NON-OVERSAMPLED PHYSICAL MODELING FOR VIRTUAL ANALOG SIMULATIONS

A DISSERTATION
SUBMITTED TO THE DEPARTMENT OF MUSIC
AND THE COMMITTEE ON GRADUATE STUDIES
OF STANFORD UNIVERSITY
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY

François G. Germain
June 2019

© 2019 by Francois Georges Germain. All Rights Reserved.
Re-distributed by Stanford University under license with the author.

This work is licensed under a Creative Commons Attribution-Noncommercial 3.0 United States License.
http://creativecommons.org/licenses/by-nc/3.0/us/

This dissertation is online at: http://purl.stanford.edu/xd967xc4156

I certify that I have read this dissertation and that, in my opinion, it is fully adequate in scope and quality as a dissertation for the degree of Doctor of Philosophy.

Julius Smith, III, Primary Adviser

I certify that I have read this dissertation and that, in my opinion, it is fully adequate in scope and quality as a dissertation for the degree of Doctor of Philosophy.

Jonathan Abel

I certify that I have read this dissertation and that, in my opinion, it is fully adequate in scope and quality as a dissertation for the degree of Doctor of Philosophy.

Chris Chafe

Approved for the Stanford University Committee on Graduate Studies.

Patricia J. Gumport, Vice Provost for Graduate Education

This signature page was generated electronically upon submission of this dissertation in electronic format. An original signed hard copy of the signature page is on file in University Archives.

Abstract

Physical modeling of real-world systems involves building computer models from continuous-time differential equations. These equations are derived from the physical principles governing the dynamics of these systems. To do so, it uses a systematic process to build discrete-time differential equations that approximate such dynamics. In audio research, this process is the focus of the field referred to as *virtual analog* modeling, and this thesis presents several contributions to the field.

Converting the continuous-time differential equations into discrete-time equations is typically achieved using one of the many *discretization* methods available in the literature, e.g., numerical integration methods, finite difference methods, or finite element methods. One particular parameter of interest in the definition of the discrete-time equations is the rate of update of the discrete-time quantities present in these equations. A general principle for all discretization methods is that increasing the update rate associated with the discrete-time computer model generally decreases the residual error between the quantities computed in the discrete-time model and their equivalent quantities in the real, continuous-time system. Indeed, such methods are generally built as *convergent* methods so that the residual error is asymptotically guaranteed to approach zero as the update rate increases. More sophisticated approaches modulate the update rate of the discrete-time equations dynamically as a function of pre-established error estimators.

Augmenting or modulating the update rate of the discrete-time equations is generally considered to be the most straightforward way to modulate the accuracy of a discrete-time computer model. It is, however, generally associated with a non-trivial increase in the computational cost necessary to simulate the system over a set period of time. For advanced methods with modulated update rates, it has the added drawback of limiting the predictability of the computational cost.

While significant computational costs are often acceptable in fields where accuracy is primordial (e.g., meteorology or aeronautics), virtual analog simulations do not generally need to prioritize accuracy as strongly. The first reason for this is that discrete-time audio models have a "natural" update rate around $40 - 50\,\text{kHz}$. Indeed, the human auditory range is limited to a finite range typically approximated as the range $20\,\text{Hz} - 20\,\text{kHz}$, so that Shannon's theorem guarantees that quantities updated at the aforementioned rate are sufficient to encode all the audible signal information along with limited amounts of inaudible information. The second reason is that in many audio

applications, it is important to achieve real-time computational costs (i.e., the computational time associated with deriving an output sample is shorter than the update rate) and limited latency (i.e., the amount of time between when an input sample is fed to the computer model and the time when the corresponding output sample is computed). Altogether, the determination of the appropriate update rate for a virtual analog simulation should then lead to different types of compromises.

One particular class of interest in audio systems is the class of lumped systems. In that class, we find most audio electrical circuits, e.g., analog oscillators, analog filters, guitar amplifiers, and guitar distortion effects. We also find many mechanical systems of interest in audio, e.g., loudspeakers, microphones, piano bridges, mechanical excitators, and resonators. Systems in that class are characterized by physical variables which are solely function of time, so their dynamics can be expressed as *ordinary differential equations* (ODE) or, exceptionally, as *differential algebraic equations* (DAE).

This thesis presents several results on the topics of analysis and optimization of discretization methods applied to lumped audio systems, informed by the atypical compromises of virtual analog applications regarding the model update rate. In particular, we focus on *non-oversampled* methods, meaning methods where the update rate is fixed and accuracy must be controlled through alternative methods.

The first chapter presents a framework to properly define signal identity in the context of a non-oversampled computer simulation. Indeed, a proper equivalency relationship has to be established between the continuous-time quantities we want to approximate and the intrinsically discrete-time quantities that we compute in order to unambiguously define an optimal model. Multiple options can be derived for such equivalency, but only one can be used at any one time since they lead to incompatible definitions of optimality if the update rate is kept fixed. Our framework details mathematically how to unambiguously define and apply a chosen equivalency option. A by-product of this framework is a broader definition of the concept of aliasing, attached to the intrinsic limitation in the information amount encoded by discrete-time sequences, and the resulting implications regarding the analysis of any continuous-time system of interest.

The second chapter presents new analytical and empirical developments for the analysis of aliasing error in the numerical models of lumped audio systems. It allows for an advanced and efficient comparison of discretization approaches and antialiasing methods. The framework discusses aliasing distortion in the context of different types of input periodic waveforms, from cosinusoidal to arbitrary. The analytical analysis extends the mathematical framework presented in Thornburg [1999] to dynamical systems and arbitrary waveforms. The proposed empirical analysis method uses an extension of harmonic balance methods in order to efficiently retrieve all the harmonic components of the output waveform of the model and the original system.

The third chapter presents a generalization of known one-step discretization methods using the formalism of Möbius transforms. It formalizes particular subsets of these transforms to provide for

new discretization approaches for nonlinear lumped systems, such as the α-transform and the parametric bilinear transform. It also formalizes several design criteria to choose the most appropriate candidate of that family for a given virtual analog application. In particular, it presents the novel criterion of damping monotonicity. These methods are of particular interest due to two properties. First, they can be readily interpreted as one-to-one mapping functions between the s-plane used to describe continuous-time systems and the z-plane used to describe discrete-time computer models, facilitating the derivation and interpretation of parametrization rules. Second, they preserve the order of the update equations so that the computational costs of models generated with any of these methods are comparable. In particular, this class generalizes common virtual analog methods such as the bilinear transform and the Euler methods.

The fourth chapter presents a novel framework to optimize the discretization methods applied to individual elements of a lumped linear audio system in order to systematically obtain a more accurate computer model. Deriving physical models for linear systems can provide access to the equations relative to all the dynamical elements forming the system and its model responses. However, discretization procedures are generally applied globally to the system equations, allowing very limited control on the reproduction error through available free parameters in the discretization. We show how to leverage optimizing such free parameters at the individual element level to generate a much more accurate physical model while preserving the system structure and the model computational costs. In particular, our approach can perform a joint optimization of an arbitrary number of output variables using either a Kirchhoff domain or a wave domain formulation.

Acknowledgments

Over the duration of my stay at Stanford, I have met many incredible people who, in bigger and smaller ways, have shaped my experience and by extension this present work. I can say without a doubt that, however small, they all contributed to getting me over the finish line.

First and foremost, I would like to thank my advisor, Julius Smith, for his unwavering support over the long road of my Ph.D. graduation at the Center for Computer Research in Music and Acoustics (CCRMA). Right from the day I was admitted, he made sure to call me right away as I was walking in the streets of Montreal, to make sure I learned the good news as soon as possible. He has encouraged me in all my projects and has provided invaluable feedback and guidance. I want to thank my reader Jonathan Abel, for his feedback, support and career advice and for bringing me on the technical team of the 2016 Cappella Romana concert. And thanks to my reader Chris Chafe, for his feedback and support, and for a tremendous job leading CCRMA during my time there. And I wish to thank Ge Wang and Brian Wandell for volunteering their time to participate in my oral defense and for their feedback.

I also want to thank the Dorothy Culver Haynie Music Fellowship Fund for their generous financial support of my degree. I also want to thank Julius Smith, CCRMA, the Music Department, and the IEEE Signal Processing Society for helping to fund conference trips that were essential for giving visibility to my work. And I want to thank the Friends of Music, Gil Ellenberger, William Mahrt, Joan Mansour, John Planting for supporting my lessons in the Music Department.

At this point, I must also mention the exceptional contribution of my dear friend and colleague Kurt J. Werner. Kurt was in many ways the inspiration for this project, as I observed him pour his heart and soul in his wave digital filter research. While he was doing that, he still found the time to provide support at every step of my degree, willingly volunteering his time to give extensive feedback, and serve as sounding board on my research over many years. He was an invaluable resource due to his extensive knowledge of the virtual analog literature, and provided essential feedback in making this present document as good as it it. He also generously invited me at Queen's University Belfast to work on parts of this project.

Over the course of my many years at Stanford, I had the privilege to collaborate with many remarkable people in and out of the university. The incredible team of virtual analog researchers was

instrumental in driving forward the topic, and by extension this project, over my years at CCRMA. With their unabashed passion, they proved themselves the worthy successors of the long tradition of excellence of CCRMA into the field and I was honored to witness their accomplishments. Besides Kurt, I would like to give a shout out to Ross Dunkel, Michael Olsen and Maximilian Rest. I would also like to thank my internship mentors, Gautham Mysore from Adobe Research, Alan Seefeldt from Dolby Laboratories and Vladlen Koltun from the Intelligent Systems Lab at Intel. They put their unconditional trust in my abilities and offered me amazing opportunities to experience advanced research in an industry context, adding an essential piece to my graduate education, and they put me in the presence of amazing expert researchers in the fields of audio processing, machine learning and spatial sound. I also need to thank Brian Wandell for welcoming me in his VISTA research group for 6 months and teaching me essential skills, especially as he prepped me to give the best conference presentations I have given to this day. And I wish to thank Takako Fujioka for her help with perceptual experiments, but also for attentively checking up on me over all this year. I would also like to have a thought for Margot Gerritsen for giving the most inspiring course on numerical methods I have had the opportunity to attend.

Completing my work would have been all but impossible without the joy my continuing music practice brought me while I was working long hours month after months. As such, I would like to thank people who supported my musical endeavors at Stanford and beyond: my music teachers Elaine Thornburg and John Dornenburg who made me discover the hidden beauties of the harpsichord and the viol, my composition teachers Giancarlo Aquilanti and Jonathan Berger who helped me discover my inner muse, Eric Wu and Sarah Smith who premiered my first piece, Anna Wittstruck who granted me my lifelong dream of playing in an orchestra, all my friends from the Stanford Symphony Orchestra, and the members of Queen's University Belfast Viol Consort who welcomed me while I was working in Northern Ireland.

I also want to express my deepest gratitude to all the staff and administrators from CCRMA, the Music Department and all the institutions who welcomed me for their consideration, patience and professionalism, particularly Nette Worthey, Debbie Barney, Velda Williams, Ardis Walling, Mario Champagne, Carlos Sánchez and my late friend Carr Wilkerson. I must give a special shout out to Fernando Lopez-Lezcano, CCRMA's ambisonics wizard and attentive watchman of the CCRMA computers I depended on for so many years. I also want to thank the teachers I was privileged to assist and learn from: Dave Berners, Marina Bosi, George Barth, Giancarlo Aquilanti, Fernando Lopez-Lezcano, Chris Chafe and Andrew Ng. And I want to thank Charles Kronengold and Jonathan Berger for volunteering their time to get me ready for my qualifying examinations. And I want to give a thought to all the dear classmates, colleagues, collaborators I met along the way and shared a bit of this path: Jens Ahrens, Iretiayo Akinola, Erick Arenas, Micah Arvey, Jack Atherton, Constantin Basica, Michael Berger, Alberto Bernardini, Eren Bilir, Myles Borins, Stephen Boyd, Maddie Brown, Nick Bryan, Eoin Callery, Victoria Chang, Alex Chechile, Qifeng Chen, Hongchan

Choi, John Chowning, Zoran Cvetkovic, John D'Arcy, Luke Dahl, Philippe Depalle, Sonia El Hedri, Turgut Ercetin, Joyce Farrell, Grégoire Faucher, Vincent Fréour, Nick Gang, Chet Gnegy, Victoria Grace, Victoria Grace, Andrew Greenwald, Rob Hamilton, Brian Hamilton, Alex Hay, Woody Herman, Jorge Herrera, Madeline Huberth, Kurt Isaacson, Christopher Jette, Blair Kaneshiro, Jarek Kapuscinski, Hyung-Suk Kim, Minje Kim, Carolin Krahn, Steven Lansel, Sasha Leitman, Nolan Lem, Dawen Liang, Jessie Marino, Sara Martín, Lydia Mayne, Sarah McCarthy, Hunter McCurry, Romain Michon, Prateek Murgai, Juhan Nam, Chryssie Nanou, Iván Naranjo, Tim O'Brien, Jieun Oh, Jack Perng, Mark Rau, Stephen Reid, Irán Román, Spencer Salazar, Mayank Sanganeria, Stephen Sano, Craig Sapp, Charlie Sdraulig, Kitty Shi, Laura Steenberge, Colin Sullivan, Dennis Sun, Qiyuan Tian, Maarten van Walstijn, Marcelo Wanderley, Michael Wilson, David Yeh, the CCRMA Police soccer team, and the Anthro/CCRMA/DLCL soccer team. I want to thank my EE364 study group Shubho Sengupta and Gearoid OBrien who carried me through this wonderful class. And I want to give a special shout out to the lifelong friends I made along the way. Besides Kurt, I am compelled to mention Suddhaseel Sen for sharing with me his passion for classical music and tea. I have to mention Emily Graber who, aside from being a brilliant researcher, also generously shared her immense talent at the violin to play alongside me at several harpsichord recitals. I also have a thought for her partner Josh Gevirtz, always willing to ramble with me about the state of the world. And Elliot Canfield-Dafilou, always around to bounce ideas, discuss the latest international soccer results or commiserate on our condition. I have a kind thought for Melissa Kagen, Kurt's partner, for sharing her uncompromising and passionate drive for research and excellence in all things, including mini-golf. I also need to thank John Granzow for letting me correct his French, and accompanying me on the long bus ride to the Milk Pail market, and sharing memories about French cuisine, as well as being such an inspiring and creative spirit, one of the true souls of CCRMA. And the list could not be complete without thanking Alessandra Aquilanti, always watching (out) for me, primarily from the first rows of the Stanford Symphony Orchestra viola section, cheering for me and lifting my spirits up every time I needed it.

I thank my brother Laurent Germain and my grand-parents Frdric and Raymonde Germain and Gilberte Ben Assayag for a life of love and kindness. Finally, I cannot thank enough my parents Gérard Ben Assayag and Yvette Germain for a life of unrelenting encouragements, love and support, and for visiting me more often than any son could hope for after moving more than 9,000 km away from home. They are and will always be the true inspirations of my work and my life.

Contents

List of Tables

List of Figures

Chapter 0

Introduction

Modeling lumped audio systems is an historical thread of research in the music technology community. It has generally served multiple complementary objectives. Accurate modeling of these systems has allowed for their preservation and reproduction of their characteristics and capabilities while the original system can be difficult to find and/or expensive to build. Concurrently, the modeling of existing systems has often been considered as a useful perspective for the design of novel effect designed through extensions or modifications of an existing system. In this document, we present a series of results gathered over the course of my doctoral studies at the Center for Computer Research in Music and Acoustics at Stanford University. Such results offer improvements on typical physical modeling approaches for the modeling of lumped audio systems, focusing on optimizing the response of the generated systems with limited additional computational costs.

0.1 Lumped audio systems

Lumped systems refer to systems where the information travels instantaneously across the system. As a result, the behavior of the system is generally adequately described through a system of differential algebraic equations (DAE). They are defined by opposition to distributed systems where information travels at finite speed, such that physical variables are described as function of time and space, usually through systems of partial differential equations [Smith III 2010]. The lumped system formalism can generally apply to all systems where the speed of propagation of the physical perturbations is large compared to the general size of the system itself. As such, electric circuits are generally considered to belong to the class of lumped systems thanks to the fact that electrical information travels at the speed of light. Many small point-like mechanical systems are also adequately formalized with lumped system hypothesis.

Many audio systems of interest, mechanical and/or electrical, can be described adequately as lumped systems. As a result, we find a great variety of such examples in the published literature.

These include most audio devices based on electrical circuits such as synthesizer effects [Stilson and Smith III 1996, Huovilainen 2004, Hélie 2006, Fontana 2007, Hélie 2010, Germain 2011, Daly 2012, D'Angelo and Välimäki 2013, D'Angelo and Välimäki 2014a,b], guitar distortions [Yeh et al. 2007a,b, 2008, Yeh and Smith III 2008, Yeh 2009, Holmes and van Walstijn 2015, Eichas et al. 2015, Eichas and Zölzer 2016] and amplifiers [Pakarinen et al. 2009, Pakarinen and Yeh 2009, Dempwolf et al. 2009, Pakarinen and Karjalainen 2010, Paiva et al. 2011, Mačák and Schimmel 2010, 2011, Mačák 2012, Dunkel et al. 2016], drum machines [Werner et al. 2014a,b,c,d, Werner 2016], as well as many mechanical devices such as loudspeakers [Fränken et al. 2001, Falaize and Hélie 2014, Falaize et al. 2015], microphone capsules [Wang et al. 2004, Chen et al. 2007], and some acoustical instrument parts such as violin or piano bridges [Kartofelev et al. 2013, Maestre et al. 2013].

0.2 Modeling strategies

To approach the modeling of lumped audio systems, a wide range of methods have been studied in the literature. The first objective of these methods is to generate a computer model functioning on a discrete time scale, when the behavior of the original system is generally described in a continuous time context. Typically, researchers have attempted to classify these methods among two main categories: black-box and white-box modeling techniques.

0.2.1 Black-box modeling

Black-box modeling refers to modeling approaches that focus on modeling the behavior of the target system without considering the inner mechanics of that system. The principal motivation for the black-box modeling strategies is often due to the fact that we do not have access to a complete description of the inner mechanics of the system under study. Whether we have access to a qualitative description of the output behavior of the system, or we have access to direct measurements of that output, a black-box model can be designed to attempt to reproduce it.

Other motivations exist, such as the need to generate a low-complexity model. Indeed, while an exact reproduction of the system's inner mechanics should yield an accurate model, the computational complexity of that model is sometimes too high for the intended use of that model. For example, high-accuracy white-box models can be too complex to run on mobile platforms. Black-box modeling strategies are often better suited to generate approximate solutions while controlling the model's computational complexity.

Black-box modeling of linear lumped systems is generally referred to as filter design. In this context, the system's behavior is generally captured by the knowledge of its frequency response (or its impulse response), possibly as a function of some control parameters. In case where we have access to the system for measurements, it is often possible to measure its frequency response by recording the system's output for a specific test signal. Such test signals generally need to possess

some specific properties, such as having spectral content at all frequencies, and having a known inverse filter (i.e., a signal which returns a perfect impulse when convolved with the test signal). Typical examples of such test signals are white noise sequences, Golay codes and sine sweeps [Foster 1986, Farina 2007, Novak et al. 2010c, Germain 2011]. Once a target frequency response is available, it is then possible to apply one of the many published methods to create a linear filter with a similar response. Such methods are generally divided in two categories depending on whether they produce a finite-impulse response (FIR) filter or an infinite-impulse response (IIR) filter. Generating approximate FIR models is generally the more straightforward process, with simple approaches such as window-based design [Oppenheim and Schafer 2009], or more complex optimized design through methods such as the Parks-McClellan algorithm or other approaches [Herrmann 1970, Herrmann and Schuessler 1970, Hofstetter et al. 1971, Parks and McClellan 1972a,b, Rabiner 1972a,b, McClellan and Parks 1973]. Methods to generate IIR filters are generally more complex and include for example Prony's method [Markel and Gray Jr. 1976, Smith III 2007b]. With these methods, a simple measure of the computational complexity of the resulting model is well captured by the chosen order of the model linear filter, though comparing the complexity of FIR-based and IIR-based models should consider possible optimizations (e.g., FIR filtering can be accelerated through the use of spectral-domain convolution).

The black-box modeling of nonlinear lumped systems is somewhat more complex, due to the fact that we no longer have access to a summarizing quantity of the system as simple as the frequency response for linear systems. The equivalent formalism for a generic nonlinear system generally corresponds to the decomposition in Volterra series [Billings 1980, Hélie 2006, 2010]. In this framework, any system response can be decomposed in an infinite superposition of convolutions for each polynomial order through the formula

$$y[n] = h_0 + \sum_{p=1}^{P} \sum_{\tau_1 \in \mathbb{Z}} \cdots \sum_{\tau_P \in \mathbb{Z}} h_p(\tau_1, \ldots, \tau_p) \prod_{j=1}^{p} x(n - \tau_j). \tag{0.1}$$

However, no efficient method exists to estimate this decomposition from direct system measurements, though it is technically possible using white-noise test signals [Billings 1980]. In general, research has relied on lower-complexity architecture, even though they are generally unlikely to capture every details of the target system. General architectures include the so-called Hammerstein, Wiener and Wiener-Hammerstein models and their polynomial versions (see Fig. 0.1) [Janczak 2004]. These models use one static nonlinearity (or many in the case of polynomial versions) combined with pre- and/or post-filtering depending on the architecture type (see Fig. 0.1). Ad hoc combinations of static nonlinearities and linear filters [Eichas and Zölzer 2016] and/or more complex combinations, such as neural networks [Janczak 2004, Moin 2010] are also possible. A variety of algorithms have been developed to parametrize all these algorithms, typically with the complexity of the procedure increasing as the number of static nonlinearity blocks increases in the system.

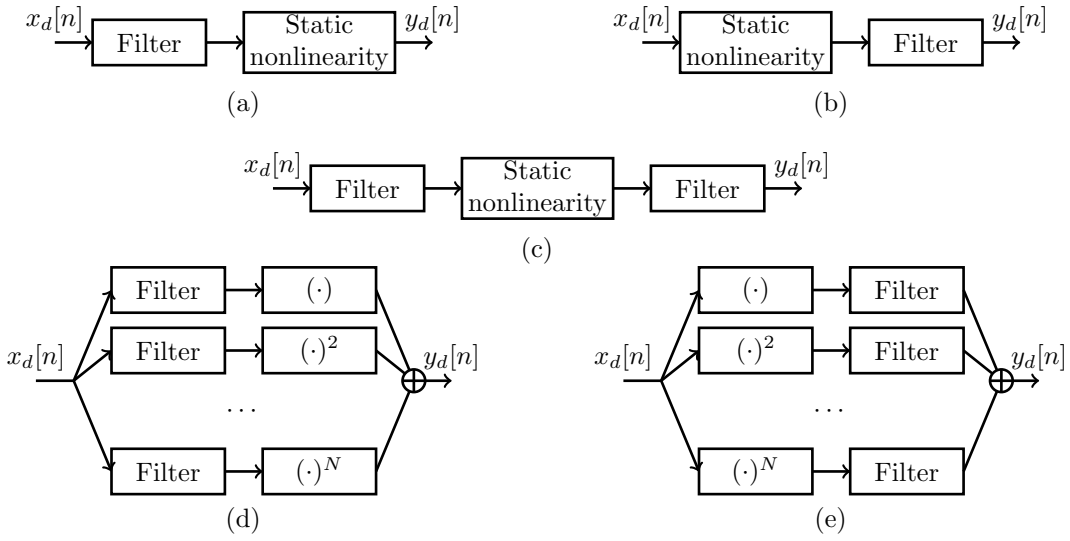


Figure 0.1: Examples of parametric nonlinear models for black-box modeling: (a) Wiener, (b) Hammerstein, (c) Wiener-Hammerstein, (d) polynomial Wiener, and (e) polynomial Hammerstein models.

One particularly popular approach for the modeling of nonlinear lumped audio models is polynomial Hammerstein models. Indeed, it is possible to parametrize an entire system through a simple procedure, where a single sine sweep is fed to the system and its output is recorded and processed accordingly [Novak et al. 2009, 2010a,b,c]. The computational complexity of the resulting model can be easily controlled as it is generally tied to the number and computational complexity of the filtering blocks in the system. Another potential benefit to these nonlinear models is that aliasing can sometimes be detected, analyzed and, ideally, suppressed [Hélie 2006].

0.2.2 White-box modeling

While black-box modeling offers a straightforward way to control the computational complexity of a model, its accuracy is often more difficult to evaluate a priori. Additionally, because its parametrization does not reflect any relation with any real physical quantities in the system (e.g., voltage, mass), it is often necessary to restart the parameter estimation from scratch as soon as any parameter is altered (e.g., if a resistor is changed due to a control knob). White-box modeling, also called physical modeling, is meant to circumvent these limitations in the scenarios where we have access to the behavioral equations of the target systems.

These approaches attempt to directly model the physics of the original system through these equations. As such, they make it easier to deploy and analyze the degree of approximation in the system. The core mathematical toolbox of white-box modeling is discretization numerical methods

intended to map equations expressed in continuous time to equations expressed in discrete time. Many numerical discretization approaches have been studied in the literature. Generally, more computational complexity (either as part of the method formula, or due to a shorter time stepping scheme) results in a more accurate model. However, classes of methods have also been tailored to better deal with some particular class of systems. One typical example of that is the class of stiff systems that we will discuss in more details in Ch. 3 for which many special methods have been designed to deliver higher accuracy at a lower computational cost. Additionally, some classes of methods can be preferred due to specific properties. The typical illustration of that fact is the variety of stability properties (e.g., A-stability, L-stability) associated with some classes of methods. Geometry-preserving methods which focus on preserving some invariants between original system and model are another example of these specialized methods [Hairer et al. 1993, Hairer and Wanner 1996, Hairer et al. 2006].

One of the core advantages of white-box modeling compared to black-box modeling is the flexibility of the models. Indeed, as the model integrates explicitly the physics of the system, it is generally trivial to include time-varying quantities (e.g., control signals) and to alter the models in physically meaningful way (e.g., change the size of a system).

0.2.3 Gray-box modeling

As we may expect, all existing methods do not always neatly fit into these two categories. For example, several black-box models of lower complexity can be organized together in a way reflecting the macro structure of the studied system and build a system of higher complexity [Eichas and Zölzer 2018, Kiiski et al. 2016]. Alternatively, the knowledge of the physics of the studied system can be partial (e.g., some physical quantities are unknown) and need to be deducted through input-output measurement analysis [Holmes and van Walstijn 2016]. It is also sometimes possible to use parametric black-box models to approximate directly the system equations, but focus on replicating the qualitative behavior of the system rather than create an equivalent system. In the case of linear systems, this kind of approaches includes the matched Z-transform [Golden 1968], the impulse-invariant and step-invariant methods [Parks and Burrus 1987, Smith III 2007b, Oppenheim and Schafer 2009] or integrator approximation methods [Al Alaoui 1993, Šekara and Stojić 2005, Al Alaoui 2006, Šekara 2006]. For weakly nonlinear systems, Volterra series can be parametrized analytically to form compact models with explicit aliasing control [Hélie 2006, 2010]. In general, there is no single definition of what qualifies as a gray-box modeling approach.

In this document, we primarily focus on methods that fall somewhat between white-box and gray-box modeling approaches, where we have access to the physical equations describing the behavior of the system.

0.3 Physical modeling formalisms

As outlined in Sec. 0.2.2, physical modeling generally refers to the direct modeling of the physics of a target system through the conversion of its behavioral continuous-time equations into discrete-time equations. Once the system's structure is identified and analyzed, multiple formalisms have been proposed to set the equations describing the relationship between the various physical quantities. The choice of a formalism also somewhat drives the way that identifying the equations and discretizing them is handled.

0.3.1 State-space formalism

The state space formalism focuses on summarizing the system's behavior through the differential equations that rule the trajectory of variable called state variables. By design, the values of these variables are necessary and sufficient predictors of the future behavior of the system for any input and control signals, without the need of any knowledge of the past trajectory of any quantity in the system. Often, these state variables will be tied to physical quantities that describe the distribution of the energy stored in the system. For example, in linear electric circuits, state variables can correspond to the voltage across the circuit's capacitors and the current through the circuit's inductors, as these quantities relate directly to the electrical energy stored in these components.

The system of equations to solve is a set of differential algebraic equations (though it can be turned into a set of ordinary differential equations in many cases). The first subset of equations links the time derivative of the state variables $\dot{\boldsymbol{x}}$ at time t to some function of the value of these state variables $\boldsymbol{x}$ at time t and the value of some input and/or control variables $\boldsymbol{u}$ at time t as well. The second subset links the value of one or several output variables of interest $\boldsymbol{y}$ at time t as a function of the state, input and/or control variables at time t. Altogether, it becomes

$$\begin{aligned} \dot{\boldsymbol{x}}(t) &= \boldsymbol{g}(t, \boldsymbol{x}(t), \boldsymbol{u}(t)), \text{ and} \\ \boldsymbol{y}(t) &= \boldsymbol{h}(t, \boldsymbol{x}(t), \boldsymbol{u}(t)). \end{aligned} \tag{0.2}$$

Deriving the system equations in this form is often impractical, as forming the function $\boldsymbol{g}$ and $\boldsymbol{h}$ often requires inverting systems without closed-form analytical solutions. As a result, several state-space-like formulation have been proposed that generally cover most of the lumped audio systems of interest, such as the K-method and its variants, namely the nodal K-method and the discrete K-method [Yeh et al. 2010, Yeh 2012], or the generalized state-space formulation [Holters and Zölzer 2015]. The literature provides systematic methods to generate the corresponding system of equations from typical system analysis methods such as modified nodal analysis [Vlach and Singhal 1993, Vlach 2002]. These methods have generally focused on two properties: isolate the nonlinear implicit equations from the linear parts of the circuits, and isolate the time-varying elements to mitigate the computational cost of parameter change [Holters and Zölzer 2016]. The state-space

formalism is somewhat attractive in most scenarios as it relates most directly to the way typical physical equations are laid out, and it can leverage a much wider sets of numerical tools from the literature, due to its prevalence across many fields.

0.3.2 Wave variable formalism

Wave variable formalism is primarily known in the context of wave digital filter theory [Fettweis 1971]. It was initially developed to systematically approach the discretization of electrical filter circuits in lattice or ladder configurations, but the formalism can be extended to any physical lumped system [Werner 2016] and to the continuous-time domain. Wave variable formalism is organized around describing each physical element in the system as transforming incident "wave" quantities $\boldsymbol{a}$ into reflected "wave" quantities $\boldsymbol{b}$ expressed at the element's "ports". In this context, the wave quantities are expressed as a linear combination of power conjugate variables (e.g., voltage-current for electrical circuits, force-velocity for mechanical systems), and that linear combination is defined up to a free parameter (sometimes referred to as port resistance $\boldsymbol{r}_p$). Several different options exist in the literature, such as voltage waves that are defined as

$$\boldsymbol{a}(t) = \boldsymbol{v}(t) + \boldsymbol{r}_p \circ \boldsymbol{i}(t) \tag{0.3}$$

for the *incident* wave and

$$\boldsymbol{b}(t) = \boldsymbol{v}(t) - \boldsymbol{r}_p \circ \boldsymbol{i}(t) \tag{0.4}$$

for the reflected wave, with $\boldsymbol{v}$ as branch voltages, $\boldsymbol{i}$ as branch currents and $\circ$ as the Hadamard (i.e., element-wise) product.

Each element e forms an implicit equation between its incident and reflected wave and their time derivatives at time t. The global behavior of the system is then fully described by connecting ports, i.e., matching reflected and incident wave quantities between various elements, with the constraint that the free parameter of two connected ports must be identical. Altogether, the system is described as

$$\begin{aligned} \mathbf{0} &= \boldsymbol{g}_e\Big(\dot{\boldsymbol{b}}_e(t), \dot{\boldsymbol{a}}_e(t), \boldsymbol{b}_e(t), \boldsymbol{a}_e(t)\Big), \quad \text{for all elements } e, \text{ and} \\ \boldsymbol{a}(t) &= \mathbf{C} \cdot \boldsymbol{b}(t) \end{aligned} \tag{0.5}$$

with $\mathbf{C}$ the symmetric connection matrix, i.e., a square matrix with dimensions equal to the total number of ports in the system, and with a single 1 in each column and row, and 0s everywhere else (the 1 indicating the connection between two ports). Recently published literature [Werner et al. 2015d, Werner 2016, Werner et al. 2018] provides a way to derive these equations using the modified nodal analysis of the system. For typical elements found in lumped systems, the MNA can be filled

up systematically using so-called "stamps" that can be used to fill out the relevant number in the MNA matrices.

When the system is expressed in discrete-time, the discretized elements are referred to as wave digital filters. Wave digital filters have found interest in the community thanks to a few desirable properties. First, the formalism favors a very modular description of the system by factoring the system using special elements called adapters which represent typical interactions between other types of elements (e.g., series or parallel connections in electrical circuits). That modularity also allows us to intuitively and efficiently segregate sections of the system that require the solving of implicit update equations, ensuring no superfluous variable is involved in their solving, and ensuring a minimal computational complexity load. Second, the update equations formed in the wave digital filter context have good numerical properties, in the sense that they allow for a port-by-port control of numerical passivity, and that when running the model in fixed-point arithmetic ensuring no "artificial energy" is injected into the system by quantization. Finally, recent developments have brought mathematical developments to the original theory (e.g., R-adaptors) that allowed for their application to most audio systems of interest [Abel et al. 2013, Werner and Smith III 2015, Werner et al. 2015d,a,b,c, Werner 2016, Werner et al. 2016a,b, Dunkel et al. 2016, Olsen et al. 2016, Bogason 2018].

0.3.3 Port-Hamiltonian formalism

The Port-Hamiltonian formalism focuses on discretizing the equations describing the dynamics of the so-called Hamiltonian of the target system. In most systems of interest for us, which are so-called closed systems (e.g., they do not exchange energy with any external system), the Hamiltonian happens to describe the total energy in the system. The literature has particularly examined the case where the port-Hamiltonian system admits an explicit realization [Falaize and Hélie 2016, Falaize 2017], in which case the system can be expressed as a system of equations between the time derivatives of the state variables $\dot{\boldsymbol{x}}$ (in the same sense as in the state-space formalism), the "dissipative" variables $\boldsymbol{w}$ (attached to dissipative elements, e.g., resistors) and the outputs $\boldsymbol{y}$ at time t on the one side, and the state variables $\boldsymbol{x}$, the "dissipative" functions $\boldsymbol{z}(\boldsymbol{w})$ and the input variables $\boldsymbol{u}$ (e.g., power sources) on the other side as

$$\begin{bmatrix} \dot{\boldsymbol{x}}(t) \\ \boldsymbol{w}(t) \\ -\boldsymbol{y}(t) \end{bmatrix} = \begin{bmatrix} \mathbf{J}_{\boldsymbol{x}} & -\mathbf{K} & -\mathbf{G}_{\boldsymbol{x}} \\ \mathbf{K}^{\mathsf{T}} & \mathbf{J}_{\boldsymbol{w}} & -\mathbf{G}_{\boldsymbol{w}} \\ \mathbf{G}_{\boldsymbol{x}}^{\mathsf{T}} & \mathbf{G}_{\boldsymbol{w}}^{\mathsf{T}} & \mathbf{J}_{\boldsymbol{y}} \end{bmatrix} \begin{bmatrix} \boldsymbol{x}(t) \\ \boldsymbol{z}(\boldsymbol{w}(t)) \\ \boldsymbol{u}(t) \end{bmatrix} \tag{0.6}$$

where the conversion matrix is skew-symmetric and indicates the energetic exchanges in the system between input sources, dissipative elements and storage elements. The published literature [Falaize and Hélie 2016, Falaize 2017] also provides a procedure to form this system of equations from the

physical equations governing the system, for example from the Kirchhoff's laws in the case of an electric circuit.

By keeping track explicitly of that quantity in the target system and its model, it becomes possible to design methods that ensure the preservation of the property of conservation of the Hamiltonian (i.e., for our systems, the conservation of energy). Hence, conservative systems (e.g., oscillators) can be guaranteed to be modeled as conservative discrete-time systems. A by-product of that fact is that these methods also guarantee that dissipative (or passive) systems are modeled as dissipative discrete-time systems (though without guarantee that the amount of dissipation is the same). While this formalism is somewhat less intuitive than the two exposed earlier, the formalism was developed to allow for a systematic discretization of many audio systems of interest such as loudspeakers, guitar effects or oscillator circuits [Falaize-Skrzek and Hélie 2013, Falaize and Hélie 2014, Falaize et al. 2014, Falaize and Hélie 2015, Falaize et al. 2015, Lopes et al. 2015, Falaize and Hélie 2016].

0.4 Numerical discretization for ordinary differential equations

Once a formalism has been selected and the systems equations have been identified, the next step is to select a procedure to convert these equations into discrete-time update equations. The goal is to find a set of discrete-time update equations for discrete-time versions of the variables in the original continuous-time systems. For example, we want to form the update equations for a series of discretized states $\boldsymbol{x}_d[n]$ at index n (i.e., time nT_s for a system discretized at sampling period T_s).

0.4.1 Classes of methods

As stated earlier, a wide variety of methods exist to perform this process. In general, methods can be categorized in three categories: numerical differentiation, numerical integration and ODE solvers. These categories are not exclusive, as some methods can be explained through the lenses of two or more categories. We will also mention a very different category of methods, the nonlinear methods.

Numerical differentiation

Numerical differentiation focuses on approximating the first-order time derivative found in all the equations using some linear combination of discrete-time samples. In the typical case of a fixed sampling period T_s, that approximation can be simply expressed as

$$\dot{\boldsymbol{x}}(nT_s) \approx \sum_{m=-\infty}^{\infty} a_m x[n-m]. \tag{0.7}$$

The accuracy of these methods are generally evaluated using the Taylor expansions of the quantities $x((n-m)T_s)$ around the time point nT_s, with the general objective that a larger number of non-zero a_m coefficients means more terms of the Taylor expansion residual are eliminated. More generally, we can approach the discretization of the differential operator as a discrete-time filter, such that

$$\dot{\boldsymbol{x}}(nT_s) \approx (h_\Delta * x)[n]. \tag{0.8}$$

In that case, Eq. (0.7) illustrates the case when that filter is a finite-impulse response filter, but the more general case allows for infinite-impulse response implementations as well. This kind of method can also be found under the denomination of s-to-z mapping, due to the fact that first-order discretization is equivalent to the variable s found in the context of the Laplace transform of linear equations, and that the method can be considered as replacing instances of s in the Laplace transform of the system by the z-transform of the filter h_Δ.

Another notable recent development in this category are the anti-derivative anti-aliasing differentiation operators developed in Bilbao et al. [2017a]. In cases where we have access to the anti-derivative functions of the nonlinearities in the system equations, we can form approximations of the differentiation operator that more specifically lower the amount of energy in higher (and aliased) harmonics for the model output, hence reducing the amount of aliasing. However, these methods still require some level of compromise as stronger aliasing suppression also results in higher distortion (i.e., coloration) in the lower harmonics [Bilbao et al. 2017b].

Numerical integration

An alternative view on the discretization process is numerical integration. Indeed, as we saw earlier, most methods end up having to solve equations of the form

$$\dot{\boldsymbol{x}}(t) = \boldsymbol{f}(t, \boldsymbol{x}(t)). \tag{0.9}$$

This equation can be rewritten after integration on both sides between nT_s and $(n+1)T_s$ such that

$$\boldsymbol{x}((n+1)T_s) = \boldsymbol{x}(nT_s) + \int_{nT_s}^{n(T_s+1)} \boldsymbol{f}(\tau, \boldsymbol{x}(\tau))d\tau. \tag{0.10}$$

In most scenarios, the integral cannot be calculated analytically. Numerical integration methods form approximations for that integral. Many approaches exist to approach that problem from simple fixed linear methods (i.e., where the integral becomes a fixed weighted sum of time-dependent factors, generally values of the function $\boldsymbol{f}$ evaluated at given time points) to more complicated methods such as extrapolation approaches and/or adaptive approaches. Many approaches generate approximate integrals by first finding a polynomial approximation of a given order for the relevant trajectories, and then integrate the approximate polynomial in closed-form.

Another notable member of this category is the so-called vector-field averaging methods [Muller and Hélie 2017]. The first-order version of these methods is equivalent to the recently published anti-derivative anti-aliasing methods [Parker et al. 2016, Zavalishin 2018], and to the discrete-Gradient method developed in the context of the port-Hamiltonian formalism [Falaize 2017, Muller and Hélie 2017]. In cases where the system nonlinearities have known anti-derivatives, these methods approximate the variable trajectories in the input space of the nonlinearities, rather than in the output space for other methods. This class of approaches has been shown to generally reduce the amount of energy in higher (and aliased) harmonics though they can introduce non-negligible coloration in the lower harmonics.

ODE solvers

While the two previous categories are essentially about using general-purpose approximations of relevant mathematical operators to generate discrete-time update equations, a different class of methods is specifically targeted at the problem of generating these update equations. Many well-known numerical tools belong to this category, such as the Runge-Kutta methods, or more generally many predictor-corrector methods. These methods are based on forming intermediary estimators for points at different time stamps along the trajectory of the different system variables and combine them to form high-accuracy update equations.

Nonlinear methods

The various categories mentioned above cover the major part of the discretization tools used in the present day literature. There do exist methods that do not belong to these categories, in the sense that they are not linear methods. The term linear here refers to the fact that methods can be fully characterized by a series of system-independent weights that are applied to various weighted sum formulas in order to form the discretization procedure. But several nonlinear methods exist in the literature, meaning methods whose parameters change depending on the value of variables in the system. One notable example is the case of exponential integrators [Hochbruck and Ostermann 2010, Hélie 2010]. In that case, the trajectories of the system variables are computed as the exponential solution for the Jacobian of the system at the current time, which would correspond to the analytic continuous-time solution if the system were linear. The extrapolation is then stopped at the next update point, the Jacobian gets updated and a new segment of extrapolation is generated. This method is then nonlinear since all the coefficients in the update equations are dependent on the current state of the system at the current time.

0.4.2 Stability, order of accuracy, consistency and convergence

Traditional analysis of discretization approaches relies on the concepts of stability, accuracy, consistency and convergence.

Stability

Stability generally refers to the fact that the overall error between the simulated quantities and their equivalents in the original system does not grow over time when the system inputs are bounded [Moin 2010]. Unconditional stability refers to the case where we can show the error will not grow independently of the studied system. Conditional stability is the more common case, when the error stays bounded only when the system verifies some condition. While this general definition about bounded output error matches the typical definition of A-stability, it is also the weakest. Stronger definitions exist in the literature, such as L-stability, to characterize the behavior of the error for some classes of systems such as stiff systems [Hairer and Wanner 1996]. The criteria to establish the stability of a method is generally assessed for the case of time-invariant linear systems so that stability is generally not truly guaranteed for nonlinear and/or time-varying systems, but these criteria are still generally good guidelines to compare methods.

Order of accuracy

Assessing objectively the true accuracy of a method is generally intractable. One accessible metric that is often used in the literature is the asymptotic concept of order of accuracy [Moin 2010]. It corresponds to the order in time of the leading term of the truncation error, i.e., the residual when analyzing the Taylor expansion of the discretization method around a given time index. Reaching higher orders of accuracy generally requires more complex methods in order to eliminate more terms in the residual. While the order of accuracy offers an easy way to compare methods and is an essential part of some approaches (e.g., adaptive time-step algorithms), it is actually a poor predictor of accuracy in many practical settings, especially in the audio context where the sampling period is fixed and relatively large, but the literature has failed to provide an alternative.

Consistency

Consistency is another asymptotic property of discretization methods [Moin 2010]. It refers to the property that the truncation error vanishes as the time step goes to zero. It means the update equations converges to the true differential equation. As a result, it is equivalent to say that the method is at least first-order accurate. As for the order of accuracy, it is a property of limited use in contexts where shortening the time step is unlikely.

Convergence

Convergence is the last asymptotic property generally studied in the context of discretization methods [Moin 2010]. It refers to the fact that for identical boundary and/or initial conditions, and as the time step vanishes, the trajectories of the discrete-time variables converges to the trajectories of their equivalents in the original continuous-time system.

0.4.3 Explicit and implicit methods

Another general criterion of classification for various approaches to discretization is the distinction between implicit and explicit methods. These two classes generally lead to very contrasting compromises between stability and computational complexity.

