FBFM oscillator evens out the characteristic oscillating spectral evolution of FM. Higher modulation indices produce excessive aliasing up to the point where FBFM can simulate noise. The commercial implementations arranged FM oscillators into complex predefined structures (i.e., algorithms) that contain both multi-carrier and multi-modulator arrangements [Chowning and Bristow, 1986]. Software synthesizers such as FM-8[2] and Sytrus[3] allow free-form oscillator patching. The cross modulation technique used in analog synthesizers produces exponential FM [Hutchins, 1975], whose bandwidth for sinusoidal modulation was recently derived by Timoney and Lazzarini [2011].

The undulating nature of dynamic FM spectra is not optimal for automatic synthesis parameter optimization [Beauchamp, 1982; Horner et al., 1993; Horner 1998; Tan and Lim, 1996; Lai et al., 2006; Mitchell, 2010; Zhang and Chen, 2012] because acoustic instruments have a smaller magnitude of variation in their individual harmonics. Timoney et al. [2008] and Lazzarini and Timoney [2010b] propose an alternative FM formulation, called ModFM, where partial magnitudes are bound to modified Bessel functions that are non-oscillatory and more naturally decaying than the Bessel functions of the first kind. The formulation is implemented using exponential waveshaping and ring modulation and is thus

[2] http://www.native-instruments.com/#/products/producer/fm8
[3] http://www.image-line.com/documents/sytrus.html

related to the single-sided [Moorer, 1977] and asymmetric [Palamin et al., 1988; Tan and Gan, 1993] forms of FM. ModFM extensions are able to produce formant regions and sharp resonances, and it is possible to morph between simple FM, the proposed form, and single-sided spectra of Moorer [1977]. A phase vocoder implementation using a bank of ModFM operators was described in [Lazzarini and Timoney, 2011]. Macret, Pasquier and Smyth [2012] used genetic algorithms to match the first ten harmonics of several acoustic instrument tones with two, four, and six carrier simple FM and ModFM. ModFM compared favorably with 13% faster convergence time on the average and having a better fitness score in 30 experiments out of 45.

Another FM reformulation called split-sideband synthesis (SpSB) [Lazzarini et al., 2008a; 2008b] segregates the output of the simple FM of Eq. (10) into four independent channels. First, odd and even components are separated using the waveshaping interpretation of PM (see Section 2.5). Second, the upper and lower sidebands of the resulting component spectra are split at f_0 using single-sideband modulation (see Section 2.4.2). Since each of the outputs may be used separately, or in any combination, multi-oscillator SpSB-arrangements expand the sonic palette of FM, while simultaneously offering more precise control over the produced timbres.

Self-adaptive digital audio effects [Verfaille et al., 2006; Pakarinen et al., 2011] extract control features from the input audio signal, which is then processed using a DSP algorithm driven by the extracted time varying parameters. This retains the delicate gestural control properties of musical instruments, enriching their sound with hybrid acoustic-synthetic forms. Adaptive FM (AdFM) by Lazzarini, Timoney and Lysaght [2007; 2008c; 2008d] apply sinusoidal PM to an arbitrary carrier signal, thereby complementing the work of Poepel and Dannenberg [2005]. In AdFM, the carrier is fed through a variable-length delay line, whose length is modulated by a sinusoid. The frequency and depth of modulation are controlled by the fundamental frequency (and amplitude) of the external carrier signal.

2.4.2 Amplitude modulation

Multiplication of two digital signals yields amplitude modulation (AM) [Dodge and Jerse, 1997, pp. 90-96]. The generic AM formula, with reference to Fig. 4(b), is given by

$$w_{\mathrm{AM}}(n) = \left[A_{\mathrm{c}} + A_{\mathrm{m}} a(n)\right] w(n), \tag{11}$$

where $w(n)$ is the carrier with fundamental frequency f_0, $a(n)$ is the modulator with fundamental frequency f_{m}, and A_{c} and A_{m} are the scaling coefficients in the range [0,1]. Consider first the sinusoidal carrier and modulation signals. If $A_{\mathrm{m}} = 0$, there is no modulation and the output of Eq. (11) is the scaled carrier signal $A_{\mathrm{c}}w(n)$. If A_{m}

> 0 and $A_c = 0$, from the trigonometric identity $\cos(\theta_1)\cos(\theta_2) = [\cos(\theta_1 - \theta_2) + \cos(\theta_1 + \theta_2)] / 2$, the generated spectrum is seen to consist of two sideband components $f_0 \pm f_m$, centered symmetrically around the carrier frequency. This special form of AM, with the carrier component missing and energy split equally between the sidebands, is called ring modulation (RM). Making $A_c > 0$ produces various forms of suppressed-carrier AM. At $A_c = 1$ and $A_m > 0$, the classic double-sideband full-carrier AM is generated. Naturally, the output of Eq. (11) should be scaled appropriately to avoid clipping. The magnitudes of the sidebands increase with A_m, reaching up to the maximum level of 6 dB below the magnitude of the carrier. The spectrum is inharmonic for irrational f_m / f_0 relations.

Sidebands are generated in a similar way for complex carrier and modulation signals: for each partial in each signal, two additional components are produced. The highest partial of the generated spectrum is located at a frequency which is the sum of the highest component of the carrier and the highest one of the modulator. Aliasing may thus be produced even with low modulation amounts, unless the inputs are oversampled and/or kept relatively simple. However, with pure sinusoidal inputs, the bandwidth of the amplitude modulated signal spectrum is too low to be of musical use.

The bandwidth can be increased by feeding the amplitude modulated signal back to the amplitude input of the oscillator, i.e., by using the output of the oscillator as the modulator. This method appeared in [Risset, 1969, recipe #510.1] and [Layzer, 1971] and was later briefly discussed in [Valsamakis and Miranda, 2005], but otherwise left unexplored. A preliminary study by the present author and his colleagues [Lazzarini et al., 2009b] augmented Layzer's original formulation with modulation index β and settled for a unit delay feedback length. The sinusoidal feedback AM oscillator

$$y(n) = \left[1 + \beta y(n-1)\right]\cos\left[2\pi\phi(n)\right] \tag{12}$$

was found to produce a pulse-like waveform whose spectral bandwidth is directly related to β (see Fig. 9). The spectral evolution is smooth, which was also observed by Tomisawa for FBFM [1981]. Also, five variations to Eq. (12) were introduced, their spectral morphologies were discussed, and the paper was complemented with practical application examples. Nevertheless, the theory of feedback AM was still left unattached. Publications P-4 and P-5 provide a detailed investigation of the feedback amplitude modulation (FBAM) synthesis technique in its first- and second-order forms by interpreting the algorithm as a coefficient-modulated one-pole IIR filter (see Section 2.6.2).

Single-sideband modulation (SSB) provides separate outputs for the lower and the upper sidebands of the classic RM spectrum [Bode and Moog, 1972]. SSB was recently employed in [Lazzarini et al., 2008a; 2008b; 2008d; 2009a], where a pair of sixth order allpass filters (implemented as a cascade of first-order sections) were used to generate the required phase quadrature signal. Kleimola [2008a] replaced the signal multiplication of Eq. (11) with a bitwise logical operation (AND, OR, XOR) to produce modulo arithmetic influenced staircased signals. Because of the sharp edges, the method generates an excessive amount of aliasing. The aliasing is suppressed considerably using the method proposed in Publication P-6 (see the accompanying web page). The classic analog synthesizer ARP Odyssey[4] employed XOR circuits for RM-like effects. Virtual analog models for diode- and transistor-based RM implementations were explored by Hoffmann-Burchardi [2008; 2009] and Parker [2011].

2.4.3 Waveshape parameter modulation

Direct evaluation of the waveform generator function $F(\cdot)$ may provide additional parameters for timbral domain manipulation. The most well-known example of this is the classic pulse width modulation (PWM) technique, where the duty width parameter of a pulse wave is manipulated at control rate [Guinee, 2007], or more infrequently, at audio rates [Hutchins and Ludwig, 1980]. The term waveshape parameter modulation was coined by Mitsuhashi in [1980], where he modulated the zero-crossing point of a parabolic sinusoid approximation. Lang [2004] introduced a related technique that modulates the control points of a constrained Bézier curve.

2.5 Convergence

The abstract sound synthesis techniques based on nonlinear shaping and audio-rate modulation are closely related. Their canonical configurations deform sinusoids to generate complex spectra. The amount of deformation is controlled by a distortion or modulation index parameter, and the produced spectra are either purely harmonic or organized around the carrier frequency into upper and lower sidebands, where the partial distance is determined by the modulation frequency. The negative frequencies reflect from 0 Hz back to the positive frequencies, molding harmonic amplitudes or giving rise to inharmonic spectra.

The principal equations of the classic methods are closed form representations of geometric trigonometric series, similar to the discrete summation formulas (DSF) of [Moorer, 1976; 1977] and [Winham and Stieglitz, 1970]. The multisine oscillator recently described in [Zölzer, 2012] augments the DSF-related buzz oscillator with a parameter for adjusting the slope of the produced spectrum.

The classic methods have also a deeper connection, since the seemingly disparate equations may be interpreted by other more general abstract synthesis techniques.

---

[4] http://www.hylander.com/moogschematics.html

For example, phaseshaping, including the PD synthesis method, may be expressed in terms of classic phase modulation [Timoney et al., 2009a; 2009b, Lazzarini et al., 2009a]. To see the connection, the phaseshaper $P(\cdot)$ and the resulting phase signal $\phi_{PS}(n)$ are first decomposed into linear $\phi(n)$ and nonlinear $P_M[\phi(n)]$ terms, given by

$$\phi_{PS}(n) = P[\phi(n)] = \phi(n) + P_M[\phi(n)]. \tag{13}$$

The linear and nonlinear terms are depicted by the dashed and dotted lines in Fig. 6(a), showing that the nonlinear term is complex, i.e., non-sinusoidal. When $\phi_{PS}(n)$ is scaled and fed through a sinusoidal WG, the familiar PM expression emerges (see Eq. (10)):

$$w(n) = F[2\pi\phi_{PS}(n)] = \sin\{2\pi\phi(n) + 2\pi P_M[\phi(n)]\}. \tag{14}$$

Phaseshaping is thus an intuitive form of complex modulator phase-synchronous PM. The amount of modulation I depends on the phaseshaper function $P_M(\cdot)$. For the classic phase distortion method, $I_{PD} = 2\pi(0.5 - d)$ [Lazzarini et al., 2009a]. By analogy, the modulation index of a single inflection point VPS is $I_{VPS} = 2\pi(v - d)$, which reveals that VPS affords higher amounts of modulation compared to PD. Since phaseshaping operates on a single period, PM may be expressed in terms of phaseshaping only if its spectrum is harmonic, i.e., when f_m/f_c is integral or rational. VPS is able to produce inharmonic timbres as well, but that is achieved using audio-rate inflection point modulation.