Explicit methods

Explicit methods refers to methods for which the value of the individual state variables at a given time can be expressed as an closed-form explicit equation for all systems [Moin 2010], i.e., an expression of the form

$$\boldsymbol{x}_d[n+1] = \boldsymbol{g}_d(t_{n+1}, \ldots, t_{n-M}, \boldsymbol{x}_d[n], \ldots, \boldsymbol{x}_d[n-M], \boldsymbol{u}[n+1], \ldots, \boldsymbol{u}[n-M]). \tag{0.11}$$

The better known methods in this category include the forward Euler method, the leapfrog method, the Adams-Bashforth methods and the explicit Runge-Kutta methods (in particular, the ubiquitous Runge-Kutta 4 method). The main drawback of these methods is generally their poor stability properties, which makes them unsuitable to simulate strongly nonlinear systems with a typical audio sampling period. However, the update equations they generate are generally much cheaper to solve computationally, so that they are often preferred in the context of adaptive-step methods. Indeed, it is sometimes as computationally costly to solve an explicit method update equation over a smaller sampling period as to solve an equally accurate implicit method update equation over the nominal sampling period [Yeh 2009].

Implicit methods

Implicit methods correspond to methods when finding the individual state variables requires solving a coupled set of equations where a closed-form explicit equation is not guaranteed [Moin 2010], i.e., an implicit expression of the form

$$\boldsymbol{x}_d[n+1] = \boldsymbol{g}_d(t_{n+1}, \ldots, t_{n-M}, \boldsymbol{x}_d[n+1], \ldots, \boldsymbol{x}_d[n-M], \boldsymbol{u}[n+1], \ldots, \boldsymbol{u}[n-M]). \tag{0.12}$$

The better known methods in this category include the backward Euler method, the trapezoidal method, the Adams-Moulton methods and the implicit Runge-Kutta methods. It also includes the

vector-field averaging methods mentioned earlier. These methods can generally be designed to have strong stability properties. For example, the trapezoidal method and the backward Euler method are two typical implicit methods who are unconditionally A-stable [Dahlquist 1963]. Hence, it is generally possible to run sampling periods large enough for audio applications. On the other hand, their computational complexity is generally higher due to the necessity to solve the implicit update equations. Additionally, stability does not guarantee good accuracy.

In the typical case where we cannot invert the function $\boldsymbol{g}_d$ in closed form, the most accurate approach is to use a numerical root finder, such as Newton-Raphson. In some cases however, implicit methods can sometimes still lead to reasonable explicit approximations. The wave digital filter formalism (see also Secs. 0.3.2 and 4.2.2 for details) attempts to isolate an implicit equation of dimension as small as possible while allowing for an explicit update of the rest of the system [Werner 2016]. A similar motivation applies to state-space-like approaches such as the ones using the K-method [Yeh et al. 2010]. If the dimensionality of the implicit equations is low enough, it is often possible to build reasonable approximate based on piecewise interpolation or tabulation. The way the equation that is fed to the root solver is formed can also non-trivially affect the overall computation cost of the system [Yeh et al. 2010, Holmes and van Walstijn 2015, Olsen et al. 2016, Werner et al. 2018]. Finally, the parameters for some classes of numerical methods can be tuned to reveal an explicit update with the proper change of variables [Lopes et al. 2015].

Semi-implicit methods

If we can express the iterations of the numerical root finder in closed form (such as in the case of Newton-Raphson), it is possible to form a nonlinear method with explicit update equations based on a fixed amount of root finder iterations. For example, for update equations with a differentiable $\boldsymbol{g}_d$, we can form an explicit update equations with the help of the update Jacobian by integrating a single Newton-Raphson iteration [Yeh 2009]. These methods are however somewhat rarely used, in part because their computational cost comes with limited accuracy gains, and in part because, as nonlinear methods, they are somewhat more complex to analyze.

0.5 Modeling considerations in lumped audio systems

Altogether, the choice of formalism and discretization method is a compromise between accuracy, flexibility, and general complexity, both in terms of derivation and computation. In the case of the formalisms, while more naive implementations are more straightforward to derive systematically, careful selection of model structure and variables can allow for a lower operation count [Werner et al. 2018], and the identification and isolation of a minimal set of implicit equations in order to compute as many variables as possible with simple explicit updates. In the case of the discretization method, the user must make a compromise choice to select a method stable for its application, and

sufficiently accurate while limiting the computational cost. As part of that selection, the user must select a time-stepping strategy, fixed or variable, and adapted to the target application. In audio applications, an additional concern is the problem of aliasing distortion (see Sec. 1.3.3 and Ch. 2) which also needs to be considered in the model and time-step selection.

0.6 Summary of the dissertation

The following manuscript is structured as follows. In Chapter 1, we describe a consistent framework for the design and evaluation of discrete-time models of continuous-time systems. This framework rigorously resolves the intrinsic indeterminacy involved in comparing continuous-time and discrete-time system, and allows for a generalized definition of aliasing. In Chapter 2, we derive new analytical and empirical tools that allow for the advanced and efficient comparison of aliasing distortion among discretization approaches for a given system. In Chapter 3, we present a generalized approach to one-step discretization methods based on the Möbius transform formalism. In that context, we describe different criteria for selecting a preferred parametrization of these transforms for a given system of interest. In Chapter 4, we develop a novel approach to the design of the discrete-time models for linear electrical systems. It is designed to improve the accuracy of these models by jointly tuning the discretization of the individual constitutive elements of that system.

Chapter 1

Signal similarity definition and aliasing

In this chapter, we present a unified discussion of the concept of signal similarity in the context of fixed-rate computer simulation from a mathematical and perceptual perspective. We show how understanding and properly defining this similarity naturally connects to the well-known concept of aliasing. This discussion allows us to draw guidelines for the design of an unambiguous comparison framework among different methods and approaches when optimizing the computer model of a given analog system. Designing such framework is an essential step to present a well-posed modeling problem.

First and foremost, the definition of similarity for signals must acknowledge the essential difference in nature between the *continuous-time* signals existing in the physical (analog) audio systems we would like to model, and the *discrete-time* sequences used on our computer models as input, control and output, as can be seen in Figs. 1.1 and 1.2. The process to reconcile these two domains raises a variety of questions, especially the question of aliasing. Analyzing and reducing that aliasing must then be discussed in that context.

Second, the definition of similarity for signals must be informed by the nature of human auditory perception, as we should generally be guided by achieving, at minimum, perceptual similarity. This

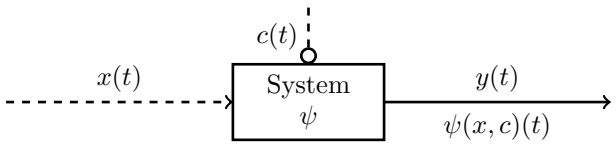


Figure 1.1: Generic real-world audio system, with (optional) continuous-time input $x(t)$ and control $c(t)$ signals and continuous-time output signal $y(t)$.

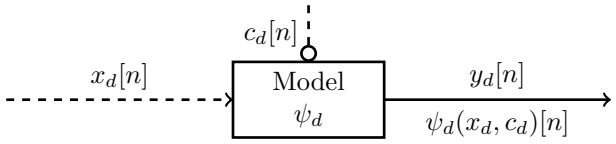


Figure 1.2: Generic audio computer model, with (optional) discrete-time input $x_d[n]$ and control $c_d[n]$ signals and discrete-time output signal $y_d[n]$.

means that we would like to design a rule such that two signals can only be considered identical if they are *perceptually identical*.

1.1 Human perception and perceptual similarity

When listening to audio, humans perceive only frequencies in a finite range, often summarized as roughly between 20 Hz and 20 kHz. As a result, the audio content outside that range is generally inaudible and could be ignored in the context of audio modeling, in the sense that two signals with identical content *in the audible band* could generally be considered identical *overall*, because a human listener should not be able to perceive any difference between these signals. Thus, to be considered equivalent, two systems only need to generate signals with matching spectral content in the limited audible band. This "band-limited" objective is an important distinction with other disciplines, and hints at a different approach when drawing guidelines for design and comparison of computer audio models, especially in the context of fixed audio rate simulation.

In mathematical terms, if we have two systems that generate two signals x_1 and x_2, we can define a necessary condition for similarity using their Fourier transforms X_1 and X_2 as

$$x_1 \equiv x_2 \quad \Rightarrow \quad X_1(f) = X_2(f),\ \forall f \in [20\,\text{Hz}, 20\,\text{kHz}]\,. \tag{1.1}$$

Obviously, this condition would be trivially verified if the two signals were equal. However, as we will see in later chapters, building a computer model of a system that satisfies this condition is generally very difficult in the context of fixed-rate simulation, so this condition can form a lower bound on signal similarity definition.

Note that such a condition could be considered conservative, considering the complexity of human auditory perception. Indeed, mechanisms such as temporal and spectral maskings create additional layers of errors imperceptible to human listeners, and such mechanisms can be exploited as in the case of audio coding [Bosi and Goldberg 2002]. Describing these in mathematical terms is however quite complex compared to the bound in Eq. (1.1), so we set it as our benchmark for perceptual similarity in the rest of the manuscript. The following sections will show how this discussion of signal similarity between systems fits within the broader problem of rigorously defining the relationship between the continuous-time and the discrete-time domains.

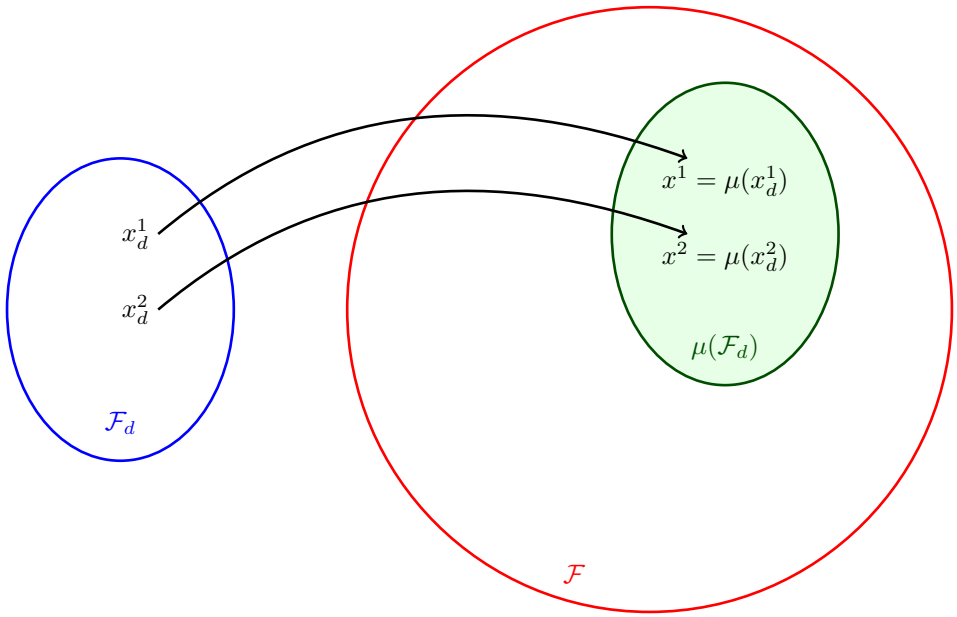


Figure 1.3: Illustration of interpolation as bijective mapping μ between the set of discrete-time sequences $\mathcal{F}_d$ and then a subset $\mu(\mathcal{F}_d)$ of the set of continuous-time signals $\mathcal{F}$

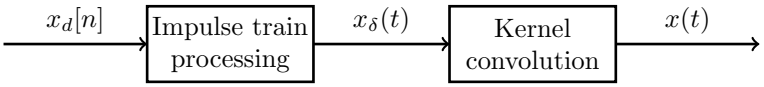


Figure 1.4: Two-step decomposition of interpolation with an impulse train transform followed by a convolution by a chosen interpolation kernel.

1.2 Discrete-to-continuous conversion

1.2.1 Interpolation

Interpolation methods are designed to convert discrete-time sequences into continuous-time signals. Multiple methods have been proposed in the literature, many of them being applicable to the case of fixed-rate sample sequences.

The nature of the interpolation method defines a subset of continuous-time signals that can be *represented* by a discrete-time sequence. In general, for a given sampling rate f_s, it is impossible to represent the entire set of continuous-time signals as a discrete-time sequence. On the other hand, typical interpolation methods are such that the discrete-time sequence associated with a given continuous-time function is unique, meaning interpolation methods are generally an injective mapping μ from the set of discrete-time sequences $\mathcal{F}_d$ to the set of continuous-time functions $\mathcal{F}$, and hence are a bijective mapping from $\mathcal{F}_d$ to a subset $\mu(\mathcal{F})$ as illustrated in Fig. 1.3.

For many typical interpolation methods, it is often interesting to analyze the relationship between

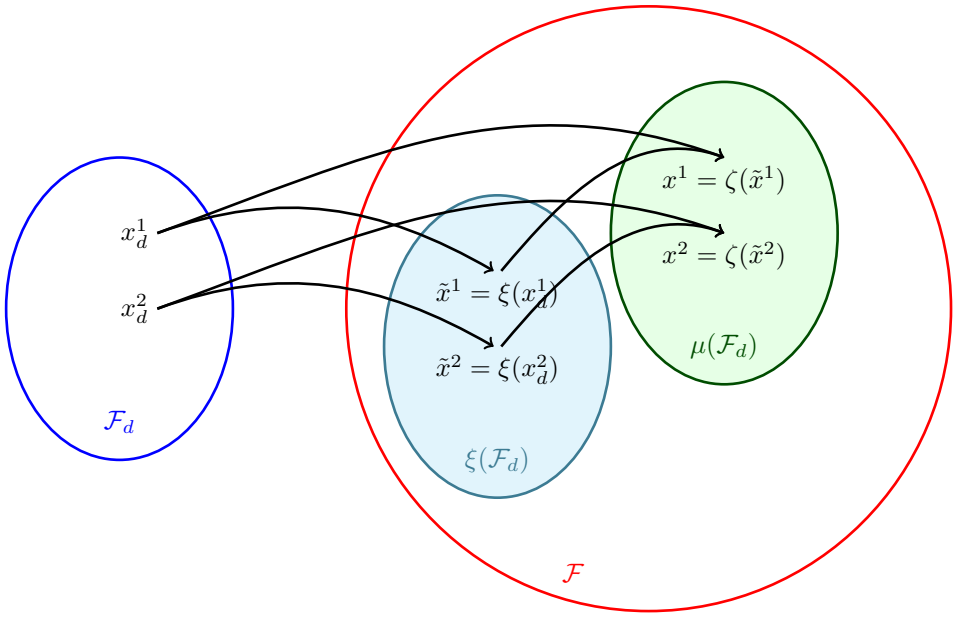


Figure 1.5: Illustration of interpolation as two successive bijective mappings: ξ between the set of discrete-time sequences $\mathcal{F}_d$ and the subset $\xi(\mathcal{F}_d)$ of the set of continuous-time signals $\mathcal{F}$, followed by ζ between the subset $\xi(\mathcal{F}_d)$ and the final subset $\mu(\mathcal{F}_d)$.

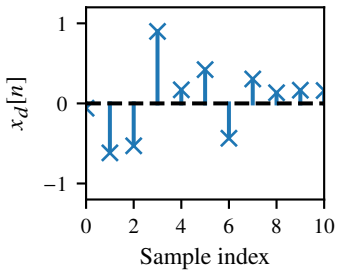


Figure 1.6: Example of discrete-time sequence. Samples outside of the displayed range are assumed to be zero for the purpose of interpolation.

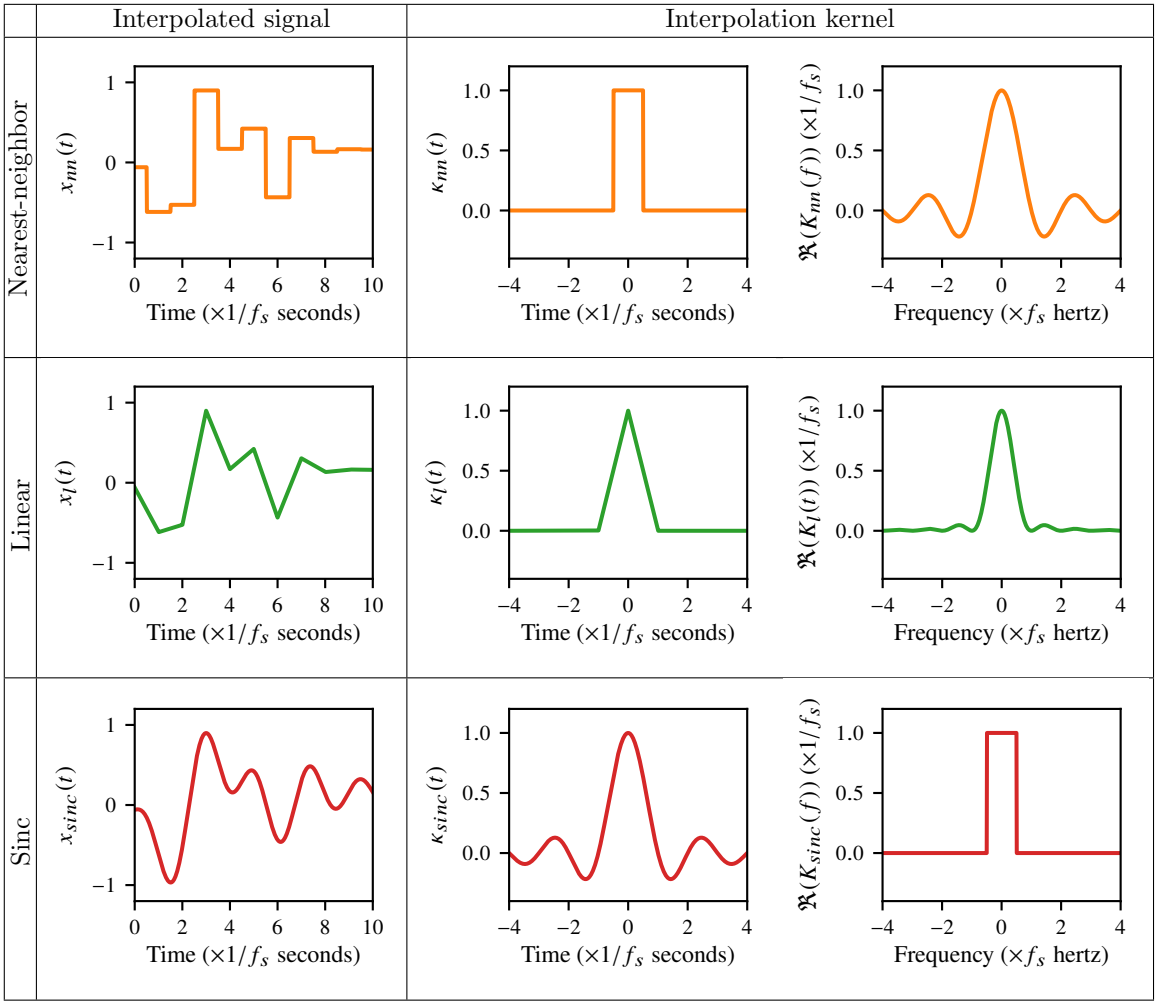


Table 1.1: Interpolated signal from the discrete sequence example in Fig. 1.6, and interpolation kernel for nearest-neighbor, linear and sinc interpolation methods. For the kernels, we provide the time-domain representation and the real part of their frequency-domain representation. The imaginary part is uniformly zero.

the discrete-time Fourier transform $X_d = \mathrm{DTFT}(x_d)$ of the discrete-time sequence x_d and the Fourier transform $X = \mathrm{FT}(x)$ of its interpolated counterpart x. Indeed, these methods can be interpreted as the 2-step process shown in Fig. 1.4:

1. Map the discrete-time sequence x_d to an impulse train $\tilde{x}$ with each impulse matching its corresponding sample in the sequence, as

$$\tilde{x}(t) = \xi(x_d)(t) = \sum_{n\in\mathbb{Z}} \delta(tf_s - n)x_d[n]. \tag{1.2}$$

2. Convolve $\tilde{x}$ with a time-domain interpolation kernel κ, such that the interpolated signal $\mu(x_d)$ is

$$\mu(x_d)(t) = \zeta(\tilde{x})(t) = (\tilde{x} * \kappa)(t) = \int_{\tau\in\mathbb{R}} \tilde{x}(\tau).\kappa(t-\tau)d\tau \tag{1.3}$$

From that process, we get the following:

- Eq. (1.2) means that the discrete-time Fourier transform $X_d = \mathrm{DTFT}(x_d)$ and the Fourier transform $\widetilde{X} = \mathrm{FT}(\tilde{x})$ are the same, i.e.,

$$X_d(f) = \widetilde{X}(f),\ \forall f \in \mathbb{R}. \tag{1.4}$$

  In particular, this implies the well-known fact that $\widetilde{X}$ is f_s-periodic, i.e.,

$$\widetilde{X}(f) = \widetilde{X}(f + f_s),\ \forall f \in \mathbb{R}. \tag{1.5}$$

- Eq. (1.3) means that the Fourier transform $X = \mathrm{FT}(x)$ can be computed as the Fourier transform $\widetilde{X}$ multiplied by the Fourier transform K of the interpolation kernel κ, as

$$X(f) = \widetilde{X}(f) \times K(f),\ \forall f \in \mathbb{R}. \tag{1.6}$$

Altogether, we get that the Fourier transform of the interpolated signal $x = \mu(x_d)$ can be summarized as a multiplication of the discrete-time Fourier transform $X_d = \mathrm{DTFT}(x_d)$ by a frequency-domain interpolation kernel K as

$$X(f) = X_d(f) \times K(f),\ \forall f \in \mathbb{R}. \tag{1.7}$$

Describing the system as a two-step process means that we first map the discrete-time sequences bijectively to a fixed subset $\xi(\mathcal{F}_d)$ of the continuous-time functions (i.e., the set of impulse train functions), and then re-map them through a second bijective mapping (through convolution) to the final subset $\mu(\mathcal{F}_d)$ of continuous-time functions.

Below, we present three common types of interpolation methods and their associated properties. Using the discrete-time sequence example shown in Fig. 1.6, we show the resulting interpolated continuous-time signals for each method, along with their associated interpolation kernel in the time and frequency domains in Tab. 1.1.

Nearest-neighbor interpolation

For a given sampling rate f_s, the nearest-neighbor interpolation dictates that a discrete-time sequence x_d gets mapped to $\mu_{nn}(x_d)$ defined as

$$\mu_{nn}(x_d)(t) = x_d\left[\text{floor}\left(tf_s + \tfrac{1}{2}\right)\right],\ \forall t \in \mathbb{R} \tag{1.8}$$

which means that, assuming that the "time stamp" of the nth sample in the discrete-time sequence is n/f_s, we assign to each time point of the continuous-time function the value of the sequence sample whose time stamp is the closest (picking the lowest time stamp in case the time point is equidistant from two samples).

The resulting continuous-time functions present a piecewise constant shape, as they are by design constant over each interval $\left[\frac{n-1/2}{f_s}, \frac{n+1/2}{f_s}\right[,\ \forall n \in \mathbb{Z}$. Furthermore, the subset $\mu_{nn}(\mathcal{F})$ actually corresponds to set of all continuous-time functions that are piecewise constant over the intervals $\left[\frac{n-1/2}{f_s}, \frac{n+1/2}{f_s}\right[,\ \forall n \in \mathbb{Z}$.

As for the Fourier transform $X = \text{FT}(x)$, this interpolation corresponds to a rectangular time-domain kernel, such that

$$\kappa_{nn}(t) = 1_{[-1,1[}\left(\frac{tf_s}{2}\right),\ \forall t \in \mathbb{R} \tag{1.9}$$

so that, in the frequency domain, it corresponds to the sinc function kernel, i.e.,

$$K_{nn}(f) = \frac{\sin\left(\pi f/f_s\right)}{\pi f},\ \forall f \in \mathbb{R}. \tag{1.10}$$

Linear interpolation

For a given sampling rate f_s, the linear interpolation dictates that a discrete-time sequence x_d gets mapped to $\mu_l(x_d)$ defined as

$$\begin{aligned}\mu_l(x_d)(t) = {} & \left(\text{floor}\left(tf_s + 1\right) - tf_s\right) x_d\left[\text{floor}\left(tf_s\right)\right] \\ & + \left(tf_s - \text{floor}\left(tf_s\right)\right) x_d\left[\text{floor}\left(tf_s + 1\right)\right],\ \forall t \in \mathbb{R}.\end{aligned} \tag{1.11}$$

which means that, assuming that the "time stamp" of the nth sample in the discrete-time sequence is n/f_s, we assign to each time point of the continuous-time function the value linearly interpolated between the sample immediately before and the sample immediately after in the sequence as per

their time stamps.

The resulting continuous-time functions present the well-known piecewise linear shape, as they are by design linear over each interval $\left[\frac{n}{f_s}, \frac{n+1}{f_s}\right[$, $\forall n \in \mathbb{Z}$. Furthermore, the subset $\mu_l(\mathcal{F})$ actually corresponds to set of all continuous-time functions that are piecewise linear over the intervals $\left[\frac{n}{f_s}, \frac{n+1}{f_s}\right[$, $\forall n \in \mathbb{Z}$.

As for the Fourier transform $X = \mathrm{FT}(x)$, this interpolation corresponds to a triangular time-domain kernel, such that

$$\kappa_l(t) = \max\left(0, 1 - |t|\right), \ \forall t \in \mathbb{R} \tag{1.12}$$

and

$$K_l(f) = f_s \times \left(\frac{\sin\left(\pi f / f_s\right)}{\pi f}\right)^2, \ \forall f \in \mathbb{R}. \tag{1.13}$$

As a reminder, we actually have the relation

$$\kappa_l(t) = f_s \times (\kappa_{nn} * \kappa_{nn})(t), \ \forall t \in \mathbb{R} \tag{1.14}$$

and consequently, as can be easily observed, the relation

$$K_l(f) = f_s \times (K_{nn}(f))^2, \ \forall f \in \mathbb{R}. \tag{1.15}$$

Sinc interpolation

For a given sampling rate f_s, the sinc interpolation dictates that a discrete-time sequence x_d gets mapped to $\mu_{\mathrm{sinc}}(x_d)$ defined as

$$\mu_{\mathrm{sinc}}(x_d)(t) = \sum_{n \in \mathbb{R}} \frac{\sin[\pi(t f_s - n)]}{\pi(t f_s - n)} x_d[n], \ \forall t \in \mathbb{R}. \tag{1.16}$$

The name of the interpolation comes from the fact that this interpolation corresponds to a sinc time-domain interpolation kernel κ_{sinc} such that

$$\kappa_{\mathrm{sinc}}(t) = \frac{\sin[\pi t f_s]}{\pi t f_s}, \ \forall t \in \mathbb{R}. \tag{1.17}$$

and the frequency-domain interpolation kernel is a rectangular function, i.e.,

$$K_{\mathrm{sinc}}(f) = 1_{\{-f_s/2 \leq f < f_s/2\}}, \ \forall f \in \mathbb{R}. \tag{1.18}$$

As a result, we see that the subset $\mu_{\mathrm{sinc}}(\mathcal{F})$ corresponds to the set of all continuous-time functions band-limited below $f_s/2$, i.e., functions x such that their Fourier transform $X = \mathrm{FT}(x)$ verifies

$$X(f) = 0, \ \forall f \in \mathbb{R} \setminus \left[-\tfrac{f_s}{2}, \tfrac{f_s}{2}\right[. \tag{1.19}$$

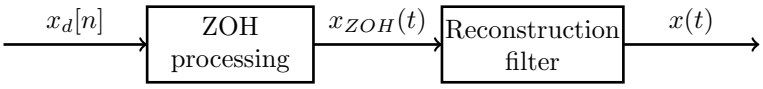


Figure 1.7: Two-step decomposition of a practical D/A convertor with a zeroth-order hold transform followed by a reconstruction filter.

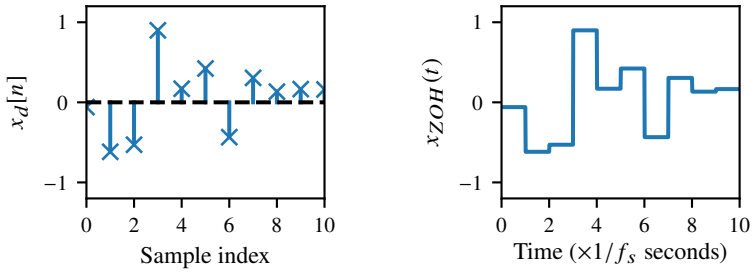


Figure 1.8: Example of zeroth-order hold converted signal x_{ZOH} (right) for the discrete sequence x_d from Fig. 1.6 (left).

Contrarily to the previously mentioned interpolation methods, this method has a band-limited frequency-domain kernel, while the time-domain kernel is not time-limited. As a result, every time point in the continuous-time function depends on every single sample in the discrete-time sequence (past *and* future). This fact means that exact sinc interpolation is generally impossible to implement exactly in practice. As we see later, it is however an important concept due to considerations related to perception and aliasing.

1.2.2 Representativity limitations due to conversion injectivity

As mentioned in Sec. 1.2.1, at best, discrete-to-continuous conversion is generally an injective, but not a bijective transform between the set of discrete-time sequences and the set of continuous-time functions, due to the limited amount of information that can be encoded in a discrete-time sequence. That means that every single continuous-time function of a subset of all continuous-time functions can be converted from a unique discrete-time sequence, but not all continuous-time functions can be obtained through such a conversion.

We must take note of such reality because it has practical implications on the design of a comparison framework for fixed-rate simulation. Indeed, it implies that a behavior in the computer model would appear not to be able to represent a large collection of behaviors in the continuous-time system, so that it will us require to decide what to do for these unrepresented behaviors.

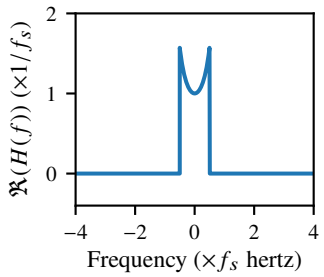


Figure 1.9: Real part of the reconstruction filter H from Eq. (1.24). The imaginary part is uniformly zero.

1.2.3 Digital-to-analog conversion in practice

In practice, the conversion between discrete-time sequences and continuous-time signals is performed using digital-to-analog (abbreviated D/A) convertors. Such devices are ubiquitous in audio hardware (e.g., soundcards) due to the fact that the devices used to deliver audio content to human listeners (e.g., loudspeakers, headphones) require continuous-time (i.e., analog) input signals.

Real D/A convertors need to deal with two problems at once:

- The discrete nature of the computer signals *in time* (i.e., sampling), and
- The discrete nature of the computer signals *in quantity* (i.e., quantization).

The discussion regarding the issue of quantization is beyond the scope of this document, but it is however not to be outright dismissed in well-designed system. In particular, this is of interest for systems based on integer arithmetic and low bit depth. We refer the reader to [Bosi and Goldberg 2002, Pelgrom 2013] for more information on the impact of quantization on signal processing algorithms. Additional considerations beyond our scope are the imperfect aspects of hardware elements used in such convertors (e.g., component tolerances, electronic noise).

A typical architecture for D/A converters resembles the 2-step process described at the end of Sec. 1.2.1 and in Fig. 1.4, following instead the process shown in Fig. 1.7 as:

1. The discrete-time sequence is converted into a coarse continuous-time waveform. Instead of the impulse train we used in Sec. 1.2.1, historical converters use a *zeroth-order hold* (ZOH) which returned a piecewise stepwise signal x_{ZOH} such that

$$x_{ZOH}(t) = x_d\left[\text{floor}(tfs)\right], \ \forall t \in \mathbb{R}. \tag{1.20}$$

   An example of zeroth-order hold processing is shown in Fig. 1.8. Note that this looks similar to the nearest-neighbor interpolation discussed in Sec. 1.2.1, except that it instead uses the nearest *previous* neighbor to decide the value of each time point in the output signal. This

has the critical advantage of making the system *causal*, while limiting the added *latency* (i.e., delay).

2. Smoothing the signal by applying a *reconstruction filter* of impulse response h (and frequency response $H = \mathrm{FT}(h)$) to x_{ZOH}, which has a similar function as our interpolation kernel and produces the final continuous-time output signal.

In that process, the relationship between $X_d = \mathrm{DTFT}(x_d)$ and $X = \mathrm{FT}(x)$ becomes

$$X(f) = \frac{\sin(\pi f/f_s)}{\pi f} \underbrace{\exp(-j\pi f/f_s)}_{\mathrm{e}} H(f) X_d(f), \ \forall f \in \mathbb{R}. \tag{1.21}$$

The linear phase term translates the $1/2f_s$ latency (corresponding to half a sampling period) introduced by the zeroth-order hold operation. We see that this process is compatible with the 2-step process described in Fig. 1.4 if we write the interpolation kernel as

$$K(f) = \frac{\sin(\pi f/f_s)}{\pi f} \exp(-j\pi f/f_s) H(f). \tag{1.22}$$

In a typical audio converter [Smith 1997], the smoothing filter is chosen to achieve matching spectral content in the base band $\left[-\frac{f_s}{2}, \frac{f_s}{2}\right[$ (minus the $1/2f_s$ delay) between x_d and x, i.e., so that we verify

$$X(f) \exp(j\pi f/f_s) = \begin{cases} X_d(f) & \text{for } f \in \left[-\frac{f_s}{2}, \frac{f_s}{2}\right[, \\ 0 & \text{otherwise.} \end{cases} \tag{1.23}$$

Then, the reconstruction filter is designed to verify the frequency response in Fig. 1.9, i.e.,

$$H(f) = \begin{cases} \dfrac{\pi f}{\sin(\pi f/f_s)} & \text{for } f \in \left[-\frac{f_s}{2}, \frac{f_s}{2}\right[, \\ 0 & \text{otherwise.} \end{cases} \tag{1.24}$$

Of course, it is generally impossible to build hardware system that ideally perform the tasks described in the 2-step process above for various technical reasons.

In particular, any practical D/A conversion system necessarily has to be *causal*, implying that it will necessarily exhibit some level of *latency* in a real-time processing context, which differs from the interpolation methods discussed before which are non-causal but have zero latency. This means that these interpolation method can be applied as such only in an off-line processing context.

This also means that implementing a filter such as the one described in Eq. (1.24) is impossible, since its time support extends infinitely in the future (as a consequence of having a finite frequency support [Hörmander 1963]). Matching the response in Eq. (1.24) (or alternatively a filter with same magnitude response but an added linear phase term) then becomes a compromise between latency and precision.

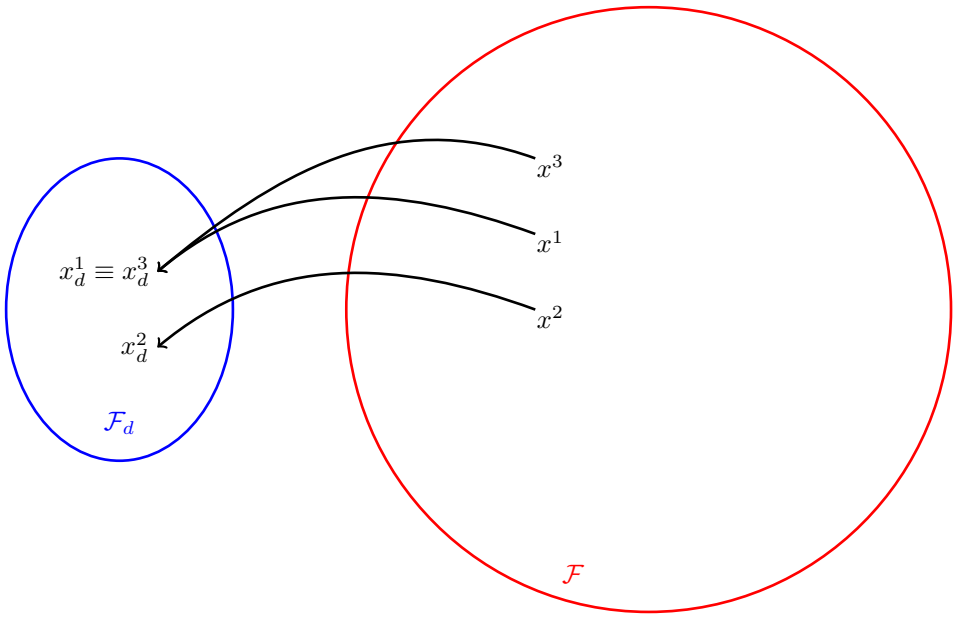


Figure 1.10: Illustration of continuous-to-discrete conversion as surjective mapping between the set of continuous-time signals $\mathcal{F}$ and the set of discrete-time sequences $\mathcal{F}_d$.

1.2.4 Digital-to-analog conversion in this discussion

Again, discussions regarding quantization or non-ideal hardware components is beyond the scope of our discussion, as we want to focus on the signal processing implications of discrete-to-continuous conversion, in particular *realizability* even as quantization and hardware imperfection concerns are absent.

Furthermore, in the rest of this discussion, we will abstract discrete-to-continuous conversion as following the impulse-train 2-step process described at the end of Fig. 1.4 rather than the zeroth-order hold process. We will also assume that the first step of the process is performed perfectly, as it is designed to be realizable. We will however allow a non-ideal second step as ideal processing generally necessitates non-realizable processing. However, we state that most of the discussion applies readily to zeroth-order hold-based systems, even though the design of the processing for their second step is somewhat more complex.

1.3 Continuous-to-discrete conversion

1.3.1 Ambiguity and conversion surjectivity

Due to the limited amount of information that can be encoded in a discrete-time sequence, it is impossible to convert each continuous-time functions into a unique discrete-time sequence. At best,

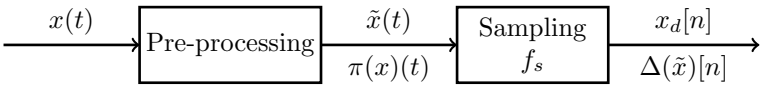


Figure 1.11: Two-step decomposition of continuous-to-discrete conversion with a pre-processing stage followed by waveform sampling at sampling rate f_s.

we can expect the continuous-to-discrete conversion to be surjective, meaning that all discrete-time sequences can be obtained from the conversion of a least one continuous-time functions, but more than one continuous-time function gets converted into the same discrete-time sequence as illustrated in Fig. 1.10.

This reality also has practical implications on the design of a comparison framework for fixed-rate simulation. Indeed, it implies that multiple different behaviors in a continuous-time system could end up represented with the same behavior in the discrete-time computer model, so that deciding on if the behavior of the computer model is optimal will require deciding with which of the continuous-time behaviors we are comparing it.

1.3.2 Two-step process with fixed-rate sampling

To convert a continuous-time function $x(t)$ into a discrete-time sequence $x_d[n]$ at a fixed sampling rate f_s, we reduce the procedure to the two-step process shown in Fig. 1.11, as:

1. Pre-process the continuous-time function to obtain another continuous-time function $\tilde{x}(t) = \pi(x)(t)$, and

2. Sample the continuous-time function $\tilde{x}(t)$ at rate f_s to form the discrete sequence $x_d[n] = \Delta(\tilde{x})[n]$, so that
$$x_d[n] = \tilde{x}(n/f_s). \tag{1.25}$$

Direct sampling is achieved when the pre-processing function π is the identity function, so that $x \equiv \tilde{x}$.

1.3.3 Fixed-rate sampling and aliasing

Fixed-rate sampling is a surjective conversion operator Δ between the set of continuous-time functions and the set of discrete-time sequences as the conversions described in Sec. 1.3.1. Indeed, the sampling operation at rate f_s applied to a continuous-time function x produces the discrete-time sequence x_d such that
$$x_d[n] = x(n/f_s). \tag{1.26}$$

It is then obvious that a necessary and sufficient condition for two continuous-time functions x_1

and x_2 to be converted into the same discrete-time sequence x_d is that we have

$$x_1(n/f_s) = x_2(n/f_s),\ \forall n \in \mathbb{Z}. \tag{1.27}$$

We can also easily show that any discrete-time sequence can be matched to at least one continuous-time functions that could be converted into it, as fixed-rate sampling is surjective.

The operation of sampling is also the source of the phenomenon referred to as *aliasing*[1]. This phenomenon is best described by looking at the relationship between the Fourier transform $X = \mathrm{FT}(x)$ of continuous-time function x and the discrete-time Fourier transform $X_d = \mathrm{DTFT}(x_d)$ of the discrete-time sequence $x_d = \Delta(x)$ obtained by fixed-rate sampling of x.

First, we remind that for discrete-time sequences at fixed rate f_s, the discrete-time Fourier transform is necessarily periodic, with periodicity f_s so that

$$X_d(f) = X_d(f + kf_s),\ \forall k \in \mathbb{Z}. \tag{1.28}$$

Then, we can express X_d using the well-known derivation

$$\begin{aligned} X_d(f) &= \sum_{n\in\mathbb{Z}} x_d[n] e^{-j2\pi n \frac{f}{f_s}} \\ &= \sum_{n\in\mathbb{Z}} x(n/f_s) e^{-j2\pi n \frac{f}{f_s}} \\ &= \sum_{n\in\mathbb{Z}} \int_{\mathbb{R}} \delta(t - n/f_s) x(t) e^{-j2\pi t f} dt \\ &= \int_{\mathbb{R}} \left[\sum_{n\in\mathbb{Z}} \delta(t - n/f_s) \right] x(t) e^{-j2\pi t f} dt \\ &= \int_{\mathbb{R}} \left[\sum_{m\in\mathbb{Z}} \delta(s + mf_s) \right] X(f - s) ds \end{aligned} \tag{1.29}$$

giving us the result

$$X_d(f) = \sum_{m\in\mathbb{Z}} X(f + mf_s),\ \forall f \in \mathbb{R}. \tag{1.30}$$

We can then decompose the right-hand term as

$$X_d(f) = X(f) + \underbrace{\sum_{m\in\mathbb{Z}^*} X(f + mf_s)}_{\text{aliased frequency terms}},\ \forall f \in \mathbb{R} \tag{1.31}$$

where the sum corresponds to the so-called "aliased" frequency terms, meaning spectral contributions

[1] Note however that aliasing can be more generally defined in the context of signal downsampling, meaning when converting a discrete-time sequence into another discrete-time sequence at a lower sampling rate. Continuous-time signals can then be interpreted as the asymptotic case with infinite sampling rate. Such topic is out of the scope of our discussion.

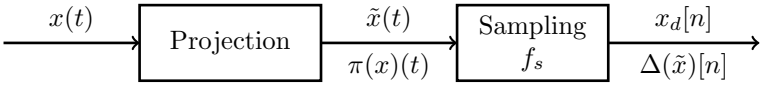


Figure 1.12: Projection-sampling decomposition of continuous-to-discrete conversion.

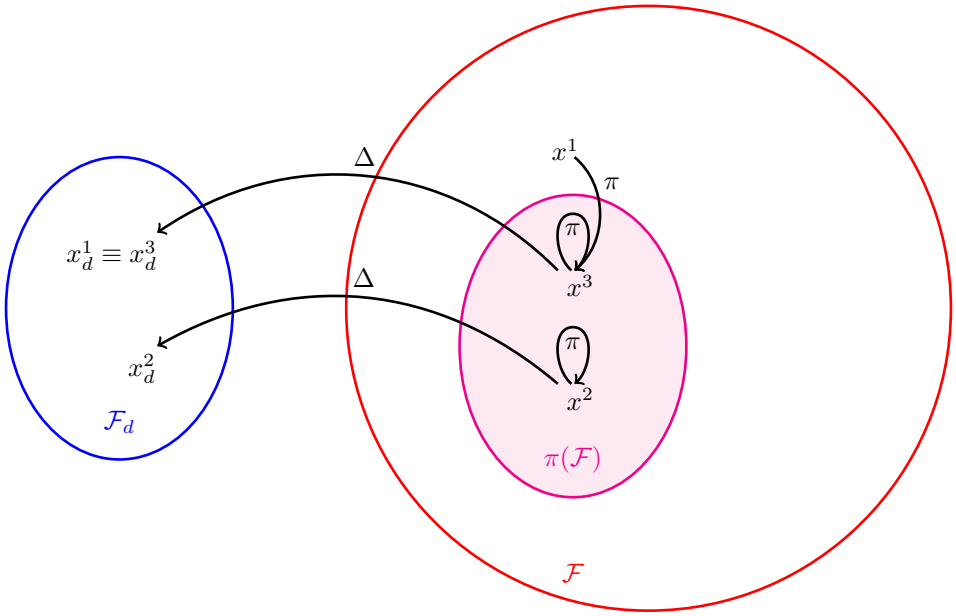


Figure 1.13: Illustration of a projection-sampling continuous-to-discrete conversion as mapping from the set of continuous-time signals $\mathcal{F}_d$ onto the set of discrete-time sequences $\mathcal{F}$. We have π a surjective projection from the set of continuous-time signals $\mathcal{F}$ onto the subset of continuous-time signals $\pi(\mathcal{F})$, and Δ the bijective sampling mapping from the subset of continuous-time function $\pi(\mathcal{F})$ onto the set of discrete-time sequences $\mathcal{F}_d$.

from a different frequency in X than the examined frequency in X_d. Thus, each frequency component in X has "aliases" that are added at all frequency components distant from an integer number of period f_s in X_d. It is also said that the spectral content of X is "folded" into a single spectral band of length f_s and then periodically repeated with period f_s.

1.3.4 Continuous-to-continuous projection and projection-sampling conversion

As mentioned in Sec. 1.3.3, the fixed-rate sampling operator Δ is surjective from the set of continuous-time function to the set of discrete-time sequences, creating ambiguity in the continuous-to-discrete conversion as discussed in Sec. 1.3.1.

Following these observations, a desirable way to design the two-step conversion process described in Sec. 1.3.2 is to make it such that the sampling operation is bijective (or at least injective) from

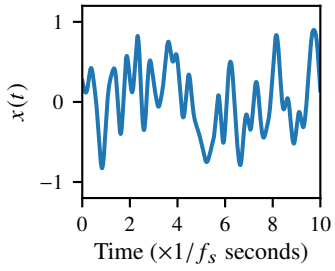
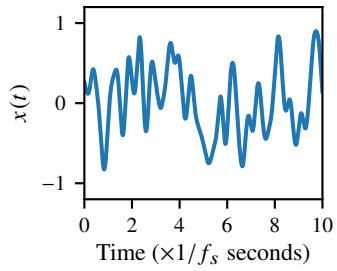
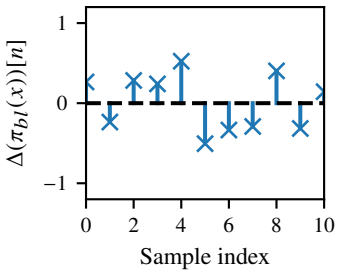
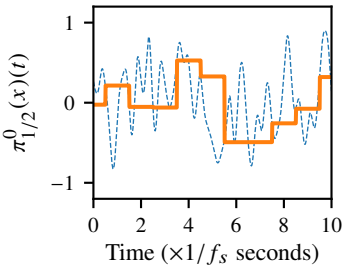
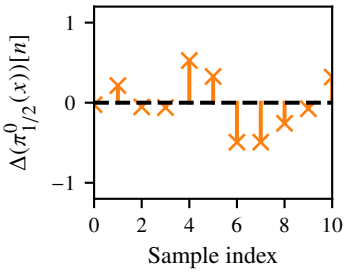
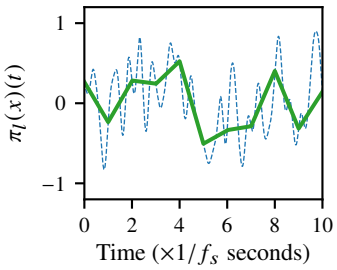
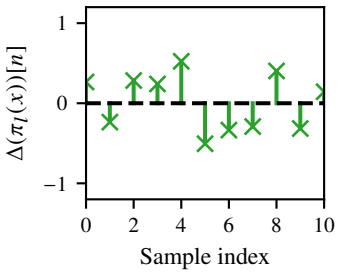
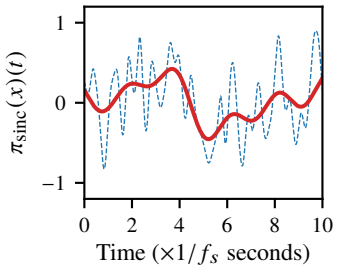
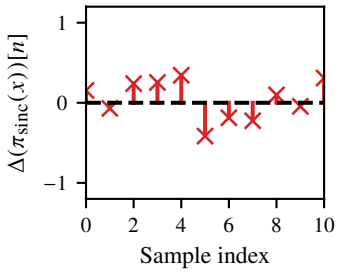
Continuous-time signal	Projected signal	Projected-sampled sequence	
			No projection
			$\pi^0_{1/2}$
			π_l
			π_{sinc}

Table 1.2: Projection and sampling of the continuous-time signal shown in the left column for no projection, and the projections $\pi^0_{1/2}$ (Eq. (1.33)) onto $\mathcal{F}_{1/2}$, π_l (Eq. (1.36)) onto $\mathcal{F}_l$ and π_{sinc} (Eq. (1.38)) onto $\mathcal{F}_{bl}$.

the subset of pre-processed continuous-time functions $\pi(\mathcal{F})$ to the set of discrete-time sequences $\mathcal{F}_d$. If the sampling operation is indeed bijective, the conversion surjectivity will be due to the necessary surjectivity of the mapping between $\mathcal{F}$ and $\pi(\mathcal{F})$.

One particular class of transforms that are of interest for the pre-processing steps are *projection* transforms. Indeed, their *idempotent* property is desirable, as it means that the pre-processing is the identity mapping from $\pi(\mathcal{F})$ onto itself.

The benefit of formalizing the continuous-to-discrete conversion as the projection-sampling two-step process shown in Fig. 1.12 is that the ambiguity of the conversion is entirely summarized by the projection operation as shown in Fig. 1.13. The sampling operation being bijective between the projected subset of continuous-time functions $\pi(\mathcal{F})$ and the set of discrete-time sequences $\mathcal{F}_d$, there is no ambiguity on how to compare a continuous-time behavior belonging to $\pi(\mathcal{F})$ with a discrete-time behavior. Below are a few examples of typical projected subsets of interest and candidates and potential projection operators. Some of these are also shown in Tab. 1.2. All the projection operators we describe here are *linear*, meaning that the projection of a linear combination of continuous-time functions is equal to the linear combination with identical weights of the individual projections of these continuous-time functions.

Projection on a set of piecewise constant functions

One class of sets for which the sampling operator becomes bijective are the sets of piecewise constant functions $\mathcal{F}_\alpha$, for $\alpha \in]-1, 1]$, defined as

$$x(t) = x\left(\frac{\text{floor}\,(tf_s + \alpha)}{f_s}\right), \forall t \in \mathbb{R} \tag{1.32}$$

or equivalently functions that are piecewise constant on the $\frac{1}{f_s}$-long segments $\left[\frac{n-\alpha}{f_s}, \frac{n+1-\alpha}{f_s}\right[$, $\forall n \in \mathbb{Z}$.

Many possible projection operators from $\mathcal{F}$ onto $\mathcal{F}_\alpha$ exist, two obvious candidates are:

- The class of projection π_α^β, for $\beta \in [0, 1[$, defined as

$$\tilde{x}(t) = \pi_\alpha^\beta(x)(t) = x\left(\frac{\text{floor}\,(tf_s + \alpha) - \alpha + \beta}{f_s}\right), \forall t \in \mathbb{R} \tag{1.33}$$

  where the function is mapped to a projected piecewise constant function that matches the original function at one time point in each segment.

- The averaging projection π_α^{mean}, defined as

$$\tilde{x}(t) = \pi_\alpha^\beta(x)(t) = f_s \int_{\frac{n}{f_s}}^{\frac{n+1}{f_s}} x(\tau)d\tau,\ \forall t \in \mathbb{R} \tag{1.34}$$

  with $n = \text{floor}(tf_s + \alpha)$. The function is then mapped to a constant value over each segment

equal its average value over that segment $\left[\frac{1}{f_s}(n-\alpha), \frac{1}{f_s}(n+1-\alpha)\right[, \forall n \in \mathbb{Z}$.

Projection on a set of piecewise linear functions

One other set for which the sampling operator becomes bijective is the set of piecewise linear functions $\mathcal{F}_l$ defined as

$$x(t) = (\text{floor}(tf_s) + 1 - tf_s)\, x\left(\frac{\text{floor}\,(tf_s)}{f_s}\right) + (tf_s - \text{floor}(tf_s))\, x\left(\frac{\text{floor}\,(tf_s)+1}{f_s}\right), \forall t \in \mathbb{R} \quad (1.35)$$

or equivalently functions that are piecewise linear on the $\frac{1}{f_s}$-long segments $\left[\frac{n}{f_s}, \frac{n+1}{f_s}\right[, \forall n \in \mathbb{Z}$.

Many possible projection operators from $\mathcal{F}$ onto $\mathcal{F}_l$ exist. An obvious candidate is the projection π_l, defined as

$$\begin{aligned} \tilde{x}(t) &= \pi_l(x)(t) \\ &= (\text{floor}(tf_s) + 1 - tf_s)\, x\left(\frac{\text{floor}\,(tf_s)}{f_s}\right) \\ &\quad + (tf_s - \text{floor}(tf_s))\, x\left(\frac{\text{floor}\,(tf_s)+1}{f_s}\right), \quad \forall t \in \mathbb{R}. \end{aligned} \quad (1.36)$$

Projection on a set of band-limited functions

Finally, one set of particular interest for which the sampling operator becomes bijective is the set of functions $\mathcal{F}_{bl}$ band-limited to the base band $\left[-\frac{f_s}{2}, \frac{f_s}{2}\right[$. Functions belonging to that set have to verify a condition best expressed through their Fourier transform as

$$X(f) = 0, \ \forall f \in \mathbb{R} \setminus \left[-\tfrac{f_s}{2}, \tfrac{f_s}{2}\right[. \quad (1.37)$$

Constructing projections to that set is more complex in general than in the previous cases, but we have one obvious candidate in the projection π_{sinc} defined as

$$\begin{aligned} \tilde{x}(t) &= \pi_{\text{sinc}}(x)(t) \\ &= \left(\frac{\sin(\pi f_s \cdot)}{\pi \cdot} * x\right)(t) = \int_{\mathbb{R}} \frac{\sin(\pi f_s \tau)}{\pi\tau} x(t-\tau)d\tau, \quad \forall t \in \mathbb{R}. \end{aligned} \quad (1.38)$$

Since we have

$$\text{FT}\left(\frac{\sin(\pi f_s \cdot)}{\pi \cdot}\right)(f) = \int_{\mathbb{R}} \frac{\sin(\pi t f_s)}{\pi t} e^{-j2\pi tf} dt = 1_{[-1,1[}\left(2\frac{f}{f_s}\right) \quad (1.39)$$

we get:

$$\widetilde{X}(f) = \text{FT}(\tilde{x})(f) = \begin{cases} X(f) & \text{for } f \in \left[-\frac{f_s}{2}, \frac{f_s}{2}\right[, \\ 0 & \text{otherwise.} \end{cases} \quad (1.40)$$

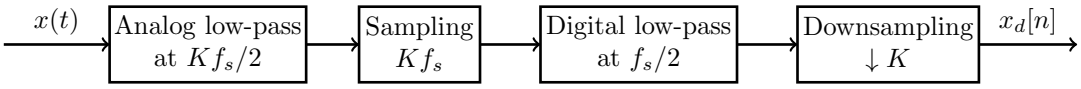


Figure 1.14: Example of conversion process decomposition for modern A/D convertors where the signal is first band-limited at $Kf_s/2$ and sampled at a higher sampling rate Kf_s, before being further band-limited at $f_s/2$ and downsampled at the final rate f_s.

1.3.5 Analog-to-digital conversion in practice

In practice, the conversion between discrete-time sequences and continuous-time signals is performed using analog-to-digital (abbreviated A/D) convertors. Similarly as described in Sec. 1.2.3 for D/A convertors, such devices are ubiquitous in audio hardware (e.g., soundcards) in order to encode the continuous-time analog audio (e.g., microphone) and control (e.g., knob, slider) signals delivered by typical audio interfaces.

Here again, such convertors actually need to discretize signals both in time and amplitude, but the quantization component remains out of the scope of our discussion. We also still ignore issues related to hardware issues such as component tolerances. Regarding the discretization in time, system generally follow the projection-sampling described in Sec. 1.3.4 and Fig. 1.12, where the projection operation corresponds to a low-pass filter with cutoff at $f_s/2$, in order to emulate the ideal projection on the set of band-limited signals described at the end of Sec. 1.3.4. We refer the reader to the literature to know more about the various algorithms designed to perform signal sampling in practice (e.g., [Pelgrom 2013]).

Furthermore, in some modern systems, the process is slightly more complicated and follows the four-step process shown in Fig. 1.14, because of desirable properties in the design of digital filters compared to their analog counterparts [Pelgrom 2013]. Such considerations are also beyond the scope of our discussion at this stage though they could be the subject of future research. For now, we will only consider the two-stage approach mentioned above.

Finally, the same way the filtering operation in the interpolation process can not be perfectly performed in the case of practical D/A conversion (see Sec. 1.2.3), the band-limiting process can not be perfectly performed here. Indeed, it is well-known the filter described in Eq. (1.39) is not realizable, in particular because the ideal brick-wall low-pass filter is not causal and its impulse response has infinite time support. In practical systems, realizable (i.e., causal) filters necessarily introduce latency in the process. Matching the filter response from Eq. (1.39) (or, again, a filter with same magnitude response but an added linear phase term) then also becomes a compromise between latency and precision.

1.3.6 Analog-to-digital conversion in this discussion

Again, discussions regarding quantization or non-ideal hardware components is beyond the scope of our discussion, as we want to focus on the signal processing implications of continuous-to-discrete conversion, in particular realizability even as quantization and hardware imperfection concerns are absent.

Furthermore, in the rest of this discussion, we will abstract continuous-to-discrete conversion as following the projection-sampling 2-step process described at the end of Fig. 1.12. We will also assume that the sampling step of the process is performed perfectly, as it is designed to be realizable. We will however allow a non-ideal second step as ideal processing generally necessitate non-realizable processing. However, we state that most of the ideas in our discussion apply readily to many more complex conversion systems.

1.4 Similarity definition for fixed-rate audio

Now equipped with a formal framework for the conversion between continuous-time and discrete-time quantities, we can now proceed to formalizing mathematically the concept of similarity and compare it to our earlier discussion in Sec. 1.1.

1.4.1 General considerations

We have seen in Sec. 1.2 and Sec. 1.3 that there are some intrinsic challenges in the definition of similarity due to the dimensionality difference between the set of continuous-time functions and the set of discrete-time sequences (or equivalently a difference in information capacity), namely:

- Ambiguity: several continuous-time functions could be matched to the same discrete-time sequences.
- Representativity: a discrete-time sequence can be matched to one continuous-time function at most, meaning many continuous-time functions cannot be matched.

Hence, similarity cannot be based on a bijective mapping between the two. Instead, we base similarity on a bijective mapping between the discrete-time sequences and a subset of the continuous-time functions of same dimension (e.g., functions band-limited to $f_s/2$). Then, we need a mapping function to transform continuous-time functions outside the chosen representable subset into a representable continuous-time function.

1.4.2 Input and control signals

In the case of input and control signals (respectively x and c in Figs. 1.1 and 1.2), the issue is ambiguity. Indeed, the ambiguity means that two pairs of continuous-time input and control signals

(x_1, c_1) and (x_2, c_2) could be mapped through continuous-to-discrete conversion to the same pair of discrete-time sequences (x_d, c_d). The issue then is that, in general, the continuous-time system would produce two distinct continuous-time output signals $y_1 \equiv \psi(x_1, c_1)$ and $y_2 \equiv \psi(x_2, c_2)$. However, for both scenarios, the discrete-time model will output the sequence $y_d \equiv \psi_d(x_d, c_d)$. Hence, we run the risk that our similarity definition makes it impossible to simultaneously verify $y_d \sim y_1$ and $y_d \sim y_2$, making it difficult to trivially define a modeling objective.

1.4.3 Output signals

In the case of the output signals (y in Figs. 1.1 and 1.2), the issue is representativity. Indeed, for a given discrete-to-continuous conversion process, we cannot guarantee that a desired continuous-time output signal y can be obtained through the conversion of any discrete-time sequence y_d, meaning that y is not representable. This implies that no continuous-time signal $\tilde{y}$ obtained by the conversion of any discrete-time sequence can achieve zero error, i.e., $y \equiv \tilde{y}$, also making it difficult to trivially define a modeling objective.

1.4.4 Similarity definition

In order to account for the observations listed above, we ground our similarity definition in the choice of discrete-to-continuous and continuous-to-discrete conversion processes.

First, we choose a subset $\tilde{\mathcal{F}}$ of the set of continuous-time functions $\mathcal{F}$. This subset is chosen so that it is possible to define a bijection between $\tilde{\mathcal{F}}$ and the set of discrete-time sequences $\mathcal{F}_d$.

Second, we choose:

- an bijective interpolation operator μ from F_d onto $\tilde{\mathcal{F}} = \mu(\mathcal{F}_d)$ to perform discrete-to-continuous conversion.

- a surjective projection operator π from $\mathcal{F}$ onto $\tilde{\mathcal{F}} = \pi(\mathcal{F})$, associated with the sampling operation Δ, to perform continuous-to-discrete conversion.

We then define similarity among continuous-time functions, and between continuous-time functions and discrete-time sequences with respect to the dataset $\tilde{\mathcal{F}}$, so that:

- Among continuous-time functions, we have

$$\pi(x_1) \equiv \pi(x_2) \qquad \Leftrightarrow \qquad x_1 \sim x_2, \qquad \forall x_1, x_2 \in \mathcal{F}. \tag{1.41}$$

  The idempotent property of the projection operator π then guarantees that any continuous-time functions in $\tilde{\mathcal{F}}$ is both identical and equal to its projection (i.e., itself), while any function

in $\tilde{\mathcal{F}}$ is not identical to any other function in that set, so that

$$\begin{array}{lll} x \equiv \pi(x) & \Leftrightarrow & x \sim \pi(x), \quad \forall x \in \tilde{\mathcal{F}}, \text{ and} \\ x_1 \not\equiv x_2 & \Leftrightarrow & x_1 \not\sim x_2, \quad \forall x_1, x_2 \in \tilde{\mathcal{F}}. \end{array} \tag{1.42}$$

- Between continuous-time functions and discrete-time sequences, we have

$$\mu(x_d) \equiv \pi(x) \quad \Leftrightarrow \quad x_d \sim x, \quad \forall x \in \mathcal{F},\ x_d \in \mathcal{F}_d. \tag{1.43}$$

1.5 Audio modeling problem definition

1.5.1 Well-posed problem

Now that we have a similarity definition as presented in Sec. 1.4.4, we can exploit it to propose a well-posed audio modeling problem, by converting similarity into an equivalency. That means:

- For input and control signals, we want to have the similarities $x \sim x_d$ and $c \sim c_d$. As such, there are two possible approaches, depending if we choose to use the set of continuous-time functions $\mathcal{F}$ or the set of discrete-time sequences $\mathcal{F}_d$ as starting point:

  - For $\mathcal{F}$, for a given function pair (x, c), we cannot feed it directly to the continuous-time system. Instead, we feed the projected signals $(\pi(x), \pi(c))$. For the discrete-time model, we consider that the proper corresponding function pair (x_d, c_d) to be fed to the continuous-time system should be defined using the equivalencies

$$\pi(x) \equiv \mu(x_d) \ \text{ and } \ \pi(c) \equiv \mu(c_d) \tag{1.44}$$

    so that the two outputs we want to consider when comparing the two pipelines should be based on

$$y_d \equiv \psi_d(\nu(x), \nu(c)) \tag{1.45}$$

    for the discrete-time model, with ν the continuous-to-discrete conversion operation (i.e., the composition of the pre-processing operation π and the sampling operation Δ as defined in Sec. 1.3.2), and

$$y \equiv \psi(\pi(x), \pi(c)) \tag{1.46}$$

    for the continuous-time system.

  - For $\mathcal{F}_d$, for a given sequence pair (x_d, c_d) fed to the discrete-time model, we consider that the proper corresponding function pair (x, c) to be fed to the continuous-time system

should be defined using the equivalencies

$$x \equiv \mu(x_d) \quad \text{and} \quad c \equiv \mu(c_d) \tag{1.47}$$

so that the two outputs we want to consider when comparing the two pipelines should be based on

$$y_d \equiv \psi_d(x_d, c_d) \tag{1.48}$$

for the discrete-time model, and

$$y \equiv \psi(\mu(x_d), \mu(c_d)) \tag{1.49}$$

for the continuous-time system.

In particular, both approaches resolve the ambiguity issue at the input and control signal level.

- For output signals, the issue of representativity can be solved if for a continuous-time output y and a discrete-time output y_d, we consider that the proper comparable quantities is between the continuous-time function $\pi(y)$ and the continuous-time function $\mu(y_d)$. In that context, we can define typical error quantities (e.g., mean square error) applied between these two quantities, knowing that for any $y \in \mathcal{F}$, there always exists a unique $y_d \in \mathcal{F}_d$ such that $\pi(y) \equiv \mu(y_d)$, meaning zero error. Such definition then removes the problem of representativity by mapping all output quantities to compare onto the same subset of functions.

1.5.2 Comparable pipeline description

From the features of a well-posed problem described in the previous section, we can then define two comparable pipelines for the continuous-time system and the discrete-time model. It is done by feeding comparable input/control quantities and making comparable output quantities through the proper applications of conversion operators.

The two options for comparable pipelines are shown in Fig. 1.15 (depending which set is chosen as starting point for input/control signals). The corresponding set mappings are displayed in Fig. 1.16. We see that, thanks to taking into account our similarity, we have removed the ambiguity and representativity issues discussed in this chapter, and we now have a well-defined error measure when running the model against the system.

One interesting thing to notice is that, in option (a) as shown in Fig. 1.15, we find the pipeline that is practically in use when using a computer model in conjunction with external analog input and output devices, with n analog-to-digital convertor for the input/control signals to feed to the model, and a digital-to-analog convertor for the output signal to deliver to a transducer.

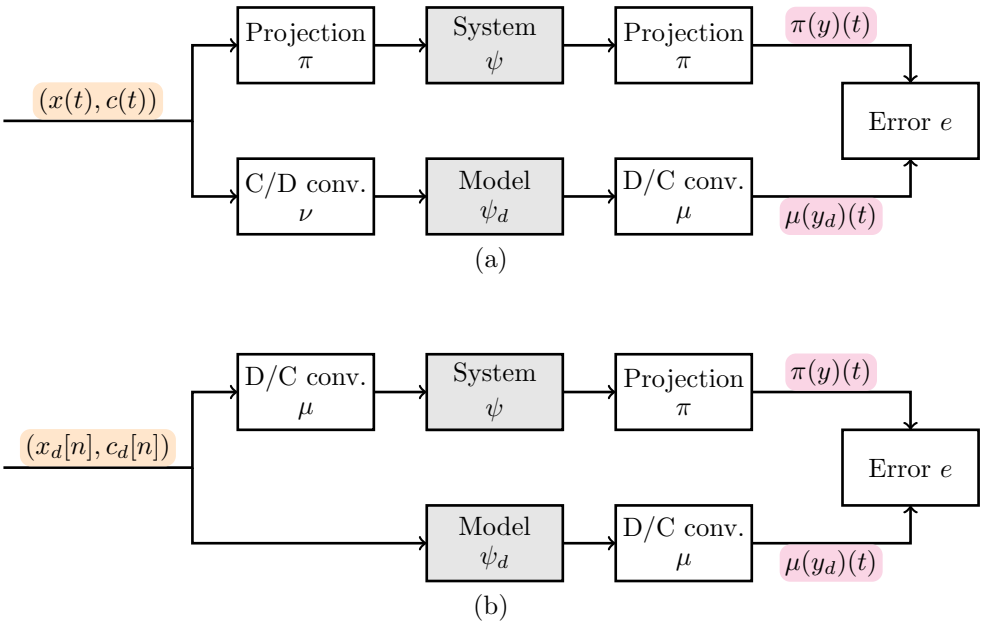


Figure 1.15: Signal chains for well-posed comparable pipelines when using (a) $\mathcal{F}$ or (b) $\mathcal{F}_d$ as departure set for the input/control quantities.

1.5.3 Equivalent continuous-time systems

Another consequence of this comparison framework is that multiple continuous-time systems end up being equivalent. For example, for a given continuous-time system ψ, we can add a pre-processing stage ψ_{pre} and a post-processing stage ψ_{post} and obtain an identical behavior with respect to our similarity measure as long as

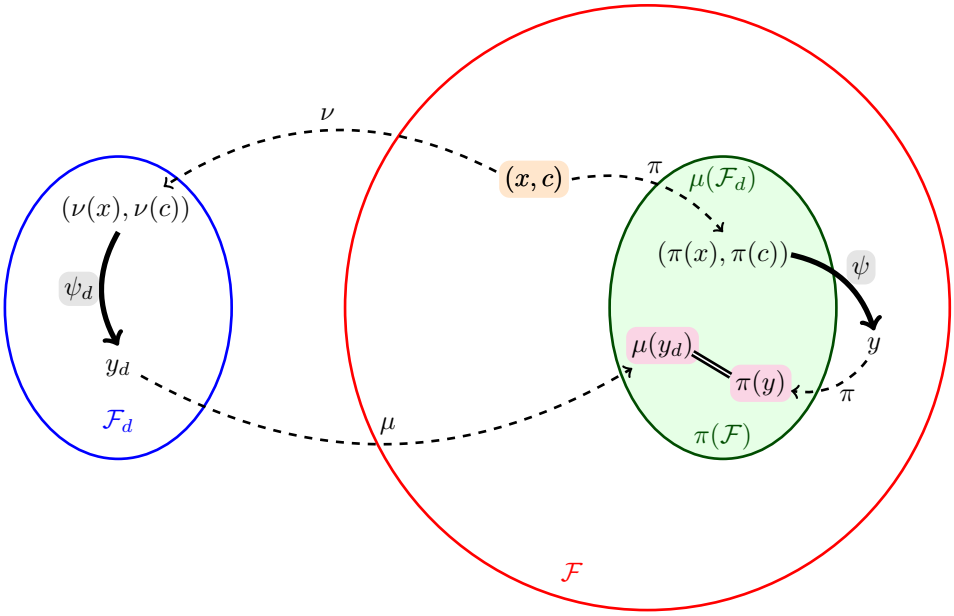
$$\pi \circ \psi_{\text{pre}} \equiv \pi \quad \text{and} \quad \psi_{\text{post}} \circ \pi \equiv \pi. \tag{1.50}$$

Indeed, we would then get

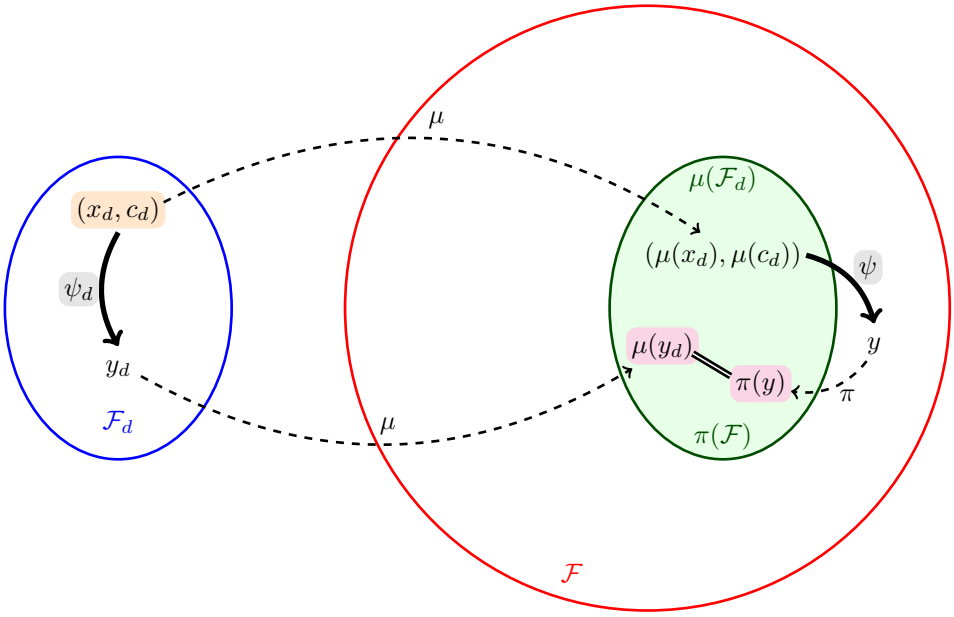
$$\pi \circ [\psi_{\text{pre}} \circ \psi \circ \psi_{\text{post}}] \circ \pi \equiv \pi \circ \psi \circ \pi \tag{1.51}$$

so that both system effectively deliver the same output signal as far as our similarity measure is concerned in the context of pipeline (a) in Fig. 1.15.

In particular, using the fact that $\pi \circ \pi \equiv \pi$ by definition of projection operators, we see that we can absorb the projection in the system description to form the equivalent system $\widetilde{\psi} \equiv \pi \circ \psi \circ \pi$. In doing so, we actually get a system for which similarity of input, control and output signals becomes equivalency without the need for the projection operators on the continuous-time pipeline (a) in Fig. 1.15, resulting in the pipeline shown in Fig. 1.17.



(a)



(b)

Figure 1.16: Set mappings for well-posed comparable pipelines when using (a) $\mathcal{F}$ or (b) $\mathcal{F}_d$ as departure set for the input/control quantities.

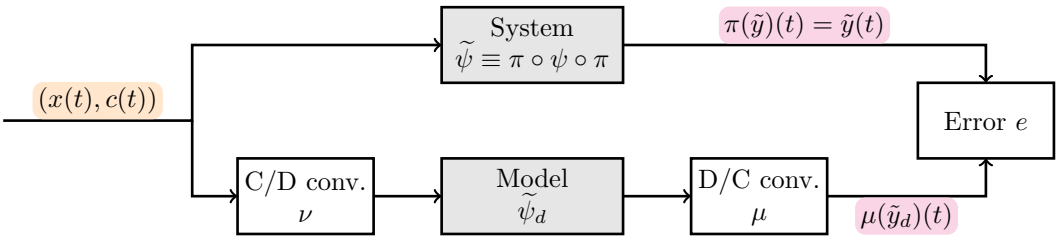


Figure 1.17: Equivalent continuous-time pipeline for pipeline (a) in Fig. 1.15 with absorbed projection operators in the continuous-time signal description.

1.5.4 Band-limited continuous-time functions and perceptual similarity

As stated in Sec. 1.1, a basic perceptual similarity measure for audio signals is that for two continuous-time audio signals x_1 and x_2, we define similarity using their Fourier transform $X_1 = \mathrm{FT}(x_1)$ and $X_2 = \mathrm{FT}(x_2)$ as

$$x_1 \sim x_2 \qquad \Leftrightarrow \qquad X_1(f) = X_2(f), \quad \forall f \in [20\,\mathrm{Hz}, 20\,\mathrm{kHz}]\,. \tag{1.52}$$

We then see that this perceptual similarity definition somewhat matches the mathematical similarity definition we defined in Sec. 1.4.4 if the chosen target subset of the continuous-time functions is the set of band-limited functions, which is

$$x_1 \sim x_2 \qquad \Leftrightarrow \qquad X_1(f) = X_2(f), \quad \forall f \in \left[-\tfrac{f_s}{2}, \tfrac{f_s}{2}\right]. \tag{1.53}$$

In particular, we see that for sampling frequencies above 40 kHz, our mathematical definition of similarity is a necessary condition for perceptual similarity. Furthermore, for typical audio sampling frequencies (e.g., 44.1 kHz, 48 kHz) result in a high degree of correlation between perceptual and mathematical similarity measures.

In the rest of our discussion, we will use the comparison framework corresponding to band-limited signals for a rate at $f_s = 48\,\mathrm{kHz}$ unless stated otherwise. As projection π, we use the brick-wall filtering operation π_{sinc} described in Eq. (1.38). For the interpolation μ, we use the sinc interpolation kernel described in Eq. (1.9). We note that, in this context, the operation of convolution by the sinc interpolation kernel happens to match exactly the operator π_{sinc}.

We point out that this fact does not preclude the model to run at a higher rate, as is sometimes done in numerical modeling. However, this approach does not alter our general pipeline: the higher rate only affects the design of the model ψ_d. Designing a model at a higher rate then corresponds to a similar approach as *oversampling* [Thornburg 1999].

1.6 Case studies

1.6.1 Linear time-invariant systems

Linear time-invariant systems are the simplest class of systems found in audio modeling. They are however ubiquitous in audio hardware in the form of filters composed of linear electric components such as resistors, inductors and capacitors. Such systems are not necessarily truly time-invariant in practice due to the presence of parameter controls (e.g., cutoff frequencies) through variable components (e.g., potentiometers), but they can still be considered as such on the time scale we need to consider.

Linear systems are typically characterized by their continuous-time impulse response $h(t)$. The output y is then given by the result of the convolution of the input x by that impulse response, i.e.,

$$y(t) = (h * x)(t), \ \forall t \in \mathbb{R}. \tag{1.54}$$

We can also express the relation between the Fourier transform of these quantities so that

$$Y(f) = H(f) \times X(f), \ \forall f \in \mathbb{R}. \tag{1.55}$$

Traditionally, continuous-time linear time-invariant systems are modeled as discrete-time linear time-invariant systems. These systems are characterized by the discrete-time impulse response $h_d[n]$. The output y_d is then given by the result of the convolution of the input x_d by the impulse response as

$$y_d[n] = (h_d * x_d)[n], \ \forall n \in \mathbb{Z}. \tag{1.56}$$

Here also, we can express the relation between the f_s-periodic discrete-time Fourier transform of these quantities so that

$$Y_d(f) = H_d(f) \times X_d(f), \ \forall f \in \mathbb{R}. \tag{1.57}$$

For a given input signal x, we apply the pipeline (a) from Fig. 1.15. On the continuous-time chain, we get the input signal $\tilde{x} \equiv \pi_{\text{sinc}}(x)$ with Fourier transform

$$\widetilde{X}(f) = X(f) \times 1_{[-1,1]}\left(2\tfrac{f}{f_s}\right), \ \forall f \in \mathbb{R}. \tag{1.58}$$

The output y then is such that

$$Y(f) = H(f) \times \widetilde{X}(f) = H(f) \times X(f) \times 1_{[-1,1]}\left(2\tfrac{f}{f_s}\right), \ \forall f \in \mathbb{R}. \tag{1.59}$$

Finally, the projected output $\pi_{\text{sinc}}(y)$ is equal to y as y is already properly band-limited, so that their spectral content is identical as well.

On the discrete-time pipeline, we get the discretized sequence $x_d = \nu(x)$ such that its discrete-time Fourier transform is defined as

$$X_d(f) = \sum_{k\in\mathbb{Z}} X(f + kf_s) \times 1_{[-1,1]}\left(2\left(\frac{f}{f_s} + k\right)\right),\ \forall f \in \mathbb{R} \tag{1.60}$$

so that the output discrete-time Fourier transform is

$$Y_d(f) = H_d(f) \times \sum_{k\in\mathbb{Z}} X(f + kf_s) \times 1_{[-1,1]}\left(2\left(\frac{f}{f_s} + k\right)\right),\ \forall f \in \mathbb{R}. \tag{1.61}$$

The final step is to interpolate the signal as a continuous-time signal $\tilde{y} \equiv \mu(y_d)$, and its Fourier transform is such that

$$\widetilde{Y}(f) = H_d(f) \times X(f) \times 1_{[-1,1]}\left(2\frac{f}{f_s}\right),\ \forall f \in \mathbb{R}. \tag{1.62}$$

Hence, we see that comparing the outputs of both signal chains in the pipeline essentially boils down to comparing the spectral content of the impulse responses of the system and the model. For example, if we measure the mean square error of the system, the error is expressed as:

$$\begin{aligned} ||y - \tilde{y}||_{L^2} &= \int_{\mathbb{R}} |y(t) - \tilde{y}(t)|^2\, dt \\ &= \int_{\mathbb{R}} \left|Y(f) - \widetilde{Y}(f)\right|^2 df \\ &= \int_{\mathbb{R}} \left|(H(f) - H_d(f)) \times X(f) \times 1_{[-1,1]}\left(2\frac{f}{f_s}\right)\right|^2 df \\ &= \int_{-f_s/2}^{f_s/2} |H(f) - H_d(f)|^2 \times |X(f)|^2\, df. \end{aligned} \tag{1.63}$$

Hence, we see that a sufficient condition to achieve zero error for all possible input signals it to match the spectral content of the system and the model in the base band $\left[-\frac{f_s}{2}, \frac{f_s}{2}\right]$, i.e.,

$$H(f) = H_d(f),\ \forall f \in \left[-\tfrac{f_s}{2}, \tfrac{f_s}{2}\right]. \tag{1.64}$$

Hence, our framework results in the same objective used in historical linear time-invariant system modeling, meaning the matching of base-band spectral response for the system and the model.

1.6.2 Square distortion

We consider here a square distortion effect, meaning a system characterized by

$$y(t) = \psi(x)(t) = (x(t))^2,\ \forall t \in \mathbb{R}. \tag{1.65}$$

It is interesting to see how that output spectral content depends on the input spectral content, especially in the base band $\left[-\frac{f_s}{2}, \frac{f_s}{2}\right]$, which can be expressed as

$$Y(f) = (X * X)(f), \ \forall f \in \mathbb{R}. \tag{1.66}$$

In the context of our comparison pipeline, the projection operators however modify the actual output to be compared against, which becomes

$$Y(f) = \begin{cases} \displaystyle\int_{f-f_s/2}^{f_s/2} X(\phi)X(f-\phi)d\phi & \text{for } f \in \left[0, \frac{f_s}{2}\right], \\ \displaystyle\int_{-f_s/2}^{f+f_s/2} X(\phi)X(f-\phi)d\phi & \text{for } f \in \left[-\frac{f_s}{2}, 0\right], \\ 0 & \text{otherwise.} \end{cases} \tag{1.67}$$

Such memoryless systems have typically been modeled with the same basic operation, meaning that the model is based on

$$Y_d(f) = (X_d \circledast X_d)(f), \ \forall f \in \mathbb{R} \tag{1.68}$$

where $\circledast$ denotes circular convolution.

In the case where we use as input sequence $x_d = \nu(x)$, and interpolate the output as $\tilde{y} = \mu(y)$, we get

$$\widetilde{Y}(f) = \begin{cases} \displaystyle\int_{f-f_s/2}^{f_s/2} X(\phi)X(f-\phi)d\phi + \underbrace{\int_{-f_s/2}^{f-f_s/2} X(\phi)X(f-\phi-f_s)d\phi}_{\text{aliased terms}} & \text{for } f \in \left[0, \frac{f_s}{2}\right], \\ \displaystyle\int_{-f_s/2}^{f+f_s/2} X(\phi)X(f-\phi)d\phi + \underbrace{\int_{f+f_s/2}^{f_s/2} X(\phi)X(f-\phi+f_s)d\phi}_{\text{aliased terms}} & \text{for } f \in \left[-\frac{f_s}{2}, 0\right], \\ 0 & \text{otherwise.} \end{cases} \tag{1.69}$$

By comparing Eqs. (1.67) and (1.69), we can see the limits of memoryless modeling for distortion effects as we can see the aliasing introduced by the discrete-time model.

Hence, in our framework, the error would be measured between the target signal whose spectrum is given by Eq. (1.67) and the signal whose spectrum is given by Eq. (1.69). For a mean square error, it would be

$$\begin{aligned} ||y - \tilde{y}||_2^2 = \int_{-f_s/2}^{0} df \int_{f+f_s/2}^{f_s/2} |X(\phi)X(f-\phi+f_s)|^2 \, d\phi \\ + \int_{0}^{f_s/2} df \int_{-f_s/2}^{f-f_s/2} |X(\phi)X(f-\phi-f_s)|^2 \, d\phi. \end{aligned} \tag{1.70}$$

This error will obviously be non-zero for a wide-variety of signals due to the complex interaction between the different frequencies in the input signal. We know however that the error can be zero for a certain class of input signal, in particular in the case where input signal are band-limited to $\left[-\frac{f_s}{4}, \frac{f_s}{4}\right]$. More generally, we retrieve the fact that, in the context where the chosen projection operator corresponds to the band-limiting operator to $\left[-\frac{f_s}{4}, \frac{f_s}{4}\right]$, the static nonlinearity function in the discrete-time domain is a poor computer model for that same function as a continuous-time domain system. It is however trivial to see that for many other simple projection operators (e.g., projections on piecewise constant or piecewise linear functions as defined in Sec. 1.3.4), the static nonlinearity becomes an exact discrete-time model. In practice, it will thus be crucial to identify the context in which this model is inserted, and which definition is most relevant.

1.7 Conclusion

In this chapter, we discussed the general concept of similarity when comparing signals involved in a continuous-time system and the equivalent sequences in a fixed-rate discrete-time model of that system. We show how defining such similarity relies heavily on a well-defined context for the use of the model. That context allows to establish the relevant mathematical operators to convert continuous-time quantities into discrete-time ones, and back. These operators then establish a well-posed procedure to measure error between these quantities, once they are converted into the same space (either continuous-time functions or discrete-time functions). In particular, we also showed how the concept of aliasing can be more generally understood as translating the ambiguity in converting continuous-time quantities into discrete-time ones, since dimensionality commands that multiple continuous-time candidate signals will map to the same discrete-time sequence. As such, aliasing becomes the measure of the difference between any of these candidate signals and the one candidate that is set as reference for belonging to a preferred subset of the continuous-time signals. That subset is chosen such that there is a one-to-one correspondence with the discrete-time sequences when using the proper conversion operation. Finally, we show how this mathematical framework to signal similarity leads to a definition that is compatible with the general understanding of perceptual similarity between continuous-time systems and fixed-rate discrete-time models in the audio context. These various concepts are illustrated on the two cases of modeling a linear time-invariant system, and a static square nonlinear system.

Chapter 2

Advanced aliasing and anti-aliasing analysis of single-input systems

In this chapter, we provide contributions around two main topics:

- A generalization of the analytical results presented in Thornburg [1999] to a wider variety of input waveforms and systems to provide some insight on analytical tools for aliasing analysis, and

- An empirical framework for the evaluation of the harmonic response of continuous-time systems and various discretization strategies in order to efficiently extract comprehensive information regarding:

  - The accuracy of the harmonic reconstruction of the target continuous-time system response for non-aliased harmonic components, and
  - The efficiency of aliasing suppression for aliased harmonic components in the discrete-time model response.

  By harmonic response, we mean the response of a continuous-time system and/or a discrete-time model to a periodic input waveform of any kind, either simple (e.g., cosinusoidal waveform) or more complex (e.g., triangle waveform, arbitrary waveform). Our empirical framework is generally inspired by the research thread around "harmonic balance" methods [Urabe 1965, Gilmore and Steer 1991a,b] which were initially designed to compute the approximate steady-state response of continuous-time microwave circuit.

In prior work, the typical analysis approach to analyze the harmonic response of a system would somewhat split in two broad categories:

- Chirp-response analysis: one would record the response of the system (and its model(s)) to a sinusoidal chirp signal. The response would then be convolved by the time-reversed version of that signal, and the output of that process would then generally be plotted in the time-frequency domain (see for example in Yeh [2012]). It is well known that this procedure, when done with a reasonably designed chirp signal in terms of frequency range and chirp duration displays reasonable qualitative information regarding the harmonic response of the signal. Furthermore, that response can even be used to fit a black-box model of any nonlinear system [Novak et al. 2010c,a,b], albeit under a strong polynomial Hammerstein structure hypothesis.

- The second approach would simply be to simulate the system over a relatively long period of time, then compute the Fourier transform of the resulting response in order to obtain a high accuracy estimate of the frequency distribution of the signal, thanks to the length of the signal being transformed, the prior knowledge of the harmonic location (since they are generally expected to be located at harmonics of the input signal) and the application of an appropriate windowing function with strong sidelobe suppression, such as a Chebyshev window. An example of such procedure is used in Välimäki [2005] where a segment of 1.0 s was used for the Fourier transform along with a Chebyshev window with 120dB side-lobe suppression. Such approach has been used in the literature for the purpose of assessing the signal-to-noise ratio (signal being the non-aliased part of the signal, and noise being the aliased part of the signal).