The original PM formulation may in turn be expressed in terms of ring modulation and waveshaping, as noted by Le Brun [1979], Arfib [1979], and Lazzarini et al. [2008b]. Using a basic trigonometric identity, Eq. (10) may be rewritten as

$$y(n) = \sin[2\pi\phi_c(n)]\cos\{I\sin[2\pi\phi_m(n)]\} + \cos[2\pi\phi_c(n)]\sin\{I\sin[2\pi\phi_m(n)]\}. \tag{15}$$

The carrier phase term $2\pi\phi_c(n)$ is here taken out of the WG, and the spectrum shifting is performed using ring modulation, i.e., by multiplying the signals in the time domain. The PM modulation index I then corresponds to the distortion index a of the waveshaping technique. By replacing the cos(·) and sin(·) terms above with analytic signals results in SSB modulation, leading to the split-sideband synthesis method of Lazzarini et al. [2008a; 2008b]. Le Brun [1979] further shows that DSF has waveshaping interpretations.

On the other hand, abstract nonlinear sound synthesis techniques may be aggregated into a powerful digital oscillator structure by combining the models of Figs. 4(a) and 4(b) together, as shown in Fig. 8. The output $y(n)$ of the oscillator may subsequently drive the modulation signal inputs $p(n)$ and $a(n)$ of other similar structures, enabling complex modular constellations. Such a versatile synthesis engine, excluding the phaseshaper block, is available in the Korg Kronos Waveshaping VPM synthesis module [Korg, 2011, p.338].

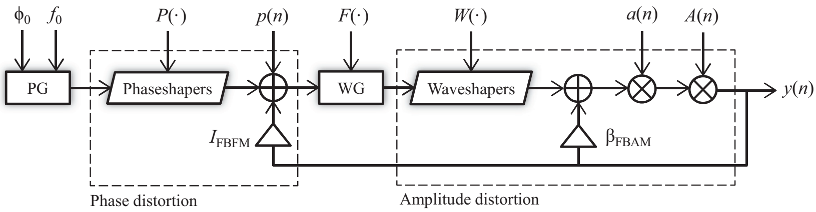


Fig. 8. Aggregate digital oscillator operator for abstract sound synthesis.

2.6 Coefficient modulation

This section interprets abstract sound synthesis methods as time-varying digital filters. The alternative perspective provides an analysis tool and leads to the discovery of novel synthesis techniques. Publication P-3 explores a coefficient-modulated allpass filter chain as a new synthesis method, while Publications P-4 and P-5 analyze the FBAM technique as a coefficient-modulated recursive filter. These synthesis techniques are discussed in subsections 2.6.1 and 2.6.2, respectively.

2.6.1 Allpass filters

Slowly time-varying allpass filters have been used in physical modeling synthesis contexts, such as, passive nonlinearity modeling [Pierce and Van Duyne, 1997], guitar body model [Penttinen et al., 2000], dispersion [Evangelista, 2000], and tension modulation [Välimäki et al., 1998; Tolonen et al., 2000; Pakarinen et al., 2005; Bank, 2009].

Pekonen [2008b] investigated a coefficient-modulated first-order allpass filter as an audio effect and concluded that audio-rate modulation induces phase distortion in its sinusoidal input. Timoney et al. [2009b] noted that the phenomenon is related to the classic PD synthesis method, introduced its allpass-based variation, and used the novel formulation to imitate sawtooth and pulse waveforms in [2009a]. The allpass variation is especially beneficial when processing arbitrary input signals [Lazzarini et al., 2009a].

Välimäki et al. [2009] introduced a novel spectral delay audio effect implementation that cascades a high order allpass structure — constructed from a long chain of identical low order sections — with an equalizing filter. Since allpass filters are dispersive, an input signal entering this spectral delay filter (SDF) faces a frequency dependent delay. Pekonen et al. [2009] extended the SDF structure with a feedback connection and modulated the allpass coefficients at control rates. They noted that the SDF structure remains stable when the modulating signal is within a signed unit interval. In a later work, Pekonen et al. [2011b] simulated the rotating Leslie speaker effect using a pair of control rate modulated SDFs cascaded with individual AM units.

Publication P-3 explores SDF as a novel abstract sound synthesis method by raising the coefficient modulation rate to audio frequencies. In contrast to [Pekonen, 2008b] and [Timoney et al., 2009b], each allpass section of P-3 employs a modified direct form I difference equation

$$y(n) = x(n-1) + x(n)m(n) - y(n-1)m(n)\,, \tag{16}$$

where $x(n)$ is the input and $m(n)$ is the modulator sinusoid. The modification is on the rightmost term, which replaces the delayed modulator signal $m(n-1)$ with $m(n)$. This reduces transient generation for rapid modulation signal changes, and for the continuous sinusoids used in this work, the effect of the approximation is negligible. For an example of the produced effect, see [Lazzarini 2010a].

The terms of Eq. (16) are interpreted as the delayed input, ring modulated input, and ring modulated feedback signals. By using phase synchronous sinusoids as input and modulation signals, comparing Eq. (16) to Eq. (12) reveals that the leftmost and the rightmost terms produce a FBAM-like spectrum, whose carrier f_x and first sideband components $f_x \pm f_m$ are further enforced by the middle term. When many sections are cascaded, the backward leak of each allpass section results in a complex dynamic behavior, whose effect is understood by considering the allpass chain as a dispersive variable length delay line. By rearranging the terms of Eq. (16) as

$$y(n) + y(n-1)m(n) = x(n-1) + x(n)m(n)\,, \tag{17}$$

the weighted sum in each side is seen to approximate a sub-sample time shift in a complementary manner: the closer the modulation signal is to –1 or +1, the larger the discrepancy between the input and output time shifts. The time delay between input and output increases with the length of the allpass chain, and because the modulation signal is time varying, frequency modulation is produced. The shape of the produced FM was noted to be non-sinusoidal. This novel synthesis method called "Coefficient Modulation" (CM) generalizes the methods that produce FM by modulating the output tap position of an interpolated unit delay-based variable length delay line [Van Duyne and Smith, 1992; Lazzarini et al., 2007]. In CM, the output tap position modulation is done explicitly, since it is an inherent property of the method. When CM is not modulated, the allpass chain acts as a dispersive, or as a special case, conventional non-dispersive delay line.

CM has a characteristic timbre which differs from the other modulation-based methods described in Section 2.4. Because of its roots in SDF, the produced timbres have a unique spectral effect-like quality, which are well suited for thick evolving pad and rich source material synthesis. The synthesis parameters of CM include input and modulation signal frequencies f_x and f_m, the modulation index M, and the length of the allpass chain N. The spectral structure is controlled by f_x and f_m, while M defines the bandwidth of the spectrum within the limits imposed by N. CM may also be used in waveguide constructs to produce similar chorus-like fattening observed in [Van Duyne and Smith, 1992], [Stilson, 1995], and [Kleimola, 2008b].

2.6.2 First- and second-order recursive PLTV filters

Linear time-invariant (LTI) system theory cannot be applied directly in the study of filters with time-varying coefficients. Instead, Publications P-4 and P-5 employed periodically linear time-variant (PLTV) filter theory to explore the properties of first- and second-order IIR filters when their coefficients are modulated at audio rates. PLTV[5] filter theory is thoroughly investigated in [Cherniakov, 2003], which lays out a unified framework that converges related research on time-varying systems, as reviewed by Donskoi and Cherniakov [1999]. The framework focuses on systems where the coefficient variation is periodic and assumes further that the period is fixed to a multiple of the sampling period. The periodicity constraint simplifies the derivation of the main characteristics of a PLTV system, since the impulse response, generalized transfer function, and generalized frequency response are thereby periodic as well.

To the knowledge of the authors, Publication P-4 is the first work that explores Cherniakov's PLTV framework in audio synthesis contexts. P-4 interpreted the first-order feedback amplitude modulation (FBAM) oscillator discussed in Section 2.4.2 as a recursive PLTV filter by rewriting Eq. (12) as

$$y(n) = x(n) + \beta a(n) y(n-1)\,, \tag{18}$$

where β is the modulation index, $x(n)$ is the input, and $a(n)$ is the modulation signal, with $x(n) = a(n) = \cos(\omega_0 n)$ and $\omega_0 = 2\pi f_0/f_s$. According to Cherniakov [2003, p.50], a PLTV filter has a 2-D impulse response $h(m,n)$, because the system output depends on the moment of observation n and the moment of input signal application m. Additionally, since the impulse response (IR) of a resursive PLTV filter is periodic, it is computable with a finite number of calculations. For a first-order recursive PLTV filter, the IR is given by [Cherniakov, 2003, pp.123-124]

$$h(m,n) = \begin{cases} 1, & m = n \\ \prod_{k=m+1}^{n} a(k) = \dfrac{g(n)}{g(m)}, & m < n, \\ 0, & n < 0, m > n. \end{cases} \qquad g(n) = \prod_{k=1}^{n} a(k) \tag{19}$$

[5] Acronym PLTV denotes Cherniakov's framework; related research refers to LPTV.

Figure 9 depicts the 2-D IR of the filter given in Eq. (18) for one period of modulation ($N = P_0 = 25$). For better visualization, the cosine modulator and the diagonal time axis (dashed) run from front to back.

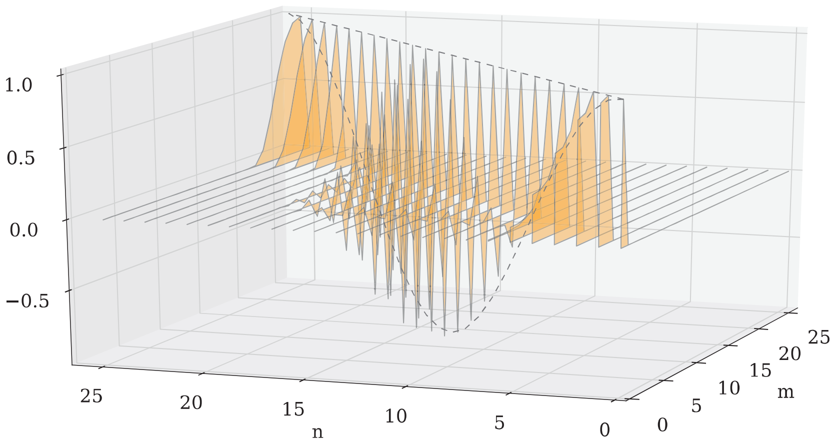


Fig. 9. FBAM impulse response for one period of modulation ($N = 25$, $f_0 = 1764$ Hz, $\beta = 1$).

The IR of a stable system should decrease in time, which happens when $|\, g(N) \,| < 1$, i.e., when the product of the instantaneous coefficient values over a period is less than unity [Cherniakov, 2003, p. 125]. Thus, individual coefficient values may exceed one, which is a more relaxed constraint than the one posed for a stable LTI system. This was observed to be indeed the case, since the FBAM system remained stable even when the modulation index β approached 2. For FBAM, the stability condition is given by

$$\left| \beta \prod_{n=0}^{N\text{-}1} \cos(2\pi n \frac{f_0}{f_s}) \right| < 1, \tag{20}$$

and therefore, the stability is frequency dependent. The approximate stability limit for the first-order FBAM system was concluded to be given by $\beta_{stable} \approx 1.9986 - 0.00003532(f_0 - 27.5)$, which agrees with observations. However, the practical limit for β is determined by the tolerable level of aliasing. In a related work, Berdahl and Harris [2010] noticed that inserting a time-variant block (e.g., a frequency shifter) into the feedback path of an amplification system increases the maximum stable gain of the system significantly.