The first category of approaches is generally difficult to generalize to arbitrary waveforms, and it relies on rather strong assumptions regarding the underlying system as a polynomial Hammerstein structure. On the other hand, the second category of approaches is rather inefficient as it requires running the system for an undetermined amount of time, without a very intuitive way to assess convergence. By enforcing periodicity explicitly in our response analysis, thanks to the modeling of that response as a superposition of periodic Fourier elements, we aim at obtaining an efficient and accurate measure of the harmonic weights directly from the solution to a (potentially nonlinear) root finding problem. One straightforward application of our approach would be replacing the method used in Välimäki [2005] to compute harmonic components. Our method computes these harmonic weights in a much more straightforward manner and allow for their efficient and intuitive estimation across a wide frequency range. For example, we could then efficiently compute the A-weighted noise-to-mask ratio metric [Lehtonen et al. 2012] across that same frequency band and get a good assessment the general intensity of aliasing found in a model, in order to do a similar performance analysis as in Esqueda et al. [2017].

2.1 Memoryless continuous-time nonlinear systems

Here, we examine the case of a single-input/single-output memoryless nonlinear system in the continuous-time domain. Its input-output relationship is expressed as

$$y(t) = g(u(t)) \tag{2.1}$$

with $y(t)$ the output signal, g the nonlinear function characterizing the system, and u the input signal.

2.1.1 Periodic analysis in continuous time

Analytical periodic analysis for cosinusoidal input

Along with the mathematical formulation for oversampling, Thornburg [1999] also provides the mathematical formulation expressing the equations relative to the harmonic components of the nonlinearity output signal as a function of the nonlinearity and the input amplitude in the case of a cosinusoidal input signal. We provide here an extended derivation of these results for clarity with the trivial addition of the phase term in the input signal. In this case, the input signal is expressed as

$$u(t) = A\cos(\omega_0 t + \phi) \tag{2.2}$$

where A is the (positive) amplitude, ω_0 the radian frequency, and ϕ the phase of the cosinusoidal input signal u.

The output y of the system in Eq. (2.1) in the case where the input signal is periodic (which obviously includes the case where the input signal is cosinusoidal) can be shown to be necessarily periodic as well (with the same period as the input signal). This means that it can be written as the infinite sine and cosine series as

$$y(t) = \frac{c_0}{2} + \sum_{k=1}^{\infty} c_k \cos(k\omega_0 t) + d_k \sin(k\omega_0 t) \tag{2.3}$$

where c_k and d_k corresponds to the weights of the kth harmonic of the signal y (While d_0 is absent in this formulation, we can equivalently define it as $d_0 \equiv 0$ without altering the following derivations).

The core purpose of this derivation is find an expression of the coefficients c_k and d_k following the reasoning of Thornburg [1999].

Finding c_k and d_k relies on the Jacobi-Anger identity [Olver et al. 2010] which states that

$$e^{jz\cos\theta} = \sum_{m\in\mathbb{Z}} j^m J_m(z) e^{jm\theta} = J_0(z) + 2\sum_{m=1}^{\infty} j^m J_m(z)\cos(m\theta) \tag{2.4}$$

where J_m is mth Bessel function of the first kind [Olver et al. 2010].

Using the fact that the nonlinear function g can be alternatively expressed as a function of its Fourier transform G as

$$g(x) = \int_{\mathbb{R}} G(\omega) e^{j\omega x} d\omega, \tag{2.5}$$

we find that the output y can be written as

$$\begin{aligned} y(t) &= g(A\cos(\omega_0 t + \phi)) \\ &= \int_{\mathbb{R}} G(\omega) e^{j\omega A\cos(\omega_0 t+\phi)} d\omega \\ &= \int_{\mathbb{R}} G(\omega) J_0(A\omega) d\omega + 2\sum_{m=1}^{\infty} j^m \cos(m(\omega_0 t + \phi)) \int_{\mathbb{R}} G(\omega) J_m(A\omega) d\omega. \end{aligned} \tag{2.6}$$

By definition, we have

$$\begin{aligned} c_k &= \frac{2}{T_0}\int_0^{T_0} y(t)\cos(-k\omega_0 t)dt, \text{ and} \\ d_k &= \frac{2}{T_0}\int_0^{T_0} y(t)\sin(-k\omega_0 t)dt \end{aligned} \tag{2.7}$$

so that, combined with Eq. (2.6), we get

$$\begin{aligned} c_k &= 2j^k\cos(-k\phi)\int_{\mathbb{R}} G(\omega)J_k(A\omega)d\omega, \text{ and} \\ d_k &= 2j^k\sin(-k\phi)\int_{\mathbb{R}} G(\omega)J_k(A\omega)d\omega. \end{aligned} \tag{2.8}$$

Finally, using the results from App. A, we can apply Parseval's theorem [Olver et al. 2010] to get the expression

$$\begin{aligned} c_k &= \cos(-k\phi)\frac{2}{\pi A}\int_{-A}^{A} g(x)\frac{T_k(x/A)}{\sqrt{1-(x/A)^2}}dx, \text{ and} \\ d_k &= \sin(-k\phi)\frac{2}{\pi A}\int_{-A}^{A} g(x)\frac{T_k(x/A)}{\sqrt{1-(x/A)^2}}dx. \end{aligned} \tag{2.9}$$

Another proof of this result relies on the orthogonality of the sequence of Chebyshev polynomials of the first kind $T_n(x)$ with respect to the weight $\frac{1}{\sqrt{1-x^2}}$ on the interval $[-1, 1]$ [Olver et al. 2010]. In particular, we have

$$\int_{-1}^{1} \frac{T_n(x)T_m(x)}{\sqrt{1-x^2}}dx = \frac{\delta(m-n)}{2-\delta(m)}\pi. \tag{2.10}$$

Through a trivial change of variable, we get that for any positive scalar A the polynomial sequence $T_n(x/A)$ is orthogonal with respect to the weight $\frac{1}{\sqrt{1-(x/A)^2}}$ on the interval $[-A, A]$. In particular,

we have

$$\int_{-A}^{A} \frac{T_n(x/A)T_m(x/A)}{\sqrt{1-(x/A)^2}}dx = \frac{\delta(m-n)}{2-\delta(m)}\pi A. \tag{2.11}$$

As such, we know that there exists a sequence α_k such that

$$g(x) = \frac{\alpha_0}{2}T_0(x/A) + \sum_{l=1}^{\infty} \alpha_l T_l(x/A) \tag{2.12}$$

with:

$$\alpha_l = \frac{2}{\pi A}\int_{-A}^{A} g(x)\frac{T_l(x/A)}{\sqrt{1-(x/A)^2}}dx. \tag{2.13}$$

For the cosinusoidal input $u(t)$, we have, by definition of the Chebyshev polynomials of the first kind,

$$T_l(u(t)/A) = \cos(l(\omega t + \phi)). \tag{2.14}$$

Combining Eqs. 2.12 and 2.14, we get

$$g(u(t)) = \frac{\alpha_0}{2} + \sum_{l=0}^{\infty} \alpha_l \cos(l(\omega t + \phi)) \tag{2.15}$$

so that, by definition of the coefficients c_k and d_k, we have

$$\begin{aligned} c_k &= \cos(-k\phi)\alpha_k, \text{ and} \\ d_k &= \sin(-k\phi)\alpha_k. \end{aligned} \tag{2.16}$$

Combining this result with Eq. (2.13), we find again the result from Eq. (2.9), with

$$\begin{aligned} c_k &= \cos(-k\phi)\frac{2}{\pi A}\int_{-A}^{A} g(x)\frac{T_k(x/A)}{\sqrt{1-(x/A)^2}}dx, \text{ and} \\ d_k &= \sin(-k\phi)\frac{2}{\pi A}\int_{-A}^{A} g(x)\frac{T_k(x/A)}{\sqrt{1-(x/A)^2}}dx. \end{aligned} \tag{2.17}$$

Analytical periodic analysis for band-limited periodic input

The aliasing analysis framework in [Thornburg 1999] can be extended to deal with the more general case of a generic band-limited periodic input, including inputs such as band-limited rectangular, triangular, or saw waveforms. For a generic band-limited periodic input waveform of radian frequency ω_0, the input signal u can be described through the series

$$u(\omega) = \frac{a_0}{2} + \sum_{k=1}^{K} a_k \cos(\omega - k\omega_0) + b_k \sin(\omega - k\omega_0) \tag{2.18}$$

where a_k and b_k corresponds to the weight of the kth harmonic of the signal u. Equivalently, u can be represented as a superposition of cosinusoidal waveforms as

$$u(t) = A_0 + \sum_{k=1}^{K} A_k \cos(k\omega_0 t + \phi_k), \tag{2.19}$$

with $A_0 = 2a_0$, $A_k = \sqrt{a_k^2 + b_k^2}$, and $\phi_k = \angle(a_k - jb_k)$.

Then, the output of the system is written as

$$y(t) = g\left(A_0 + \sum_{k=1}^{K} A_k \cos(k\omega_0 t + \phi_k) \right). \tag{2.20}$$

From the Fourier transform of g, we then get

$$\begin{aligned} y(t) &= \int_{\mathbb{R}} G(\omega) e^{j\omega\left(A_0 + \sum_{k=1}^{K} A_k \cos(k\omega_0 t + \phi_k)\right)} d\omega \\ &= \int_{\mathbb{R}} G(\omega) e^{j\omega A_0} \prod_{k=1}^{K} e^{j\omega A_k \cos(k\omega_0 t + \phi_k)} d\omega. \end{aligned} \tag{2.21}$$

We can then apply the Jacobi-Anger identity to obtain

$$\begin{aligned} y(t) &= \int_{\mathbb{R}} G(\omega) e^{j\omega A_0} \prod_{k=1}^{K} \sum_{m\in\mathbb{Z}} j^m J_m(A_k\omega) e^{jmk\omega_0 t} e^{jm\phi_k} d\omega \\ &= \sum_{m_1=-\infty}^{\infty} \cdots \sum_{m_K=-\infty}^{\infty} e^{j\sum_{k=1}^{K} m_k(k\omega_0 t + \phi_k)} \int_{\mathbb{R}} G(\omega) e^{j\omega A_0} \prod_{k=1}^{K} j^{m_k} J_{m_k}(A_k\omega) d\omega \end{aligned} \tag{2.22}$$

where we can once again use Parseval's theorem

$$y(t) = \sum_{m_1=-\infty}^{\infty} \cdots \sum_{m_K=-\infty}^{\infty} e^{j\sum_{k=1}^{K} k(m_k\omega_0 t + \phi_k)} \int_{\mathbb{R}} g(x) FT\left\{ e^{j\cdot A_0} \prod_{k=1}^{K} j^{m_k} J_{m_k}(A_k\cdot) \right\}(x) dx \tag{2.23}$$

with

$$FT\left\{ e^{j\omega A_0} \prod_{k=1}^{K} j^{m_k} J_{m_k}(A_k\omega) \right\}(x) = [h_{1,m_1} * \cdots * h_{K,m_K}](x - A_0) \tag{2.24}$$

and

$$h_{k,m}(x) = \mathrm{FT}(j^m J_m(A_k\cdot))(x) = \mathbf{1}_{[-1,1]}\left(\frac{x}{A_k}\right) \frac{1}{\pi A_k} \frac{T_m\left(\frac{x}{A_k}\right)}{\sqrt{1 - (x/A_k)^2}} \tag{2.25}$$

so that

$$y(t) = \sum_{m_1=-\infty}^{\infty} \cdots \sum_{m_K=-\infty}^{\infty} e^{j\omega_0 t \sum_{k=1}^{K} k(m_k\omega_0 t + \phi_k)} \int_{\mathbb{R}} g(x)[h_{1,m_1} * \cdots * h_{K,m_K}](x - A_0) dx. \tag{2.26}$$

Here again, we see that the harmonic components of the output signal y is controlled by the projection of the nonlinearity g on the set of time-limited projectors $[h_{1,m_1} * \cdots * h_{K,m_K}](x - A_0)$. These projectors are time-limited as finite convolutions of the time-limited functions h_{k,m_k}.

A lot of typical audio waveforms (e.g., square, triangle, sawtooth) have the property of having a particular phase structure. In the band-limited case, they are of the form

$$u(t) = \sum_{k=1}^{\infty} \gamma_k \cos(k(\omega_0 t + \phi)) \tag{2.27}$$

with γ_k real weights (possibly negative) and ϕ a real phase term.

In this case, Eq. (2.21) becomes

$$y(t) = \int_{\mathbb{R}} G(\omega) e^{j\omega \sum_{k=1}^{K} \gamma_k \cos(k(\omega_0 t + \phi))} d\omega. \tag{2.28}$$

It is possible to express the term $e^{j\omega \sum_{k=1}^{K} \gamma_k \cos(k(\omega_0 t + \phi))}$ using multi-variate Bessel functions as defined by Dattoli et al. [1996], which follow the generalized Jacobi-Anger identity

$$e^{j \sum_{k=1}^{K} z_k \cos(k\theta)} = \sum_{n \in \mathbb{Z}} j^n e^{jn\theta} J_n^{(K)}(\{z_k\}) \tag{2.29}$$

where $J_n^{(K)}$ is the nth multi-variate Bessel function of the first kind for K variables. As a result, we have

$$y(t) = \sum_{n \in \mathbb{Z}} j^n e^{jn(\omega_0 t + \phi)} \int_{\mathbb{R}} G(\omega) J_n^{(K)}(\{\gamma_k \omega\}) d\omega. \tag{2.30}$$

Empirical periodic analysis for arbitrary periodic input

While the analytical results in Eqs. 2.6 and 2.26 can provide valuable insight on the harmonic structure of the continuous-time system output, computing their value for a given nonlinearity can be impractical, for example due to the need of numerical integration techniques to compute the various integrals.

Another common analysis approach which can be applied for any arbitrary periodic input waveform is to compute the Fourier coefficients c_k and d_k as defined in Eq. (2.3) using Eq. (2.7). As periodic input waveform, the only constraint on u is to verify

$$u(t) \equiv u\left(t - \frac{2\pi}{\omega_0}\right), \ \forall t \in \mathbb{R}. \tag{2.31}$$

Solving Eq. (2.7) numerically can done using any numerical integration methods, including high-accuracy adaptive methods [Gander and Gautschi 2000, Shampine 2008, Moin 2010]. One straightforward alternative is provided when approximating the output signal as having $K - 1$ harmonic

components such that

$$y(t) \approx \frac{\tilde{c}_0}{2} + \sum_{k=1}^{K-1} \tilde{c}_k \cos(k\omega_0 t) + \tilde{d}_k . \sin(k\omega_0 t) \tag{2.32}$$

We then discretize the signal period in $N+1$ equidistant points (i.e., separated by a time interval of $\frac{2\pi}{N\omega_0}$, with $N = 2K$) and leverage the computational speed of the fast Fourier transform (FFT) algorithm [Smith III 2007a] so that we approximate c_k and d_k as

$$\begin{aligned} c_k \approx \tilde{c}_k &= \frac{2}{N} \sum_{n=0}^{N-1} y\left(\frac{2\pi n}{N\omega_0}\right) \cos\left(-\frac{kn}{N}\right) = \frac{2}{N} \sum_{n=0}^{N-1} g\left(u\left(\frac{2\pi n}{N\omega_0}\right)\right) \cos\left(-\frac{kn}{N}\right), \text{ and} \\ d_k \approx \tilde{d}_k &= \frac{2}{N} \sum_{n=0}^{N-1} y\left(\frac{2\pi n}{N\omega_0}\right) \sin\left(-\frac{kn}{N}\right) = \frac{2}{N} \sum_{n=0}^{N-1} g\left(u\left(\frac{2\pi n}{N\omega_0}\right)\right) \sin\left(-\frac{kn}{N}\right). \end{aligned} \tag{2.33}$$

One advantage of such approach is that the error due to this approximation can be straightforwardly described through the aliasing of the sequences c_k and d_k, such that

$$\tilde{c}_0 = c_0 + 2 \sum_{m=1}^{\infty} c_{mK} + d_{mK} \tag{2.34}$$

and, for $k = 1 \dots K$,

$$\begin{aligned} \tilde{c}_k &= c_k + \sum_{m=1}^{\infty} c_{mN+k} + c_{mN-k}, \text{ and} \\ \tilde{d}_k &= d_k + \sum_{m=1}^{\infty} d_{mN+k} - d_{mN-k}. \end{aligned} \tag{2.35}$$

In that spirit, N should be chosen carefully to limit that error, by selecting N high enough, in the sense that we need to select N such that the coefficients b_k are negligible outside of the frequency band $\left]-\frac{N\omega_0}{4\pi}, \frac{N\omega_0}{4\pi}\right[$, i.e., b_k is negligible for $|k| \geq \frac{N}{2}$. One possible approach to judge empirically if the chosen N is sufficient is to compute the $\tilde{b}_k$ weights for another $N' > N$ that is coprime to N and verify that the computed weights for N and N' for $|k| < \frac{N}{2}$ do not change by a non-negligible amount.

2.1.2 Discretization and anti-aliasing methods

Direct discretization

The straightforward technique to discretize a given SISO continuous-time nonlinear system is by direct discretization of the time dimension at a fixed sampling period $T_s = 1/f_s = 2\pi/\omega_s$, such that the nth sample of the discrete-time output waveform y_d is given as a function of the discrete-time

input waveform u_d as

$$y_d[n] = g(u_d[n]) \tag{2.36}$$

with u_d is derived from a reference continuous-time waveform through sampling as $u_d[n] = u(nT_s)$. When a reference continuous-time waveform is not provided, the general practice is to consider that inferred reference waveform is the band-limited sinc interpolation of the sequence u_d (see Sec. 1.2.1 and in particular Eq. (1.16)).

In the direct discretization framework, the periodic analysis results found in Sec. 2.1.1 generally apply here, except for the straightforward fact that all the harmonic components become subject to aliasing, in the sense that

$$Y_d(\omega) = \sum_{m=-\infty}^{\infty} Y(\omega - m\omega_s) \tag{2.37}$$

where Y follows the expressions found in Sec. 2.1.1 in relation to a desired reference continuous-time input waveform (e.g., cosinusoidal, band-limited periodic).

Anti-aliasing by oversampling [Thornburg 1999]

The aliasing error in the literature generally follows our the framework we provided in Sec. 1.5.4, in the sense that the error is measured as a function of $Y_d(\omega) - Y(\omega)$ for $\omega \in \left[-\frac{\omega_s}{2}, \frac{\omega_s}{2}\right]$. with $Y_d(\omega) - Y(\omega) \equiv 0$ corresponding to no error. In that sense, we see that the error is captured by the coefficients c_k and d_k for which $k \geq \frac{\omega_s}{2\omega_0}$.

This observation is at the basis of the well-known anti-aliasing technique of oversampling as described in Thornburg [1999], where the system is essentially using direct discretization, but while using a smaller sampling period $T_s' < T_s$ ($\omega_s' > \omega_s$), followed by an anti-aliasing linear filter and a down-sampling operation to eliminate the contribution of the coefficients c_k and d_k for which $k \in \left[\frac{\omega_s}{2\omega_0}, \frac{\omega_s'}{2\omega_0}\right]$ to the aliasing error. Guidelines regarding the choice of a good sampling period T_s' are provided in Thornburg [1999] for typical distortions and to a lesser extent in Yeh [2009] for saturating nonlinearities.

Anti-derivative anti-aliasing [Parker et al. 2016]

A novel approach to anti-aliasing was proposed in Parker et al. [2016] for the purpose of aliasing mitigation without or with limited oversampling in the context of static nonlinearity modeling. The method relies on the use of the anti-derivative(s) of the nonlinearity in a system to build discretization operators. It was shown that these methods generally present lower amounts of aliasing distortion in the resulting model, but at the expense of increased error ("coloration") of the target harmonic response below the Nyquist frequency. Interesting alternatives to the design of the weighting of the different anti-derivatives were additionally proposed in Bilbao et al. [2017a] and Bilbao et al. [2017b], resulting in different trade-offs between aliasing distortion and baseband coloration.

By far the most practical method out of this family of methods is the one obtained at the first order, where a static nonlinearity operation in a discrete-time model $g(u_d[n])$ becomes

$$\frac{g^{(-1)}(u_d[n]) - g^{(-1)}(u_d[n-1])}{u_d[n] - u_d[n-1]} \tag{2.38}$$

where $g^{(-1)}$ is the first-order anti-derivative of g. This formula has generally been preferred to higher-order approaches despite its limited amount of aliasing mitigation. This is mostly due to the fact that the higher-order methods are much more computationally expensive, in particular due to the fact that few nonlinearities have closed-form formulas for their anti-derivative beyond the first-order one.

One particularly interesting aspect of this method is the fact that, for a linear function g, the method simplifies to a linear interpolation between two consecutive samples. Indeed if $g(u_d[n]) = \lambda u_d[n]$, its anti-derivative is $g^{(-1)}(u_d[n]) = \lambda(u_d[n])^2/2$, so that

$$\frac{g^{(-1)}(u_d[n]) - g^{(-1)}(u_d[n-1])}{u_d[n] - u_d[n-1]} = \frac{u_d[n] + u_d[n-1]}{2} \tag{2.39}$$

which, in passing, is equal to the linearized term that is used in Parker et al. [2016] and the rest of the literature as approximate update when the denominator becomes small, i.e., $|u_d[n] - u_d[n-1]| << 1$. Eq. (2.39) also extends trivially to the case of a multi-dimensional static non-linearity.

Eq. (2.39) is important in the context of lumped audio system modeling, since it proves the following. The first-order anti-derivative method can be applied to the static nonlinearity term in the discretization process any dynamical equation of the form $\dot{u}(t) = g(u(t))$ to form the following discrete model update equation

$$\frac{u_d[n] - u_d[n-1]}{T_s} = \frac{g^{(-1)}(u_d[n]) - g^{(-1)}(u_d[n-1])}{u_d[n] - u_d[n-1]}, \tag{2.40}$$

using the typical first-order differential approximation to form the left-hand term from $\dot{u}$. Then, the fact that the right hand term becomes as in Eq. (2.39) for linear systems implies that this method behaves similarly as the bilinear transform/trapezoidal method in terms of numerical stability (see Secs. 0.4.2 and 3.7.2 for additional context). In particular, it means the method is so-called *unconditionally stable* or *A-stable* and it can generally be applied safely to any system with static nonlinear terms without the risk of the simulation being unstable. For example, it means this very application of the method done in Paschou et al. [2017] will be well-behaved as far as stability is concerned, as would be its application to any other system.

2.1.3 Analytical periodic analysis of anti-aliasing techniques

We want to compare the periodic analysis of direct discretization approaches to the a similar analysis performed on the output waveform y_d obtained using an anti-derivative technique. Parker et al. [2016] observes that the method can generally be interpreted as applying a (smoothing) filtering kernel on a continuous-time waveform obtained through a piecewise stepwise interpolation (see Sec. 1.2.1 of the chosen input sequence u_d. But this interpretation only provide a partial information regarding the harmonic structure of the output.

Here, we propose examining the output with a similar approach as developed in Sec. 2.1.1 above. As proof of concept, we focus here on the first-order version of the anti-derivative methods, where the output y_d is expressed as

$$y_d[n] = \frac{g^{(-1)}(u_d[n]) - g^{(-1)}(u_d[n-1])}{u_d[n] - u_d[n-1]} \tag{2.41}$$

where $g^{(-1)}$ is the anti-derivative of g, meaning the function such that

$$g^{(-1)}(x) = \int_{x_0}^{x} g(y)dy \tag{2.42}$$

with x_0 an arbitrary constant. x_0 is generally chosen as a function of g to make the integral tractable (changing the chosen x_0 shifts the function by a constant, producing another anti-derivative candidate). Since the method is designed around the difference of two anti-derivative evaluations, the choice of x_0 has no impact on the behavior of the method.

Our goal, here again, is to find the sequences c'_k and d'_k such that

$$y_d[n] = \frac{c'_0}{2} + \sum_{k=1}^{\infty} c'_k \cos k\omega_0 nT_s + d'_k \sin k\omega_0 nT_s. \tag{2.43}$$

Again, we can define $d'_0 \equiv 0$ to help the notation in the derivation. First, we look at the system when its input is a cosinusoidal waveform, such that

$$u_d[n] = A\cos(\omega_0 nT_s + \phi). \tag{2.44}$$

For the periodic analysis, we rearrange Eq. (2.41) as

$$(u_d[n] - u_d[n-1])y_d[n] = g^{(-1)}(u_d[n]) - g^{(-1)}(u_d[n-1]). \tag{2.45}$$

For the left-hand side of this equation, we get

$$\frac{(u_d[n] - u_d[n-1])}{A\sin\frac{\omega_0 T_s}{2}} y_d[n] = c'_1 \sin\left(\tfrac{\omega_0 T_s}{2} - \phi\right) - d'_1 \cos\left(\tfrac{\omega_0 T_s}{2} - \phi\right)$$

$$+ \quad \sum_{k=1}^{\infty}\big(\big(c'_{k+1}+c'_{k-1}\big)\sin\big(\tfrac{\omega_0 T_s}{2}-\phi\big)+\big(d'_{k-1}-d'_{k+1}\big)\cos\big(\tfrac{\omega_0 T_s}{2}-\phi\big)\big)\cos k\omega_0 nT_s \\ + \big(\big(d'_{k+1}+d'_{k-1}\big)\sin\big(\tfrac{\omega_0 T_s}{2}-\phi\big)+\big(c'_{k+1}-c'_{k-1}\big)\cos\big(\tfrac{\omega_0 T_s}{2}-\phi\big)\big)\sin k\omega_0 nT_s \tag{2.46}$$

while, using the Jacobi-Anger identity [Olver et al. 2010] and the right-hand side becomes

$$g^{(-1)}(u_d[n])-g^{(-1)}(u_d[n-1])=\sum_{k=1}^{\infty}\sin\tfrac{k\omega_0 T_s}{2}\left[4j^k\int_{\mathbb{R}}\mathrm{FT}\{g^{(-1)}\}(\omega)J_m(A\omega)d\omega\right] \\ \times\big[\sin\big(\tfrac{k\omega_0 T_s}{2}-k\phi\big)\cos k\omega_0 nT_s-\cos\big(\tfrac{k\omega_0 T_s}{2}-k\phi\big)\sin k\omega_0 nT_s\big]. \tag{2.47}$$

We can use Parseval's theorem [Olver et al. 2010] and integration by parts to express the integral as

$$4j^m\int_{\mathbb{R}}\mathrm{FT}\{g^{(-1)}\}(\omega)J_m(A\omega)d\omega=\frac{2}{\pi m}\int_{-A}^{A}g(x)\frac{T_{m-1}(x/A)-T_{m+1}(x/A)}{\sqrt{1-(x/A)^2}}dx \\ =\frac{A}{m}(\alpha_{m-1}-\alpha_{m+1}). \tag{2.48}$$

Here again, we see that the harmonic weights, and as a result the aliasing error, for the anti-derivative method is strongly related to the weights α_m of the nonlinear function g decomposition over the sequence of Chebyshev polynomials of the first kind T_m, as

$$g^{(-1)}(u_d[n])-g^{(-1)}(u_d[n-1])=\sum_{k=1}^{\infty}\frac{A}{k}\sin\tfrac{k\omega_0 T_s}{2}(\alpha_{m-1}-\alpha_{m+1}) \\ \times\big[\sin\big(\tfrac{k\omega_0 T_s}{2}-k\phi\big)\cos k\omega_0 nT_s-\cos\big(\tfrac{k\omega_0 T_s}{2}-k\phi\big)\sin k\omega_0 nT_s\big] \tag{2.49}$$

which leads to the linear system of equations:

$$c'_1\sin\big(\tfrac{\omega_0 T_s}{2}-\phi\big)-d'_1\cos\big(\tfrac{\omega_0 T_s}{2}-\phi\big)=0 \tag{2.50}$$

and, for $m\geq 1$,

$$k\frac{\sin\frac{\omega_0 T_s}{2}}{\sin\frac{k\omega_0 T_s}{2}}\big[\big(c'_{k+1}+c'_{k-1}\big)\sin\big(\tfrac{\omega_0 T_s}{2}-\phi\big)+\big(d'_{k-1}-d'_{k+1}\big)\cos\big(\tfrac{\omega_0 T_s}{2}-\phi\big)\big]= \\ \sin\big(\tfrac{k\omega_0 T_s}{2}-k\phi\big)(\alpha_{k-1}-\alpha_{k+1}), \tag{2.51}$$

and

$$k\frac{\sin\frac{\omega_0 T_s}{2}}{\sin\frac{k\omega_0 T_s}{2}}\big[\big(d'_{k+1}+d'_{k-1}\big)\sin\big(\tfrac{\omega_0 T_s}{2}-\phi\big)+\big(c'_{k+1}-c'_{k-1}\big)\cos\big(\tfrac{\omega_0 T_s}{2}-\phi\big)\big]= \\ \cos\big(\tfrac{k\omega_0 T_s}{2}-k\phi\big)(\alpha_{k+1}-\alpha_{k-1}). \tag{2.52}$$

We also see that the harmonic weights here are tied to the harmonic weights found for direct discretization following Eq. (2.9) as

$$\begin{aligned} g^{(-1)}(u_d[n]) - g^{(-1)}(u_d[n-1]) = A\sum_{k=1}^{\infty} \frac{\sin\frac{k\omega_0 T_s}{2}}{k} \\ \big[c_{k-1}\sin\big(\tfrac{k\omega_0 T_s}{2}-\phi\big) - c_{k+1}\sin\big(\tfrac{k\omega_0 T_s}{2}+\phi\big)\big]\cos k\omega_0 nT_s \\ +\big[d_{k-1}\cos\big(\tfrac{k\omega_0 T_s}{2}-\phi\big) - d_{k+1}\cos\big(\tfrac{k\omega_0 T_s}{2}+\phi\big)\big]\cos k\omega_0 nT_s \\ +\big[d_{k-1}\sin\big(\tfrac{k\omega_0 T_s}{2}-\phi\big) - d_{k+1}\sin\big(\tfrac{k\omega_0 T_s}{2}+\phi\big)\big]\sin k\omega_0 nT_s \\ -\big[c_{k-1}\cos\big(\tfrac{k\omega_0 T_s}{2}-\phi\big) - c_{k+1}\cos\big(\tfrac{k\omega_0 T_s}{2}+\phi\big)\big]\sin k\omega_0 nT_s \end{aligned} \tag{2.53}$$

which leads to the linear system of equations:

$$c'_1\sin\big(\tfrac{\omega_0 T_s}{2}-\phi\big) - d'_1\cos\big(\tfrac{\omega_0 T_s}{2}-\phi\big) = 0. \tag{2.54}$$

and, for $k \geq 1$,

$$\begin{aligned} k\frac{\sin\frac{\omega_0 T_s}{2}}{\sin\frac{k\omega_0 T_s}{2}}\big[\big(c'_{k+1}+c'_{k-1}\big)\sin\big(\tfrac{\omega_0 T_s}{2}-\phi\big) + \big(d'_{k-1}-d'_{k+1}\big)\cos\big(\tfrac{\omega_0 T_s}{2}-\phi\big)\big] = \\ c_{k-1}\sin\big(\tfrac{k\omega_0 T_s}{2}-\phi\big) - c_{k+1}\sin\big(\tfrac{k\omega_0 T_s}{2}+\phi\big) + d_{k-1}\cos\big(\tfrac{k\omega_0 T_s}{2}-\phi\big) - d_{k+1}\cos\big(\tfrac{k\omega_0 T_s}{2}+\phi\big), \end{aligned} \tag{2.55}$$

and

$$\begin{aligned} k\frac{\sin\frac{\omega_0 T_s}{2}}{\sin\frac{k\omega_0 T_s}{2}}\big[\big(d'_{k+1}+d'_{k-1}\big)\sin\big(\tfrac{\omega_0 T_s}{2}-\phi\big) + \big(c'_{k+1}-c'_{k-1}\big)\cos\big(\tfrac{\omega_0 T_s}{2}-\phi\big)\big] = \\ d_{k-1}\sin\big(\tfrac{k\omega_0 T_s}{2}-\phi\big) - d_{k+1}\sin\big(\tfrac{k\omega_0 T_s}{2}+\phi\big) - c_{k-1}\cos\big(\tfrac{k\omega_0 T_s}{2}-\phi\big) + c_{k+1}\cos\big(\tfrac{k\omega_0 T_s}{2}+\phi\big). \end{aligned} \tag{2.56}$$

2.1.4 Empirical periodic analysis

Analyzing the harmonic structure empirically for a discretized SISO static nonlinear system can be done through the use of the discrete-time Fourier transform (DTFT) [Smith III 2007a] over long windowed signal segments using an appropriate window with low side-lobe energy level. In Välimäki [2005], signals are generated over 1.0 s and a Chebyshev window with 120 dB attenuation [Smith III 2011].

This category of methods essentially relies on a high-accuracy approximation to find the harmonic components of the signal. We propose an alternative approximation which parallels the approach we used in Sec. 2.1.1, which has the benefit of decoupling the resolution tuning from the input signal period.

In continuous time, the empirical analysis is performed through our approximation using the discrete-time Fourier transform of the sequence $y\Big(\frac{2\pi n}{N\omega_0}\Big)$, for $n = 0, \ldots, N-1$. In discrete time, we only have access to samples at times nT_s, for $n \in \mathbb{Z}$. Instead, similarly as earlier, we first assume that the signal has K components. We can then use that fact to express the known samples y_d as a

function of the unknown harmonic weights c_k and d_k as

$$y_d[n] = g(x_d[n]) \approx \frac{c_0}{2} + \sum_{k=1}^{K} c_k \cos(k\omega_0 n T_s) + d_k \sin(k\omega_0 n T_s) \tag{2.57}$$

for $n = 0, \ldots, N-1$. These form a set of linear equations to solve to find the weights c_k and d_k. The equations are independent if there is no rational relationship between ω_0 and $\omega_s = \frac{2\pi}{T_s}$. If the equations are independent, the system is under-determined for $N < 2K+1$, over-determined for $N > 2K+1$, and has a unique solution if $N = 2K+1$. Note that this approach presents some similarities with the method we will discuss in Sec. 2.2.2 for the case of dynamical systems.

2.2 Dynamical time-invariant nonlinear systems

The state-space description of a generic time-invariant dynamical nonlinear systems in continuous time is given by

$$\begin{aligned} \dot{\boldsymbol{x}}(t) &= \boldsymbol{f}_x(\boldsymbol{x}(t), u(t)), \text{ and} \\ y(t) &= f_y(\boldsymbol{x}(t), u(t)). \end{aligned} \tag{2.58}$$

2.2.1 Analytical periodic analysis

Performing analytical periodic analysis for the generic system shown above is difficult. However, most audio systems of interest exhibits a simpler structure. Following the structure found in Yeh [2009], Yeh et al. [2010], we know that typical nonlinear audio systems, especially electrical ones, can be written as the differential algebraic equations (DAEs)

$$\dot{\boldsymbol{x}}(t) = \mathbf{A}\boldsymbol{x}(t) + \mathbf{B}u(t) + \mathbf{C}\boldsymbol{i}(t), \tag{2.59a}$$

$$\boldsymbol{i}(t) = \boldsymbol{g}(\boldsymbol{v}(t)), \tag{2.59b}$$

$$\boldsymbol{v}(t) = \mathbf{D}\boldsymbol{x}(t) + \mathbf{E}u(t) + \mathbf{F}\boldsymbol{i}(t). \tag{2.59c}$$

where $\boldsymbol{g}$ is a pointwise nonlinearity (i.e., the nth term of the nonlinearity output vector depends solely the nth term in its input vector). Additionally, the output of the system y is generally formed as a linear combination of the systems variables (i.e., f_y is a linear function) so that the harmonic structure of the system is dictated by the harmonic structure of $\boldsymbol{x}$, u and $\boldsymbol{i}$.

In particular, this structure exposes the fact that the system nonlinearities are applied to linear combinations of the various systems variables, which simplifies the aliasing analysis. However, we must remind here that, on the contrary to memoryless systems, the bandwidth of the output signal is no longer constrained by the polynomial order of the nonlinear function $\boldsymbol{g}$ as long as it is applied to the state variables. In this case, the bandwidth is expected to be infinite, even for finite-order

nonlinearities.

The various quantities in the system for a periodic input are necessarily periodic as well and can all be decomposed as Fourier series

$$\begin{aligned} u(t) &= \sum_{k\in\mathbb{Z}} a_k^u e^{jk\omega_0 t}, \quad \boldsymbol{i}(t) = \sum_{k\in\mathbb{Z}} \boldsymbol{a}_k^i e^{jk\omega_0 t}, \\ \boldsymbol{v}(t) &= \sum_{k\in\mathbb{Z}} \boldsymbol{a}_k^v e^{jk\omega_0 t}, \quad \boldsymbol{x}(t) = \sum_{k\in\mathbb{Z}} \boldsymbol{a}_k^x e^{jk\omega_0 t}. \end{aligned} \tag{2.60}$$

These various quantities are linked through the systems equations. From Eq. (2.59a), we have

$$0 = (\mathbf{A} - jk\omega_0 \mathbf{I}_x)\boldsymbol{a}_k^x + \mathbf{B}a_k^u + \mathbf{C}\boldsymbol{a}_k^i, \tag{2.61}$$

and, from Eq. (2.59c), we get

$$\boldsymbol{a}_k^v = \mathbf{D}\boldsymbol{a}_k^x + \mathbf{E}a_k^u + \mathbf{F}\boldsymbol{a}_k^i. \tag{2.62}$$

Finally, from Eq. (2.59b), we have

$$\begin{aligned} \boldsymbol{a}_k^i &= \sum_{\substack{\{m_p\}\in\mathbb{Z}^P \\ \sum_p pm_p = k}} e^{j\sum_{p=1}^P m_p\left(p\omega_0 t + \angle(\boldsymbol{a}_p^v)\right)} \int_{\mathbb{R}} \mathbf{G}(\omega) e^{j\omega a_0^v} \prod_{p=1}^P j^{m_p} J_{m_p}(2|\boldsymbol{a}_k^v|\omega) d\omega \\ &= \sum_{\substack{\{m_p\}\in\mathbb{Z}^P \\ \sum_p pm_p = k}} e^{j\sum_{p=1}^P m_p\left(p\omega_0 t + \angle(\boldsymbol{a}_p^v)\right)} \int_{\mathbb{R}} \boldsymbol{g}(x)[\boldsymbol{h}_{1,m_1} * \cdots * \boldsymbol{h}_{P,m_P}](x - \boldsymbol{a}_0^v) dx \end{aligned} \tag{2.63}$$

with

$$\boldsymbol{h}_{p,m}(x) = \mathbf{1}_{[-1,1]}\left(\frac{x}{2|\boldsymbol{a}_p^v|}\right) \frac{1}{2\pi|\boldsymbol{a}_p^v|} \frac{T_m\left(\frac{x}{2|\boldsymbol{a}_p^v|}\right)}{\sqrt{1 - \left(\frac{x}{2|\boldsymbol{a}_p^v|}\right)}}. \tag{2.64}$$

Here, since the nonlinearity $\boldsymbol{g}$ is pointwise, the vectorial notations only refers to the fact that each element in $\boldsymbol{i}$ and $\boldsymbol{v}$ generates one equation relative to its respective coefficients.

2.2.2 Empirical periodic analysis

In a practical context, solving the system of equations from Eqs. (2.61), (2.62) and (2.63) would require tedious numerical solving with the appropriate level of approximations. Another more tractable approach follows the approach presented earlier in Sec. 2.1.1, by computing the FFT of the system signals at equidistant points. However, the case of dynamical systems requires an additional consideration due to the memory in the system, as it is not possible to straightforwardly derive the state waveforms $\boldsymbol{x}$ in the system.

The approach can be linked to the wider literature on steady-state analysis methods for nonlinear

systems and circuits [Vlach 2002]. The starting point of many approaches is the fact that we can run an high-accuracy circuit simulator until any transient effect becomes negligible before considering that the system has reached a steady state. Such an approach is however limited due to slow convergence issues with some systems, so that extensive research has been conducted on accelerating that convergence [Vlach and Singhal 1993].

Rather than focus on high-accuracy time-domain waveform recovery found in this literature, we want to focus on solving explicitly the harmonic structure of the signal. This approach echoes the so-called "harmonic balance" thread of research in microwave system modeling [Urabe 1965, Deuflhard 2011] This approach aims at efficiently calculating the steady-state response of nonlinear differential equation systems through an application of the Galerkin method.

The Galerkin method is a well-known approach to solving continuous problems with boundary conditions by expressing it as a discrete-problem. For example, this method is at the core of finite element analysis theory which attempts to solve partial differential equations. In this case, the main approach is referred to as the Galerkin method of weighted residuals. In essence, the Galerkin method attempts to approximately solve the continuous problem by constraining the function space of the solution to be a finite superposition of a chosen set of basis function. A common example in the case of finite element analysis is to constrain the approximate solution of a partial differential equation system to have a piecewise structure (e.g., linear, cubic) organized around fixed knots (e.g., points in two dimensions, line in three dimensions, etc...) that demarcate the boundary of each "element" (hence the name of the method).

In our case of interest, we are instead dealing with ordinary differential equations (or sometimes differential algebraic equations) which express the variation of a quantity in time. In particular, our systems of interest can generally be expressed as an *initial value problem*, for example in the state-space form

$$\begin{aligned} \dot{\boldsymbol{x}}(t) &= \boldsymbol{f}_x(\boldsymbol{x}(t), u(t)), \\ y(t) &= f_y(\boldsymbol{x}(t), u(t)), \\ \boldsymbol{x}(0) &= \boldsymbol{x}_0 \quad \text{(initial condition)}. \end{aligned} \tag{2.65}$$

In this scenario, the starting value of the physical quantities is known and we want to compute how they evolve over an undetermined amount of time (potentially ad infinitum), i.e., compute the values of $\boldsymbol{x}(t)$ and $y(t)$ for all $t \geq 0$. One important aspect here is that the system is so-called *non-autonomous*, as the nonlinear functions $\boldsymbol{f}_x$ and f_y are time-variant thanks to the input signal argument $u(t)$. In such a form, the Galerkin method cannot be employed since it applies to a different category of equation systems, the *boundary value problems*.

However, this changes as soon as we can make the hypothesis that the system solution is periodic

of period T_0, this constrain allows us to rewrite the problem as the *boundary value problem*

$$\begin{aligned} \dot{\boldsymbol{x}}(t) &= \boldsymbol{f}_x(\boldsymbol{x}(t), u(t)), \\ y(t) &= f_y(\boldsymbol{x}(t), u(t)), \\ \boldsymbol{x}(0) &= \boldsymbol{x}(T_0). \quad \text{(periodic boundary condition)} \end{aligned} \tag{2.66}$$

In this scenario, we constrain the system to verify our hypothesis of periodicity (of period T_0), so that we only need to compute the values of $\boldsymbol{x}(t)$ and $y(t)$ over the finite time interval $0 \leq t \leq T_0$. In this context, the implicit implication is that values at time outside of that interval are obtained simply by periodic extension of the one period computed from solving the boundary value problem. Effectively, our search for a periodic solution is equivalent to the objective of harmonic balance methods to find the steady-state response of a system.

An important observation at the basis of this approach is that most lumped audio systems of interest, and in particular for passive systems, will output a periodic waveform of period T_0 for a periodic input signal at that same period. A formal mathematical framework to prove the existence and uniqueness of such a periodic solution can be found in Urabe [1965].

Since our objective is to gather information regarding the periodic response of the system for a range of frequency, it is useful to rewrite Eq. (2.58) such that the boundary condition becomes independent of the period. After the change of variables $\omega = t\omega_0$, we can form the equations followed by $\tilde{\boldsymbol{x}}(\omega) = \boldsymbol{x}(\omega/\omega_0)$ and $\tilde{y} = y(\omega/\omega_0)$ as a function of the input $\tilde{u}(\omega) = u(\omega/\omega_0)$, resulting in the equations

$$\begin{aligned} \frac{d\tilde{\boldsymbol{x}}}{d\omega} &= \frac{1}{\omega_0}\boldsymbol{f}_x(\tilde{\boldsymbol{x}}(\omega), \tilde{u}(\omega)), \\ \tilde{y}(\omega) &= f_y(\tilde{\boldsymbol{x}}(\omega), \tilde{u}(\omega)), \\ \tilde{\boldsymbol{x}}(0) &= \tilde{\boldsymbol{x}}(1). \end{aligned} \tag{2.67}$$

Now that the equation system is formulated as a boundary value problem, it is suitable for the application of the Galerkin method. In particular, since the steady-state solutions of Eq. (2.67) are assumed to be periodic, we know they can be expressed through their Fourier series with fundamental radian frequency 2π as

$$\begin{aligned} \tilde{\boldsymbol{x}}(\omega) &= \sum_{k=0}^{+\infty} \tilde{\boldsymbol{b}}_k^x \cos(k\omega) + \sum_{k=1}^{+\infty} \tilde{\boldsymbol{c}}_k^x \sin(k\omega), \quad \text{and} \\ \tilde{y}(\omega) &= \sum_{k=0}^{+\infty} \tilde{b}_k^y \cos(k\omega) + \sum_{k=1}^{+\infty} \tilde{c}_k^y \sin(k\omega). \end{aligned} \tag{2.68}$$

A natural approach is then to do the same thing we did earlier in this chapter and attempt to

find approximate solutions as truncated Fourier series

$$\begin{aligned}\tilde{\boldsymbol{x}}(\omega) &\approx \sum_{k=0}^{K} \tilde{\boldsymbol{b}}_k^x \cos(k\omega) + \sum_{k=1}^{K-1} \tilde{\boldsymbol{c}}_k^x \sin(k\omega), \quad \text{and} \\ \tilde{y}(\omega) &\approx \sum_{k=0}^{K} \tilde{b}_k^y \cos(k\omega) + \sum_{k=1}^{K-1} \tilde{c}_k^y \sin(k\omega)\end{aligned} \tag{2.69}$$

with a choice of K large enough so that the missing terms are negligible and as a result do not create too big of a distortion on the remaining terms. Again, possible approaches to assess whether K is large enough is to compare the results of the approximation for a few different values of K, or at least to verify that the coefficient sequences b_k and c_k have tailed off to negligible values for values of k near K.

Note that there are no sinusoidal term for $k = 0$ and $k = K$. The term in $k = 0$ comes trivially from the fact that $\sin(0 \times \omega) \equiv 0$ so that the value of c_0 is irrelevant. The case $k = K$ comes from the fact that we will here discuss only the case where we sample the periodic waveform with an even number of samples (e.g., power of 2), so that there will be an intrinsic indeterminacy between the terms b_K and c_K. This indeterminacy echoes the fact that the even-long Fourier transform of a real signal has no phase on the Nyquist frequency bin. To break the indeterminacy, we arbitrarily set $c_K = 0$. In general, a proper value of K should be set such that b_K is very small so that we know our harmonic series does not present too much aliasing. Alternatively, to have both a coefficient b_K and a coefficient c_K, we will need to use an odd number of samples. In the rest of the chapter, we will only discuss the case for an even number of samples, as the extension to the case of an odd number of samples is trivial from there.

This approach is then equivalent to applying the Galerkin method to Eq. (2.67) as shown in Urabe [1965], since we are searching for approximate solutions to the system as the superposition of the finite set of functions $\cos(k\omega)$ (for k in $\{0, \ldots, K\}$) and $\sin(k\omega)$ (for k in $\{1, \ldots, K-1\}$). Urabe [1965] also provides a formal mathematical framework to verify the existence of such approximate solution.

We know that all these coefficients can be determined uniquely with the knowledge of the waveform samples at instants $\frac{2\pi n}{N}$ for $n = 0, \ldots, N-1$ and $N = 2K$ with the constraint of periodicity, meaning

$$\tilde{\boldsymbol{x}}(\omega) = \tilde{\boldsymbol{x}}(\omega + 2\pi) \quad \text{and} \quad \tilde{y}(\omega) = \tilde{y}(\omega + 2\pi). \tag{2.70}$$

We then form a system of nonlinear equations for $\boldsymbol{a}_k^x$ and a_k^y by writing Eq. (2.67) at the point

$\omega = \frac{2\pi n}{N}$ for $n = 0, \ldots, N-1$ as

$$\sum_{k=1}^{K-1} k\, \tilde{\boldsymbol{c}}_k^x \cos\left(\frac{2\pi kn}{N}\right) - \sum_{k=1}^{K} k\, \tilde{\boldsymbol{b}}_k^x \sin\left(\frac{2\pi kn}{N}\right) = \boldsymbol{f}_x\left(\sum_{k=0}^{K} \tilde{\boldsymbol{b}}_k^x \cos\left(\frac{2\pi kn}{N}\right) + \sum_{k=1}^{K-1} \tilde{\boldsymbol{c}}_k^x \sin\left(\frac{2\pi kn}{N}\right), \tilde{u}\left(\frac{2\pi n}{N}\right)\right), \tag{2.71a}$$

and

$$\sum_{k=0}^{K} \tilde{b}_k^y \cos\left(\frac{2\pi kn}{N}\right) + \sum_{k=1}^{K-1} \tilde{c}_k^y \sin\left(\frac{2\pi kn}{N}\right) = f_y\left(\sum_{k=0}^{K} \tilde{\boldsymbol{b}}_k^x \cos\left(\frac{2\pi kn}{N}\right) + \sum_{k=1}^{K-1} \tilde{\boldsymbol{c}}_k^x \sin\left(\frac{2\pi kn}{N}\right), \tilde{u}\left(\frac{2\pi n}{N}\right)\right). \tag{2.71b}$$

In the particular case where the input signal is more easily expressed in the form of its Fourier transform (e.g., if u is an ideal band-limited waveform, such as a band-limited triangle waveform), the equations would then become

$$\sum_{k=1}^{K-1} k\, \tilde{\boldsymbol{c}}_k^x \cos\left(\frac{2\pi kn}{N}\right) - \sum_{k=1}^{K} k\, \tilde{\boldsymbol{b}}_k^x \sin\left(\frac{2\pi kn}{N}\right) = \boldsymbol{f}_x\left(\sum_{k=0}^{K} \tilde{\boldsymbol{b}}_k^x \cos\left(\frac{2\pi kn}{N}\right) + \sum_{k=1}^{K-1} \tilde{\boldsymbol{c}}_k^x \sin\left(\frac{2\pi kn}{N}\right), \sum_{k=0}^{+\infty} \tilde{\boldsymbol{b}}_k^u \cos\left(\frac{2\pi kn}{N}\right) + \sum_{k=1}^{+\infty} \tilde{\boldsymbol{c}}_k^u \sin\left(\frac{2\pi kn}{N}\right)\right), \tag{2.72a}$$

and

$$\sum_{k=0}^{K} \tilde{b}_k^y \cos\left(\frac{2\pi kn}{N}\right) + \sum_{k=1}^{K-1} \tilde{c}_k^y \sin\left(\frac{2\pi kn}{N}\right) = f_y\left(\sum_{k=0}^{K} \tilde{\boldsymbol{b}}_k^x \cos\left(\frac{2\pi kn}{N}\right) + \sum_{k=1}^{K-1} \tilde{\boldsymbol{c}}_k^x \sin\left(\frac{2\pi kn}{N}\right), \sum_{k=0}^{+\infty} \tilde{\boldsymbol{b}}_k^u \cos\left(\frac{2\pi kn}{N}\right) + \sum_{k=1}^{+\infty} \tilde{\boldsymbol{c}}_k^u \sin\left(\frac{2\pi kn}{N}\right)\right). \tag{2.72b}$$

This formulation is especially relevant in the case where we are investigating the periodic response to input signals with finite Fourier series expansion, such as an ideal band-limited waveform (e.g., band-limited triangle wave, square wave, etc...).

Eq. (2.71a) (or Eq. (2.72a)) corresponds to a system of nonlinear fixed-point equations that would need to be solved using a numerical solver such as a Newton–Raphson solver [Press et al. 2007] or a trust-region solver [Coleman and Li 1996, Byrd et al. 2000, Waltz et al. 2006]. Such algorithm often require providing a Jacobian and/or a Hessian. The Jacobian with respect to $\tilde{\boldsymbol{b}}_k^x$ ($k = 0, \ldots, K$), of

Eq. (2.71a) is

$$-k\sin\left(\frac{2\pi kn}{N}\right)-\cos\left(\frac{2\pi kn}{N}\right)\nabla_{\boldsymbol{x}}\boldsymbol{f}_x(\tilde{\boldsymbol{x}}(\omega),\tilde{u}(\omega)), \tag{2.73}$$

and the Jacobian with respect to $\tilde{\boldsymbol{c}}_k^x$ $(k=1,\dots,K-1)$

$$k\cos\left(\frac{2\pi kn}{N}\right)-\sin\left(\frac{2\pi kn}{N}\right)\nabla_{\boldsymbol{x}}\boldsymbol{f}_x(\tilde{\boldsymbol{x}}(\omega),\tilde{u}(\omega)). \tag{2.74}$$

The Hessian with respect to $\tilde{\boldsymbol{b}}_k^x$ and $\tilde{\boldsymbol{b}}_{k'}^x$ $(k,k'=0,\dots,K)$ is

$$-\cos\left(\frac{2\pi kn}{N}\right)\cos\left(\frac{2\pi k'n}{N}\right)\nabla_{\boldsymbol{x},\boldsymbol{x}}\boldsymbol{f}_x(\tilde{\boldsymbol{x}}(\omega),\tilde{u}(\omega)), \tag{2.75}$$

the Hessian with respect to $\tilde{\boldsymbol{b}}_k^x$ and $\tilde{\boldsymbol{c}}_{k'}^x$ $(k=0,\dots,K,\ k'=1,\dots,K-1)$ is

$$-\cos\left(\frac{2\pi kn}{N}\right)\sin\left(\frac{2\pi k'n}{N}\right)\nabla_{\boldsymbol{x},\boldsymbol{x}}\boldsymbol{f}_x(\tilde{\boldsymbol{x}}(\omega),\tilde{u}(\omega)), \tag{2.76}$$

and the Hessian with respect to $\tilde{\boldsymbol{c}}_k^x$ and $\tilde{\boldsymbol{c}}_{k'}^x$ $(k=1,\dots,K-1,\ k'=1,\dots,K-1)$ is

$$-\sin\left(\frac{2\pi kn}{N}\right)\sin\left(\frac{2\pi k'n}{N}\right)\nabla_{\boldsymbol{x},\boldsymbol{x}}\boldsymbol{f}_x(\tilde{\boldsymbol{x}}(\omega),\tilde{u}(\omega)). \tag{2.77}$$

After solving for $\tilde{\boldsymbol{b}}_k^x$ and $\tilde{\boldsymbol{c}}_k^x$, we can find the coefficients $\tilde{b}_k^y$ and $\tilde{c}_k^y$ from Eq. (2.71b) (or Eq. (2.72b)) using the fast Fourier transform to compute the discrete-time Fourier transform of the sequence $f_y\big(\tilde{\boldsymbol{x}}\big(\frac{2\pi n}{N}\big),\tilde{u}\big(\frac{2\pi n}{N}\big)\big)$. Alternatively, it is actually possible to compute a larger number of harmonics for y by extrapolating x and u at a higher sampling rate if f_y is (strongly) nonlinear.

While the literature generally focuses on this approach just as a method to generate periodic solutions to system, it serves the dual purpose of providing us with an (albeit approximate) understanding of the distribution of the signal in the system's response among harmonics. And, in the context of system modeling, we can better visualize the distribution of the signal between the base band (below the sampling frequency) and outside of the base band.

In practice, we generally aim at analyzing the harmonic distribution of the signal across a wide frequency band. It is then practical to compute the response of the system by sweeping the frequency range using contiguous frequency values, and use the solution for one frequency as initial guess to compute the solution for the next one. As the two frequencies are chosen to be close, we expect these solutions to have a similar harmonic structure.

2.2.3 Discretization analysis

Interestingly enough, all the results from the previous section can be somewhat easily extended to the case of the discrete-time update equations of a discretized model of a given system. We can then

leverage this analysis to compare the breakdown of the signal across harmonics between the target continuous-time system and various discretization options. Then, this becomes a powerful tool for fast and accurate harmonic analysis of any discrete-time model.

To illustrate the concept of applying the harmonic balance approach to a discretized model, we derive the equations for the trapezoidal method (or equivalently standard bilinear transform, see Ch. 3). The derivation can be straightforwardly extended to a variety of discretization approach, in particular any linear multi-step discretization methods, as well as implicit midpoint-like methods (see Ch. 3).

It is well-known that discretizing the equation system shown in Eq. (2.65) leads to the discrete-time initial value problem

$$\begin{aligned} \frac{\boldsymbol{x}_d[n] - \boldsymbol{x}_d[n-1]}{T_s} &= \frac{\boldsymbol{f}_x(\boldsymbol{x}_d[n], u_d[n]) + \boldsymbol{f}_x(\boldsymbol{x}_d[n-1], u_d[n-1])}{2}, \\ y_d[n] &= f_y(\boldsymbol{x}_d[n], u_d[n]), \\ \boldsymbol{x}_d[0] &= \boldsymbol{x}_0 \quad \text{(initial condition)}. \end{aligned} \tag{2.78}$$

Here, we generally rely on two observable properties:

- For a periodic continuous-time input signal u, common practice is to use a discrete-time input signal u_d that is periodic as well, and,
- For a periodic discrete-time input signal u_d, the signals $\boldsymbol{x}_d$ and y_d will be periodic as well.

However, despite the similarities with the continuous-time case so far (see Sec. 2.2.2), we generally cannot rewrite the system in the form of Eq. (2.66). Indeed, except for period values T_0 that are integer multiple of the sampling period T_s, there will not be a discrete-time sample of any index n corresponding to instant T_0. Instead, we need to proceed as follows.

In the discrete-time case, the periodicity condition above still means that the discrete-time signals can still be expressed through Fourier series as

$$\begin{aligned} \boldsymbol{x}_d[n] &= \sum_{k=0}^{+\infty} \boldsymbol{b}_k^x \cos\left(2\pi k n \frac{T_s}{T_0}\right) + \sum_{k=1}^{+\infty} \boldsymbol{c}_k^x \sin\left(2\pi k n \frac{T_s}{T_0}\right), \quad \text{and} \\ y[n] &= \sum_{k=0}^{+\infty} b_k^y \cos\left(2\pi k n \frac{T_s}{T_0}\right) + \sum_{k=1}^{+\infty} c_k^y \sin\left(2\pi k n \frac{T_s}{T_0}\right). \end{aligned} \tag{2.79}$$

In such case, the periodicity condition can be understood as equivalent to the fact that the signals $\boldsymbol{x}_d$ and y_d can be written in the Fourier series form shown in Eq. (2.79). Indeed, the fact that there exists infinite coefficient series $\boldsymbol{b}^x$, $\boldsymbol{c}^x$, b^y, and c^y verifying that equation is a necessary and sufficient condition for periodicity.

Once again, in practice, it is not possible to find these infinite coefficient sequences, but we can

attempt to find a finite approximation as

$$\begin{aligned} \boldsymbol{x}_d[n] &\approx \sum_{k=0}^{K} \boldsymbol{b}_k^x \cos\left(2\pi k n \frac{T_s}{T_0}\right) + \sum_{k=1}^{K-1} \boldsymbol{c}_k^x \sin\left(2\pi k n \frac{T_s}{T_0}\right), \quad \text{and} \\ y[n] &\approx \sum_{k=0}^{K} b_k^y \cos\left(2\pi k n \frac{T_s}{T_0}\right) + \sum_{k=1}^{K-1} c_k^y \sin\left(2\pi k n \frac{T_s}{T_0}\right). \end{aligned} \tag{2.80}$$

The expression from Eq. (2.80) can then be inserted back into the system equations to find the conditions needed to be verified for the approximation sequences, as

$$\begin{aligned} \frac{4}{T_s}\left[\sum_{k=1}^{K-1} \boldsymbol{c}_k^x \sin\left(\pi k \frac{T_s}{T_0}\right) \cos\left(\pi k(2n-1)\frac{T_s}{T_0}\right) - \sum_{k=1}^{K} \boldsymbol{b}_k^x \sin\left(\pi k \frac{T_s}{T_0}\right) \sin\left(\pi k(2n-1)\frac{T_s}{T_0}\right)\right] = \\ \boldsymbol{f}_x\left(\sum_{k=0}^{K} \boldsymbol{b}_k^x \cos\left(2\pi k n \frac{T_s}{T_0}\right) + \sum_{k=1}^{K-1} \boldsymbol{c}_k^x \sin\left(2\pi k n \frac{T_s}{T_0}\right), u_d[n]\right) \\ + \boldsymbol{f}_x\left(\sum_{k=0}^{K} \boldsymbol{b}_k^x \cos\left(2\pi k(n-1) \frac{T_s}{T_0}\right) + \sum_{k=1}^{K-1} \boldsymbol{c}_k^x \sin\left(2\pi k(n-1) \frac{T_s}{T_0}\right), u_d[n]\right), \end{aligned} \tag{2.81a}$$

and

$$\begin{aligned} \sum_{k=0}^{K} b_k^y \cos\left(2\pi k n \frac{T_s}{T_0}\right) + \sum_{k=1}^{K-1} c_k^y \sin\left(2\pi k n \frac{T_s}{T_0}\right) = \\ f_y\left(\sum_{k=0}^{K} \boldsymbol{b}_k^x \cos\left(2\pi k n \frac{T_s}{T_0}\right) + \sum_{k=1}^{K-1} \boldsymbol{c}_k^x \sin\left(2\pi k n \frac{T_s}{T_0}\right), u_d[n]\right). \end{aligned} \tag{2.81b}$$

While, in general, this system of equations can be made to be as large as needed to estimate all the Fourier series coefficients by computing enough n, it will degenerate for the case where the sampling period T_s is an integer multiple of T_0. In that case, we can only form as many independent equations as that multiple. This is in part due to the fact that the sampled sequence u_d has an ambiguous content as aliasing (see Ch. 1) superposes the harmonic contributions in the base band and those out of the base band. Additionally, even if such an integer relationship is not present, this formulation can easily suffer from forming ill-conditioned equations which could slow down numerical solvers, and possibly prevent their convergence. These issues can be alleviated if we set the harmonic structure of u_d unambiguously for example through the exact definition of its Fourier series. In that case, we can actually interpolate the discrete-time systems between samples and generate new independent equations by leveraging the periodic extension from Eq. (2.80), and using

the interpolants

$$\begin{aligned}
\tilde{\boldsymbol{x}}_d(t) &= \sum_{k=0}^{K} \boldsymbol{b}_k^x \cos\left(2\pi k \frac{t}{T_0}\right) + \sum_{k=1}^{K-1} \boldsymbol{c}_k^x \sin\left(2\pi k \frac{t}{T_0}\right), \quad \text{and} \\
\tilde{y}_d(t) &= \sum_{k=0}^{K} b_k^y \cos\left(2\pi k \frac{t}{T_0}\right) + \sum_{k=1}^{K-1} c_k^y \sin\left(2\pi k \frac{t}{T_0}\right), \quad \text{and} \\
\tilde{u}_d(t) &= \sum_{k=0}^{+\infty} b_k^u \cos\left(2\pi k \frac{t}{T_0}\right) + \sum_{k=1}^{+\infty} c_k^u \sin\left(2\pi k \frac{t}{T_0}\right).
\end{aligned} \tag{2.82}$$

Alternatively, we can to be provided with a continuous-time interpolant $\tilde{u}_d(t)$ of $u_d[n]$, which should verify $u_d[n] = \tilde{u}_d(nT_s)$. This option is especially useful if the Fourier series describing u_d is intractable (e.g., if the coefficient sequences are infinite). We can then see here how this discussion echoes our broader discussion in the first chapter regarding the proper choice of interpolant/sampling so that the behavior of u_d and $\tilde{u}_d$ are consistent with the practical usage of the continuous-time system.

Once these interpolants are known, we can construct the system with the same sampling grid that we picked for the continuous time case (i.e., $\frac{nT_0}{N}$ for all n integer between 0 and $N-1$) in order to sample uniformly the period T_0, which leads us to the equations

$$\begin{aligned}
&\frac{4}{T_s}\left[\sum_{k=1}^{K-1} \boldsymbol{c}_k^x \sin\left(\pi k \frac{T_s}{T_0}\right) \cos\left(\pi k \left[\frac{2n}{N} + \frac{T_s}{T_0}\right]\right) - \sum_{k=1}^{K} \boldsymbol{b}_k^x \sin\left(\pi k \frac{T_s}{T_0}\right) \sin\left(\pi k \left[\frac{2n}{N} + \frac{T_s}{T_0}\right]\right)\right] = \\
&\qquad \boldsymbol{f}_x\left(\sum_{k=0}^{K} \boldsymbol{b}_k^x \cos\Big(2\pi k \frac{n}{N}\Big) + \sum_{k=1}^{K-1} \boldsymbol{c}_k^x \sin\Big(2\pi k \frac{n}{N}\Big), \tilde{u}\left(\frac{nT_0}{N}\right)\right) \\
&\quad + \boldsymbol{f}_x\left(\sum_{k=0}^{K} \boldsymbol{b}_k^x \cos\left(2\pi k \left[\frac{n}{N} + \frac{T_s}{T_0}\right]\right) + \sum_{k=1}^{K-1} \boldsymbol{c}_k^x \sin\left(2\pi k \left[\frac{n}{N} + \frac{T_s}{T_0}\right]\right), \tilde{u}\left(\frac{(n-1)T_0}{N}\right)\right),
\end{aligned} \tag{2.83a}$$

and

$$\begin{aligned}
&\sum_{k=0}^{K} b_k^y \cos\Big(2\pi k \frac{n}{N}\Big) + \sum_{k=1}^{K-1} c_k^y \sin\Big(2\pi k \frac{n}{N}\Big) = \\
&\qquad f_y\left(\sum_{k=0}^{K} \boldsymbol{b}_k^x \cos\Big(2\pi k \frac{n}{N}\Big) + \sum_{k=1}^{K-1} \boldsymbol{c}_k^x \sin\Big(2\pi k \frac{n}{N}\Big), \tilde{u}\left(\frac{(n-1)T_0}{N}\right)\right).
\end{aligned} \tag{2.83b}$$

Here again, we may want an explicit formulation of the Jacobian and Hessian with respect to the unknown Fourier series coefficients to feed to numerical solvers. The gradient of Eq. (2.81a) with respect to $\boldsymbol{b}_k^x$ is

$$-\frac{4}{T_s}\sin\left(\pi k\frac{T_s}{T_0}\right)\sin\left(\pi k(2n-1)\frac{T_s}{T_0}\right)-\cos\left(2\pi kn\frac{T_s}{T_0}\right)\nabla_{\boldsymbol{x}}\boldsymbol{f}_x(\boldsymbol{x}_d[n],\boldsymbol{u}_d[n])$$
$$-\cos\left(2\pi k(n-1)\frac{T_s}{T_0}\right)\nabla_{\boldsymbol{x}}\boldsymbol{f}_x(\boldsymbol{x}_d[n-1],\boldsymbol{u}_d[n-1]) \quad (2.84)$$

and with respect to $\boldsymbol{c}_k^x$

$$\frac{4}{T_s}\sin\left(\pi k\frac{T_s}{T_0}\right)\cos\left(\pi k(2n-1)\frac{T_s}{T_0}\right)-\sin\left(2\pi kn\frac{T_s}{T_0}\right)\nabla_{\boldsymbol{x}}\boldsymbol{f}_x(\boldsymbol{x}_d[n],\boldsymbol{u}_d[n])$$
$$-\sin\left(2\pi k(n-1)\frac{T_s}{T_0}\right)\nabla_{\boldsymbol{x}}\boldsymbol{f}_x(\boldsymbol{x}_d[n-1],\boldsymbol{u}_d[n-1]). \quad (2.85)$$

Then the Hessian with respect to $\boldsymbol{b}_k^x$ and $\boldsymbol{b}_{k'}^x$ is

$$-\cos\left(2\pi kn\frac{T_s}{T_0}\right)\cos\left(2\pi k'n\frac{T_s}{T_0}\right)\nabla_{\boldsymbol{x},\boldsymbol{x}}\boldsymbol{f}_x(\boldsymbol{x}_d[n],\boldsymbol{u}_d[n])$$
$$-\cos\left(2\pi k(n-1)\frac{T_s}{T_0}\right)\cos\left(2\pi k'(n-1)\frac{T_s}{T_0}\right)\nabla_{\boldsymbol{x},\boldsymbol{x}}\boldsymbol{f}_x(\boldsymbol{x}_d[n-1],\boldsymbol{u}_d[n-1]) \quad (2.86)$$

and, with respect to $\boldsymbol{b}_k^x$ and $\boldsymbol{c}_{k'}^x$

$$-\cos\left(2\pi kn\frac{T_s}{T_0}\right)\sin\left(2\pi k'n\frac{T_s}{T_0}\right)\nabla_{\boldsymbol{x},\boldsymbol{x}}\boldsymbol{f}_x(\boldsymbol{x}_d[n],\boldsymbol{u}_d[n])$$
$$-\cos\left(2\pi k(n-1)\frac{T_s}{T_0}\right)\sin\left(2\pi k'(n-1)\frac{T_s}{T_0}\right)\nabla_{\boldsymbol{x},\boldsymbol{x}}\boldsymbol{f}_x(\boldsymbol{x}_d[n-1],\boldsymbol{u}_d[n-1]) \quad (2.87)$$

and, finally, with respect to $\boldsymbol{c}_k^x$ and $\boldsymbol{c}_{k'}^x$

$$-\sin\left(2\pi kn\frac{T_s}{T_0}\right)\sin\left(2\pi k'n\frac{T_s}{T_0}\right)\nabla_{\boldsymbol{x},\boldsymbol{x}}\boldsymbol{f}_x(\boldsymbol{x}_d[n],\boldsymbol{u}_d[n])$$
$$-\sin\left(2\pi k(n-1)\frac{T_s}{T_0}\right)\sin\left(2\pi k'(n-1)\frac{T_s}{T_0}\right)\nabla_{\boldsymbol{x},\boldsymbol{x}}\boldsymbol{f}_x(\boldsymbol{x}_d[n-1],\boldsymbol{u}_d[n-1]). \quad (2.88)$$

We can get the equivalent quantities for Eq. (2.83a), for which the gradient with respect to $\boldsymbol{b}_k^x$ is

$$-\frac{4}{T_s}\sin\left(\pi k\frac{T_s}{T_0}\right)\sin\left(\pi k\left[\frac{2n}{N}+\frac{T_s}{T_0}\right]\right)-\cos\left(2\pi k\frac{n}{N}\right)\nabla_{\boldsymbol{x}}\boldsymbol{f}_x\left(\tilde{\boldsymbol{x}}_d\left(\frac{nT_0}{N}\right),\tilde{u}_d\left(\frac{nT_0}{N}\right)\right)$$
$$-\cos\left(2\pi k\frac{n-1}{N}\right)\nabla_{\boldsymbol{x}}\boldsymbol{f}_x\left(\tilde{\boldsymbol{x}}_d\left(\frac{(n-1)T_0}{N}\right),\tilde{u}_d\left(\frac{(n-1)T_0}{N}\right)\right), \quad (2.89)$$

and the one with respect to $\boldsymbol{c}_k^x$ is

$$\frac{4}{T_s}\sin\left(\pi k\frac{T_s}{T_0}\right)\cos\left(\pi k\left[\frac{2n}{N}+\frac{T_s}{T_0}\right]\right)-\sin\left(2\pi k\frac{n}{N}\right)\nabla_{\boldsymbol{x}}\boldsymbol{f}_x\left(\tilde{\boldsymbol{x}}_d\left(\frac{nT_0}{N}\right),\tilde{u}_d\left(\frac{nT_0}{N}\right)\right)$$

$$- \sin\Big(2\pi k\frac{n-1}{N}\Big)\nabla_{\boldsymbol{x}}\boldsymbol{f}_x\Big(\tilde{\boldsymbol{x}}_d\Big(\frac{(n-1)T_0}{N}\Big),\tilde{u}_d\Big(\frac{(n-1)T_0}{N}\Big)\Big). \quad (2.90)$$

Finally, the Hessian with respect to $\boldsymbol{b}_k^x$ and $\boldsymbol{b}_{k'}^x$ is

$$\begin{aligned}&- \cos\Big(2\pi k\frac{n}{N}\Big)\cos\Big(2\pi k'\frac{n}{N}\Big)\nabla_{\boldsymbol{x},\boldsymbol{x}}\boldsymbol{f}_x\Big(\tilde{\boldsymbol{x}}_d\Big(\frac{nT_0}{N}\Big),\tilde{u}_d\Big(\frac{nT_0}{N}\Big)\Big)\\ &\quad - \cos\Big(2\pi k\frac{n-1}{N}\Big)\cos\Big(2\pi k'\frac{n-1}{N}\Big)\nabla_{\boldsymbol{x},\boldsymbol{x}}\boldsymbol{f}_x\Big(\tilde{\boldsymbol{x}}_d\Big(\frac{(n-1)T_0}{N}\Big),\tilde{u}_d\Big(\frac{(n-1)T_0}{N}\Big)\Big), \quad (2.91)\end{aligned}$$

and, with respect to is $\boldsymbol{b}_k^x$ and $\boldsymbol{c}_{k'}^x$ is

$$\begin{aligned}&- \cos\Big(2\pi k\frac{n}{N}\Big)\sin\Big(2\pi k'\frac{n}{N}\Big)\nabla_{\boldsymbol{x},\boldsymbol{x}}\boldsymbol{f}_x\Big(\tilde{\boldsymbol{x}}_d\Big(\frac{nT_0}{N}\Big),\tilde{u}_d\Big(\frac{nT_0}{N}\Big)\Big)\\ &\quad - \cos\Big(2\pi k\frac{n-1}{N}\Big)\sin\Big(2\pi k'\frac{n-1}{N}\Big)\nabla_{\boldsymbol{x},\boldsymbol{x}}\boldsymbol{f}_x\Big(\tilde{\boldsymbol{x}}_d\Big(\frac{(n-1)T_0}{N}\Big),\tilde{u}_d\Big(\frac{(n-1)T_0}{N}\Big)\Big), \quad (2.92)\end{aligned}$$

and, finally, with respect to is $\boldsymbol{c}_k^x$ and $\boldsymbol{c}_{k'}^x$ is

$$\begin{aligned}&- \sin\Big(2\pi k\frac{n}{N}\Big)\sin\Big(2\pi k'\frac{n}{N}\Big)\nabla_{\boldsymbol{x},\boldsymbol{x}}\boldsymbol{f}_x\Big(\tilde{\boldsymbol{x}}_d\Big(\frac{nT_0}{N}\Big),\tilde{u}_d\Big(\frac{nT_0}{N}\Big)\Big)\\ &\quad - \sin\Big(2\pi k\frac{n-1}{N}\Big)\sin\Big(2\pi k'\frac{n-1}{N}\Big)\nabla_{\boldsymbol{x},\boldsymbol{x}}\boldsymbol{f}_x\Big(\tilde{\boldsymbol{x}}_d\Big(\frac{(n-1)T_0}{N}\Big),\tilde{u}_d\Big(\frac{(n-1)T_0}{N}\Big)\Big). \quad (2.93)\end{aligned}$$

As in the continuous-time case, once the coefficients for the Fourier series of $\boldsymbol{x}_d$ are found, we can use the second part of the system of equations (Eqs. (2.81b) and (2.83b)) to find the coefficients for the Fourier series of y_d. Here again, as in the continuous-time case, we actually do not have to match the Fourier series order for $\boldsymbol{x}_d$ and y_d and we can straightforwardly increase the order of y_d to accommodate for very nonlinear functions f_y. As in the continuous-time case, in a practical implementation, we would sweep the input period over the range of interest and use the computed solution from one period as initial guess for the next one.

Other discretization methods would lead to different equations through an equivalent derivation process as here. Additionally, while it may seem like solving the equations would become computationally challenging for large systems, we can leverage a lot of the findings from the harmonic balance literature to handle such large systems (this observation applies also to the continuous-time case). In particular, a typical approach is to divide any studied system in linear and nonlinear contributions, since the linear equations can be quickly converted back and forth between a Fourier series representation and a time-domain representation using the fast Fourier transform [Gilmore and Steer 1991a,b]. A nice thing to notice then is that most of the modeling formalism in the virtual analog literature already aim at efficiently splitting linear and nonlinear contributions to the system dynamics (e.g., nodal K-method [Yeh et al. 2010], wave digital filters [Werner 2016], generalized

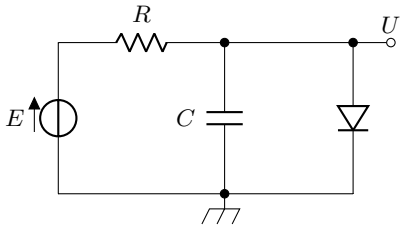


Figure 2.1: Schematic of a diode clipper circuit.

Variable	Value
R	$2.2\,\mathrm{k\Omega}$
C	$10\,\mathrm{nF}$
I_s	$2.52\,\mathrm{nA}$
V_T	$25.85\,\mathrm{mV}$
T_s	$10.417\,\mu\mathrm{s}$ ($f_s = 96\,\mathrm{kHz}$)

Table 2.1: Component and sampling period values for the diode clipper circuit (see Fig. 2.1) and its models.

state-space [Holters and Zölzer 2015]) so that the design of efficient equations to solve for arbitrary solutions and for periodic solutions are essentially the same.

2.2.4 Case study

In this section, we apply our empirical approach to the periodic analysis of a diode clipper circuit and three typical discretization approaches. The circuit we are modeling is shown in Fig. 2.1. As shown in Germain and Werner [2015], for the source input waveform $E(t)$, the measured node voltage $U(t)$ verifies the equation

$$\dot{U}(t) = \frac{E(t) - U(t)}{RC} - \frac{\phi(U(t))}{C}, \quad \text{with} \quad \phi(U) = I_s\Big(e^{U/V_T} - 1\Big), \tag{2.94}$$

assuming the diode is modeled using the Shockley ideal diode equation [Shockley 1949], where I_s and V_T are respectively the saturation current and the thermal voltage as is defined for this typical model of diode electrical elements. This system is an example of a dynamic nonlinear lumped audio system. The output variable $o(t)$ is equal to $U(t)$.

We then formulate the discretization methods we investigate:

- The bilinear transform (BT), or trapezoidal method. In this case, the discrete-time model of the system verifies the equation

$$\frac{U_{BT}[n] - U_{BT}[n-1]}{T_s} = \frac{E_d[n] - U_{BT}[n]}{2RC} + \frac{E_d[n] - U_{BT}[n-1]}{2RC}$$

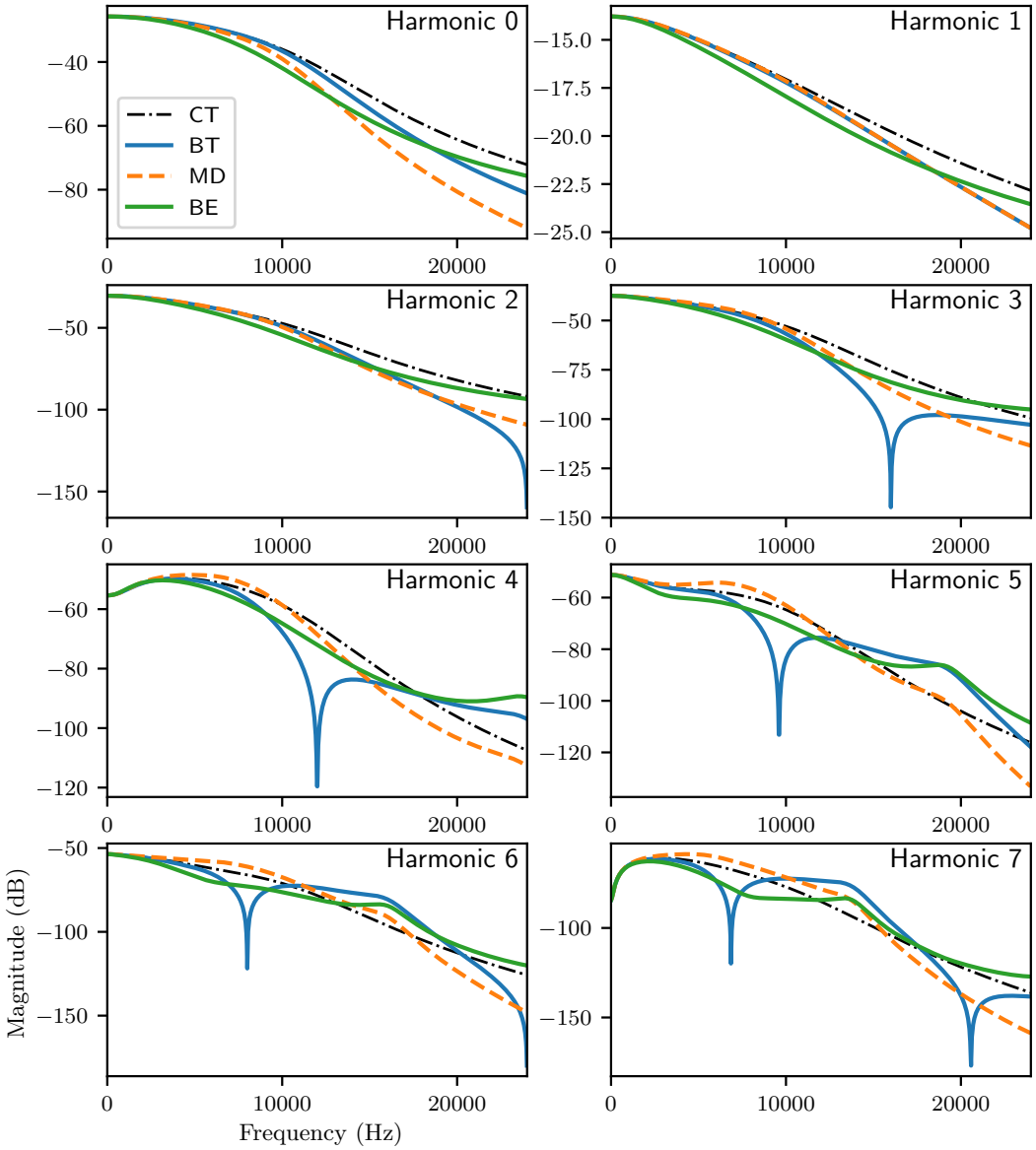


Figure 2.2: Estimated magnitude response for the first 7 harmonic components of the response associated with the diode clipper circuit for a cosinusoidal input of amplitude 0.5 as computed using our empirical estimation method. We display the response for the continuous-time system (CT, black) and 3 different numerical models: a bilinear transform (BT, blue), an implicit midpoint (MD, orange) and a backward Euler (BE, green) models. The magnitude is displayed in dB, as a function of the input cosinusoidal waveform frequency.

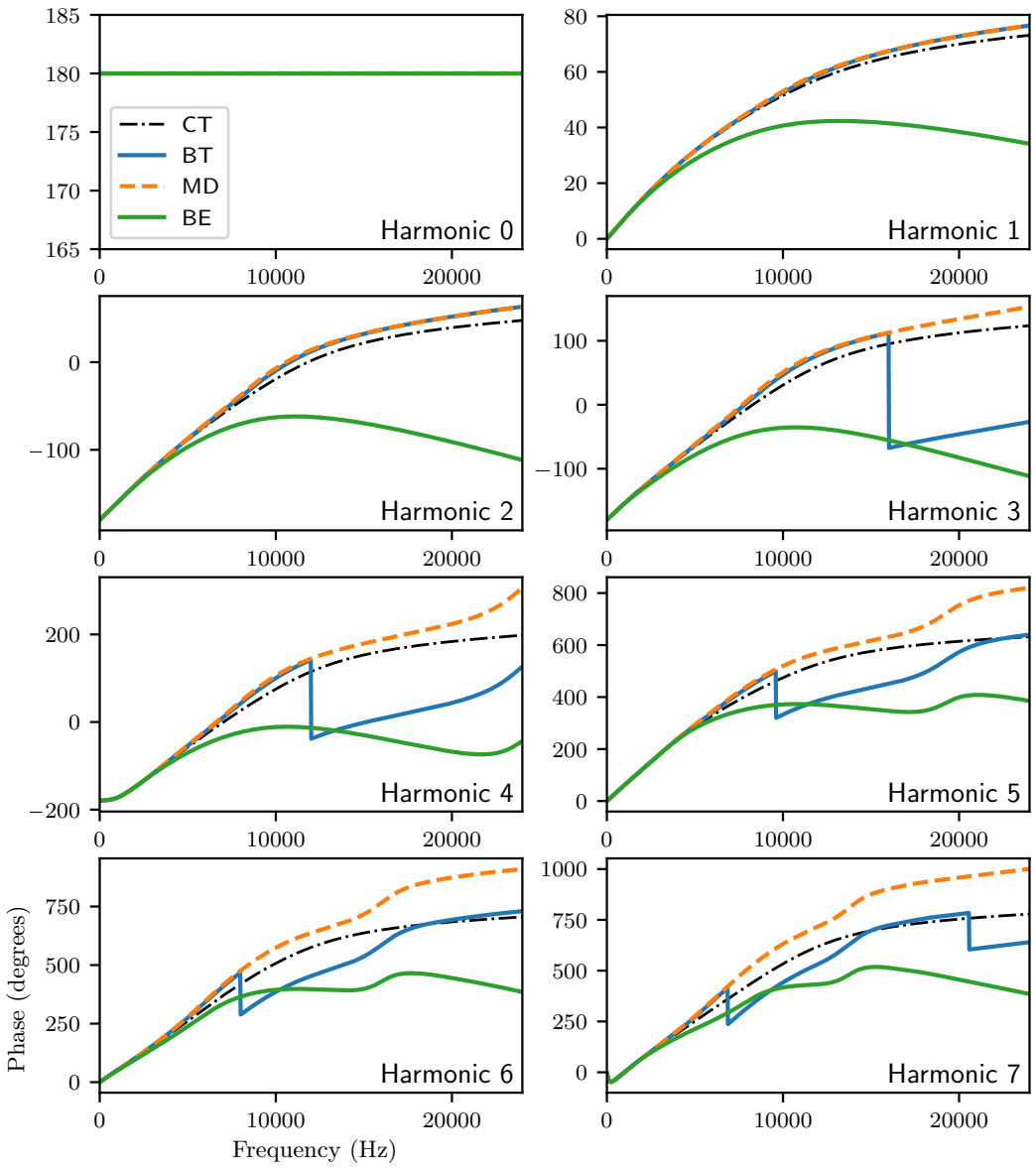


Figure 2.3: Estimated phase response for the first 7 harmonic components of the response associated with the diode clipper circuit for a cosinusoidal input of amplitude 0.5 as computed using our empirical estimation method. We display the response for the continuous-time system (CT, black) and 3 different numerical models: a bilinear transform (BT, blue), an implicit midpoint (MD, orange) and a backward Euler (BE, green) models. The phase is unwrapped and displayed in degrees, as a function of the input cosinusoidal waveform frequency.

$$- \frac{1}{2C}\phi(U_{BT}[n]) - \frac{1}{2C}\phi(U_{BT}[n-1]). \quad (2.95)$$

- The midpoint method (MD). In this case, the discrete-time model of the system verifies the equation

$$\frac{U_{MD}[n] - U_{MD}[n-1]}{T_s} = \frac{E_d[n] - U_{MD}[n]}{2RC} + \frac{E_d[n] - U_{MD}[n-1]}{2RC} - \frac{1}{C}\phi\left(\frac{U_{MD}[n] + U_{MD}[n-1]}{2}\right). \quad (2.96)$$

- The backward Euler method (BE). In this case, the discrete-time model of the system verifies the equation

$$\frac{U_{BE}[n] - U_{BE}[n-1]}{T_s} = \frac{E_d[n] - U_{BE}[n]}{RC} - \frac{1}{C}\phi(U_{BE}[n]). \quad (2.97)$$

Following our framework, we approximate the solutions to the system and its models as Fourier series with $K = 65$ harmonics when fed with a cosinusoidal input of amplitude 0.5, i.e.,

$$E(t) = \tilde{E}_d(t) = 0.5\cos(\omega_0 t) \quad (2.98)$$

where $\tilde{E}_d$ is the continuous interpolation used to find the input samples of the discrete-time model(s) $E_d[n] = \tilde{E}_d(nT_s)$. We sample 1,000 frequency values between 0 Hz and 48 kHz where to compute the harmonic weights. The system and model parameters are shown in Tab. 2.1. We compute the models at a sampling frequency of 96 kHz which, in current audio practice, generally corresponds to 2-times oversampling.