In the steady state, the time-varying generalized transfer function (GTF) of a recursive first-order PLTV filter is given by [Cherniakov, 2003, pp. 126-127]

$$H(z,n) = \frac{\sum_{k=0}^{N-1} h(n-k,n) z^{-k}}{1 - g(N) z^{-N}}. \tag{21}$$

Thus, the coefficient modulated first-order IIR filter may be represented as an equivalent structure that cascades a FIR filter of order $N - 1$ with a constant-coefficient IIR filter. The GTF of Eq. (20) is periodic in N, and as a special case, setting $N = 1$, reduces Eq. (21) to the well-known transfer function of the first-order IIR with a static coefficient, i.e., $H(z) = 1/(1 - az^{-1})$.

The frequency domain characteristics of the PLTV system are described by the time-varying generalized frequency response (GFR) and the bifrequency function (BF). The GFR of a recursive first-order PLTV filter is obtained from Eq. (20) with the substitution $z = e^{j\omega}$ giving

$$H(\omega,n) = \frac{\sum_{k=0}^{N-1} h(n-k,n)e^{-jk\omega}}{1-g(N)e^{-jN\omega}}. \tag{22}$$

The GFR represents the system response to an analytic input signal $e^{j\omega m}$ with frequency $\omega = 2\pi f_0/f_s$ [Cherniakov, 2003, p. 56]. It should be pointed out that Eq. (22) does not describe the spectrum of the output signal, as is the case with frozen time approximations, which perform poorly when coefficients are updated rapidly. Instead, PLTV theory employs the bifrequency function $H(\omega',\omega)$ to transform the input spectrum components $X(\omega)$ into the output spectrum $Y(\omega')$ [Cherniakov, 2003, pp. 56-57]. The bifrequency function is the 2-D discrete time Fourier transform of the IR $h(m,n)$, given by

$$H(\omega',\omega) = \sum_{n=-\infty}^{\infty} \sum_{m=-\infty}^{\infty} h(m,n)e^{j(\omega' m-\omega n)} = \sum_{n=-\infty}^{\infty} H(\omega,n)e^{jn(\omega'-\omega)}. \tag{23}$$

To verify the theory, for the first-order FBAM with the sinusoidal input $\cos(\omega_0 n)$, $g(N) = a_N$, and the impulse response $h(m,n)$ shown in Fig. 9, the system response can be reconstructed as amplitude modulation $A(n) = |H(\omega_0,n)|$ and phase modulation $p(n) = \arg[H(\omega_0,n)]$. Figure 10(a) compares the GFR reconstruction $y(n) = A(n)\cos(\omega_0 n + p(n))$ and the GTF reconstruction of Eq. (21) with the first-order FBAM signal from difference Eq. (18). As can be seen, the signals match each other. The GFR reconstruction represents steady state and does not therefore contain the initial transient.

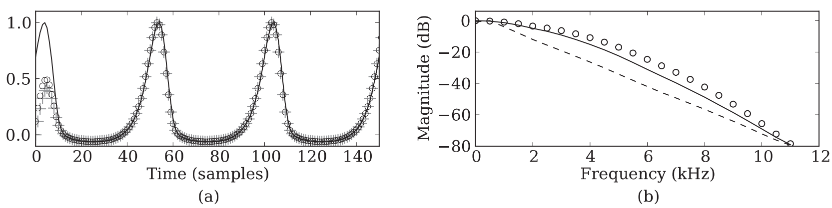


Fig. 10. (a) Normalized FBAM signal (crosses, f_0 = 441 Hz), its GTF (circles) and GFR (solid) reconstruction, and (b) FBAM spectral envelope (solid, $\beta = 1.62$, $f_0 = f_m$ = 500 Hz) compared to the spectral envelope of ModFM (circles, $k = 25$) and PAF (dashed, $\delta = 0.164$).

The scaling of FBAM is more challenging than in FBFM, where output normalization is performed automatically by the sinusoidal waveshaping of the carrier. However, the scaling issue was resolved by using approximate peak scaling based on lookup tables or using an alternative generalized method based on a RMS power balancing algorithm.

The phase modulation component of the FBAM GFR reconstruction had only a minor effect on the produced timbre, and as expected, FBAM is more related to the exponential complex modulator RM techniques such as PAF [Puckette, 1995] and the more recent ModFM [Timoney et al., 2008; Lazzarini and Timoney, 2010b]. FBAM shares the non-oscillating nature of the ModFM spectral evolution, while the steady-state decay rate of the FBAM spectral envelope was observed to be between PAF and ModFM (see Fig. 10(b)). The bandwidth of the FBAM spectrum can be increased by adding partials to the modulation signal, e.g., by raising it to the third power, as proposed in P-4. The cost of this is two multiplications per output sample, which is feasible since the computational load of FBAM compares favorably with PAF and ModFM.

However, FBAM, PAF, and ModFM are complementary methods whose timbral spaces overlap only in part. Publication P-4 and its preliminary investigation [Lazzarini et al., 2009b] extended the basic FBAM timbral palette with several variations, including feed-forward and allpass configurations, heterodyning, waveshaping, nonunitary feedback periods, and separation of the input and modulation signals. The last enables arbitrary modular FBAM constellations similar to the operator algorithms of commercial FM implementations [Chowning and Bristow, 1986; Kim and Park, 2010] and complex carrier and modulator setups explored in the related FM literature (see Section 2.4.1). P-4 demonstrated the flexibility of the method in the following applications: elementary subtractive synthesis without filters, formant synthesis, abstract physical modeling, and adaptive audio effect generation.

The versatile extensions, coupled with the naturally decaying dynamic spectral envelope, direct binding between the modulation index and the spectral brightness, and the increased understanding of the properties of the technique make FBAM a prominent candidate for future automatic parameter matching exploration.

Publication P-5 provided the initial investigation into the second-order FBAM form and its extensions. The PLTV formulation of the basic second-order equation rewrites Eq. (18) as

$$y(n) = x(n) + \beta_1 a(n) y(n-1) + \beta_2 a(n) y(n-2), \tag{24}$$

where β_1 and β_2 are the modulation indices. The bandwidth of the basic first-order FBAM equation and its variations increased with the added unit delay stage. The extra stage enabled novel extensions as well, such as those based on a second-order LTI resonator structure [Steiglitz, 1996, pp. 89-96]. Direct modulation of the pole angle produced rich spectra but posed stability problems especially when the radius approached unity. Keeping the Q factor constant and modulating the center frequency resulted in a more stable filter. Further experiments were performed with the Chamberlin state variable filter's lowpass topology [Chamberlin, 1985, pp. 489-491; Dattorro, 1997], modulating its center frequency while, again, maintaining a constant Q factor. This setup produces a steep spectral envelope with an adjustable format region.

2.7 Aliasing suppression

Nonlinear shaping and audio-rate modulation techniques distort a simple waveform to generate a rich spectrum. This process may also produce unwanted distortion in terms of aliasing artefacts, which come forth when the generated partials exceed the Nyquist limit, or when they enter into the negative frequency domain at 0 Hz. Such partials fold back into the audio band as aliased components. The folded components may be subsequently reflected at 0 Hz and the Nyquist limit, producing the second, third, etc. generation of aliasing, thereby contaminating the signal with unpleasant noise. It should be noted, however, that the first generation of aliasing around 0 Hz is sometimes considered a desired feature, since it enables the generation of inharmonic spectra [Chowning, 1973; Välimäki and Huovilainen, 2007].

Anti-aliasing algorithms strive to minimize the amount of the undesired aliasing noise and therefore focus on folding at the Nyquist limit. Välimäki and Huovilainen [2007] present a taxonomy that categorizes the anti-aliasing solutions into bandlimited, quasi-bandlimited, and spectral tilt modification algorithms. Pekonen and Välimäki [2008a] introduced later a fourth category of anti-aliasing algorithms, where digital filtering is applied to aliasing-contaminated input signals.

Bandlimited algorithms avoid aliasing entirely by producing partials only below the Nyquist limit. This can be achieved using additive synthesis, and in the context of the present section, with classic polynomial waveshaping and specific forms of DSF. Nam et al. [2009] describe an inharmonic oscillator based on a feedback delay loop and an impulse train excitation.

2.7.1 Quasi-bandlimited algorithms

Quasi-bandlimited algorithms produce signals where the amount of aliasing is low and mainly located at high frequencies where human perception is less sensitive [Pekonen et al., 2010a; Lehtonen et al., 2012]. For an overview of such methods, consider first the impulse train, which is the derivative of a continuous time sawtooth signal where each impulse corresponds to the discontinuity of the sawtooth. Methods based on bandlimited impulse trains (BLITs) [Stilson and Smith, 1996] apply anti-aliasing to the discrete time unit impulses and then integrate the result to generate sawtooth (unipolar BLIT) or pulse and triangular (bipolar BLIT) waveforms with reduced aliasing. Non-uniform impulse sequences have uses in PWM, hardsync and supersaw implementations [Nam et al., 2010].

Approximately bandlimited impulse trains are created using lowpass filtering [Stilson and Smith, 1996; Nam et al., 2010; Pekonen et al., 2010b] or abstract algorithms [Timoney et al., 2008; Winham and Stieglitz, 1970]. An alternative BLIT formulation based on IIR filters is described in [Tassart, 2013].

Certain BLIT-based methods are computationally intensive. For example, Schimmel [2012] concluded that for perceptually alias-free synthesis, a hybrid scheme comprising a 64-fold oversampled trivial sawtooth for low and lookup table-based additive synthesis for high fundamental frequencies (threshold $f_0 = 587$ Hz) is computationally more efficient than BLIT-SWS [Stilson and Smith, 1996].

The leaky integration step of BLIT methods may be performed in advance before sample computation time. This is used in another class of quasi-bandlimited algorithms, which approximate the ideal bandlimited step function (BLEP), i.e., the sine integral [Välimäki et al., 2012], using an accumulated windowed sinc [Brandt, 2001] or efficient low-order polynomial functions [Välimäki and Huovilainen, 2007]. A correction function is then prepared by subtracting an ideal unit step from the approximation. At sample computation time, the correction function is mixed with the aliasing-contaminated input signal when discontinuities are observed in the input, i.e., the method is applicable to any signal with discontinuities. Välimäki et al. [2012] derived higher order polynomial correction functions from integrated Lagrange and B-spline interpolators, which were further optimized by Pekonen et al. [2012].

Publication P-2 introduced a post-processing algorithm to reduce aliasing of single inflection point VPS waveforms, which is related to the half-cosine interpolation [Mitsuhashi, 1982]. In VPS, the non-integral vertical inflection point parameter v generates an incomplete period, which in turn produces a discontinuity in the output signal. The proposed algorithm modifies the phaseshaper when the phase is inside the incomplete period so that incomplete segment is rendered as a scaled and offset full-cycle sinusoid. The asymmetry of the full-cycle replacement is controlled by the horizontal inflection point parameter d, which provides trade off adjustment between the amount of aliasing and suppression of high frequency spectral energy.

Figure 11(a) shows the output of a single oscillator half-cosine hard-sync algorithm of Publication P-1, using an elementary phaseshaper $P(x) = 1.25x$ and a trivial half-cosine WG without aliasing suppression. The discontinuities at the simulated slave oscillator wrap-around point and at the point of synchronization produce excessive amount of aliasing, as shown in Fig. 11(b). The corresponding VPS implementation using the aliasing suppression method of Publication P-2 is shown in Figs. 11(c) and 11(d). As can be seen, the aliasing has been suppressed considerably. The trade-off between the amount of aliasing and the high-frequency spectral energy is thus controlled using the horizontal inflection point parameter d, which here is $d = 0.85$ at the wrap-around and $d = 0.7$ at the synchronization point.

In a related quasi-bandlimited phaseshaping application, Lazzarini and Timoney [2010a] reduced aliasing of sawtooth and pulse waveforms by employing bandlimited phaseshapers.

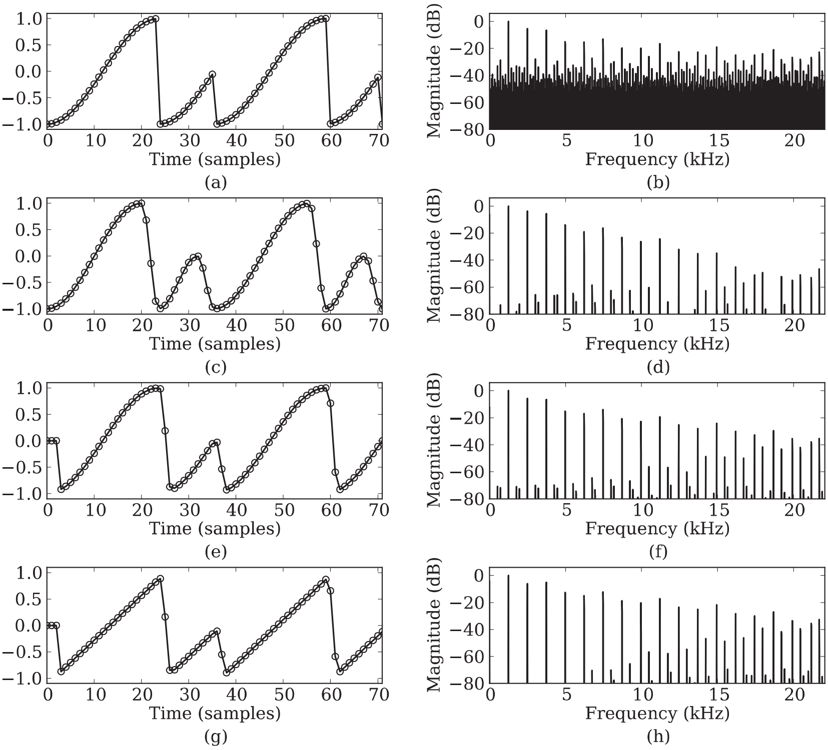


Fig. 11. Waveform and spectrum of the single oscillator half-cosine hard sync algorithm, f_0 = 1245 Hz and $f_{\mathrm{SLAVE}} = 1.25\, f_0$. (a, b) Trivial, (c, d) VPS implementation, (e, f) PTR implementation with W = 3, and (g, h) corresponding PTR sawtooth hard sync algorithm.

2.7.2 Algorithms based on spectral tilt modification

The anti-aliasing algorithms in this class first generate a signal that has an identical set of harmonics but a steeper spectral slope than the desired target signal. Filtering is applied afterwards to bend the slope towards that of the target. Lane et al. [1997] distorted sine waveforms with absolute value waveshapers and applied lowpass and highpass filtering to approximate sawtooth, triangle, and pulse waveforms. Välimäki [2005], Huovilainen and Välimäki [2005], and Välimäki and Huovilainen [2006] utilized a second-order piecewise parabolic waveform for similar purposes.

The differentiated polynomial waveform (DPW) algorithm extends the latter method into higher polynomial orders N [Välimäki et al., 2010]. The DPW algorithm operates in four stages as follows. First, the unipolar phase signal $\phi(n)$ is phaseshaped into a trivial bipolar sawtooth signal $s(n)$. The sawtooth is then waveshaped using an N^{th}-order polynomial transfer function s_N to produce a signal that decays about $6N$ dB per octave. The transfer functions s_N are formed from successive integrals of $s(n)$, e.g., $s_2 = s^2$, $s_3 = s^3 - s$, and $s_4 = s^4 - 2s^2$. In the third stage, the waveshaped signal is reverted to the sawtooth-like shape by differentiating it $N - 1$ times. This stage restores the spectral tilt towards that of the sawtooth. The differentiated signal is finally multiplied with a scaling factor to recover the original signal level. The alias-suppressed sawtooth signal is thus given by

$$y(n) = \frac{P_0^{N-1}\nabla^{N-1}s_N\left[2\phi(n)-1\right]}{2^{N-1}N!}, \tag{25}$$

where ∇^{N-1} is the backwards-difference operator applied $N - 1$ times.

Publication P-6 analyzed the time-domain properties of the DPW algorithm and noticed that although all samples go through the processing steps given by Eq. (25), DPW only modifies samples in a finite region around each discontinuity which is $W = N - 1$ samples wide. The rest of the samples are simply offset from the trivial sawtooth waveform $s(n)$. This observation led to the definition of the polynomial transition region (PTR) algorithm, which is described as

$$y(n) = \begin{cases} p_W(n) - c_{\mathrm{dc}}, & \text{when } \phi(n) < WT_0 \\ x(n) - c_{\mathrm{dc}}, & \text{when } \phi(n) \geq WT_0, \end{cases} \tag{26}$$

where $p_W(n)$ [see P-6, Table I] is the transition polynomial that replaces the aliasing-contaminated input signal $x(n)$ when the phase counter $\phi(n)$ is within the transition region, and c_{dc} is an offset.

The proposed PTR algorithm has three advantages over the DPW method. First, it provides substantial savings in computational cost, because outside the transition region, which is usually substantially wider than W, the DPW algorithm reduces to a modulo counter. PTR also compares favorably to the previous BLIT-FDF [Nam et al., 2010] and the second-order PolyBLEP [Välimäki and Huovilainen, 2007] methods as demonstrated in P-6. Second, PTR enables transient-free response to rapid input signal changes, because it does not employ the differentiator state variables that are required in DPW. Third, PTR scales to arbitrary transition heights and is applicable to other waveforms besides the trivial sawtooth, even to those that are nondifferentiable.

Figures 11(e) and 11(f) apply PTR to a hard-synced half-cosine waveform ($W = 3$), using transition polynomials that are optimized for a sawtooth waveform. As can be seen, the PTR algorithm preserves more of the high-frequency spectral energy than the method introduced in Publication P-2. Figures 11(g) and 11(h) show that PTR transition polynomials that are optimal for sawtooth waveforms are even more effective in suppressing the low-frequency aliasing noise.

Figure 12 compares the optimal PTR transition polynomials for a rising edge sawtooth signal (solid) to the delayed integrated B-spline interpolation polynomials (crosses) given in [Välimäki et al., 2012]. As can be seen, the polynomials match. Therefore, the DPW method of Välimäki et al. [2005, 2010] – which was used to derive the depicted PTR transition polynomials – and the PolyBLEP method based on integrated B-spline interpolation polynomials share identical aliasing suppression properties. However, the PTR form of DPW is computationally more efficient than the equivalent PolyBLEP implementation [P-6].

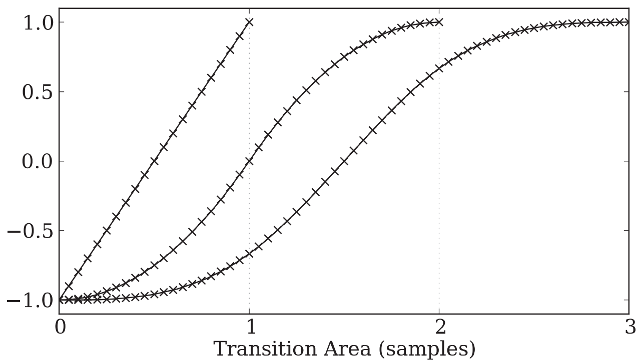


Fig. 12. PTR transition polynomials (solid) of orders 1, 2, and 3 and the corresponding delayed integrated B-spline interpolation polynomials (crosses).

3. Performative modulation

The sound synthesis methods described in Section 2 provide a flexible sonic palette for DMI design. Gestural control of the exposed synthesis parameters enable dynamic spectra and smooth morphing between timbres. The DMI design challenge then is to devise interaction models that approach the direct and intimate, yet powerful and expressive performative properties of traditional acoustic musical instruments (AMIs).

The human computer interaction (HCI) field provides paradigms that are useful when designing methods for controlling the multi-dimensional parameter space of the described sound synthesis methods [Wanderley and Orio, 2002]. For example, the classic linguistic model by Foley and Wallace [1974] introduced a multi-layered design methodology that separates lexical, syntactic, semantic, and conceptual levels of interaction. This helps in structuring the multi-level mapping mechanisms when performer's low level gestures and high level intention are to be transformed into timbral variances of the synthesized sound. Furthermore, the direct manipulation technique [Schneiderman, 1983], where the syntactic layer of interaction is kept minimal, has a parallel in the intimate timbral control prevalent in AMIs. The recent natural user interaction (NUI) paradigm, where the user interface adapts to exploit the existing skill set of the user [Buxton, 2010; Blake, 2011], unfolds into environments that support discovery and learning ever more powerful means of interaction. This is closely related to the process of mastering a musical instrument through practice.

Wanderley and Depalle [2004] present a DMI model (see Fig. 2) where the sound synthesis and the gestural controller units are separate independent entities, which are bound together via mapping strategies. According to Wanderley and Depalle, a successful DMI design balances the analysis of both independent units. Since the units are separated, however, the mapping strategy between performer gestures and the sound synthesis parameters becomes of paramount importance. A common language and a communication channel to mediate the mappings are therefore required. These issues are addressed by the Musical Instrument Digital Interface

(MIDI) [MMA, 1983] and the Open Sound Control (OSC) [Wright and Freed, 1997; Freed and Schmeder, 2009] de facto standards.

MIDI events are mostly controller-centric, i.e., the specification predefines a fixed set of controller-specific message types which carry instantaneous controller values with 7 or 14 bit resolution. Despite criticism concerning the low bandwidth, temporal imprecision [Loy, 1985; Moore, 1988], insufficient parameter resolution, and limited extensibility, MIDI is still the most widely used communication protocol between the controller and sound synthesis unit.

OSC is a content format where each time-stamped message consists of a string-format address part and a typed data vector. The address string identifies the source or target of the message in a human-readable form (e.g., "/synth/LPF/cutoff"), while the data vector carries the parameter values with improved resolution. Unlike MIDI, OSC does not predefine message types, so the address spaces are entirely application specific. The physical transmission layer is likewise undefined, although OSC messages are usually transmitted as UDP/IP packets. The benefits of OSC include intuitiveness, extensibility, improved resolution, and a stateless communication paradigm that simplifies the setup procedure. These advancements have found uses especially in experimental DMI and platform independent NUI implementations, where OSC has gained a de facto status as TUIO-based multi-touch applications [Kaltenbrunner et al., 2005].

Fraietta [2008] argued that the intuitive address strings are inefficient since the computers, which carry out the transmission and parsing actions, are more adapted for numerical representations such as MIDI. Fraietta did not provide a detailed solution, but suggested that the internet name resolution or the local scope address resolution protocol could provide a basis for an improvement. However, this would break the compatibility with the current OSC implementations.