We set the equation systems as described in Eq. (2.72) for the continuous-time system, and as in Eq. (2.83) for the models (with the appropriate modifications relative to each method). We use the Newton trust-region solver [Nocedal and Wright 2006] from Julia's `NLSolve.jl` version 1.0.1 package[1]. The results for the first 8 harmonics (corresponding b_k for $k \in \{0, \ldots, 7\}$ and c_k for $k \in \{1, \ldots, 7\}$) are presented in terms of magnitude and phase in Figs. 2.2 and 2.3. It is then interesting to notice some similarities and discrepancies between methods. For example, we see that the responses for harmonic 1 are identical between the midpoint and the bilinear transform method. While this would be true for a linear system, it is interesting to see that the property is preserved for a nonlinear one as well. Another interesting point is to see how the magnitude response for all harmonics of the backward Euler method is not dramatically different from the ones of the midpoint and the bilinear transform method, but their phase response is radically different. Finally, we see this interesting feature from the bilinear transform that the response for the harmonics presents a comb filtering (which is visible in the dips of the magnitude response as well as the 180 degrees discontinuities in the phase response). We assume this behavior is the result from the "filtering" effect

[1] https://github.com/JuliaNLSolvers/NLsolve.jl

of the linear interpolation between the nonlinear values at time steps n and $n-1$ in the bilinear transform expression, since such filtering is not present in either the midpoint nor the backward Euler formulation. In some aspects, this study could be thought of as adding to the comparison discussion regarding behavioral differences between the bilinear transform and the midpoint methods as presented in Germain [2017].

2.3 Conclusion

In this chapter, we presented tools to enable the advanced analysis of the periodic response(s) of continuous-time systems and of their discrete-time models, i.e., their response when the input signal(s) are periodic. These tools fall into two categories:

- An analytical discussion of the harmonic contributions in the output waveform as a function of the harmonic contributions in the input waveform based on an extension of the results in Thornburg [1999], and,
- An empirical approach to efficiently compute these harmonic contributions inspired by the harmonic balance framework [Urabe 1965, Gilmore and Steer 1991a,b].

These tools were described for various cases of interest in the lumped audio system context, such the cases of static and dynamic nonlinear systems, and various categories of input waveforms (e.g., cosinusoidal, band-limited, etc...). The case of periodic input waveforms is an essential scenario in the context of audio systems, and the accurate knowledge of the harmonic response of these systems provides an invaluable insight in the advantages and drawbacks of various discretization strategies. In particular, it allows for the fast comparison of:

- The accuracy of the reproduction of harmonic contributions below the Nyquist frequency (i.e., the timbral "coloration" of the method), and,
- The suppression of harmonic contributions above the Nyquist frequency (e.g., in the case of anti-aliasing methods).

The known harmonic weights can in particular used to compute state-of-the-art aliasing distortion measures, such as the Total Harmonic Distortion (THD) [Blagouchine and Moreau 2011] or the A-weighted Noise-to-Mask Ratio (ANMR) [Lehtonen et al. 2012].

Chapter 3

Möbius transformation-based discretization design

In this section, we present our work regarding the analysis and the design of numerical methods based on Möbius transformations. These methods are a generalization of typical discretization methods such as the bilinear transform or the backward Euler method. It also encompasses methods more specific to the audio field such as the parametric bilinear transform which is often used to mitigate the frequency warping distortion of the bilinear transform. We derive the generic properties of these methods as a function of their parametrization, with a particular focus on how they map the poles of the target continuous-time system to the poles of the corresponding discrete-time model and how it may result in differences of behavior between the system and its model.

To illustrate the potential of this approach, we focus part of our analysis on the particular case the α-transform, which corresponds to a subclass of the Möbius transforms with a single free parameter, and encompassing the standard bilinear transform, the backward Euler method and the forward Euler method. Additionally, we describe a novel design criterion for parametrizing these methods in order to preserve damping monotonicity in the pole mapping as a function of the system dynamics. Finally, we apply it to the discretization of three historical lumped audio systems to validate it.

Core parts of the work in this chapter were previously published in Germain and Werner [2015] and Germain [2017].

3.1 Prior art

Some variations of the α-transform as shown in Sec. 3.7.3 have appeared before in the literature. The main area of development has dealt with issues of filter design and s-to-z mapping.

In Stilson and Smith III [1996], it was shown that by setting by hand the equivalent of the

parameter we will describe later in the context α-transform, the authors could tune a qualitative linear model of the Moog voltage-controlled filter to have a more consistent Q-factor at its resonance across the entire frequency band. In Stilson [2006], this degree of freedom is again mentioned as a possible avenue for designing constant-Q discrete-time filters. It was also proposed as an approach to generate improved approximations of the integrators, i.e., approximations of systems of the form

$$\frac{d^n x}{dt^n}(t) = u(t) \tag{3.1}$$

or its Laplace domain equivalent

$$s^n X(s) = U(s) \tag{3.2}$$

for all n [Al Alaoui 1993, Šekara and Stojić 2005, Šekara 2006, Al Alaoui 2006]. Finally, we find mentions of a general form of s-to-z transforms in the switched circuit literature [Dostál and Pospíšil 1985, Biolkova and Biolek 1999, Biolek and Biolkova 2001] where it was presented as a generalization of the bilinear transform, and the Euler mappings for switched circuit design. These articles provide for a mathematical methodology to derive a discrete-time transfer functions (see Eq. (3.10)) from its continuous-time transfer function (see Eq. (3.5)) based on the generalized Pascal matrix associated with the s-to-z mapping, but do not provide criteria for choosing the mapping parameters.

3.2 Time-invariant systems and pole/zero representation

3.2.1 Linear time-invariant continuous-time system representation

System equations

It is well-known that the dynamics of a continuous-time linear time-invariant system can be expressed using the differential equation

$$\sum_{k=0}^{K} a_k \frac{d^k}{dt^k} y(t) = \sum_{l=0}^{L} b_l \frac{d^l}{dt^l} x(t), \ \forall t \in \mathbb{R} \tag{3.3}$$

where x is the input signal and y is the output signal. For physical systems, the coefficients a_k and b_l are real numbers. These dynamics can alternatively be expressed using the Laplace transform, so that

$$\left[\sum_{k=0}^{K} a_k s^k\right] Y(s) = \left[\sum_{l=0}^{L} b_l s^l\right] X(s), \ \forall s \in \mathbb{C} \tag{3.4}$$

with X (respectively Y) the Laplace transform of x (resp. y).

Transfer function

The system behavior is then generally summarized by its transfer function $H(s)$ defined as the quantity $Y(s)/X(s)$. For a system whose behavior is summarized by Eq. (3.4), this means that we have

$$H(s) = \frac{\sum_{l=0}^{L} b_l s^l}{\sum_{k=0}^{K} a_k s^k}, \ \forall s \in \mathbb{C} \tag{3.5}$$

and the frequency response at radian frequency $\Omega \in \mathbb{R}$ is given by

$$H(j\Omega) = \frac{\sum_{l=0}^{L} b_l (j\Omega)^l}{\sum_{k=0}^{K} a_k (j\Omega)^k}, \ \forall \Omega \in \mathbb{R}. \tag{3.6}$$

Pole/zero decomposition

These expressions are ratios of polynomials, and the complex roots of the denominator (respectively numerator) polynomial are called *poles* (resp. *zeroes*). The location of these roots (especially the poles) are generally considered to be essential descriptors of the system behavior. In particular, typical physical linear systems are necessarily *passive* (or *energy dissipative*). This property translates in the property of *stability* for their descriptor (e.g., here, their transfer function). This property can be read readily in the pole locations, as a sufficient condition for a system to be stable is that the imaginary part of all of its poles is strictly negative (the condition becomes necessary if no zero is co-located with a pole).

The polynomials in the transfer function in Eq. (3.5) can also be factored to make the roots appear as

$$H(s) = \kappa \frac{\prod_{l=0}^{L} (s - q_l)}{\prod_{k=0}^{K} (s - p_k)}, \ \forall s \in \mathbb{C}. \tag{3.7}$$

A pole location is also given an *order*, corresponding to the number of poles at that location. For example, a second order pole location p means that there are exactly two poles p_k and $p_{k'}$ that verify $p_k = p_{k'} = p$. Zero locations can similarly be labeled depending on the number of zeroes at that location.

For physical systems, as the coefficients a_k and b_l are real, we know that all the complex pole locations p_k must verify either:

- p_k is real, or
- both p_k and $\overline{p_k}$ are pole locations of equal order. Equivalently, every non-real pole at location p_k must be associated with another pole at the conjugate location $\overline{p_k}$.

The same property is verified by the zero locations.

3.2.2 Linear time-invariant discrete-time system representation

System equations

On the other hand, a discrete-time linear time-invariant system is generally described by an equation of the form

$$\sum_{k=0}^{K_d} a_k y_d[n-k] = \sum_{l=0}^{L_d} b_l x_d[n-l],\ \forall n \in \mathbb{Z} \tag{3.8}$$

or the system z-transform:

$$\left[\sum_{k=0}^{K_d} a_k z^{-k}\right] Y_d(z) = \left[\sum_{l=0}^{L_d} b_l z^{-l}\right] X_d(z),\ \forall z \in \mathbb{C}. \tag{3.9}$$

Transfer function

The system z-transform then leads to the system's transfer function defined in a similar fashion as the continuous-time transfer function in Sec. 3.2.1 as $H_d(z) = Y_d(z)/X_d(z)$, leading for systems described by Eq. (3.9) to

$$H_d(z) = \frac{\sum_{l=0}^{L_d} b_l z^{-l}}{\sum_{k=0}^{K_d} a_k z^{-k}} \tag{3.10}$$

and its frequency response at radian frequency ω, expressed as

$$H_d\big(e^{j\omega T_s}\big) = \frac{\sum_{l=0}^{L_d} b_l \exp(-jl\omega T_s)}{\sum_{k=0}^{K_d} a_k \exp(-jk\omega T_s)},\ \forall \omega \in \mathbb{R} \tag{3.11}$$

where T_s is the sampling period of the discrete-time model sequences.

Pole/zero decomposition

Here again, poles and zeroes denote the roots of the denominator and numerator of the transfer function, and the transfer function in Eq. (3.10) can be factorized as

$$H_d(z) = \kappa \frac{\prod_{l=0}^{L}(1 - q_l z^{-1})}{\prod_{k=0}^{K}(1 - p_k z^{-1})},\ \forall z \in \mathbb{C}. \tag{3.12}$$

3.2.3 Pole properties

Basic pole properties are generally described using the solution to the linear test ordinary differential equation

$$\frac{d}{dt} y(t) = p\, y(t) + x(t),\ \forall t \in \mathbb{R} \tag{3.13}$$

with $p \in \mathbb{C}$. Such equation corresponds to a one-pole continuous-time system with one pole located at p, meaning a transfer function $H(s) = \frac{1}{s-p}$. The impulse response of such system is simply given

by

$$h(t) = \begin{cases} \exp(pt), & \text{for } t \geq 0 \\ 0, & \text{otherwise.} \end{cases} \tag{3.14}$$

The expression of $h(t)$ for $t \geq 0$ can then be decomposed as

$$h(t) = \underbrace{\exp(\sigma t)}_{\substack{\text{exponential} \\ \text{envelope}}} \times \underbrace{\exp(j\Omega t)}_{\substack{\text{oscillatory} \\ \text{function}}}, \quad \text{for } t \geq 0 \tag{3.15}$$

with $p = \sigma + j\Omega$, i.e., $\sigma = \Re(p)$ and $\Omega = \Im(p)$. A passive (or stable) system is then characterized by $\sigma < 0$, meaning that the exponential envelope is decaying over time. The decay rate (or *damping*) associated with the pole corresponds to the constant quantity $\frac{1}{|h(t)|}\frac{d|h|}{dt}(t) = \frac{d\log|h|}{dt}(t)$ which is constant for this impulse response for $t \geq 0$ as

$$\frac{1}{|h(t)|}\frac{d|h|}{dt}(t) = \frac{d\log|h|}{dt}(t) = \sigma. \tag{3.16}$$

On the other hand, the pole *radian frequency* is described by the quantity $\frac{d}{dt}\angle h(t)$ is also constant for $t \geq 0$ and equal to

$$\frac{d}{dt}\angle h(t) = \Omega. \tag{3.17}$$

Interestingly, this quantities are also equal to the quantities expressed with the discrete difference operator, i.e.:

$$T_s(\log|h(t+T_s)| - \log|h(t)|) = \sigma, \tag{3.18}$$

and

$$T_s(\angle h(t+T_s) - \angle h(t)) = \Omega. \tag{3.19}$$

These two expression allow to compare these quantities to the results of the discrete-time test equation given by

$$y_d[n] = p\, y_d[n-1] + x_d[n], \ \forall n \in \mathbb{Z} \tag{3.20}$$

with $p \in \mathbb{C}$. Such equation corresponds to a one-pole discrete-time system with one pole located at p, meaning a transfer function $H_d(z^{-1}) = \frac{1}{1-pz^{-1}}$. The impulse response of such system is simply given by

$$h_d[n] = \begin{cases} p^n, & n \in \mathbb{Z} \text{ and } n \geq 0 \\ 0, & n \in \mathbb{Z} \text{ and } n < 0. \end{cases} \tag{3.21}$$

The expression of $h_d[n]$ for $n \geq 0$ can then be decomposed as

$$h_d[n] = \underbrace{\exp(\log(r)n)}_{\substack{\text{geometric} \\ \text{envelope}}} \times \underbrace{\exp(j\omega n)}_{\substack{\text{oscillatory} \\ \text{function}}}, \quad \text{for } n \in \mathbb{Z} \text{ and } n \geq 0 \tag{3.22}$$

with $p = r\exp(j\omega)$, i.e., $r = |p|$ and $\omega = \angle p$. A passive (i.e., stable) system is then characterized by $r < 1$, meaning that the geometric envelope is decaying over samples. If we consider the system to be sampled at sampling period T_s, we can express an equivalent definition for damping and radian frequency. The damping is defined as $T_s \log\left(\frac{|h_d[n+1]|}{|h_d[n]|}\right)$, and is given as

$$T_s \log\left(\frac{|h_d[n+1]|}{|h_d[n]|}\right) = \log(r)T_s \tag{3.23}$$

and the radian frequency is defined as $T_s(\angle h_d[n+1] - \angle h_d[n])$, and given as

$$T_s(\angle h_d[n+1] - \angle h_d[n]) = \omega T_s. \tag{3.24}$$

3.3 Modeling as s-to-z mapping

3.3.1 Pole-zero conversion through s-to-z mapping

Over the years, many systematic procedures to perform such conversion were devised, such as impulse invariant designs or matched Z-transform design [Golden 1968, Parks and Burrus 1987, Smith III 2007b] . One category of interest to us are s-to-z mappings.

In that context, we obtain the transfer function $H_d(z^{-1})$ of the model from the transfer function $H(s)$ of the system by performing the substitution corresponding to the mapping $s \mapsto \mathcal{T}(z^{-1})$, so that

$$H_d(z) = H(\mathcal{T}(z)). \tag{3.25}$$

If the mapping is invertible, we can find the continuous-time system associated with a given model by applying the inverse substitution $z \mapsto \mathcal{T}^{-1}(s)$, i.e.,

$$H(s) = H_d\big(\mathcal{T}^{-1}(s)\big). \tag{3.26}$$

We can find the poles of the discrete-time system using Eq. (3.25). Indeed, these are defined as p such that $H_d(p) = \infty$, because a necessary and sufficient condition for p to be a pole of the discrete-time model is that $\mathcal{T}(p)$ is a pole of the continuous-time system. Similarly, the zeroes are defined as q such that $H_d(q) = 0$ if and only if $\mathcal{T}(q)$ is a zero of the continuous-time system.

Similarly, in the case of an invertible mapping, it is possible to deduce the poles and zeroes of a continuous-time system from the poles and zeroes of a converted discrete-time model following Eq. (3.26).

3.3.2 Modeling as pole-zero conversion procedure

Many modeling methods for linear time-invariant system rely on building a discrete-time system such that its number of poles and zeroes verifies

$$\max(K_d, L_d) = \max(K, L). \tag{3.27}$$

Another common version of such condition is made a bit differently by stating first that both systems have identical number of poles and zeroes ($K = L$ and $K_d = L_d$) by adding any "missing" pole or zero with one at $s = \infty$ and $z = 1$, so that the condition becomes that the transfer function of the continuous-time system and the discrete-time system have polynomial of identical order at the numerator and denominator so that

$$K = L = K_d = L_d. \tag{3.28}$$

Such methods then rely on describing what is essentially a "conversion" procedure between the s-plane and the z-plane. Because system stability is often one of the core properties of continuous-time systems and their discrete-time models, the methods often focus first on the locations of the poles in order to ensure that stable continuous-time systems are somewhat guaranteed to be modeled as stable discrete-time systems.

3.4 Linear one-step discretization methods

Linear one-step discretization methods are methods that solve the vector ODE $\dot{\boldsymbol{y}}(t) = \boldsymbol{g}(t, \boldsymbol{y}(t))$ using the discrete-time model written as

$$\boldsymbol{y}_d[n] + a_1 \boldsymbol{y}_d[n-1] = T_s[b_0 \boldsymbol{g}(t[n], \boldsymbol{y}_d[n]) + b_1 \boldsymbol{g}(t[n-1], \boldsymbol{y}_d[n-1])] \tag{3.29}$$

where a_1, b_0 and b_1 are fixed coefficients selected to fulfill some form of criterion.

In a typical application where the system is nonlinear time-invariant, its ordinary differential equation can be equivalently rewritten as the differential algebraic equation $\dot{\boldsymbol{y}}(t) = \boldsymbol{h}(\boldsymbol{y}(t), \boldsymbol{x}(t))$, in which case, linear one-step methods produce the discrete update equation

$$\boldsymbol{y}_d[n] + a_1 \boldsymbol{y}_d[n-1] = T_s[b_0 \boldsymbol{g}(\boldsymbol{y}_d[n], \boldsymbol{x}_d[n]) + b_1 \boldsymbol{g}(\boldsymbol{y}_d[n-1], \boldsymbol{x}_d[n-1])]. \tag{3.30}$$

The typical approach in numerical analysis to choose the fixed coefficients is to set all non-zero coefficients such that a maximum number of terms in the Taylor series expansion associated with the method are eliminated. This guarantees the maximum speed of convergence to the true solution when we reduce the sampling period of the model. Hence, for the general form, this formulation

leads to the trapezoidal method ($a_1 = -1$, $b_0 = b_1 = 1/2$) as a 2nd-order accurate method. If we set $b_0 = 0$, the formulation leads to the 1st-order accurate forward Euler method ($a_1 = -1$, $b_1 = 1$) and if we set $b_1 = 1$, the formulation leads to the 1st-order accurate backward Euler method ($a_1 = -1$, $b_0 = 1$).

3.5 Möbius transformations

Möbius transformations (also called homographic transformations, linear fractional transformations, fractional linear transformations or bilinear transformations) corresponds to rational functions from the complex plane $\mathbb{C}$ of the form

$$w(v) = \frac{\gamma_1 v + \gamma_2}{\gamma_3 v + \gamma_4}, \ \forall v \in \mathbb{C} \tag{3.31}$$

where the coefficients generally have to verify $\gamma_1\gamma_4 - \gamma_2\gamma_3 \neq 0$ to ensure that the function is invertible, with inverse

$$v(w) = \frac{\gamma_4 w - \gamma_2}{\gamma_1 - \gamma_3 w}, \ \forall w \in \mathbb{C}. \tag{3.32}$$

Möbius transformations describe *conformal* mappings of the complex plane, meaning that they map circles to circles (with lines considered as degenerate circles of infinite radius) and preserve right angles between curves. Möbius transformations are important tools in complex geometry. While the transformations are parametrized by four coefficients, they actually have three degrees of freedom, since for all nonzero complex scaling χ, the transformations parametrized by $(\chi\gamma_1, \chi\gamma_2, \chi\gamma_3, \chi\gamma_4)$ and $(\gamma_1, \gamma_2, \gamma_3, \gamma_4)$ describe the same mapping.

3.6 Möbius transformations as s-to-z mapping

3.6.1 Transfer function conversion

Möbius transformations as shown in Sec. 3.5 can be used to describe invertible s-to-z mappings between the s-plane and the z-plane as discussed in Sec. 3.3.1.

In that context, we obtain the transfer function $H_d(z)$ of the model from the transfer function $H(s)$ of the system by performing the substitution

$$s \ \mapsto \ \mathcal{T}(z) = \frac{\gamma_1 z + \gamma_2}{\gamma_3 z + \gamma_4} \tag{3.33}$$

meaning that

$$H_d(z) = H\left(\frac{\gamma_1 z + \gamma_2}{\gamma_3 z + \gamma_4}\right). \tag{3.34}$$

On the other hand, we can find the continuous-time system associated with a given model by

applying the substitution

$$z \;\mapsto\; \mathcal{T}^{-1}(s) = \frac{\gamma_1 - \gamma_3 s}{\gamma_4 s - \gamma_2} \tag{3.35}$$

meaning that:

$$H(s) = H_d\left(\frac{\gamma_1 - \gamma_3 s}{\gamma_4 s - \gamma_2}\right). \tag{3.36}$$

3.6.2 Pole/zero locations in Möbius transformations

If we express the substitution from Eq. (3.34) on the pole-zero factorization in Eq. (3.7), it becomes

$$H_d(z) = \kappa \frac{\prod_{l=0}^{L}(\gamma_1 - \gamma_3 q_l)}{\prod_{k=0}^{K}(\gamma_1 - \gamma_3 p_k)} \left[\left(\gamma_3 + \gamma_4 z^{-1}\right)^{K-L} \frac{\prod_{l=0}^{L}\left(1 - \frac{\gamma_4 q_l - \gamma_2}{\gamma_1 - \gamma_3 q_l} z^{-1}\right)}{\prod_{k=0}^{K}\left(1 - \frac{\gamma_4 p_k - \gamma_2}{\gamma_1 - \gamma_3 p_k} z^{-1}\right)} \right]. \tag{3.37}$$

The poles are then deduced from the poles p_k of the continuous-time system as

$$\mathcal{T}^{-1}(p_k) = \frac{\gamma_4 p_k - \gamma_2}{\gamma_1 - \gamma_3 p_k} \tag{3.38}$$

as predicted in Sec. 3.3.1. Similarly, the zeroes are deduced from the zeroes q_l of the continuous-time system as $\mathcal{T}^{-1}(q_l)$. Hence, we get the well-known property that the poles of the discrete-time model can be directly deduced from the poles of the continuous-time system by applying the same Möbius transformation to those pole and zero locations. If $K > L$, we see in the expression in Eq. (3.37) that $K - L$ additional zeroes appear at $z = -\frac{\gamma_4}{\gamma_3}$ so that $K_d = L_d = K$. Through reversing the mapping, we see that this additional zeroes can also be interpreted as a mapping of "silent" zeroes at infinity in the s-plane since the transformation maps $s = \infty$ to $z = -\frac{\gamma_4}{\gamma_3}$. A similar process happens in the case where $L > K$, except we know find $L - K$ poles added at $z = -\frac{\gamma_4}{\gamma_3}$.

As mentioned in Sec. 3.3.1, we can also do the converse deduction for the pole and zero locations of continuous-time system from knowledge of the pole and zero locations of the discrete-time model thanks to the invertibility of the conformal mapping described by the Möbius transformation.

3.6.3 Constraints on coefficients

While the only constraint on Möbius transformation required for it to be a conformal mapping operator is the one outlined in Sec. 3.5, i.e., $\gamma_1\gamma_4 - \gamma_2\gamma_3 \neq 0$, other constraints are often desirable in the context of pole-zero conversion procedures.

Real-axis symmetry

Verifying real-axis symmetry is generally desirable, meaning that the mapping verifies

$$\mathcal{T}(\overline{z}) = \overline{\mathcal{T}(z)}. \tag{3.39}$$

This property guarantees:

- That real poles in the s-plane are mapped to real poles in the z-plane and conversely, and
- That continuous-time transfer functions with real coefficients get converted into discrete-time transfer functions with real coefficients and conversely.

A necessary and sufficient condition to verify real-axis symmetry is that the coefficients γ_1, γ_2, γ_3 and γ_4 are real numbers. In the remaining of this chapter, we will assume that this property is always verified.

Frequency sign conservation

A stronger constraint on top of real-axis symmetry is frequency sign conservation, meaning insuring that positive frequencies in the s-plane (i.e., $\Omega \in \mathbb{R}^+$) map to "positive" frequencies in the z-plane (positive in the sense that $\omega \in [0, \pi]$ *mod* 2π). A necessary and sufficient condition to verify such property is to have

$$\gamma_1\gamma_4 - \gamma_2\gamma_3 > 0 \tag{3.40}$$

with γ_1, γ_2, γ_3 and γ_4 real numbers.

DC mapping

Another possible desirable property is to preserve "DC mapping", meaning that a pole with zero frequency and zero damping in the s-plane ($s = 0$) is mapped to a pole with zero frequency and zero damping in the z-plane ($z = 1$), i.e.,

$$\mathcal{T}(1) = 0 \tag{3.41}$$

which corresponds to the necessary and sufficient condition

$$\gamma_2 = -\gamma_1 \tag{3.42}$$

meaning that the Möbius transformation is reduced to

$$\mathcal{T}(z) = \gamma_1 \frac{z-1}{\gamma_3 z + \gamma_4}. \tag{3.43}$$

This condition updates the Möbius transformation condition to

$$\gamma_1(\gamma_4 + \gamma_3) \neq 0 \tag{3.44}$$

meaning that $\gamma_1 \neq 0$ and $\gamma_3 + \gamma_4 \neq 0$.

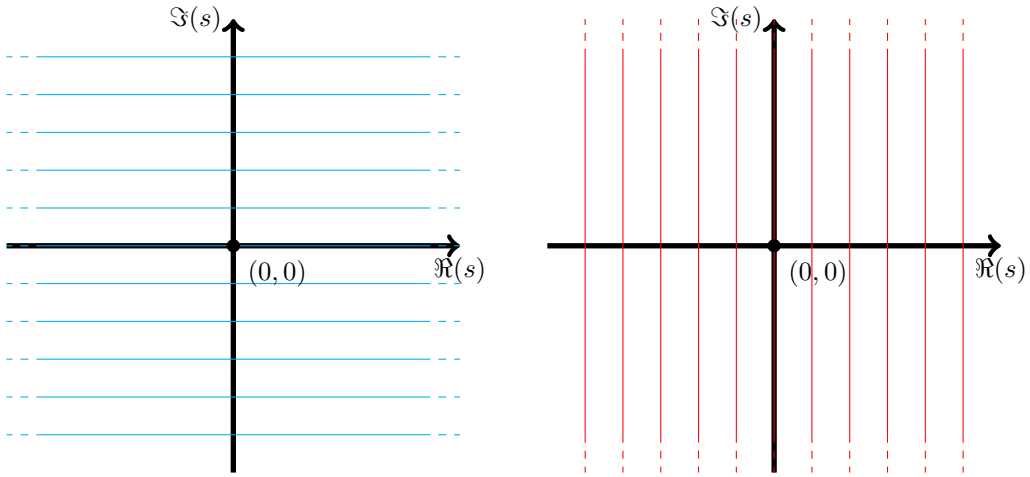


Figure 3.1: Pole iso-contours for frequency (left) and damping (right) in the case of continuous-time systems.

Zero-damping mapping

Another desirable property is to preserve "zero-damping mapping" meaning that a pole with zero damping in the s-plane (i.e., $\sigma = 0$) is mapped to a pole with zero damping in the z-plane (i.e., $|z| = 1$).

Since we assume that the mapping is real-axis symmetrical, it means that we have two possible cases:

1. $\mathcal{T}(1) = 0$ and $\mathcal{T}(-1) = \infty$. The necessary and sufficient conditions are given by $\gamma_2 = -\gamma_1$ and $\gamma_3 = \gamma_4$. In that case, the Möbius transformation condition becomes $\gamma_1\gamma_3 \neq 0$ meaning $\gamma_1 \neq 0$ and $\gamma_3 \neq 0$. The transformations then have the form

$$\mathcal{T}(z) = \frac{\gamma_1}{\gamma_3}\frac{z-1}{z+1}. \tag{3.45}$$

2. $\mathcal{T}(1) = \infty$ and $\mathcal{T}(-1) = 0$. The necessary and sufficient conditions are given by $\gamma_2 = \gamma_1$ and $\gamma_3 = -\gamma_4$. In that case, the Möbius transformation condition becomes the same as in the first case, meaning $\gamma_1\gamma_3 \neq 0$, or equivalently $\gamma_1 \neq 0$ and $\gamma_3 \neq 0$. The transformations then have the form

$$\mathcal{T}(z) = \frac{\gamma_1}{\gamma_3}\frac{z+1}{z-1}. \tag{3.46}$$

3.6.4 Pole iso-contours

From their definition, we can observe the well-known fact that pole iso-contours are different for continuous-time systems and discrete-time models.

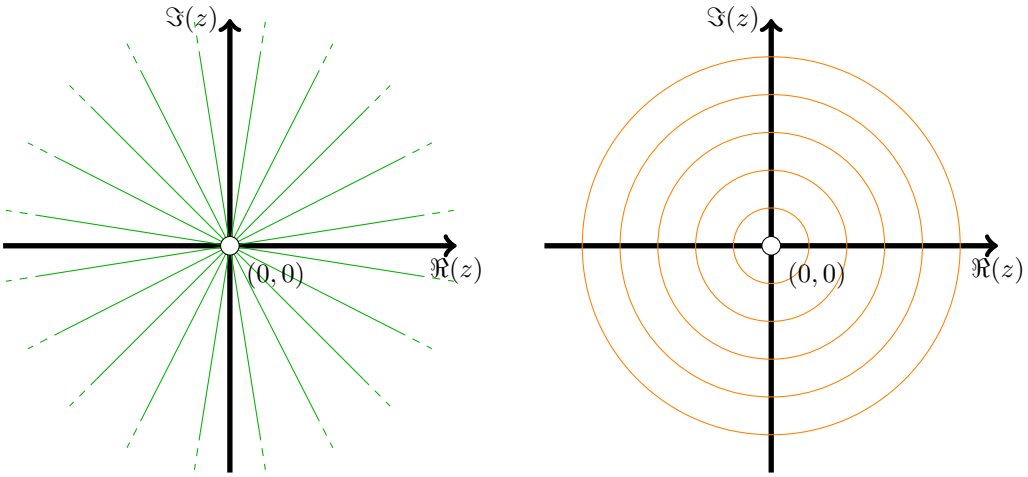


Figure 3.2: Pole iso-contours for frequency (left) and damping (right) in the case of discrete-time systems.

For continuous-time systems, frequency iso-contours correspond to lines with constant imaginary part (horizontal lines in the s-plane), and damping iso-contours correspond to lines with constant real part (vertical lines in the s-plane). Both types of iso-contours are shown in Fig. 3.1.

For discrete-time systems, frequency iso-contours correspond to semi-straight lines starting from the origin point $z = 0$ (radial semi-straight lines in the z-plane), and damping iso-contours correspond to circle centered around $z = 0$ (concentric circles in the z-plane). If we group frequencies using a π-modulo instead of 2π-modulo, the iso-contours become radial lines going through the origin $z = 0$. Both types of iso-contours are shown in Fig. 3.2.

3.6.5 Iso-contour projections through Möbius transformation

If we write points in the s-plane as $s = \sigma + j\Omega$ and points in the z-plane as $z = re^{j\omega}$, we can write the conversion equations from s to z for those parameters as

$$\begin{cases} r(\sigma,\Omega) = \sqrt{\dfrac{(\gamma_1-\sigma\gamma_3)^2+(\Omega\gamma_3)^2}{(\gamma_2-\sigma\gamma_4)^2+(\Omega\gamma_4)^2}} \\ \tan\left(\dfrac{1}{2}\omega(\sigma,\Omega)\right) = \dfrac{1}{\sqrt{((\sigma\gamma_4-\gamma_2)(\sigma\gamma_3-\gamma_1)+\Omega^2\gamma_3\gamma_4)^2+\Omega^2(\gamma_1\gamma_4-\gamma_2\gamma_3)^2}} \\ \qquad\qquad \times \dfrac{\Omega(\gamma_1\gamma_4-\gamma_2\gamma_3)}{1-\dfrac{(\sigma\gamma_4-\gamma_2)(\sigma\gamma_3-\gamma_1)-\Omega^2\gamma_3\gamma_4}{\sqrt{((\sigma\gamma_4-\gamma_2)(\sigma\gamma_3-\gamma_1)+\Omega^2\gamma_3\gamma_4)^2+\Omega^2(\gamma_1\gamma_4-\gamma_2\gamma_3)^2}}} \end{cases} \tag{3.47}$$

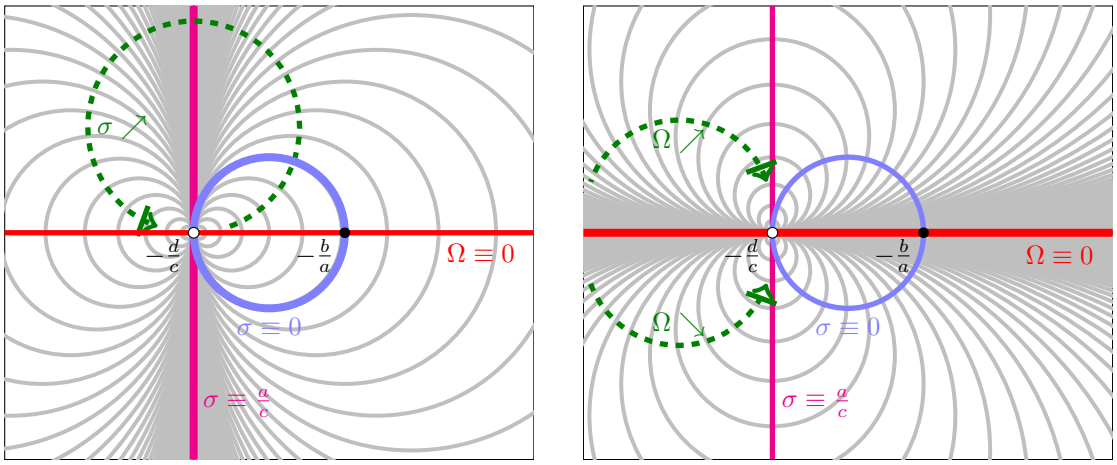


Figure 3.3: Projections of the continuous-time iso-contour in the z-plane for damping (left) and frequency (right). The example here correspond to the case where $\gamma_1\gamma_4 - \gamma_2\gamma_3 > 0$.

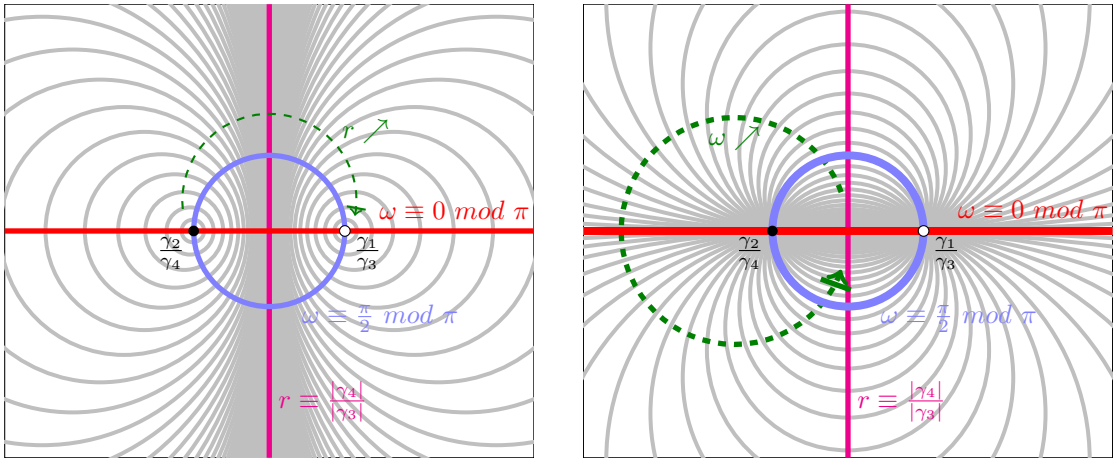


Figure 3.4: Projections of the discrete-time iso-contour in the s-plane for damping (left) and frequency module π (right). The example here correspond to the case where $\gamma_1\gamma_4 - \gamma_2\gamma_3 > 0$.

and from z to s as

$$\begin{cases} \sigma(r,\omega) = \dfrac{r^2\gamma_1\gamma_3 + \gamma_2\gamma_4 + r\cos(\omega)(\gamma_1\gamma_4 + \gamma_2\gamma_3)}{\gamma_4^2 + r^2\gamma_3^2 + 2r\cos(\omega)\gamma_3\gamma_4} \\ \Omega(r,\omega) = \dfrac{r\sin(\omega)(\gamma_1\gamma_4 - \gamma_2\gamma_3)}{\gamma_4^2 + r^2\gamma_3^2 + 2r\cos(\omega)\gamma_3\gamma_4} \end{cases} \tag{3.48}$$

More interesting is to look at the projection of iso-contours from one plane into the other. By property of conformal mapping, as the iso-contours are all circles or lines, we know that they get mapped to circles or lines.

Continuous-time damping iso-contours

We have two main cases to consider for the mapping of the continuous-time damping iso-contours (vertical lines):

- if $\gamma_3 = 0$, the transformation is reduced to the translation $\frac{\gamma_4}{\gamma_1}s - \frac{\gamma_2}{\gamma_1}$, so each iso-contour is mapped to the vertical line corresponding to $\Re(z) = \frac{\gamma_4}{\gamma_1}\sigma - \frac{\gamma_2}{\gamma_1}$,

- otherwise, the iso-contours are mapped to a circle of center ζ_σ and radius R_σ such that

$$\begin{aligned} \zeta_\sigma &= \frac{1}{2}\left(\frac{\gamma_4\sigma - \gamma_2}{\gamma_1 - \gamma_3\sigma} - \frac{\gamma_4}{\gamma_3}\right) \text{ and} \\ R_\sigma &= \frac{|\gamma_1\gamma_4 - \gamma_2\gamma_3|}{2|\gamma_3(\gamma_3\sigma - \gamma_1)|} \end{aligned} \tag{3.49}$$

  except for the iso-contour corresponding to $\sigma = \gamma_1/\gamma_3$ which maps to the vertical line corresponding to $\Re(z) = -\gamma_4/\gamma_3$.

Continuous-time frequency iso-contours

Here again, we have the two same cases to consider for the mapping of the continuous-time frequency iso-contours (horizontal lines):

- if $\gamma_3 = 0$, the transformation is reduced to a translation so that each iso-contour is mapped to the horizontal line corresponding to $\Im(z) = \frac{\gamma_4}{\gamma_1}\Omega$,

- otherwise, the iso-contours are mapped to a circle of center ζ_Ω and radius R_Ω such that

$$\begin{aligned} \zeta_\Omega &= -\frac{\gamma_4}{\gamma_3} + j\frac{\gamma_1\gamma_4 - \gamma_2\gamma_3}{2\gamma_3^2\Omega} \text{ and} \\ R_\Omega &= \frac{|\gamma_1\gamma_4 - \gamma_2\gamma_3|}{2\gamma_3^2|\Omega|}. \end{aligned} \tag{3.50}$$

Discrete-time damping iso-contours

There is only one case to consider for the mapping of the discrete-time damping iso-contours (concentric circles centered at $z = 0$), which are mapped to a circle of center ζ_r and radius R_r such that

$$\begin{aligned} \zeta_r &= \frac{\gamma_1\gamma_3 r^2 - \gamma_2\gamma_4}{\gamma_3^2 r^2 - \gamma_4^2} \quad \text{and} \\ R_r &= \frac{r|\gamma_1\gamma_4 - \gamma_2\gamma_3|}{|\gamma_3^2 r^2 - \gamma_4^2|} \end{aligned} \tag{3.51}$$

except for the iso-contour corresponding to $r = |\gamma_4|/|\gamma_3|$ which is mapped to the vertical line corresponding to $\Re(s) = \frac{\gamma_1\gamma_4+\gamma_2\gamma_3}{2\gamma_3\gamma_4}$.

Discrete-time frequency iso-contours

The discrete-time frequency iso-contours defined for frequencies modulo π (radial lines from $z = 0$) are mapped to a circle of center ζ_ω and radius R_ω such that:

$$\begin{aligned} \zeta_\omega &= \frac{\gamma_1\gamma_4 + \gamma_2\gamma_3}{2\gamma_3\gamma_4} - j\frac{\gamma_1\gamma_4 - \gamma_2\gamma_3}{2\gamma_3\gamma_4 \tan\omega} \quad \text{and} \\ R_\omega &= \frac{|\gamma_1\gamma_4 - \gamma_2\gamma_3|}{2|\gamma_3\gamma_4 \sin\omega|} \end{aligned} \tag{3.52}$$

except for the iso-contour corresponding to $\omega = m\pi$ with $m \in \mathbb{Z}$ (horizontal line $\Im(z) = 0$) which is mapped to itself, meaning $\Im(s) = 0$. This circle is divided in two arcs corresponding to the mapping from the radial semi-straight line corresponding to $\omega + 2m\pi$ and the radial semi-straight line corresponding to $\omega + (2m + 1)\pi$ with $m \in \mathbb{Z}$. The two arcs are connect at the points $s = \frac{\gamma_2}{\gamma_4}$ and $s = \frac{\gamma_1}{\gamma_3}$ that belong to all the projected iso-contours.

3.7 Linear one-step discretization methods and Möbius transformations

3.7.1 Numerical methods as pole mapping

It is a well-known property that, when the system is linear time-invariant, i.e., when the equations of the system are written as

$$\dot{\boldsymbol{y}}(t) = \mathbf{A}\boldsymbol{y}(t) + \mathbf{B}\boldsymbol{x}(t), \tag{3.53}$$

the formula in Eq. (3.30), which becomes

$$\boldsymbol{y}_d[n] + a_1\boldsymbol{y}_d[n-1] = T_s\mathbf{A}[b_0\boldsymbol{y}_d[n] + b_1\boldsymbol{y}_d[n-1]] + T_s\mathbf{B}[b_0\boldsymbol{x}_d[n] + b_1\boldsymbol{x}_d[n-1]] \tag{3.54}$$

shows that the formula in Eq. (3.30) is equivalent to the s-to-z mapping

$$s \mapsto \frac{1}{T_s} \frac{1 + a_1 z^{-1}}{b_0 + b_1 z^{-1}}. \tag{3.55}$$

For this reason, these discretization methods are often summarized using the equivalent s-to-z mapping, leading for example to the equivalence between the bilinear transform and the trapezoidal method.

3.7.2 Stability, consistency and order of accuracy

We mentioned in the introduction (see Sec. 0.4.2) the concepts found in numerical analysis regarding the stability, consistency and convergence of discretization methods. Stability is generally analyzed by analyzing the discretization method when applied to the test equation

$$\dot{x}(t) = p\, x(t), \tag{3.56}$$

for p in the set of complex numbers with a negative real part, whose solution has a decaying exponential envelope as a function of time.

Most numerical methods are linear, meaning they will convert linear time-invariant continuous-time equations such as Eq. (3.56) to a linear time-invariant discrete-time equation (see Eq. (3.2.2)). That discrete-time system has one or more poles $p'_k \in \mathbb{C}$. Stability is then determined based on properties of the location of p'_k as a function of p. The most common definition of stability corresponds to the fact that a continuous-time system expected to generate a bounded output for any bounded input gets converted into a discrete-time model generating a bounded output as well for any bounded input. When we can describe the discretization process in terms of pole mapping, this condition can be expressed in terms of pole location. It translates to the fact that continuous-time poles with negative damping get mapped to discrete-time poles with modulus lesser than 1. In a few cases, a method can be *unconditionally stable*, meaning the mapping conditions (modulus lesser than 1) for stability are met for all continuous-time poles p with negative damping. But the more general case is *conditional stability* where the mapping conditions are satisfied only for some values of p that are then gathered as a set called the *stability region* [Moin 2010]. A method that is unconditionally stable for this definition is also referred to as an *absolute-stable* (or *A-stable*) method. Stronger conditions for stability can also be found in the literature, such as L-stability [Hairer and Wanner 1996] which essentially requires that that poles at infinity in the s-plane get mapped to 0 in the z-plane. Another one is Lyapounov stability [Hélie 2011], which provides a sufficient condition for model stability for a specific nonlinear system. However, Lyapounov analysis requires finding a so-called Lyapounov (or storage) function to formalize that system-specific stability condition, and no systematic procedure is generally available to do so. Additionally, both these stability conditions

are often overly conservative compared to the more common A-stability condition.