Publication P-7 proposes a detailed backwards-compatible improvement that decodes the OSC address string into a single integer and transmits the decoded token inside the data vector of the message (see Fig. 13). The target of the message may then perform the parsing operation in the more efficient integer representation, or encode the token back into the intuitive string form for human patching. The decoding process is performed either in the controller or in a centralized hub service residing in the network. Three widely used OSC toolkits were modified so that existing applications are able to take advantage of the improved efficiency with minimal reimplementation effort. Backwards compatibility with the existing OSC standard is achieved by utilizing empty address strings ("/") to denote that the message is in the proposed decoded format, and an empty data vector to request the address string – integer decoding mappings from the end points.

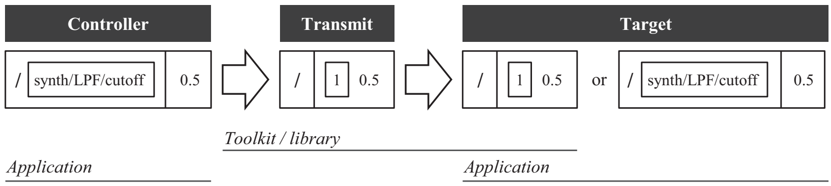


Fig. 13. The proposed data reduction technique (solid outlines). The address string of the standard OSC message (left) is decoded into integer representation and transmitted inside the data vector (middle), and finally, processed in the transmitted or encoded form in the target (right).

The proposed method has several benefits. First, the amount of transmitted data is reduced dramatically, because the address part is transmitted repeatedly with each parameter update. Thus, the bandwidth is not wasted and there is more room for the actual content. Second, the parsing operation at the receiving end is simplified and becomes independent of the length of the address string. This leaves more CPU time for the actual sound synthesis algorithms, while guaranteeing lower and unified parsing latencies across the entire address space. Third, the controllers and targets are able to publish their entire address spaces, which enables centralized mapping and routing strategies [Malloch et al., 2008]. These benefits are especially useful for resource constrained devices where CPU and power consumption is of great importance, and when the transmission channel is over a wireless medium where the risk of dropped packets is high. The exact performance measurements of the proposed method are complicated by multi-dimensional properties of the problem space, and were therefore left for future work.

The downside of the method is that the stateless nature of OSC messages is not fully preserved. However, content formats with fixed address spaces (e.g., TUIO) can provide a priori mappings at the specification level, which makes messages self-contained albeit still not truly stateless.

Finally, Publication P-7 proposes two additional enhancements for inclusion in the next major OSC version: the RESTful paradigm [Fielding, 2000] of the current OSC specification can be extended with additional verbs (e.g., PUT, ADD, DELETE) to augment the implied SET action of the current version, while the Zeroconf protocol [Cheshire and Steinberg, 2005] is adept for service discovery and name resolution tasks.

4. Main results of the thesis

P-1: Phaseshaping oscillator algorithms for musical sound synthesis

Publication P-1 proposes a new framework for pipelined phaseshaping expressions. The framework is based on function composition (Eqs. 8a–b) and semantic phaseshaper entities (Eqs. 9a–d), which are derived from a unipolar modulo counter signal. The entities are organized into a set of reusable elementary phaseshapers (Eqs. 10–19) that can be parametrized and cascaded into more elaborate shaper pipelines.

Several classic oscillator effect algorithms were composed from the proposed set of elementary phaseshapers in order to evaluate their practical applicability. These included ad-hoc sawtooth waveform emulation (Eq. 20), anti-aliasing (Eq. 21), and PWM (Eqs. 25a–b). The compositions lead to novel interpretations as well as to entirely new modular algorithms. These include single-oscillator hard and soft sync (Eqs. 22–24), triangle modulation (Eqs. 26–27), efficient supersaw simulation (Eq. 28), and sinusoidal waveshape modulation effects. The last eventually led to the discovery of the VPS technique described in Publication P-2.

Accompanying webpage: http://www.acoustics.hut.fi/go/smc2010-phaseshaping

P-2: Vector phaseshaping synthesis

PD is an intuitive abstract sound synthesis technique, but its timbral diversity is limited. Publication P-2 generalizes the PD synthesis technique and extends it in four ways to increase its expressive power and to update its interface suitable for modern multi-touch controllers. First, the inflection point is expressed as a 2-D vector (Eqs. 5–6). The added dimension allows, for example, flexible formant synthesis and enables touch-based surface interaction. Second, VPS allows multiple inflection points per cycle (Eq. 19), permitting a highly complex phase signal definition and multi-touch interaction. Third, P-2 proposes an aliasing-suppression technique to manage incomplete wave generator periods (Eqs. 11–12). Fourth, VPS extends the inflection point modulation to audio rates, which enables, among other effects, the generation of inharmonic spectra. Publication P-2 also provides an exact spectral derivation for single-inflection point VPS and its special case, the PD synthesis technique (Eqs. 16–18).

Accompanying webpage: http://www.acoustics.hut.fi/go/dafx11-vps

P-3: Sound synthesis using an allpass filter chain with audio-rate coefficient modulation

Publication P-3 introduces a novel sound synthesis method that is based on allpass filter chains whose coefficients are modulated with audio-rate signals. P-3 looks at abstract sound synthesis methods from a filtering perspective and interprets the introduced method in terms of FBAM (Eqs. 3, 12–13) and complex modulator FM (Eqs. 4, 15–17). The modified direct form I formulation of Eq. 3 was found to synthesize transient-free signals. A parametric synthesis model was constructed, and the synthesis parameters with spectral effects were described (Table 1). The spectral properties of the method were explored as well (Eqs. 5–10). The aliasing resilience of the method was found to be favorable, since the high-frequency spectral slope is steep (Fig. 3) and because the bandwidth is controllable by both the length of the chain and by the amount of modulation. The dispersive qualities of the allpass chain produce unique timbres that are rich and characteristically evolving even for otherwise static sustaining notes. The versatility of the method was shown with several application examples (Section 4).

Accompanying webpage:
http://www.acoustics.hut.fi/publications/papers/dafx09-cm

P-4: Feedback amplitude modulation synthesis

The time-varying digital filter interpretation provides a novel analysis tool for abstract sound synthesis methods. To the authors' knowledge, Publication P-4 is the first work that explores the established PLTV framework in audio synthesis context. P-4 interpreted the first-order FBAM oscillator as a recursive PLTV filter (Eqs. 9–16). The filter perspective provided new insight to the spectral (Eqs. 17–22), stability (Eq. 24), and evaluation (Section 5) aspects of the method. The aliasing (Fig. 7) and scaling (Fig. 9) analysis produced means for practical implementations.

Six variations of the basic FBAM method were explored in detail (Fig. 10). The separation of carrier and modulator signals (Fig. 18) and the operator abstraction (Fig. 19) allow arbitrary modular system constructions, which were demonstrated with several applications (Fig. 20). The versatile extensions, coupled with the naturally decaying dynamic spectral envelope, bandwidth extension (Fig. 24), direct binding between the modulation index and the spectral brightness, and the increased understanding of the properties of the technique make FBAM a prominent candidate for future automatic parameter matching exploration.

Accompanying webpage: http://www.acoustics.hut.fi/go/jasp-fbam

P-5: Aspects of second-order feedback AM synthesis

Publication P-5 continues the analysis of P-4 in second-order IIR filter topologies. The second-order FBAM oscillator was expressed as a PLTV filter (Eq. 7), and the variations were explored in second-order form (Eqs. 8–9). The spectral bandwidth was observed to increase with the added unit delay stage. The extra stage enables novel extensions, of which a second-order resonator (Eqs. 11, 13–14) and the Chamberlin state variable filter topology (Eqs. 15–16) were investigated. The stability of the second-order system is problematic, as was anticipated from the PLTV theory.

P-6: Reducing aliasing from synthetic audio signals using polynomial transition regions

Nonlinear shaping, audio-rate modulation, and virtual analog techniques are not bandlimited, and therefore produce unwanted aliasing artefacts. Publication P-6 analyzed the time-domain properties of an established aliasing-suppression algorithm (DPW) in Eqs. 4–7, and provided a novel interpretation of the method (Eq. 8, Table I). The proposed PTR algorithm has three advantages over the DPW method. First, it provides substantial savings in computational cost and compares favorably with all other established aliasing-suppression algorithms in terms of computational efficiency (Fig. 2, Table II). Second, PTR enables a transient-free response to rapid input signal changes (Fig. 3). Third, PTR scales to arbitrary transition heights and is applicable to other waveforms besides the trivial sawtooth, even to those that are nondifferentiable (Fig. 4 and accompanying webpage).

Accompanying webpage: http://www.acoustics.hut.fi/go/spl-ptr

P-7: Improving the efficiency of Open Sound Control with compressed address strings

OSC has become the de facto standard in experimental DMI, artistic installation, and platform independent multi-touch applications. Publication P-7 proposes a detailed backwards-compatible solution that reduces the amount of transmitted data dramatically by encoding the (long) address strings as integer tokens (Fig. 2). The proposed method has several benefits. First, the bandwidth of the transmission medium is not wasted, and there is more room for the actual content. Second, the parsing operation at the receiving end is simplified. This reduces latency and leaves more CPU time for the actual sound synthesis algorithms. Third, the controllers and targets are able to publish their entire address spaces, which enables centralized mapping and routing strategies. These benefits are especially useful for resource constrained devices where CPU and power consumption issues are of great importance and when the transmission channel is over a wireless medium where the risk of dropped packets is high.

Three widely used OSC toolkits were modified so that existing applications are able to take advantage of the improved efficiency with minimal reimplementation efforts (Sections 3.2–3.4). P-7 also suggests two enhancements for inclusion in the next major OSC version: addition of RESTful verbs for increased action space and the adaption of the Zeroconf protocol for service discovery and name resolution tasks.

Accompanying webpage: http://www.acoustics.hut.fi/go/smc2011-osc

5. Conclusion and future directions

This thesis explores abstract synthesis techniques in the implementation of digital musical instruments, with specific focus on nonlinear distortion and audio-rate modulation. The techniques are approached from two complementary perspectives. First, the classic view, which is based on closed form mathematical expressions, is seen to converge into a pipelined compound model where different abstract synthesis methods both generalize and reinforce each other. The frontmost component of the compound pipeline, i.e., the recently coined phaseshaping technique is investigated in detail in the form of piecewise linear phase signal transformations. In particular, the concept of nested phaseshaping developed in P-1 has led to efficient novel oscillator effect implementations, and further to the discovery of the generalized 2-D multipoint phase distortion synthesis method called vector phaseshaping synthesis introduced in P-2. The latter provides intuitive parametrization of the complex modulator PM technique and is well-suited for contemporary multi-touch controller applications.

The second perspective approaches abstract sound synthesis techniques from a time-varying filter angle. The thesis demonstrates that the filter approach leads to novel synthesis methods, such as the coefficient-modulated allpass filter chain introduced in P-3. This method shares the familiar parameters of classic audio-rate modulation techniques, but produces a unique timbral space that is different from the established methods. Publications P-4 and P-5 employ PLTV filter theory, which has not been extensively utilized in previous research, to gain new insight on the properties of the feedback amplitude modulation technique.

Since the synthesis methods explored in this thesis produce rich and generally non-bandlimited spectra, aliasing suppression is investigated as an adjacent post-processing stage. Two algorithms are proposed. Publication P-2 employs scaled and offset sinusoids to reduce aliasing generated by incomplete phaseshaping periods, while P-6 describes a new efficient and general polynomial interpolation technique that is applicable to any signal. The polynomial method called PTR is seen to provide better results, and it is currently the most efficient method for implementing anti-aliased virtual analog oscillators. However, the proposed sinusoidal technique has potential to become valuable in future research.

Finally, the thesis addressed the performative aspects of the explored sound production mechanisms by interfacing each proposed method with a set of synthesis parameters. In addition, Publication P-7 proposed a streamlined control protocol to mediate the mappings between performance controllers and the synthesis parameters.

Three potential future research directions conclude this thesis. Classic abstract sound synthesis techniques are computationally efficient and require only one or two oscillators to generate rich and dynamic spectra. The timbral space of such simple oscillator setups is surprisingly wide, but can still be made more versatile by increasing the complexity of the system. Future research should take advantage of the increased computing power of modern devices, and instead of concentrating extensively on the efficient implementation of synthesis algorithms, also consider models that are less efficient to implement. Extended parameter spaces can be reached, for example, using the compound oscillator model presented in this work. Further expansions are obtained by arranging the compound oscillators into multi-carrier and multi-modulator setups, exploring complex but more flexible equations, and investigating hybrid synthesis algorithms that combine abstract and, for instance, physical modeling-based techniques.

The number of synthesis parameters increases with the complexity of the system. This provides more precise control over the produced spectra, but unfortunately also generates more dense control streams. In classic additive synthesis, the amount of control data is reduced by approximating the time-varying harmonic weights with piecewise linear envelope functions. Similar envelopes, when applied to the phase signal of the VPS oscillator of P-2, produce more profound timbral effects. In addition, when the VPS oscillator is modified to employ half-cosine interpolated breakpoints, a single VPS oscillator can replace a cascade of PM modulators: instead of building complex modulation signals by adding multiple parametrized oscillators, the VPS method adds mutable breakpoints. Future research on automatic parameter matching algorithms may then target the breakpoint set [Vairetti, 2012]. However, multi-point and multi-carrier VPS setups are required for best results.

The graphical representation of the VPS breakpoints provides an intuitive parametrization of complex modulator PM. 2-D visualization of other abstract algorithms can provide similar benefits. For example, parametric equations with two variables spread naturally into the 2-D plane. Sampling the trajectory of the continuous-time graph in equal-angle or equidistance increments may then expand into a new class of abstract sound synthesis techniques. The starting point of future research in this direction lies in epicycloids, spirographs, and Gielis' supershapes.

Bibliography

Arfib, D., 1979. Digital synthesis of complex spectra by means of multiplication of non-linear distorted sine waves. *J. Audio Eng. Soc.*, 27(10), pp. 757–779.

Bank, B., 2009. Energy-based synthesis of tension modulation in strings. In: *Proc. Int. Conf. Digital Audio Effects*, Como, Italy, Sept. 2009, pp. 365–372.

Beauchamp, J., 1982. Synthesis by spectral amplitude and "brightness" matching of analyzed musical instrument tones. *J. Audio Eng. Soc.*, 30(6), pp. 396–406.

Beauchamp, J. and Horner, A., 1992. Extended nonlinear waveshaping analysis/synthesis technique. In: *Proc. Int. Computer Music Conf.*, San Francisco, CA, USA, Mar. 1992, pp. 2–5.

Berdahl, E. and Harris, D., 2010. Frequency shifting for acoustic howling suppression. In: *Proc. Int. Conf. Digital Audio Effects*, Graz, Austria, Sept. 2010, pp. 101–104.

Blake, J., 2011. *Natural User Interfaces in .NET*. New York, NY: Manning Publications Co.

Bode, H. and Moog, R., 1972. A high accuracy frequency shifter for professional audio applications. *J. Audio Eng. Soc.*, 20(6), pp. 453–458.

Boulanger, R. ed., 2000. *The Csound Book: Perspectives in Software Synthesis, Sound Design, Signal Processing, and Programming*. Cambridge, MA: MIT Press.

Boulanger, R. and Lazzarini, V. eds., 2010. *The Audio Programming Book*. Cambridge, MA: MIT Press.

Brandt, E., 2001. Hard sync without aliasing. In: *Proc. Int. Computer Music Conf.*, Havana, Cuba, Sept. 2001, pp. 365–368.

Buxton, B., 2010. *Natural user interfaces* interviewed by Larry Larsen [video] Channel 9, 6 Jan. 2010. Available at: http://channel9.msdn.com/posts/LarryLarsen/CES-2010-NUI-with-Bill-Buxton [Accessed 27 Jan. 2013].

Casio, 1984. *Casio CZ Series Sound Synthesis Handbook*, Tokyo: Casio Computer Co., Ltd.

Chamberlin, H. 1985. *Musical Applications of Microprocessors*. 2nd ed. Hasbrouck Heights, NJ: Hayden Book Company.

Cherniakov, M., Sizov, V.I. and Donskoi, L., 2000. Synthesis of a periodically time-varying digital filter. *IEE Proc.–Vis. Image Signal Process.*, 147(5), pp. 393–399.

Cherniakov, M., 2003. *An Introduction to Parametric Digital Filters and Oscillators*. New York, NY: John Wiley & Sons.

Cheshire, S. and Steinberg, D., 2005. *Zero Configuration Networking: the Definitive Guide*. Sebastopol, CA: O'Reilly Media.

Chowning, J., 1973. The synthesis of complex audio spectra by means of frequency modulation. *J. Audio Eng. Soc.*, 21(7), pp. 526–534.

Chowning, J. and Bristow, D., 1986. *FM Theory and Applications*. Tokyo: Yamaha Music Foundation.

Chowning, J., 1989. Frequency modulation synthesis of the singing voice. In: Mathews, M. and Pierce, J.R. eds., *Current directions in computer music research*. Cambridge, MA: MIT Press, pp. 57–63.

Collins, N., 1999. SplineSynth: an interface to low level digital audio. In: *Proc. Diderot Forum on Mathematics and Music*, Vienna, Austria, 1999, pp. 49–61.

Cook, P., 2002. *Real Sound Synthesis for Interactive Applications*. London: AK Peters.

Cooper, C.M. and Abel, J.S., 2010. Digital simulation of 'brassiness' and amplitude-dependent propagation speed in wind instruments. In: *Proc. Int. Conf. Digital Audio Effects*, Graz, Austria, Sept. 2010, pp. 89–94.

Dannenberg, R.B., 1997. Machine tongues XIX: Nyquist, a language for composition and sound synthesis. *Comput. Music J.*, 21(3), pp. 50–60.

Dannenberg, R.B., 1998. Interpolation error in waveform table lookup. In: *Proc. Int. Computer Music Conf.*, Ann Arbor, MI, USA, Oct. 1998, pp. 240–243.

Dattorro, J., 1997. Effect design, part 1: reverberator and other filters. *J. Audio Eng. Soc.*, 45(9), pp. 660–684.

De Poli, G., 1984. Sound synthesis by fractional waveshaping. *J. Audio Eng. Soc.*, 32(11), pp. 849–861.

Depalle, P., Garcia, G. and Rodet. X., 1993. Tracking of partials for additive sound synthesis using hidden Markov models. In: *Proc. IEEE Intl. Conf. Acoust., Speech, and Signal Process.*, Minneapolis, MN, USA, Apr. 1993, pp. 225–228.

Dodge, C. and Jerse, T., 1997. *Computer Music: Synthesis, Composition, and Performance*. 2nd ed. New York, NY: Schirmer Books.

Donskoi, L. and Cherniakov, M., 1999. Analysis of time-varying digital filter. In: *Proc. IEEE Region 10 Conf. (TENCON 99)*, Silla Cheju, South Korea, Sept. 1999, pp. 658–661.

Essl, G., 2010. UrSound - live patching of audio and multimedia using a multi–rate normed single-stream data-flow engine. In: *Proc. Int. Computer Music Conf.*, New York, NY, USA, June 2010, pp. 534–537.

Evangelista, G., 2000. Real-time time-varying frequency warping via short-time Laguerre transform. In: *Proc. Int. Conf. Digital Audio Effects*, Verona, Italy, Dec. 2000, pp. 7–12.

Fielding, R.T., 2000. *Architectural Styles and the Design of Network-based Software Architectures*. Ph.D. dissertation, Univ. of California, Irvine, USA.

Foley, J. and Wallace, V., 1974. The art of natural graphic man-machine conversation. *Proc. IEEE*, 62(4), pp. 462–470.

Fraietta, A., 2008. Open Sound Control: constraints and limitations. In: *Proc. Int. Conf. on New Interfaces for Musical Expression*, Genova, Italy, June 2008, pp. 19–23.

Freed, A. and Schmeder, A., 2009. Features and future for Open Sound Control version 1.1 for NIME. In: *Proc. Int. Conf. on New Interfaces for Musical Expression*, Pittsburgh, PA, USA, June 2009, pp. 116–120.

Guinee, R., 2007. A novel Fourier series simulation tool for pulsewidth modulation (PWM) in pulsed power systems. In: *Proc. IEEE Symp. Fusion Eng.*, Albuquerque, NM, USA, June 2007, pp. 1–4.

Hoffmann-Burchardi, R., 2008. Digital simulation of the diode ring modulator for musical applications. In: *Proc. Int. Conf. Digital Audio Effects*, Espoo, Finland, Sept. 2008, pp. 165–168.

Hoffmann-Burchardi, R., 2009. Asymmetries make the difference: an analysis of transistor-based analog ring modulators. In: *Proc. Int. Conf. Digital Audio Effects*, Como, Italy, Sept. 2009, pp. 98–101.

Holm, F., 1992. Understanding FM implementations: a call for common standards. *Comput. Music J.*, 16(1), pp. 34–42.

Horner, A., Beauchamp, J. and Haken, J., 1993. Genetic algorithms and their application to FM matching synthesis. *Comput. Music J.*, 17(4), pp. 17–29.

Horner, A., 1998. Nested modulator and feedback FM matching of instrument tones. *IEEE Trans. Audio, Speech Lang. Process.*, 6(4), pp. 398–409.

Huovilainen, A. and Välimäki, V., 2005. New approaches to digital subtractive synthesis. In: *Proc. Int. Computer Music Conf.*, Barcelona, Spain, Sept. 2005, pp. 399–402.

Hutchins, B., 1975. The frequency modulation spectrum of an exponential voltage-controlled oscillator. *J. Audio Eng. Soc.*, 23(3), pp. 200–206.

Hutchins, B. and Ludwig, L., 1980. A new look at pulse width modulation. *Electronotes*, 12(100), pp. 3–17.

Ishibashi, M., Casio Computer Co., 1987. *Electronic musical instrument*. U.S. Pat. 4,658,691.

Kaltenbrunner, M., Bovermann, T., Bencina, R. and Costanza, E., 2005. TUIO - a protocol for table–top tangible user interfaces. In: *Proc. Int. Workshop on Gesture in Human–Computer Interaction and Simulation*, Ile de Berder, France, May 2005.

Karplus, K. and Strong, A., 1983. Digital synthesis of plucked-string and drum timbres. *Comput. Music J.*, 7(2), pp. 43–55.

Kim, T. and Park, I., 2010. Design of a scalable and programmable sound synthesizer. *IEEE Trans. Very Large Scale Integration (VLSI) Systems*, 18(6), pp. 875–886.