Consistency on the other hand, is done as follows. First, we discretize the test equation

$$\dot{x}(t) = g(t, x(t)) \tag{3.57}$$

into

$$\dot{x_d}[n] = g(t_n, t_{n-1}, \ldots, x_d[n-1], \ldots). \tag{3.58}$$

Following this, we calculate the Taylor equation between the true solution $x(t_{n+1})$ (i.e., the solution of Eq. (3.57) at t_{n+1}) and the approximate solution $x_d[n+1]$ (i.e., the solution of Eq. (3.58) at t_{n+1}) for identical initial conditions $x(t_n) = x_d[n]$ at t_n. Consistency is then proven if we can verify the limit condition

$$|x(t_{n+1}) - x_d[n+1]| \xrightarrow[|t_{n+1}-t_n|\to 0]{} 0, \tag{3.59}$$

i.e., the fact that the difference between the true solution of the continuous-time system and the result of one iteration step of the discrete-time model converges to zero as the sampling period vanishes (i.e., gets closer to zero).

In general, a sufficient condition when analyzing the Taylor expansion is to have

$$x(t_{n+1}) - x_d[n+1] = \mathcal{O}(t_{n+1} - t_n). \tag{3.60}$$

This Taylor expansion is also used to determine the order of accuracy of the method, defined as the highest integer m such that

$$x(t_{n+1}) - x_d[n+1] = \mathcal{O}((t_{n+1} - t_n)^m). \tag{3.61}$$

Transforms with fixed coefficients

Most numerical methods have been traditionally defined using fixed coefficients. In our context, this corresponds to methods where the expression of the form as shown in Eq. (3.29) is set with fixed coefficients a_1, b_0 and b_1. The equivalent linear one-step method associated with a given Möbius transform $(\gamma_1, \gamma_2, \gamma_3, \gamma_4) \in \mathbb{R}^4$, assuming $\gamma_1 \neq 0$ can be expressed in that form as

$$x_d[n+1] + a_1 x_d[n] = T_s b_0 g(t_{n+1}, x_d[n+1]) + T_s b_1 g(t_n, x_d[n]) \tag{3.62}$$

with $a_1 = \gamma_2/\gamma_1$, $b_0 = \gamma_3/(T_s\gamma_1)$, $b_1 = \gamma_4/(T_s\gamma_1)$ and $T_s = t_{n+1} - t_n$.

For the test equation Eq. (3.56), we get that

$$p' = \frac{\gamma_4 p - \gamma_2}{\gamma_1 - \gamma_3 p} \tag{3.63}$$

in which case, we get the following:

- Methods verifying the following conditions:

$$|\gamma_2| \leq |\gamma_1| \quad \text{and} \quad |\gamma_4| \leq |\gamma_3| \tag{3.64}$$

  are A-stable, and

- for other methods, the stability region is defined as the reciprocal mapping of the unit circle, meaning the set

$$p \in \left\{ p = \frac{\gamma_4 p' - \gamma_2}{\gamma_1 - \gamma_3 p'}, \quad \text{for } |p'| < 1 \right\} \tag{3.65}$$

  which by property of mappings based on Möbius transforms must be either the inside of a circle or the outside of circle whose center lies on the real axis in the s-plane, or a half-plane whose boundary is a vertical line in the s-plane.

In terms of consistency analysis, we get the Taylor expansion formulation as

$$x(t_{n+1}) - x_d[n+1] = (1 + a_1)x(t_n) + \mathcal{O}(T_s) \tag{3.66}$$

so that consistency is verified as long as $a_1 = -1$, i.e., $\gamma_1 = -\gamma_2$.

1st-order methods can be found by continuing the Taylor expansion under the $a_1 = -1$ hypothesis, i.e.,

$$x(t_{n+1}) - x_d[n+1] = T_s(1 - b_0 - b_1)\dot{x}(t_n) + \mathcal{O}(T_s^2) \tag{3.67}$$

so that 1st-order accurate methods verify $b_0 + b_1 = 1$, i.e., $\gamma_3 + \gamma_4 = \gamma_1$.

2nd-order methods can then be found by continuing the Taylor expansion under the $a_1 = -1$ and $b_0 + b_1 = 1$ hypotheses, i.e.,

$$x(t_{n+1}) - x_d[n+1] = T_s^2 \left(\frac{1}{2} - b_0 \right) \ddot{x}(t_n) + \mathcal{O}(T_s^3) \tag{3.68}$$

so that the only 2nd-order accurate method verifies $b_0 = 1/2$, i.e., $\gamma_3 = 2\gamma_1$.

We recover the fact that mappings corresponding to the Euler methods lead to 1st-order accurate methods, and that the bilinear transform (which corresponds to the trapezoidal method) is the only 2nd-order accurate method with fixed coefficients.

Transforms with dependent coefficients

While numerical methods have been traditionally defined with fixed coefficients in forms similar to the one described above, filter design methods have shown that it is technically possible to design methods with what we will call here *dependent* coefficients. The most prevalent example of this

concept is what we call the *parametric bilinear transform* (PBT), that is often used in the context of converting continuous-time linear filter prototypes into discrete-time linear filter architectures. The parametric bilinear transform corresponds to the general mapping

$$s \mapsto \eta \frac{1 - z^{-1}}{1 + z^{-1}} \tag{3.69}$$

where η is a free parameter chosen to fit the chosen application. It is then easy to see that, when picking $\eta = 2/T$, we obtain what we denote the *standard bilinear transform* whose mapping corresponds to the well-known trapezoidal method. This method is known to create a distortion known as frequency warping, due to the way the imaginary axis in the s-plane is mapped onto the unit circle in the z-plane (see Smith III [2007b], Oppenheim and Schafer [2009] and Sec. 3.8.2).

In a transform with dependent coefficients, the coefficients a_1, b_0 and b_1 shows in Eq. (3.29) become functions of a set of chosen variables ν, possibly including the sampling period, so that we make the substitutions

$$\begin{cases} a_1 \longmapsto a_1(\nu, T_s), \\ b_0 \longmapsto b_0(\nu, T_s), \\ b_1 \longmapsto b_1(\nu, T_s). \end{cases} \tag{3.70}$$

Stability is a general concept agnostic to the sampling period of the discrete-time system. As we did for fixed coefficients, the only requirements for typical definitions of stability are conditions on the mapping between continuous-time poles and discrete-time ones. As a result, stability does not preclude using methods such as the parametric bilinear transform to design alternative numerical discretization methods. When looking at a method based on a Möbius transformation $(\gamma_1, \gamma_2, \gamma_3, \gamma_4)$, the definition of the stability region and the conditions for A-stability remain identical to the ones outlined in the section for fixed-coefficient methods.

The more complex question for dependent-coefficient methods is to work out the concept of consistency and order of accuracy. Indeed, as pointed out in Sec. 0.4.2, these concepts are *asymptotic* concepts, defined by observing the behavior of a method as the sampling period vanishes. In this case, the conditions we outlined in the paragraph on fixed coefficients have to be reformulated for dependent coefficients as follows:

- a consistent method verifies the limit condition

$$\forall \nu, \quad a_1(\nu, T_s) \xrightarrow[T_s \to 0]{} -1, \tag{3.71}$$

  i.e., $a_1(\nu, T_s)$ converges to -1 as the sampling period T_s vanishes.

- a 1st-order accurate method, in addition to the condition above, verifies the limit condition

$$\forall \nu, \quad b_0(\nu, T_s) + b_1(\nu, T_s) \xrightarrow[T_s \to 0]{} 1, \tag{3.72}$$

and,

- a 2nd-order accurate method, in addition to the conditions above, verifies the limit condition

$$\forall \nu, \quad b_0(\nu, T_s) \xrightarrow[T_s \to 0]{} 1/2. \tag{3.73}$$

As a case study, we can show how a typical parametrization of the parametric bilinear transform is actually a 2nd-order method, similarly as the standard bilinear transform. A typical approach is to set the coefficient η so that, for a given frequency $\Omega_0 < \pi/T_s$, we verify the mapping

$$j\Omega_0 \mapsto \exp(j\omega_0 T_s). \tag{3.74}$$

We know that this is achieved by setting η as dependent coefficient of Ω_0 and T_s as

$$\eta(\Omega_0, T_s) = \Omega_0 \cot\left(\frac{\Omega_0 T_s}{2}\right), \tag{3.75}$$

so that

$$\begin{cases} a_1(\Omega_0, T_s) = -1, \\ b_0(\Omega_0, T_s) = \frac{2}{\Omega_0 T_s} \tan\left(\frac{\Omega_0 T_s}{2}\right), \\ b_1(\Omega_0, T_s) = \frac{2}{\Omega_0 T_s} \tan\left(\frac{\Omega_0 T_s}{2}\right). \end{cases} \tag{3.76}$$

Since $\tan(\Omega_0 T_s/2) = \Omega_0 T_s/2 + \mathcal{O}(T_s^2)$, we get

$$\begin{cases} a_1(\Omega_0, T_s) = -1 + \mathcal{O}(T_s), \\ b_0(\Omega_0, T_s) = 1/2 + \mathcal{O}(T_s), \\ b_1(\Omega_0, T_s) = 1/2 + \mathcal{O}(T_s), \end{cases} \tag{3.77}$$

so that we have the limit properties

$$\begin{cases} a_1(\Omega_0, T_s) \xrightarrow[T_s \to 0]{} -1, \\ b_0(\Omega_0, T_s) \xrightarrow[T_s \to 0]{} 1/2, \\ b_1(\Omega_0, T_s) \xrightarrow[T_s \to 0]{} 1/2, \end{cases} \tag{3.78}$$

which means all parametric bilinear transforms parametrized using this approach correspond to consistent 2nd-order numerical methods, same as the standard bilinear transform (i.e., the trapezoidal method).

3.7.3 Classes of mapping parametrizations

Starting from the Möbius mappings corresponding to the typical numerical methods, i.e., the trapezoidal method and the Euler methods, we derive several mapping parametrizations constraining the choice for the mapping parameters $(\gamma_1, \gamma_2, \gamma_3, \gamma_4)$.

Parametric bilinear transform

We already mentioned the mapping we denote parametric bilinear transform in Sec. 3.7.2. It corresponds to mappings such that

$$s \mapsto \frac{2}{T}\frac{1-z^{-1}}{1+z^{-1}} \tag{3.79}$$

or equivalently

$$\begin{aligned} \gamma_1 &= 1 & \gamma_2 &= -1 \\ \gamma_3 &= T/2 & \gamma_4 &= T/2 \end{aligned} \tag{3.80}$$

which we generally only consider for positive values $T \geq 0$, such that the mapping for the standard bilinear transform corresponds to the particular case $T = T_s$. As fixed-parameter transform, this mapping is unconditionally A-stable, consistent and 0th-order (except for $T = T_s$ where it is 2nd-order). All the transforms maps the imaginary axis on the s-plane to the unit circle on the z-plane. All the transforms maps $s = 0$ to $z = 1$ (i.e., DC to DC) and $s = \infty$ to $z = -1$.

α-transform

Another mapping that can be found in prior art is variations of what we denote the α-transform. It corresponds to mappings such that

$$s \mapsto \frac{1+\alpha}{T_s}\frac{1-z^{-1}}{1+\alpha z^{-1}} \tag{3.81}$$

or equivalently

$$\begin{aligned} \gamma_1 &= 1 & \gamma_2 &= -1 \\ \gamma_3 &= \frac{T_s}{1+\alpha} & \gamma_4 &= \frac{\alpha T_s}{1+\alpha} \end{aligned} \tag{3.82}$$

which we generally only consider for positive values $\alpha \geq 0$. The mapping for the standard bilinear transform corresponds to $\alpha = 1$, the mapping for the forward Euler method corresponds to $\alpha \to +\infty$, and the mapping for the backward Euler method corresponds to $\alpha = 0$. As fixed-parameter transform, this mapping is unconditionally A-stable for $\alpha \leq 1$, consistent and 1st-order (except for $\alpha = 1$ where it is 2nd-order). The mapping maps $s = 0$ to $z = 1$ (i.e., DC to DC) and $s = \infty$ to $z = -\alpha$.

This mapping should not be confused with the equivalent mapping of the form

$$s \mapsto \frac{1}{T_s} \frac{1 - z^{-1}}{(1 - \alpha') + \alpha' z^{-1}} \tag{3.83}$$

for which the standard bilinear transform corresponds to $\alpha' = 1/2$, the mapping for the forward Euler method corresponds to $\alpha' = 1$, and the mapping for the backward Euler method corresponds to $\alpha' = 0$ (e.g., in Gao et al. [2003]).

Parametric α-transform

A more general mapping is what we denote the parametric α-transform, which combines the degrees of freedom from the two previous cases, with

$$s \mapsto \frac{1 + \alpha}{T} \frac{1 - z^{-1}}{1 + \alpha z^{-1}} \tag{3.84}$$

or equivalently

$$\begin{aligned} \gamma_1 &= 1 & \gamma_2 &= -1 \\ \gamma_3 &= \frac{T}{1 + \alpha} & \gamma_4 &= \frac{\alpha T}{1 + \alpha} \end{aligned} \tag{3.85}$$

which we generally only consider for positive values $T \geq 0$ and $\alpha \geq 0$. The mapping for the standard bilinear transform corresponds to $\alpha = 1$ and $T = T_s$, the mapping for the forward Euler method corresponds to $\alpha \to +\infty$ and $T = T_s$, and the mapping for the backward Euler method corresponds to $\alpha = 0$ and $T = T_s$. As fixed-parameter transform, this mapping is unconditionally A-stable for $\alpha \leq 1$, consistent and 0th-order (except for $T = T_s$ where it is 1st-order, and $\alpha = 1$ and $T = T_s$ where it is 2nd-order). The mapping maps $s = 0$ to $z = 1$ (i.e., DC to DC) and $s = \infty$ to $z = -\alpha$.

$\alpha\beta$-transform

Another general mapping with two degrees of freedom is denoted as the $\alpha\beta$-transform and corresponds to

$$s \mapsto \frac{1 + \alpha}{T_s} \frac{1 - \beta z^{-1}}{1 + \alpha z^{-1}} \tag{3.86}$$

or equivalently

$$\begin{aligned} \gamma_1 &= 1 & \gamma_2 &= -\beta \\ \gamma_3 &= \frac{T_s}{1 + \alpha} & \gamma_4 &= \frac{\alpha T_s}{1 + \alpha} \end{aligned} \tag{3.87}$$

which we generally only consider for positive values $\alpha \geq 0$ and $\beta \geq 0$. The mapping for the standard bilinear transform corresponds to $\alpha = 1$ and $\beta = 1$, the mapping for the forward Euler method corresponds to $\alpha \to +\infty$ and $\beta = 1$, and the mapping for the backward Euler method corresponds to $\alpha = 0$ and $\beta = 1$. As fixed-parameter transform, this mapping is unconditionally A-stable for

$\alpha \leq 1$ and $\beta < 1$. It is consistent and 1st-order for $\beta = 1$ (except $\alpha = 1$ and $\beta = 1$ where it is 2nd-order). The mapping maps $s = 0$ to $z = \beta$ and $s = \infty$ to $z = -\alpha$.

Iso-contours for dependent-coefficient mapping

As we will discuss in the next section, one interesting aspect of the dependent-coefficient mappings described above is the possibility of address the limitations of fixed-variable mappings when we have some a priori knowledge of the system, allowing for the design of models which are better behaved, especially in a context where the target sampling rate of the model is set. One particular topic of interest for us will be discussing how to address the issue of pole damping mismatch. As a result, it is interesting to look at the behavior of the mapping of the pole iso-contours for frequency from the s-plane to the z-plane.

In Figs. 3.5, 3.6 and 3.7, we see the differences between iso-contours for the standard bilinear transform, one example of the α-transform and the backward Euler method. In particular, we see that as σ becomes smaller and smaller (i.e., the continuous-time pole damping gets higher and higher), we reach a point where the bilinear transform maps poles to poles with smaller and smaller damping (and higher and higher frequency), thus creating a large discrepancy between the expected behavior of the continuous-time system and its model. Meanwhile, we observe the well-known frequency warping effect as frequencies get mapped to lower frequencies, with the difference increasing with frequency. The backward Euler method, thanks to its L-stability property, has no such issue, but its mapping presents other kinds of distortion. As we can see, frequencies are generally mapped to lower frequency (especially at higher frequencies), poles with low damping are over-damped (especially at higher frequencies), while poles with high damping are under-damped.

The α-transform provides some kind of compromise in between the two methods in terms of distortion. In particular, while it still presents the same non-monotonous behavior for highly damped poles as the standard bilinear transform, the level of damping σ needed to observe that problem is much higher.

Then, in Figs. 3.8 and 3.9, we can observe the influence of the parameter T in the parametric bilinear transform and the parametric α-transform. We see that, as is known when we attempt to perform frequency matching with the parametric bilinear transform (see Sec. 3.8.2), higher values of T shift the discrete-time ω higher at equal Ω for all mappings. However, another interesting but less known side-effect is how the damping trends are also affected so that, for example, increasing T for the parametric bilinear transform shifts higher the damping of the discrete-time poles, especially at lower frequencies.

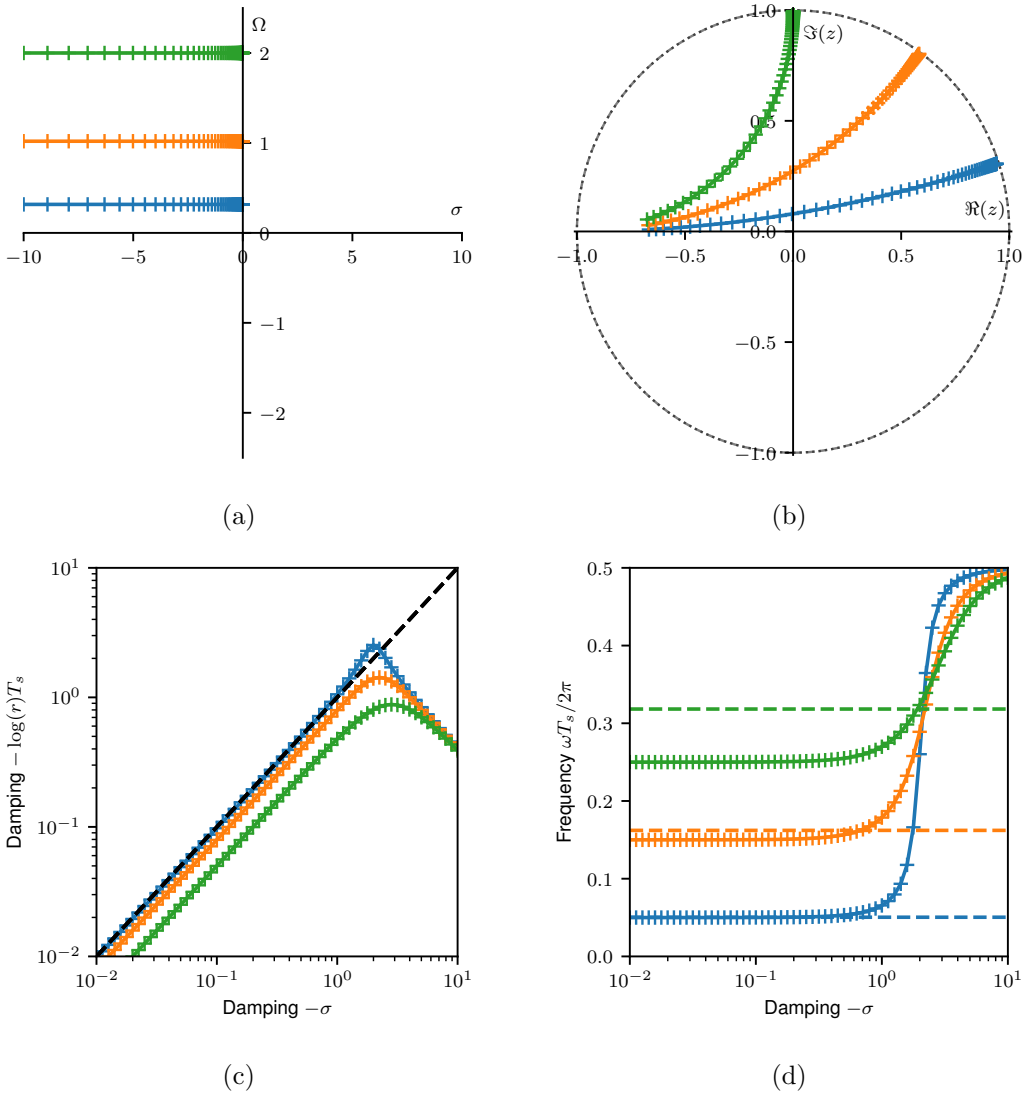


Figure 3.5: Mapping of pole iso-contours for frequency in the s-plane (in (a)) into the z plane (in (b)) for the standard bilinear transform mapping. The resulting frequency $\omega T_s/2\pi$ and damping $\log(r)T_s$ of the discrete-time poles as a function of the continuous-time damping σ is shown in (c) and (d) respectively. The dashed traces show the "ideal" mapping (i.e., equal damping and frequency between continuous-time and discrete-time poles).

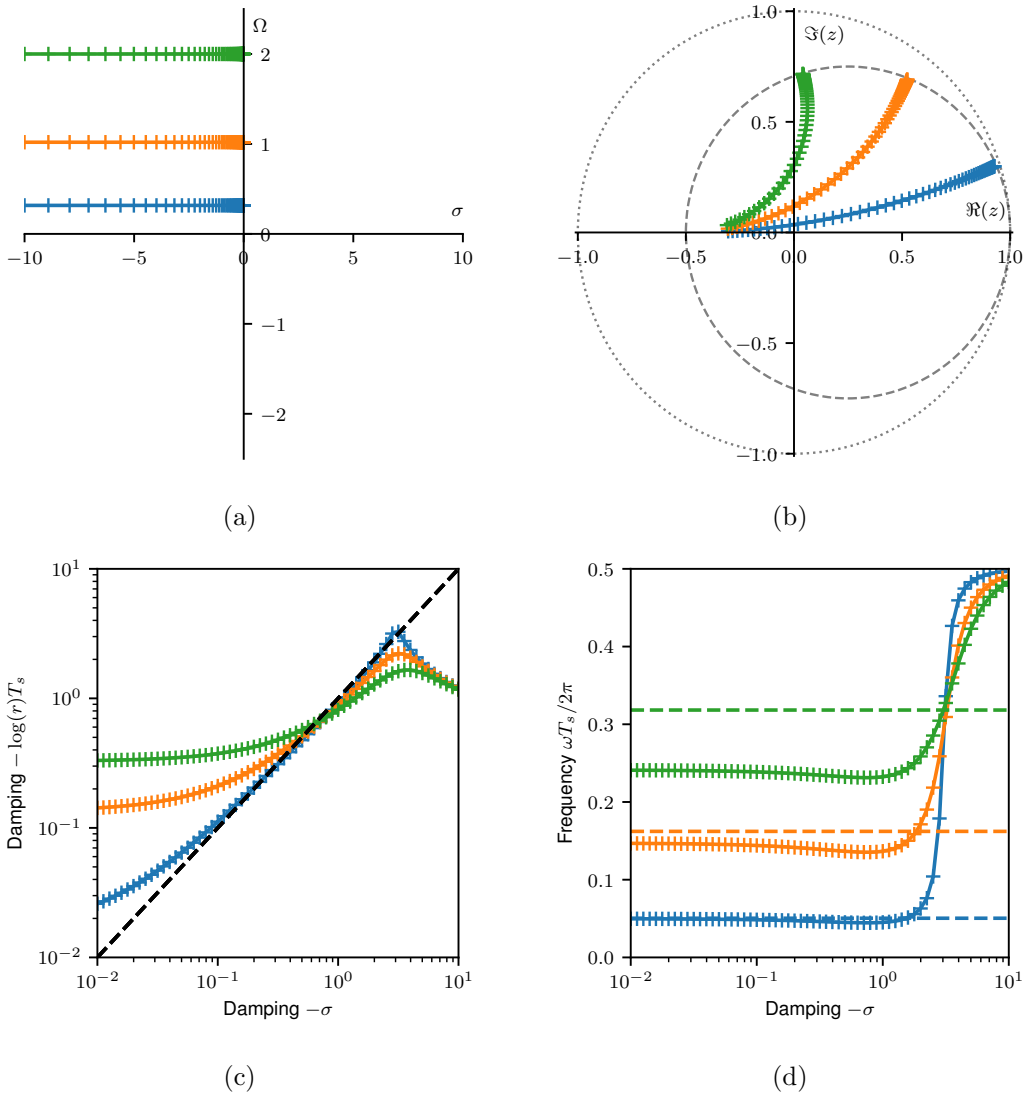


Figure 3.6: Mapping of pole iso-contours for frequency in the s-plane (in (a)) into the z plane (in (b)) for the α-transform mapping set with $\alpha = 0.5$. The resulting frequency $\omega T_s/2\pi$ and damping $\log(r)T_s$ of the discrete-time poles as a function of the continuous-time damping σ is shown in (c) and (d) respectively. The dashed traces show the "ideal" mapping (i.e., equal damping and frequency between continuous-time and discrete-time poles).

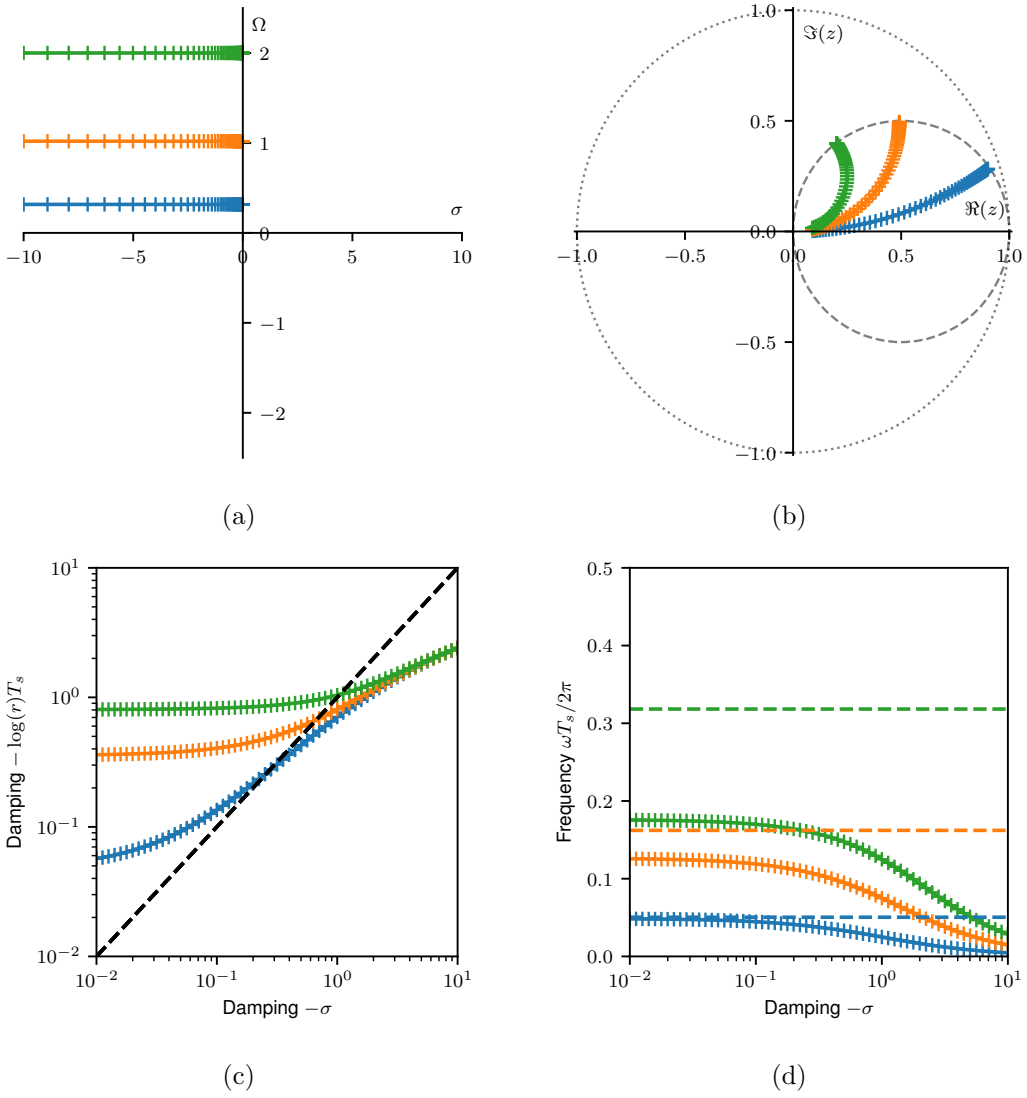


Figure 3.7: Mapping of pole iso-contours for frequency in the s-plane (in (a)) into the z plane (in (b)) for the backward Euler mapping. The resulting frequency $\omega T_s/2\pi$ and damping $\log(r)T_s$ of the discrete-time poles as a function of the continuous-time damping σ is shown in (c) and (d) respectively. The dashed traces show the "ideal" mapping (i.e., equal damping and frequency between continuous-time and discrete-time poles).

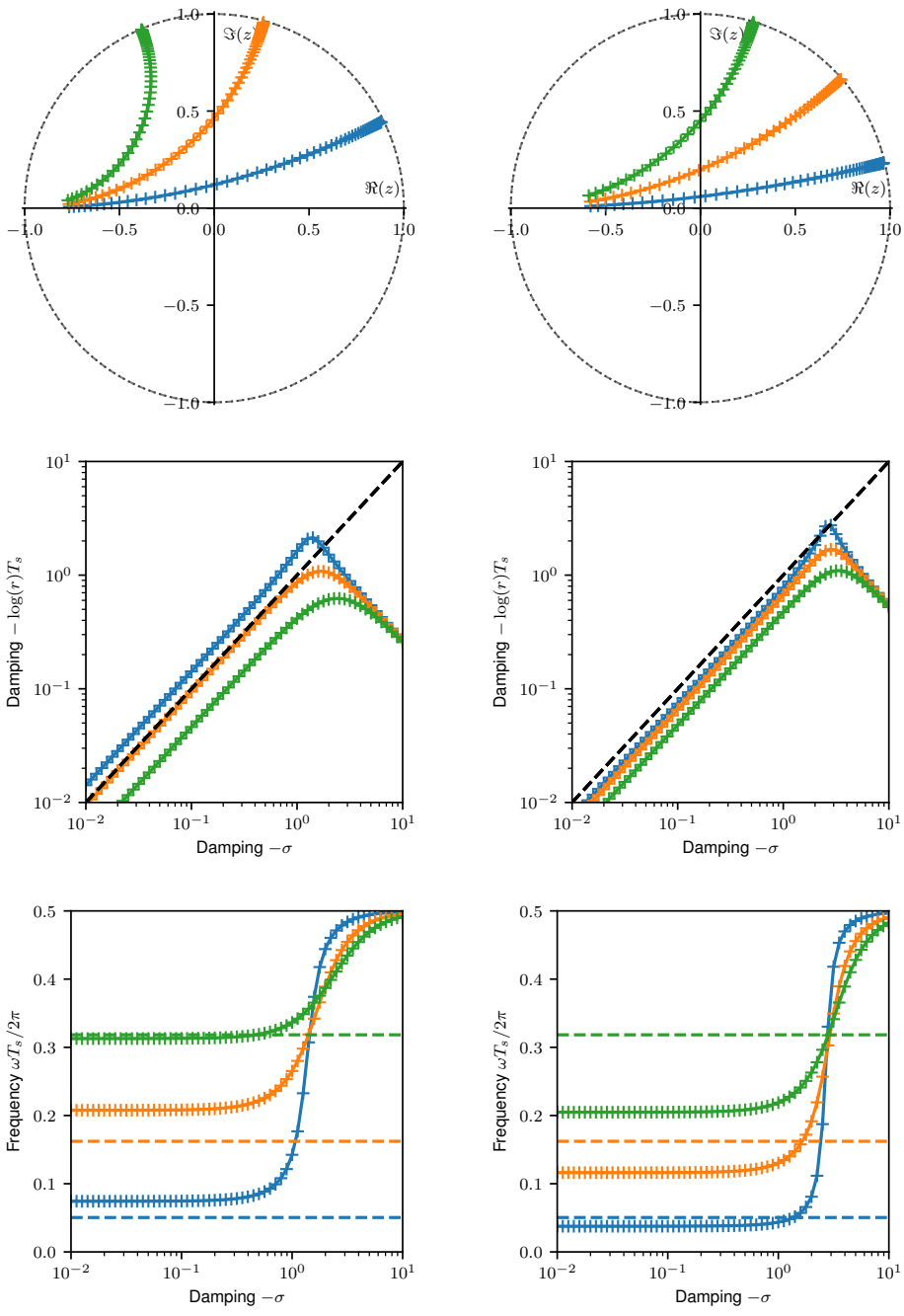


Figure 3.8: Comparison of the mapping of pole iso-contours for frequency for two parametric bilinear transform mappings: $T = 1.5T_s$ (left) and $T = 0.75T_s$ (right). The resulting pole locations, frequency $\omega T_s/2\pi$ and damping $\log(r)T_s$ of the discrete-time poles are shown in the top, middle and bottom row respectively. The dashed traces show the "ideal" mapping (i.e., equal damping and frequency between continuous-time and discrete-time poles).

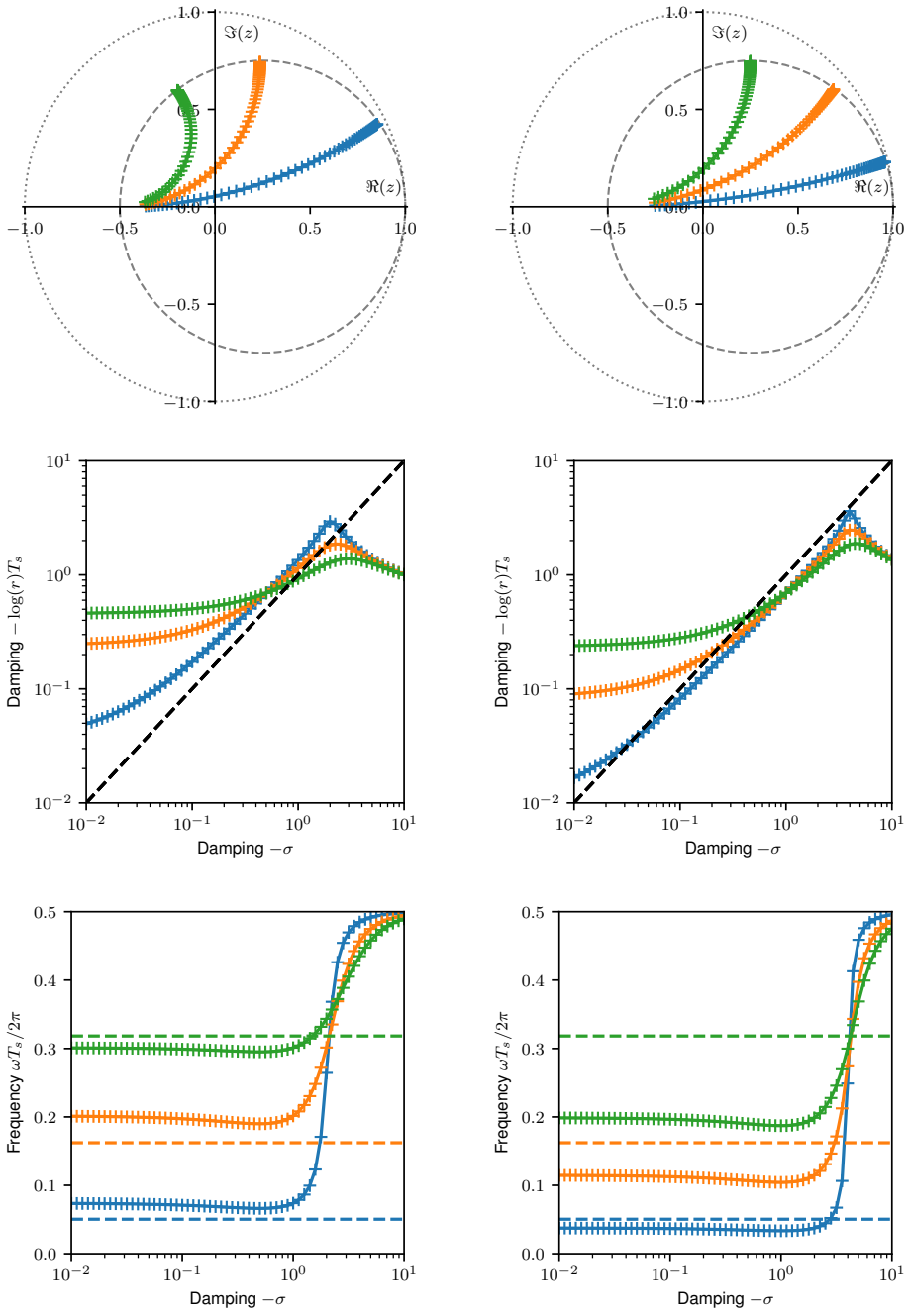


Figure 3.9: Comparison of the mapping of pole iso-contours for frequency for two parametric α-transform transform mappings (with $\alpha = 0.5$): $T = 1.5T_s$ (left) and $T = 0.75T_s$ (right). The resulting pole locations, frequency $\omega T_s/2\pi$ and damping $\log(r)T_s$ of the discrete-time poles are shown in the top, middle and bottom row respectively. The dashed traces show the "ideal" mapping (i.e., equal damping and frequency between continuous-time and discrete-time poles).

3.8 Dependent-coefficient mapping design

3.8.1 Typical fixed-coefficient mappings

Typical fixed mappings rely on the numerical analysis concepts outlined in Sec. 3.7.2, leading to the usage of well-known transforms. A straightforward approach to select Möbius transforms corresponding to consistent, 1st-order (or higher) accurate methods. These properties are generally desirable as they imply that the behavior of the numerical simulation should improve as the sampling period gets smaller. The class of mappings verifying these properties corresponds to the α-transform. In that class, the most popular mappings correspond to the Euler methods (α is 0 or $+\infty$) and the standard bilinear transform (α is 1).

The forward Euler method, while not unconditionally stable, is popular due to the fact that it presents the advantage of leading to simpler and explicit discrete-time equations without further processing needed (e.g., nonlinear function or matrix inversions) due to the fact that $b_0 = 0$. On the other hand, the standard bilinear transform is the only 2nd-order accurate approach, and is unconditionally A-stable. In the case of linear time-invariant filters, it present the advantage of reproducing qualitatively the frequency response of the original continuous-time frequency response, minus the well-known frequency warping distortion [Smith III 2007b, Oppenheim and Schafer 2009]. However, it leads to an implicit system of equations which is generally more computationally expensive to solve, though active research is ongoing into limiting this computational overhead by developing approaches to systematically re-arrange these equations towards less computationally expensive formulations. Finally, the backward Euler approach presents the same advantage as the bilinear transform in terms of being unconditionally A-stable. It is only 1st-order accurate. Additionally, it generally preserves much less of the qualitative features of the system, especially for linear time-invariant systems. However, it actually is the only α-transform verifying the stronger L-stability condition [Hairer and Wanner 1996], so that it won't present the oscillatory behavior displayed when using other α-transform with stiff systems. Another benefit is that it leads to somewhat simpler, albeit implicit, system equations due to the fact that $b_1 = 0$.

Other α values are much less common because the free parameter generally presents limited advantages in the context of fixed-variable mappings. They lead to discrete-time systems with a computational complexity similar to the ones obtained using the standard bilinear transform without some of its advantages. One rare example of fixed-variable design for the α-transform outside of the ones mentioned above is the Al-Alaoui transform [Al Alaoui 1993, 2006] where the choice of $\alpha = 1/7$ was presented as a better choice to design a minimum-phase discrete-time approximate to the ideal integrator $\dot{x}(t) = u(t)$ than either the trapezoidal integration (i.e., the standard bilinear transform) or rectangular integration (i.e., the backward Euler method).

Another alternative proposed in Liniger [1969] is to focus on the mapping of purely decaying continuous-time poles $s = \sigma$ for $\sigma \leq 0$, and optimize α to minimize the maximum error between the

position of the mapped pole, i.e.,

$$z = \frac{1 + \alpha + \sigma T_s \alpha}{1 + \alpha - \sigma T_s} \tag{3.88}$$

and the discrete-time pole with damping σ, i.e., $e^{\sigma T_s}$. Hence we pick α such that

$$\alpha = \underset{\alpha' \geq 0}{\operatorname{argmin}} \max_{\sigma \leq 0} \left| \frac{1 + \alpha' + \sigma T_s \alpha'}{1 + \alpha' - \sigma T_s} - e^{\sigma T_s} \right|. \tag{3.89}$$

It is straightforward to see that the optimal α according to that criterion does not depend on the sampling period T_s, since

$$\max_{\sigma \leq 0} \left| \frac{1 + \alpha + \sigma T_s \alpha}{1 + \alpha - \sigma T_s} - e^{\sigma T_s} \right| = \max_{\sigma' \leq 0} \left| \frac{1 + \alpha + \sigma' \alpha}{1 + \alpha - \sigma'} - e^{\sigma'} \right|. \tag{3.90}$$

According to that criterion, the optimal α is about 0.138 [Liniger 1969].

3.8.2 Dependent-coefficient design

In order to alleviate the limitations of the fixed-variable mappings outlined above in Sec. 3.8.1, more modern approaches have attempted to research design rules for dependent-coefficient mappings in order to achieve improved simulation results according to an chosen criterion for optimality.