Kleimola, J., 2008a. Audio synthesis by bitwise logical modulation. In: *Proc. Int. Conf. Digital Audio Effects*, Espoo, Finland, Sept. 2008, pp. 67–70.

Kleimola, J., 2008b. Dispersion modulation using allpass filters. In: *Proc. Int. Conf. Digital Audio Effects*, Espoo, Finland, Sept. 2008, pp. 193–197.

Korg, 2011. *Kronos Parameter Guide*. Tokyo: Korg Inc. Available at: http://www.korg.com/uploads/Support/USA_KRONOS_Param_Guide_E5.pdf [Accessed 27 Jan. 2013].

Lai, Y., Jeng, S., Liu, D. and Liu, Y., 2006. Automated optimization of parameters for FM sound synthesis with genetic algorithms. In: *Int. Workshop on Computer Music and Audio Technology*, Taipei, Taiwan, Mar. 2006, pp. 33–38.

Lane, J., Hoory, D., Martinez, E. and Wang, P., 1997. Modeling analog synthesis with DSPs. *Comput. Music J.,* 21(4), pp. 23–41.

Lang, B., 2004. Waveform synthesis using Bezier curves with control point modulation. In: *Proc. 116th AES Convention*, Berlin, Germany, May 2004.

Layzer, A., 1971. Some idiosyncratic aspects of computer synthesized sound. In: *Proc. 6th Annual ASUC Conf.*, New York: American Society of University Composers, pp. 27–39.

Lazzarini, V., Timoney, J. and Lysaght, T., 2007. Adaptive FM synthesis. In: *Proc. Int. Conf. Digital Audio Effects*, Bordeaux, France, Sept. 2007, pp. 21–26.

Lazzarini, V., Timoney, J. and Lysaght, T., 2008a. Split-sideband synthesis. In: *Proc. Int. Computer Music Conf.*, Belfast, N. Ireland, Aug. 2008.

Lazzarini, V., Timoney, J. and Lysaght, T., 2008b. Nonlinear distortion synthesis using the split-sideband method, with applications to adaptive signal processing. *J. Audio Eng. Soc.*, 56(9), pp. 684–695.

Lazzarini, V., Timoney, J. and Lysaght, T., 2008c. The generation of natural-synthetic spectra by means of adaptive frequency modulation. *Comput. Music J.*, 32(2), pp. 9–22.

Lazzarini, V., Timoney, J. and Lysaght, T., 2008d. Asymmetrical methods for adaptive FM synthesis. In: *Proc. Int. Conf. Digital Audio Effects*, Espoo, Finland, Sept. 2008, pp. 40–48.

Lazzarini, V., Timoney, J., Pekonen, J. and Välimäki, V., 2009a. Adaptive phase distortion synthesis. In: *Proc. Int. Conf. Digital Audio Effects*, Como, Italy, Sept. 2009, pp. 28–35.

Lazzarini, V., Timoney, J., Kleimola, J. and Välimäki, V., 2009b. Five variations on a feedback theme. In: *Proc. Int. Conf. Digital Audio Effects*, Como, Italy, Sept. 2009, pp. 139–145.

Lazzarini, V. and Timoney, J., 2010a. New perspectives on distortion synthesis for virtual analog oscillators. *Comput. Music J.*, 34(1), pp. 28–40.

Lazzarini, V. and Timoney, J., 2010b. Theory and practice of modified frequency modulation synthesis. *J. Audio Eng. Soc.*, 58(6), pp. 459–471.

Lazzarini, V. and Timoney, J., 2011. The ModFM synthesis vocoder. In: *Proc. Int. Computer Music Conf.*, Huddersfield, UK, July – Aug. 2011, pp. 591–594.

Le Brun, M., 1977. A derivation of the spectrum of FM with a complex modulating wave. *Comput. Music J.*, 1(4), pp. 51–52.

Le Brun, M., 1979. Digital waveshaping synthesis. *J. Audio Eng. Soc.*, 27(4), pp. 250–266.

Lehtonen, H.-M., Pekonen, J. and Välimäki, V., 2012. Audibility of aliasing distortion in sawtooth signals and its implications for oscillator algorithm design. *J. Acoust. Soc. Amer.*, 132(4), pp. 2721–2733.

Lowenfels, D., 2003. Virtual analog synthesis with a time-varying comb filter. In: *Proc. 115th AES Convention*, New York, NY, USA, Oct. 2003.

Loy, G., 1985. Musicians make a standard: the MIDI phenomenon. *Comput. Music J.*, 9(4), pp. 8–26.

Macret, M., Pasquier, P. and Smyth, T., 2012. Automatic calibration of modified FM synthesis to harmonic sounds using genetic algorithms. In: *Proc. Sound and Music Computing Conf.*, Copenhagen, Denmark, July 2012, pp. 387–394.

Maher, R.C., 2006. Wavetable synthesis strategies for mobile devices. *J. Audio Eng. Soc.*, 53(3), pp. 205-212.

Malloch, J., Sinclair, S. and Wanderley, M.M., 2008. A network-based framework for collaborative development and performance of digital musical instruments. In: Kronland-Martinet, R., Ystad, S. and Jensen, K. eds. *Computer Music Modeling and Retrieval*, LNCS 4969, pp. 401–425.

Massie, D.C., 1998, Wavetable sampling synthesis. In: Kahrs, M. and Brandenburg, K. eds., *Applications of Digital Signal Processing to Audio and Acoustics*, Norfolk, MA: Kluwer Academic, pp. 311–341.

Mathews, M., 1963. The digital computer as a musical instrument. *Science*, 142(11), pp. 553–557.

Mathews, M.V., Miller, J.E., Moore, F.R., Pierce, J.R. and Risset, J.C., 1969. *The Technology of Computer Music*. Cambridge, MA: MIT Press.

Mitchell, T.J., 2010. *An Exploration of Evolutionary Computation Applied to Frequency Modulation Audio Synthesis Parameter Optimization*. Ph.D. dissertation, Univ. of the West of England, Bristol, UK.

Mitsuhashi, Y., 1980. Waveshape parameter modulation in producing complex spectra. *J. Audio Eng. Soc.*, 28(12), pp. 879–895.

Mitsuhashi, Y., 1982. Piecewise interpolation technique for audio signal synthesis. *J. Audio Eng. Soc.*, 30(4), pp. 192–202.

MMA, 1983. *The Complete MIDI 1.0 Detailed Specification*. MIDI Manufacturers Association.

Moore, F.R., 1977a. *Real Time Interactive Computer Music Synthesis*. Ph.D. dissertation, Stanford Univ., CA, USA.

Moore, F.R., 1977b. Table lookup noise for sinusoidal digital oscillators. *Comput. Music J.*, 1(2), pp. 26–29.

Moore, F.R., 1988. The dysfunctions of MIDI. *Comput. Music J.*, 12(1), pp. 19–28.

Moore, F.R., 1990. *The Elements of Computer Music*. Englewood Cliffs, NJ: Prentice-Hall.

Moorer, J.A., 1976. The synthesis of complex audio spectra by means of discrete summation formulas. *J. Audio Eng. Soc.*, 24(9), pp. 717–727.

Moorer, J.A., 1977. Signal processing aspects of computer music: a survey. *Proc. IEEE*, 65(8), pp. 1108–1141.

Nam, J., Välimäki, V., Abel, J.S. and Smith, J.O., 2009. Alias-free virtual analog oscillators using a feedback delay loop. In: *Proc. Int. Conf. Digital Audio Effects*, Como, Italy, Sept. 2009, pp. 347–352.

Nam, J., Välimäki, V., Abel, J.S. and Smith, J.O., 2010. Efficient antialiasing oscillator algorithms using low-order fractional delay filters. *IEEE Trans. Audio Speech Lang. Process.*, 18(4), pp. 773–785.

Orlarey, Y., Fober, D. and Letz, S., 2009. Faust: an efficient functional approach to DSP programming. In: Assayag, G. and Gerzso, A. eds. *New Computational Paradigms for Computer Music*. Paris: Editions Delatour.

Palamin, J.P., Palamin, P. and Ronveaux, A., 1988. A method of generating and controlling musical asymmetrical spectra. *J. Audio Eng. Soc.*, 36(9), pp. 671–685.

Pakarinen, J., Välimäki, V. and Karjalainen, M., 2005. Physics-based methods for modeling nonlinear vibrating strings. *Acta Acustica united with Acustica*, 91(2), pp. 312–325.

Pakarinen, J., Välimäki, V., Fontana, F., Lazzarini, V. and Abel, J.S., 2011. Recent advances in real–time musical effects, synthesis, and virtual analog models. *EURASIP J. Adv. Signal Process.*, Vol. 2011.

Parker, J., 2011. A simple digital model of the diode-based ring-modulator. In: *Proc. Int. Conf. Digital Audio Effects*, Paris, France, Sept. 2011, pp. 163–166.

Pekonen, J. and Välimäki, V., 2008a. Filter-based alias reduction in classical waveform synthesis. In: *Proc. IEEE Intl. Conf. Acoust., Speech, and Signal Process.*, Las Vegas, NV, Apr. 2008, pp. 133–136.

Pekonen, J., 2008b. Coefficient–modulated first-order allpass filter as distortion effect. In: *Proc. Int. Conf. Digital Audio Effects*, Espoo, Finland, Sept. 2008, pp. 83–87.

Pekonen, J., Välimäki, V., Abel. J.S. and Smith, J.O., 2009. Spectral delay filters with feedback and time-varying coefficients. In: *Proc. Int. Conf. Digital Audio Effects*, Como, Italy, Sept. 2009, pp. 157–164.

Pekonen, J., Välimäki, V., Nam, J., Smith, J.O. and Abel, J.S., 2010a. Variable fractional delay filters in bandlimited oscillator algorithms for music synthesis. In: *Proc. Int. Conf. Green Circuits and Systems*, Shanghai, China, June 2010, pp. 148–153.

Pekonen, J., Nam, J., Smith, J.O., Abel, J.S. and Välimäki, V., 2010b. On minimizing the look-up table size in quasi-bandlimited classical waveform synthesis. In: *Proc. Int. Conf. Digital Audio Effects*, Graz, Austria, Sept. 2010, pp. 57–64.

Pekonen, J. and Välimäki, V., 2011a. The brief history of virtual analog synthesis. In: *Proc. 6th Forum Acusticum*, Aalborg, Denmark, June – July 2011, pp. 461–466.

Pekonen, J., Pihlajamäki, T. and Välimäki, V., 2011b. Computationally efficient Hammond organ synthesis. In: *Proc. Int. Conf. Digital Audio Effects*, Paris, France, Sept. 2011, pp. 19–22.

Pekonen, J., Nam, J., Smith, J.O. and Välimäki, V., 2012. Optimized polynomial spline basis function design for quasi-bandlimited classical waveform synthesis. *IEEE Signal Process. Lett.*, 19(3), pp. 159–162.

Penttinen, H., Härmä, A. and Karjalainen, M., 2000. Digital guitar body mode modulation with one driving parameter. In: *Proc. Int. Conf. Digital Audio Effects*, Verona, Italy, Dec. 2000, pp. 31–36.

Phillips, D., 1994. *Wavestation SR Player's Guide*. [pdf] Tokyo: Korg Inc. Available at: <http://www.korg.com/uploads/Support/Wavestation SR_Players Guide_634707764251818894.pdf> [Accessed 27 Jan. 2013].

Pierce, J.R. and Van Duyne, S.A., 1997. A passive nonlinear digital filter design which facilitates physics-based sound synthesis of highly nonlinear musical instruments. *J. Acoust. Soc. Am.*, 101(2), pp. 1120–1126.

Pinch, T. and Trocco, F., 2004. *Analog Days: The Invention and Impact of the Moog Synthesizer*. Cambridge, MA: Harvard University Press.

Poepel, C. and Dannenberg, R., 2005. Audio signal driven sound synthesis. In: *Proc. Int. Computer Music Conf.*, Barcelona, Spain, Sept. 2005, pp. 391–394.

Puckette, M., 1995. Formant-based audio synthesis using nonlinear distortion. *J. Audio Eng. Soc.*, 43(1–2), pp. 40–47.

Puckette, M., 2007. *The Theory and Technique of Electronic Music*. River Edge, NJ: World Scientific Press.

Risset, J.-C., 1969. *An Introductory Catalog of Computer-synthesized Sounds*. Bell Laboratories. Published in 1995 as part of "the historical CD of digital sound synthesis", Computer Music Currents 13, Wergo WER–20332.

Roads, C., 1996. *The Computer Music Tutorial.* Cambridge, MA: MIT Press.

Rodet, X. and Depalle, P., 1992. Spectral envelopes and inverse FFT synthesis. In: *Proc. 93rd AES Convention*, San Francisco, CA, USA, Oct. 1992.

Savioja, L., Välimäki, V. and Smith, J.O., 2011. Audio signal processing using graphics processing units. *J. Audio Eng. Soc.*, 59(1/2), pp. 3-19.

Schaefer, R.A., 1970. Electronic musical tone production by nonlinear waveshaping. *J. Audio Eng. Soc.*, 18(4), pp. 413–417.

Schimmel, J. 2012. Audible aliasing distortion in digital audio synthesis. *Radioeng.*, 21(1), pp. 56-62.

Schottstaedt, B., 1977. The simulation of natural instrument tones using frequency modulation with a complex modulating wave. *Comput. Music J.*, 1(4), pp. 46–50.

Schwarz, D., 2007. Corpus-based concatenative synthesis. *IEEE Signal Process. Mag.*, 24(2), pp. 92–104.

Serra, X. and Smith, J., 1990. Spectral modeling synthesis: a sound analysis/synthesis based on a deterministic plus stochastic decomposition. *Comput. Music J.,* 14(4), pp. 12-24.

Shneiderman, B., 1983. Direct manipulation: a step beyond programming languages. *IEEE Computer*, 16(8), pp. 57–69.

Smith, J.O., 1991. Viewpoints on the history of digital synthesis. In: *Proc. Int. Computer Music Conf.*, Montreal, Canada, Oct. 1991, pp. 1–10.

Smith, J.O., 1992. Physical modeling using digital waveguides. *Comput. Music J.*, 16(4), pp. 74–87.

Smyth, T. and Abel, J.S., 2012. Toward an estimation of the clarinet reed pulse from instrument performance. *J. Acoust. Soc. Am.*, 131(6), pp. 4799–4810.

Steiglitz, K., 1996. A Digital Signal Processing Primer with Applications to Digital Audio and Computer Music, Reading, MA: Addison-Wesley.

Stilson, T., 1995. General weirdness with the Karplus-Strong string. In: *Proc. Int. Computer Music Conf.*, Alberta, Canada, Sept. 1995, pp. 110–111.

Stilson, T. and Smith, J., 1996. Alias-free digital synthesis of classic analog waveforms. In: *Proc. Int. Computer Music Conf.*, Hong Kong, Aug. 1996, pp. 332–335.

Stilson, T., 2006. *Efficiently-variable Non-oversampled Algorithms in Virtual-Analog Music Synthesis – A Root-Locus Perspective*. Ph.D. dissertation, Stanford Univ., CA, USA.

Suen, C.Y., 1970. Derivation of harmonic equations in nonlinear circuits. *J. Audio Eng. Soc.*, 18(6), pp. 675–676.

Tan, B. and Gan, S., 1993. Real-time implementation of asymmetrical frequency-modulation synthesis. *J. Audio Eng. Soc.*, 41(5), pp. 357–363.

Tan, B., Gan, S., Lim, S. and Tang, S., 1994. Real–time implementation of double frequency modulation (DFM) synthesis. *J. Audio Eng. Soc.*, 42(11), pp. 918–926.

Tan, B. and Lim, S., 1996. Automated parameter optimization for double frequency modulation synthesis using the genetic annealing algorithm. *J. Audio Eng. Soc.*, 44(1/2), pp. 3–15.

Tassart, S., 2013. Band-limited impulse train generation using sampled infinite impulse responses of analog filters. *IEEE Trans. Audio, Speech Lang. Process.*, 21(3), pp. 488–497.

Timoney, J., Lazzarini, V. and Lysaght, T., 2008. A Modified-FM approach to bandlimited signal generation. In: *Proc. Int. Conf. Digital Audio Effects*, Espoo, Finland, Sept. 2008, pp. 12–20.

Timoney, J., Lazzarini, V., Carty, B. and Pekonen, J., 2009a. Phase and amplitude distortion methods for digital synthesis of classic analog waveforms. In: *Proc. 126th AES Convention*, Munich, Germany, May 2009.

Timoney, J., Lazzarini, V., Pekonen, J. and Välimäki, V., 2009b. Spectrally rich phase distortion sound synthesis using an allpass filter. In: *Proc. IEEE Int. Conf. Acoustics, Speech, and Signal Processing*, Taipei, Taiwan, Apr. 2009, pp. 293–296.

Timoney, J. and Lazzarini, V., 2011. Exponential frequency modulation bandwidth criterion for virtual analog applications. In: *Proc. Int. Conf. Digital Audio Effects*, Paris, France, Sept. 2011, pp. 115–118.

Timoney, J. and Lazzarini, V., 2012. A homomorphic interpretation of the complex FM expansion. In: *AES 45th Int. Conf.*, Helsinki, Finland, Mar. 2012.

Tolonen, T., Välimäki, V. and Karjalainen, M., 1998. Evaluation of modern sound synthesis methods. Report no. 48, Espoo, Finland: Helsinki University of Technology. Available at: http://www.acoustics.hut.fi/publications/reports/sound_synth_report.html [Accessed 27 Jan. 2013].

Tolonen, T., Välimäki, V. and Karjalainen, M., 2000. Modeling of tension modulation nonlinearity in plucked strings. *IEEE Trans. Audio, Speech Lang. Process.*, 8(3), pp. 300–310.

Tomisawa, N., Nippon Gakki Seizo Kabushiki Kaisha, 1979. *Tone production method for an electronic musical instrument*. U.S. Pat. 4,249,447.

Vairetti, G., 2012. *Vector Phaseshaping Synthesis of Harmonic Musical Instrument Tones*, MSc. thesis, Politecnico di Milano, Italy.

Valsamakis, N. and Miranda, E.R., 2005. Iterative sound synthesis by means of cross-coupled digital oscillators. *Digital Creativity*, 16(2), pp. 90–98.

Van Duyne, S.A. and Smith, J.O., 1992. Implementation of a variable pick–up point on a waveguide string model with FM/AM applications. In: *Proc. Int. Computer Music Conf.*, San Jose, CA, USA, Oct. 1992, pp. 154–157.

Verfaille, V., Zölzer, U. and Arfib, D., 2006. Adaptive digital audio effects (a-DAFx): a new class of sound transformations. *IEEE Trans. Audio, Speech Lang. Process.*, 14(5), pp. 1817–1831.

Verplank, B., Mathews, M. and Shaw, R., 2000. Scanned synthesis. In: *Proc. Int. Computer Music Conf.*, Berlin, Germany, Sept. 2000.

Välimäki, V., Huopaniemi, J., Karjalainen, M. and Jánosy, Z., 1996. Physical modeling of plucked string instruments with application to real-time sound synthesis. *J. Audio Eng. Soc.,* 44(5), pp. 331–353.

Välimäki, V., Tolonen, T. and Karjalainen, M., 1998. Signal-dependent nonlinearities for physical models using time-varying fractional delay filters. In: *Proc. Int. Computer Music Conf.*, Ann Arbor, MI, USA, Oct. 1998, pp. 264–267.

Välimäki, V., 2005. Discrete-time synthesis of the sawtooth waveform with reduced aliasing. *IEEE Signal Process. Lett.*, 12(3), pp. 214–217.

Välimäki, V. and Huovilainen, A., 2006. Oscillator and filter algorithms for virtual analog synthesis. *Comput. Music J.*, 30(2), pp. 19–31.

Välimäki, V. and Huovilainen, A., 2007. Antialiasing oscillators in subtractive synthesis. *IEEE Signal Process. Mag.*, 24(2), pp. 116–125.

Välimäki, V., Abel, J.S. and Smith, J.O., 2009. Spectral delay filters. *J. Audio Eng. Soc.*, 57(7/8), pp. 521–531.

Välimäki, V., Nam, J., Smith, J.O. and Abel, J.S., 2010. Alias-suppressed oscillators based on differentiated polynomial waveforms. *IEEE Trans. Audio, Speech Lang. Process.*, 18(4), pp. 786–798.

Välimäki, V., Pekonen, J. and Nam, J., 2012. Perceptually informed synthesis of bandlimited classical waveforms using integrated polynomial interpolation. *J. Acoust. Soc. Amer.*, 131(1), pp. 974–986.

Wanderley, M.M. and Orio, N., 2002. Evaluation of input devices for musical expression: borrowing tools from HCI. *Comput. Music J.*, 26(3), pp. 62–76.

Wanderley, M.M. and Depalle, P., 2004. Gestural control of sound synthesis. *Proc. IEEE*, 92(4), pp. 632–644.

Wilson, S., Cottle, D. and Collins, N. eds., 2011. *The SuperCollider Book*. Cambridge, MA: MIT Press.

Winham, G. and Steiglitz, K., 1970. Input generators for digital sound synthesis. *J. Acoust. Soc. Amer.*, 47(2), pp. 665–666.

Wright, M. and Freed, A., 1997. Open SoundControl: a new protocol for communicating with sound synthesizers. In: *Proc. Int. Computer Music Conf.*, Thessaloniki, Greece, Sept. 1997, pp. 101–104.

Zhang, Y. and Chen, L., 2012. Particle swarm optimization with dynamic local search for frequency modulation parameter identification. *Int. J. Advancements in Computing Technology*, 4(3), pp. 189–195.

Zicarelli, D., 2002. How I learned to love a program that does nothing. *Comput. Music J.*, 26(4), pp. 44–51.

Zölzer, U., 2012. Pitch-based digital audio effects. In: *Proc. Int. Symp. Communications Control and Signal Processing*, Rome, Italy, May 2012, pp. 1–4.

Aalto-DD 25/2013

ISBN 978-952-60-5015-7
ISBN 978-952-60-5016-4 (pdf)
ISSN-L 1799-4934
ISSN 1799-4934
ISSN 1799-4942 (pdf)

Aalto University
School of Electrical Engineering
Department of Signal Processing and Acoustics
www.aalto.fi

BUSINESS +
ECONOMY

ART +
DESIGN +
ARCHITECTURE

SCIENCE +
TECHNOLOGY

CROSSOVER

DOCTORAL
DISSERTATIONS