Frequency warping compensation using the parametric bilinear transform

A well-known issue with the standard bilinear transform is the frequency warping [Smith III 2007b, Oppenheim and Schafer 2009] we already mentioned several times above. As a reminder, this specific distortion corresponds to the following: Since the mapping corresponding to the standard bilinear transform maps the imaginary axis in the s-plane to the unit circle in the z-plane, it follows that for linear time-invariant systems, the frequency response $H_d(e^{j\omega})$ of the discrete-time model obtained using the standard bilinear transform at sampling period T_s can be simply expressed as a function of the frequency response $H(j\Omega)$ of the original continuous-time system as

$$H_d(e^{j\omega T_s}) = H(\Omega_{\mathrm{BT}}(\omega, T_s)), \quad \forall \omega \in [-\pi/T_s, \pi/T_s] \tag{3.91}$$

with the warping function

$$\Omega_{\mathrm{BT}}(\omega, T_s) = \frac{2}{T_s} \tan\left(\frac{\omega T_s}{2}\right). \tag{3.92}$$

The result is the well-known fact that the frequency is distorted with frequency Ω in the continuous-time frequency response mapped consistently to a strictly lower frequency $\omega < \Omega$ in the discrete-time frequency response (except for the DC frequency $\Omega = 0$), with the distortion increasing as the frequency Ω increases as shown in Fig. 3.10. One well-known problematic system in this case is the fact that models of resonant continuous-time systems also present resonances, but

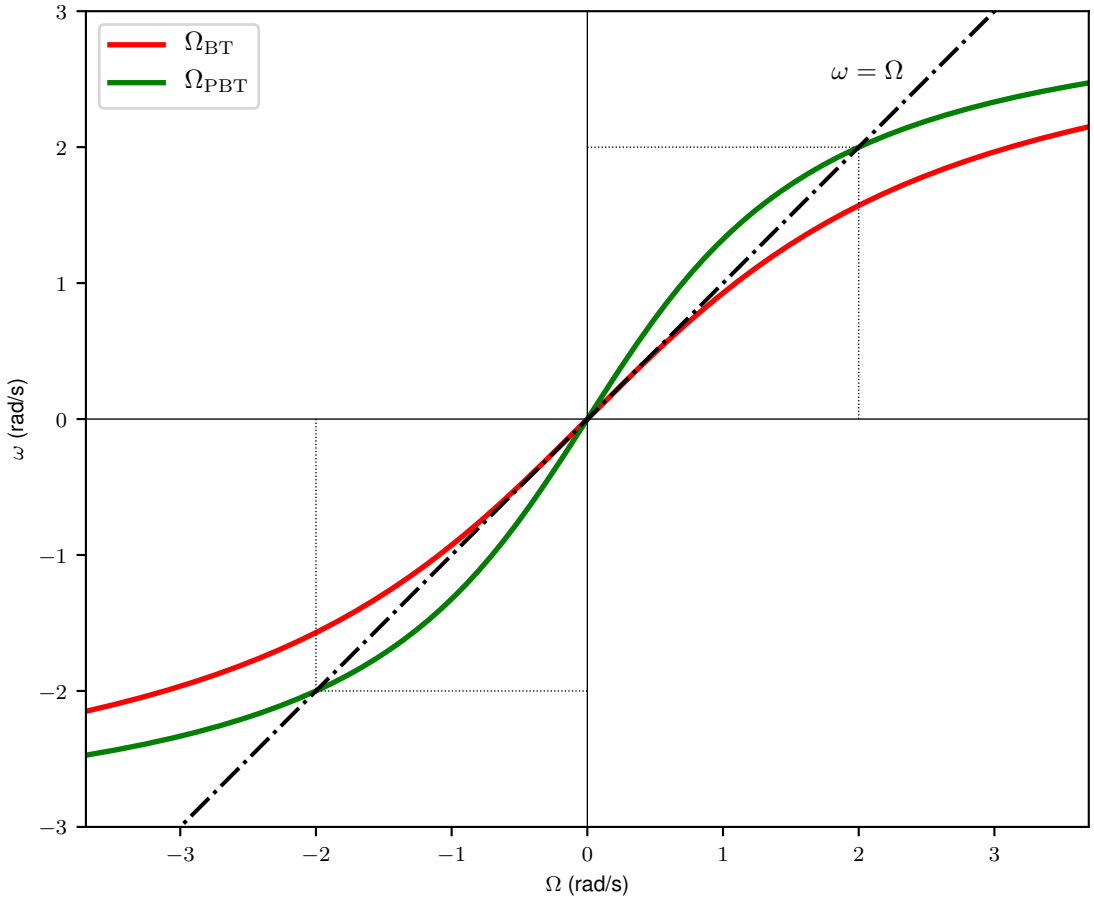


Figure 3.10: Frequency mapping $\Omega(\omega, T_s)$ (for $T_s = 1$) such that $H_d(e^{j\omega T_s}) = H(j\Omega)$ for a standard bilinear transform mapping (red), and for the parametric bilinear transform matching the frequency response at frequency $\omega_0 = 2$ (green).

these resonances have lowered frequencies. Another more subtle issue with such systems is that the Q-factor of these resonances is increased [Stilson 2006].

One way to mitigate the issue is to use a parametric bilinear transform mapping instead (see Sec. 3.7.3). As already mentioned in another form in Sec. 3.7.2, the parametric bilinear transform parameter T can be set such that one frequency ω_0 in the discrete-time model response and the continuous-time system response match (i.e., $\omega_0 = \Omega(\omega_0, T_s)$). This is achieved through setting

$$T = \frac{2}{\omega} \tan\left(\frac{\omega T_s}{2}\right) \tag{3.93}$$

which results in the alternative warping function

$$\Omega_{\mathrm{PBT}}(\omega, T_s) = \frac{2}{T} \tan\left(\frac{\omega T_s}{2}\right) \tag{3.94}$$

since the parametric bilinear transform mappings all still map the imaginary axis to the unit circle. However, in this case, the continuous-time frequency response at frequencies below the matched frequency ω_0 is mapped to higher frequencies for the discrete-time model, while frequencies above the matched frequency are still mapped to lower frequencies. Hence, if a resonant system has a resonance at the matched frequency, its discrete model will present a resonance at that same frequency. However, one limitation of that approach is that the reminding distortion of frequencies still creates artifacts. In the case of resonant systems, this distortion corresponds to a further increase in the Q factor of resonances at the matched frequency and above [Stilson 2006, Germain and Werner 2017a]. We will further illustrate this point in Ch. 4.

As stated in Sec. 3.7.2, the dependent-coefficient mapping converges asymptotically to the standard bilinear transform as the sampling period vanishes, so that this approach is unconditionally A-stable, consistent and 2nd-order accurate.

Damping warping compensation using exponential fitting in the α-transform [Liniger 1969]

As mentioned in Sec. 3.6.3, for mappings with real coefficients, the real axis in the s-plane is always mapped to the real axis in the z-plane. More importantly, for any α-transform mapping, the semi-straight line between $(0, 0)$ and $(-\infty, 0)$ (which corresponds to the purely decaying real poles) maps to the segment between $(1, 0)$ (mapped from $(0, 0)$) and $(-\alpha, 0)$ (mapped to $(-\infty, 0)$). In the z-plane, the purely decaying real poles correspond to the segment between $(1, 0)$ and $(0, 0)$.

Then, in general, there is a mismatch since some of the purely decaying poles in the s-plane can be mapped to either oscillatory decaying poles (located between $(0, 0)$ and $(-1, 0)$) or oscillatory growing poles (located between $(-1, 0)$ and $(-\infty, 0)$). Furthermore, for the part of the purely decaying real poles that map to purely decaying real poles, the damping of the poles will generally

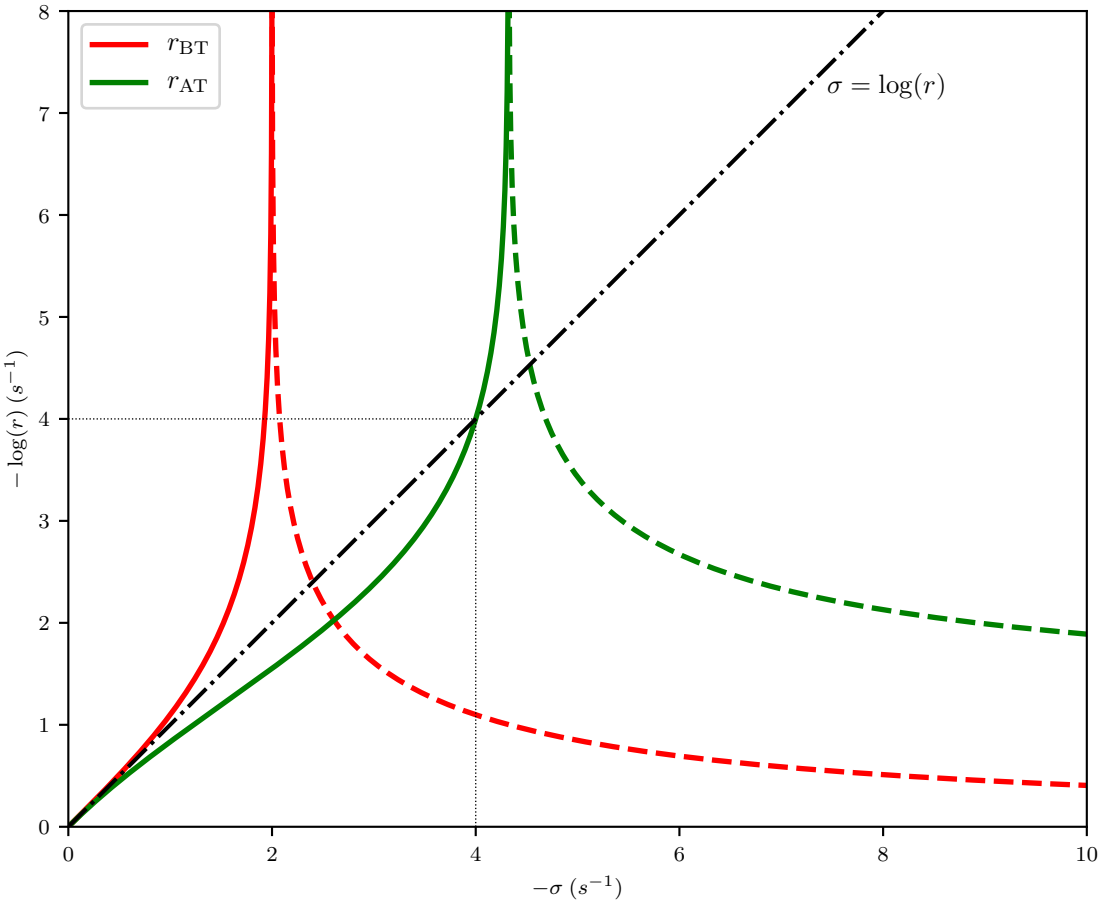


Figure 3.11: Damping mapping $r(\sigma, T_s)$ (for $T_s = 1$) for a standard bilinear transform mapping (red), and for the α-transform matching the damping $\sigma_0 = -4$, i.e., mapping $(-4, 0)$ to $(e^{-4}, 0)$ (green). Plain lines corresponds to values of σ for which purely decaying poles are mapped to purely decaying poles. Dashed lines corresponds to values of σ for which purely decaying poles are mapped to oscillatory decaying poles.

not match (except for zero damping $(0, 0)$ which maps to zero damping $(1, 0)$).

In Liniger [1969] and Liniger and Willoughby [1970], it was shown that we can set α to ensure that one purely decaying pole in the s-plane is mapped to the purely decaying pole in the z-plane of identical damping σ_0 for the sampling frequency T_s, i.e.,

$$\sigma_0 \mapsto e^{\sigma_0 T_s}. \tag{3.95}$$

This process is sometimes referred to as *exponential fitting*.

This can be achieved by setting the dependent coefficient $\alpha(\sigma_0, T_s)$ as

$$\alpha(\sigma_0, T_s) = -\frac{\left(e^{\sigma_0 T_s} - 1\right) - \sigma_0 T_s e^{\sigma_0 T_s}}{(e^{\sigma_0 T_s} - 1) - \sigma_0 T_s}. \tag{3.96}$$

We can see the distortion of the mapping for purely damped continuous-time poles, as well as an example of a compensated α-transform mapping in Fig. 3.11. We can see that, aside from the matched pole, the mapping results in regions where the discrete-time poles are under-damped, and others where the discrete-time poles are over-damped compared to the continuous-time poles.

In particular, we can verify that for all purely decaying poles (i.e., for all negative σ_0), we verify:

- $\alpha(\sigma_0, T_s) \geq 0$ due to the fact that

$$1 - e^{-\sigma_0 T_s} < \sigma_0 T_s, \quad \forall \sigma_0 \leq 0, \tag{3.97}$$

and,

- $\alpha(\sigma_0, T_s) \leq 1$ due to the fact that

$$\frac{\sigma_0 T_s}{2} < \tan\left(\frac{\sigma_0 T_s}{2}\right), \quad \forall \sigma_0 \leq 0. \tag{3.98}$$

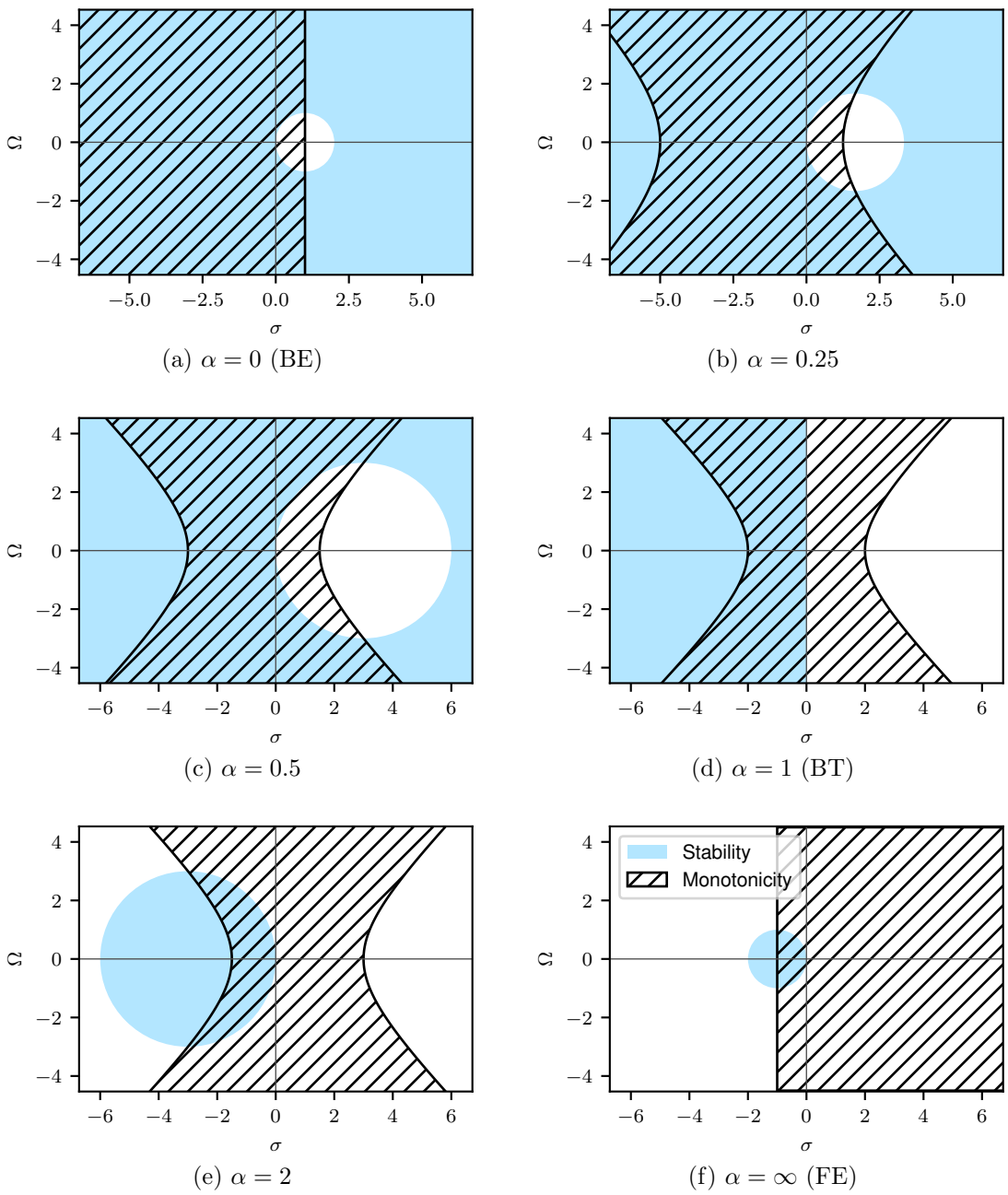
As a result, this criterion is guaranteed to lead to an unconditionally A-stable mapping. Additionally, we can verify that

$$\alpha(\sigma_0, T_s) = 1 + \mathcal{O}(\sigma_0 T_s) \tag{3.99}$$

such that the mapping will converge to a standard bilinear transform mapping as the sampling period vanishes, and by extension is a consistent 2nd-order accurate dependent-coefficient mapping.

Damping monotonicity conservation

In Germain and Werner [2015], we proposed a novel criterion to design a Möbius transform mapping, by finding the condition to enforce damping monotonicity in the mapping, meaning that, for a region of interest in the s-plane, increasing the damping of the continuous-time poles of a system results



(a) $\alpha = 0$ (BE) (b) $\alpha = 0.25$

(c) $\alpha = 0.5$ (d) $\alpha = 1$ (BT)

(e) $\alpha = 2$ (f) $\alpha = \infty$ (FE)

Figure 3.12: Stability (blue) and monotonicity (hatched) regions in the s-plane for $T_s = 1$ and different choices of parameter α

in an increased damping of the discrete-time model of that system. This criterion was designed in response to the well-known issue encountered when using the standard bilinear transform to model stiff systems (i.e., systems with highly damped poles in the context of modeling at a fixed sampling rate). As we saw in Fig. 3.5, continuous-poles with high damping in the s-plane get mapped to discrete-time poles with low damping in the z-plane. Combined with the observation that these poles also get mapped to pole locations with increasing frequency, discrete-time models then often exhibits spurious behavior with unwanted overshoots and/or oscillations (see examples in Germain and Werner [2015] and Sec. 3.10). Then, imposing a damping monotonicity criterion is meant to ensure that the model follows the general dynamical behavior of its corresponding continuous-time system, and to obtain in a much more accurate simulation.

Mathematically, the damping monotonicity condition is expressed as

$$\frac{\partial r}{\partial \sigma} \geq 0 \quad \left(\text{or equivalently } \frac{\partial \sigma}{\partial r} \geq 0\right). \tag{3.100}$$

On the contrary to the two previous approaches, which rely on the identification of one pole location of interest, this criterion attempted to focus on a more qualitative aspect of the mapping over a broad region. In particular, we notice that for many mappings, we have a definite region in the s-plane where increasing the damping of a pole in the s-plane gets mapped to poles in the z-plane with increasing damping as well (see Sec. 3.7.3 and the corresponding figures relative to pole mapping). We here set the mathematical framework to identify these regions and design the appropriate transform, in particular in the case of an α-transform mapping.

This approach is inspired by the concept of L-stability we mentioned earlier. However, the choice of actually L-stable mappings is rather restricted. The condition for L-stability requires the mapping to verify the limit condition such that

$$\frac{b_1\sigma - a_1}{1 - b_0\sigma} \quad \xrightarrow[\sigma\to-\infty]{} \quad 0. \tag{3.101}$$

Since that limit is equal to $-b_1/b_0$, this condition is equivalent to $b_1 = 0$. Unconditionally L-stable mappings additionally requires that $|a_1| \leq 1$. In particular, the only fixed-variable mapping that is unconditionally L-stable, consistent and 1st-order accurate is the backward Euler method. Our approach attempts to retain some of the benefits of L-stability while also retaining some degrees of freedom in the mapping design as the backward Euler method introduces significant simulation distortion otherwise.

In the case of the α-transform, the iso-contours equations in Eq. (3.47) show that the damping r of the discrete-time poles can be expressed as a function of the damping σ and frequency Ω of the continuous-time poles as

$$r^2 = \frac{(1 + \alpha + \alpha T_s\sigma)^2 + (\alpha T_s\Omega)^2}{(1 + \alpha - T_s\sigma)^2 + (T_s\Omega)^2}. \tag{3.102}$$

As a result, we get the relationship between variations of the dampings σ and r by looking at the derivative

$$\frac{\partial r}{\partial \sigma} = \frac{1}{2r}\frac{\partial r^2}{\partial \sigma} = 2 \cdot \frac{(\alpha+1)^2 T_s}{r} \cdot \frac{\alpha\Omega^2 T_s^2 + (\alpha+1)^2 + (\alpha^2-1)T_s\sigma - \alpha T_s^2\sigma^2}{\left((1+\alpha-T_s\sigma)^2 + (T_s\Omega)^2\right)^2}. \tag{3.103}$$

Due to the shape of the iso-contours, we know that the derivative will have two roots, which matches the fact that its numerator is a quadratic polynomial in σ. The roots σ_+ and σ_- are given as

$$\sigma_\pm = \frac{\left(\alpha^2-1\right) \pm \sqrt{(\alpha+1)^4 + (2\alpha T_s\Omega)^2}}{2\alpha T_s}. \tag{3.104}$$

From this, we can verify that, for all α:

- $\sigma_+ > 0$,
- $\sigma_- < 0$, and,
- for all $\sigma \in]\sigma_-, \sigma_+[$, we have $\frac{\partial r}{\partial \sigma} > 0$, meaning that the two damping variables have a monotonic relationship, where both increase or decrease together.

As a result, we can identify the region in the s-plane where this monotonic relationship is verified as the region for which the continuous-time poles $p = (\sigma, \Omega)$ verify

$$\left(\sigma - \frac{\alpha^2-1}{2\alpha T_s}\right)^2 - \Omega^2 \leq \left(\frac{(\alpha+1)^2}{2\alpha T_s}\right)^2. \tag{3.105}$$

We recognize here the equation describing the interior of a rectangular hyperbola of semi major axis $\frac{(\alpha+1)^2}{2\alpha T_s}$ and center $\left(\frac{\alpha^2-1}{2\alpha T_s}, 0\right)$. This degenerates for $\alpha = 0$ (backward Euler) where the monotonicity region corresponds to the half-plane $\sigma \leq 1/T$, and for $\alpha = +\infty$ (forward Euler) where the monotonicity region corresponds to the half-plane $\sigma \geq -1/T$.

That region can be compared to the region of A-stability for these methods, which is defined as:

- poles verifying
$$\left(\sigma - \frac{1}{T_s}\frac{1+\alpha}{1-\alpha}\right)^2 + \Omega^2 \geq \left(\frac{1}{T_s}\frac{1+\alpha}{1-\alpha}\right)^2 \tag{3.106}$$
for $\alpha \in [0, 1[$, which corresponds to the exterior of the circle of center $\left(\frac{1}{T_s}\frac{1+\alpha}{1-\alpha}, 0\right)$ and radius $\left|\frac{1}{T_s}\frac{1+\alpha}{1-\alpha}\right|$,
- poles verifying
$$\left(\sigma - \frac{1}{T_s}\frac{1+\alpha}{1-\alpha}\right)^2 + \Omega^2 \leq \left(\frac{1}{T_s}\frac{1+\alpha}{1-\alpha}\right)^2 \tag{3.107}$$

for $\alpha > 1$, which corresponds to the interior of the circle of center $\left(\frac{1}{T_s}\frac{1+\alpha}{1-\alpha}, 0\right)$ and radius $\left|\frac{1}{T_s}\frac{1+\alpha}{1-\alpha}\right|$, and,

- the poles verifying

$$\sigma \leq 0 \tag{3.108}$$

for $\alpha = 1$ where the circles degenerate into this half-plane.

These various areas can be visualized for different choices of positive $\alpha \geq 0$ in Fig. 3.12.

It is often possible to derive some basic knowledge regarding the location of the continuous-time poles in the relevant context for the system (in a similar fashion as the typical way of parametrizing the parametric bilinear transform we described above relies on the knowledge of a specific frequency of interest for that system). From there, we can assess which positive values of $\alpha \geq 0$ will ensure that no poles falls outside of the monotonic region. In the particular case where the continuous-time poles are purely decaying (i.e., real), the monotonicity condition simplifies by setting $\Omega = 0$ to

$$-\frac{1+\alpha}{\alpha T_s} \leq \sigma \leq \frac{1+\alpha}{T_s} \tag{3.109}$$

which means that if we know the system has real decaying poles of damping higher than $\sigma_{\min}$, we can verify:

- Stability with

$$\alpha \leq \frac{\sigma_{\min} T_s - 2}{\sigma_{\min} T_s + 2} \tag{3.110}$$

if $\sigma_{\min} T_s \leq -2$, without higher bound otherwise, and

- Monotonicity with

$$\alpha \leq -\frac{1}{1 + \sigma_{\min} T_s} \tag{3.111}$$

if $\sigma_{\min} T_s \leq -1$, without higher bound otherwise.

Similarly as the two previous approaches, the design of a dependent-coefficient α-transform that verifies the damping monotonicity criterion and is stable still leads to a consistent 2nd-order accurate method, as the monotonicity condition will eventually gets satisfied by the standard bilinear transform for a sampling rate sufficiently high.

3.8.3 Design for nonlinear systems

Instantaneous poles

All the discussion so far has been done in a context where we have some knowledge the pole locations of the system under study. While poles are well-defined in the context of linear system (see Sec. 3.2), there is no such straightforward definition in the case of nonlinear systems. A typical way around

that problem is to use what we denote the "instantaneous" poles of the system defined by linearizing the system around its operating point, so that the system

$$\dot{\boldsymbol{x}}(t) = \boldsymbol{f}(t, \boldsymbol{x}(t)) \tag{3.112}$$

is approximated around the operating point $\boldsymbol{x}_0$ as

$$\dot{\boldsymbol{x}}(t) \approx \boldsymbol{f}(t, \boldsymbol{x}_0) + (\boldsymbol{x}(t) - \boldsymbol{x}_0)\nabla_{\boldsymbol{x}}\boldsymbol{f}(t, \boldsymbol{x}_0) \tag{3.113}$$

where $\nabla_{\boldsymbol{x}}\boldsymbol{f}$ is the gradient of nonlinear function $\boldsymbol{f}$ with respect to the state vector $\boldsymbol{x}$. The approximate linearized system would have as poles the eigenvalues of $\nabla_{\boldsymbol{x}}\boldsymbol{f}(t, \boldsymbol{x}_0)$ which we denote as the instantaneous poles.

Examining these instantaneous poles is a useful way to verify whether or not a chosen method is stable, by verifying that the poles never venture out of the stability region (see Sec. 3.7.2) for the method. Using that metric is not a guarantee due to the fact that most systems quickly move away from any operating point and the linear approximation is only valid over small intervals of time. In some ways, this can be seen as a similar issue encountered in the design of stable time-varying linear filters [Laroche 2007]. However, it is generally recognized as one of the primary tool to gain an understanding the behavior of any continuous-time system and its models so that we will use it in the rest of our discussion as well.

Relevant trajectories

Once we have found the general expression of the gradient matrix, $\nabla_{\boldsymbol{x}}\boldsymbol{f}(t, \boldsymbol{x}_0)$, an important question remains. In general, most of the variable space for $(t, \boldsymbol{x}_0)$ is irrelevant to the system analysis, since most of it will not be visited by the system. Hence, part of our analysis must rely on some understanding of the system's expected usage in order to compute relevant values of the gradient matrix and its instantaneous poles. At the present, we are not aware of any general way of finding these values. In particular, even if an exhaustive knowledge of the possible trajectories of a system are known, there remains the following issues:

- Any model of the system generally visits a different set of trajectories due to the modeling error, such that the instantaneous poles provide a somewhat biased information on the dynamics of the resulting model, and,

- Due to the ambiguity between continuous-time and discrete-time quantities (see Ch. 1), it is generally unclear what form of the time-varying parts of the system should take to help understand the relevant dynamics of the system and its model(s). For example, in the typical case where the system's time-varying element can be described as an input signal $u(t)$, i.e.,

the system is described by the equation

$$\dot{x}(t) = \boldsymbol{g}(\boldsymbol{x}(t), u(t)), \tag{3.114}$$

the choice of a relevant input $u(t)$ (or class of inputs) can result in a widely different set of trajectories and instantaneous poles.

Below, we explain some of the strategies we use to determine relevant instantaneous pole locations, for the purpose of system analysis, and for the purpose of method design.

System equilibria

An important concept in the description of the dynamics of continuous-time systems. Indeed, many time-invariant systems $\dot{\boldsymbol{x}}(t) = \boldsymbol{f}(\boldsymbol{x}(t))$ will tend towards an equilibrium ("steady-state") point $\boldsymbol{x}_{eq}$, for which a necessary (but not sufficient) condition is that it verifies $0 = \boldsymbol{f}(\boldsymbol{x}_{eq})$. Additionally, systems also can have temporary equilibrium points. For example, if a system follows Eq. (3.114) and they are fed a constant input signal u_0 for a period of time, the system can tend towards an equilibrium point verifying the necessary (but, again, not sufficient) condition $0 = \boldsymbol{g}(\boldsymbol{x}_{eq}, u_0)$. As we will see in our case studies (Sec. 3.10), several systems of interest for the methods outlined in this chapter follow this structure, as their input signals correspond to step functions and/or rectangular pulses, that are then piecewise constant and create temporary equilibrium points.

One of the possible approaches to assess relevant instantaneous pole locations is then to use the equilibrium points $\boldsymbol{x}_0$ as operating point to derive the linearized approximate system in Eq. (3.113), so that the estimated instantaneous poles are calculated as the eigenvalues of $\nabla_{\boldsymbol{x}}\boldsymbol{g}(\boldsymbol{x}_{eq}, u_0)$. That approach was used in Germain and Werner [2015] in order to find highly damped poles (in our context, highly damped is meant in comparison to the quantity $\frac{1}{T_s}$) and design a damping-monotonicity preserving α-transform.

Pole analysis for autonomous systems

The first approach often fails to capture a lot of relevant information on the dynamics of the system since it captures only information about its behavior once it gets close to its steady-state. As seen in Germain [2017], and as we will see in the case studies (see Sec. 3.10), some system become stiff (i.e., have highly damped instantaneous poles) while their variables are far from their equilibrium, so that these transient trajectories are actually necessary to set the design of methods such as the one described in this section.

Still, if we are in the case where the system is autonomous (or even temporarily autonomous as for the case of constant-input system of the form in Eq. (3.114)), and that we have an idea of the general range of values for the variables $\boldsymbol{x}$, we can attempt an investigation of the general behavior of $\nabla_{\boldsymbol{x}}\boldsymbol{f}(\boldsymbol{x})$ over that range. For the case of systems that become autonomous for a constant input

signal, we also need to know the expected value or range of expected values for the constant input, in order to investigate a tractable set of gradients $\nabla_{\boldsymbol{x}}\boldsymbol{g}(\boldsymbol{x}, u_0)$. We can find an example of such analysis for the case of a simple diode clipper distortion in Germain [2017].

In practice, this approach has limits. As investigating full ranges of variables can quickly become intractable, a possible simplification is to look only at the extreme values of the ranges for $\boldsymbol{x}$ (and u if relevant), though that might remove important information about intermediary points, and that ignores the fact that extreme values in all variables are not necessarily likely to happen simultaneously in the normal usage of the system.

In general, we have observed in our empirical studies that this approach will provide estimates for the highly damped poles of the system that are much higher than what is practically needed to properly design a dependent-coefficient method following the process described in this section. Indeed, it essentially assumes the target signal will be fed instantaneous functions which is often not a realistic scenario.

Pole analysis using discretization approximations

In our experiments, we found it useful to instead estimate the relevant instantaneous pole locations using well-known fixed-variable discretization methods. The process is then to look at the response sequence $\boldsymbol{x}_d[n]$ under an (optional) relevant input sequence $u_d[n]$ for a discretized model of the system. We then perform an estimate of the instantaneous pole locations by looking at the values of $\nabla_{\boldsymbol{x}}\boldsymbol{f}(t_n, \boldsymbol{x}_d[n])$ (or $\nabla_{\boldsymbol{x}}\boldsymbol{g}(\boldsymbol{x}_d[n], u_d[n])$ if applicable). In case where the number of relevant input sequences $u_d[n]$ to test is reasonably small (see Sec. 3.10 for examples), we can then quickly collect a set of relevant instantaneous pole estimates that can then be used to apply the chosen design criterion and form the final model.

In particular, we found that the backward Euler method, being an L-stable method, is often a good investigative tool to estimate the location of the instantaneous poles with higher damping we may encounter in a practical implementation of any of our method, while the bilinear transform can provide a good qualitative picture of the location of the poles with lesser damping. We show an example of such analysis in Sec. 3.10.1.

3.9 Implementation considerations

Here are some considerations regarding the implementation of a Möbius transform-based discretization for the following typical dynamical system description formalisms as found in state-of-the-art virtual analog audio methods.

3.9.1 General system of ordinary differential equations

A general system of ordinary differential equations is written as

$$\mathbf{0} = \mathbf{G}'(t, \boldsymbol{x}'(t), \boldsymbol{x}'^{(1)}(t), \ldots, \boldsymbol{x}'^{(n)}(t)). \tag{3.115}$$

It is well-known these kind of systems of equations can be rewritten in the compact form

$$\mathbf{0} = \mathbf{G}(t, \boldsymbol{x}(t), \dot{\boldsymbol{x}}(t)). \tag{3.116}$$

This general formulation of system equations, denoted as *implicit*, is somewhat rare and generally needed only for very complex systems. A more frequency from of ordinary differential equations are *explicit* systems of equations, of the form

$$\dot{\boldsymbol{x}}(t) = \boldsymbol{g}(t, \boldsymbol{x}(t)). \tag{3.117}$$

There is actually several ways possible to turn a mapping into a discretization of this equation system. The most typical is the linear one-step discretization formula in Eq. (3.29), such that the mapping

$$s \mapsto \frac{1}{T} \frac{1 + a_1 z^{-1}}{b_0 + b_1 z^{-1}} \tag{3.118}$$

leads to the discrete-time update

$$x_d[n] = -a_1 \boldsymbol{x}_d[n-1] + T_s \left[b_0 \, \boldsymbol{g}(t_{n+1}, \boldsymbol{x}_d[n+1]) + b_1 \, \boldsymbol{g}(t_n, \boldsymbol{x}_d[n]) \right]. \tag{3.119}$$

However, alternative implementations are available, for example following the model of the implicit midpoint method [Germain 2017], i.e.,

$$\boldsymbol{x}_d[n] = -a_1 \boldsymbol{x}_d[n-1] + T_s \, \boldsymbol{g}(b_0 \, t_{n+1} + b_1 \, t_n, b_0 \, \boldsymbol{x}_d[n+1] + b_1 \, \boldsymbol{x}_d[n]). \tag{3.120}$$

3.9.2 General system of differential algebraic equations

An even more general formulation is needed to represent certain classes of systems, for example, systems presenting an hysteresis effect. We then need to express things in the form of the system of differential algebraic equations instead, such that we have

$$\begin{cases} \mathbf{0} = \mathbf{G}(t, \boldsymbol{x}(t), \dot{\boldsymbol{x}}(t), \boldsymbol{v}(t)) \\ \mathbf{0} = \mathbf{H}(t, \boldsymbol{x}(t), \boldsymbol{v}(t)) \end{cases} \tag{3.121}$$

Again, as for ordinary differential equations, the dynamic part of the expression described by G_1

can generally be expressed in the explicit form instead, i.e.,

$$\begin{cases} \dot{\boldsymbol{x}}(t) = \boldsymbol{g}(t, \boldsymbol{x}(t), \boldsymbol{v}(t)) \\ \mathbf{0} = \mathbf{H}(t, \boldsymbol{x}(t), \boldsymbol{v}(t)) \end{cases} \tag{3.122}$$

Only the dynamical part of the system needs to be discretized. The process is rather similar to the process in the previous section on ordinary differential equations, with options such as

$$\begin{cases} \boldsymbol{x}_d[n] = -a_1 \boldsymbol{x}_d[n-1] + T_s \left[b_0 \, \boldsymbol{g}(t_{n+1}, \boldsymbol{x}_d[n+1], \boldsymbol{v}_d[n+1]) + b_1 \, \boldsymbol{g}(t_n, \boldsymbol{x}_d[n], \boldsymbol{v}_d[n]) \right] \\ \mathbf{0} = \mathbf{H}(t, \boldsymbol{x}_d[n], \boldsymbol{v}_d[n]) \end{cases} \tag{3.123}$$

or alternatively

$$\begin{cases} \boldsymbol{x}_d[n] = -a_1 \boldsymbol{x}_d[n-1] + T_s \, b_0 \, \boldsymbol{h} \left(b_0 \, t_{n+1} + b_1 \, t_n, \right. \\ \qquad\qquad \left. b_0 \, \boldsymbol{x}_d[n+1] + b_1 \, \boldsymbol{x}_d[n], b_0 \, \boldsymbol{v}_d[n+1] + b_1 \, \boldsymbol{v}_d[n] \right) \\ \mathbf{0} = \mathbf{G}(t, \boldsymbol{x}_d[n], \boldsymbol{v}_d[n]) \end{cases} \tag{3.124}$$

3.9.3 State-space model

Another general form in which we find system described is the state-space form. This kind of form differs from the formalisms above in that it makes explicit the input variables $\boldsymbol{u}$ and the output variables $\boldsymbol{y}$ in the formalism as

$$\begin{cases} \mathbf{0} = \mathbf{G}(t, \boldsymbol{x}(t), \dot{\boldsymbol{x}}(t), \boldsymbol{u}(t)) \\ \mathbf{0} = \mathbf{H}(t, \boldsymbol{y}(t), \boldsymbol{x}(t), \boldsymbol{u}(t)) \end{cases} \tag{3.125}$$

The dynamical variables $\boldsymbol{x}$ are denoted as *state* variables in this formalism. Here again, the implicit form is rather rare, the following explicit form is more common:

$$\begin{cases} \dot{\boldsymbol{x}}(t) = \boldsymbol{g}(t, \boldsymbol{x}(t), \boldsymbol{u}(t)) \\ \boldsymbol{y}(t) = \boldsymbol{h}(t, \boldsymbol{x}(t), \boldsymbol{u}(t)) \end{cases} \tag{3.126}$$

The typical approach to apply a mapping results in the update equations

$$\begin{cases} \boldsymbol{x}_d[n] = -a_1 \boldsymbol{x}_d[n-1] + T_s \left[b_0 \, \boldsymbol{g}(t_{n+1}, \boldsymbol{x}_d[n+1], \boldsymbol{u}_d[n+1]) + b_1 \, \boldsymbol{g}(t_n, \boldsymbol{x}_d[n], \boldsymbol{u}_d[n]) \right] \\ \boldsymbol{y}_d[n] = \boldsymbol{h}(t_n, \boldsymbol{x}_d[n], \boldsymbol{u}_d[n]) \end{cases} \tag{3.127}$$

or alternatively

$$\begin{cases} \boldsymbol{x}_d[n] = -a_1 \boldsymbol{x}_d[n-1] + T_s \, \boldsymbol{g} \left(b_0 \, t_{n+1} + b_1 \, t_n, \right. \\ \qquad\qquad \left. b_0 \, \boldsymbol{x}_d[n+1] + b_1 \, \boldsymbol{x}_d[n], b_0 \, \boldsymbol{u}_d[n+1] + b_1 \, \boldsymbol{u}_d[n] \right) \\ \boldsymbol{y}_d[n] = \boldsymbol{h}(t_n, \boldsymbol{x}_d[n], \boldsymbol{u}_d[n]) \end{cases} \tag{3.128}$$

While this formulation corresponds to the typical approach to the discretization of state-space systems, there exists a strong ambiguity regarding how to treat the input signal(s) $\boldsymbol{u}$, echoing our discussion in Ch. 1. If we treat the state-space system as a general differential algebraic system, instead of Eq. (3.127) the discrete-time model of the system should be written as

$$\begin{cases} \boldsymbol{x}_d[n] = -a_1 \boldsymbol{x}_d[n-1] + T_s \left[b_0 \, \boldsymbol{g}(t_{n+1}, \boldsymbol{x}_d[n+1], \boldsymbol{u}(t_n)) + b_1 \, \boldsymbol{g}(t_n, \boldsymbol{x}_d[n], \boldsymbol{u}(t_n)) \right] \\ \boldsymbol{y}_d[n] = \boldsymbol{h}(t_n, \boldsymbol{x}_d[n], \boldsymbol{u}(t_n)) \end{cases} \tag{3.129}$$

and instead of Eq. (3.128), we would get

$$\begin{cases} \boldsymbol{x}_d[n] = -a_1 \boldsymbol{x}_d[n-1] + T_s \, \boldsymbol{g} \left(b_0 \, t_{n+1} + b_1 \, t_n, \right. \\ \qquad\qquad \left. b_0 \, \boldsymbol{x}_d[n+1] + b_1 \, \boldsymbol{x}_d[n], \boldsymbol{u}(b_0 \, t_{n+1} + b_1 \, t_n) \right) \\ \boldsymbol{y}_d[n] = \boldsymbol{h}(t_n, \boldsymbol{x}_d[n], \boldsymbol{u}(t_n)) \end{cases} \tag{3.130}$$

The issue here becomes how we represent the continuous-time input variable $\boldsymbol{u}(t)$ in the discrete-time domain, and to some extent whether the representation remains consistent between methods. We have already discussed extensively in Ch. 1 the various options in terms of sampling and interpolations approaches. In particular, the question is how to represent the quantity

$$\boldsymbol{u}(b_0 t_{n+1} + b_1 t_n). \tag{3.131}$$

In the context of this chapter, the treatment of the input variable has a limited impact on the design of a Möbius mapping since the design criteria generally revolve around pole locations, and these are mostly driven by the way the state variables $\boldsymbol{x}$ are treated. We refer the reader to Ch. 1 for additional details.

3.9.4 Nodal K-method

As mentioned in the introduction, the nodal K-method [Yeh et al. 2010, Yeh 2012] is a popular approach to set up circuit simulations for audio applications. It relies on the observation that many of such systems only involve static nonlinearities between a given branch voltage and a given branch circuit.

In Yeh et al. [2010], the method is laid out for systems discretized using the backward Euler method, meaning a standard bilinear transform mapping. We show below how a similar process can be followed for a more general mapping.

In the nodal DK-method, the system equations are expected of the form

$$\begin{cases} \dot{\boldsymbol{x}}(t) = \mathbf{A}\boldsymbol{x}(t) + \mathbf{B}\boldsymbol{u}(t) + \mathbf{C}\boldsymbol{i}(t) \\ \boldsymbol{i}(t) = \boldsymbol{f}(\boldsymbol{v}(t)) \\ \boldsymbol{v}(t) = \mathbf{D}\boldsymbol{x}(t) + \mathbf{E}\boldsymbol{u}(t) + \mathbf{F}\boldsymbol{i}(t) \\ \boldsymbol{y}(t) = \mathbf{L}\boldsymbol{x}(t) + \mathbf{M}\boldsymbol{u}(t) + \mathbf{N}\boldsymbol{i}(t) \end{cases} \tag{3.132}$$

As we can see, the specificity of this system of equations is to introduce the branch currents $\boldsymbol{i}$ and the branch voltages $\boldsymbol{v}$ as ancillary variables. We can then apply the mapping to these equations in a similar way as previously, which gets us

$$\begin{cases} (\mathbf{I} - b_0 T_s \mathbf{A})\boldsymbol{x}_d[n+1] = (b_1 T_s \mathbf{A} - a_1 \mathbf{I})\boldsymbol{x}_d[n] + b_0 T_s \mathbf{B}\boldsymbol{u}_d[n+1] \\ \qquad\qquad + b_1 T_s \mathbf{B}\boldsymbol{u}_d[n] + b_0 T_s \mathbf{C}\boldsymbol{i}[n+1] + b_1 T_s \mathbf{C}\boldsymbol{i}[n] \\ \boldsymbol{i}_d[n] = \boldsymbol{f}(\boldsymbol{v}_d[n]) \\ \boldsymbol{v}_d[n] = \mathbf{D}\boldsymbol{x}_d[n] + \mathbf{E}\boldsymbol{u}_d[n] + \mathbf{F}\boldsymbol{i}_d[n] \\ \boldsymbol{y}_d[n] = \mathbf{L}\boldsymbol{x}_d[n] + \mathbf{M}\boldsymbol{u}_d[n] + \mathbf{N}\boldsymbol{i}_d[n] \end{cases} \tag{3.133}$$

We can then apply a similar treatment to that system to solve it at each iteration using the K-method. By defining $\mathbf{H} = (\mathbf{I} - b_0 T_s \mathbf{A})^{-1}$, we can transform the first equation into

$$\begin{aligned} \boldsymbol{x}_d[n+1] = \mathbf{H}(b_1 T_s \mathbf{A} - a_1 \mathbf{I})\boldsymbol{x}_d[n] + b_0 T_s \mathbf{H}\mathbf{B}\boldsymbol{u}_d[n+1] \\ + b_1 T_s \mathbf{H}\mathbf{B}\boldsymbol{u}_d[n] + b_0 T_s \mathbf{H}\mathbf{C}\boldsymbol{i}[n+1] + b_1 T_s \mathbf{H}\mathbf{C}\boldsymbol{i}[n]. \end{aligned} \tag{3.134}$$

We can then express $\boldsymbol{i}_d[n+1]$ explicitly as

$$\begin{aligned} \boldsymbol{i}_d[n+1] = \boldsymbol{f}\,(\mathbf{D}\mathbf{H}(b_1 T_s \mathbf{A} - a_1 \mathbf{I})\boldsymbol{x}_d[n] + (b_0 T_s \mathbf{D}\mathbf{H}\mathbf{B} + \mathbf{E})\boldsymbol{u}_d[n+1] \\ + b_1 T_s \mathbf{D}\mathbf{H}\mathbf{B}\boldsymbol{u}_d[n] + (b_0 T_s \mathbf{D}\mathbf{H}\mathbf{C} + \mathbf{F})\boldsymbol{i}[n+1] + b_1 T_s \mathbf{D}\mathbf{H}\mathbf{C}\boldsymbol{i}[n]\,). \end{aligned} \tag{3.135}$$

We then define

$$\begin{aligned} \mathbf{K} &= b_0 T_s \mathbf{D}\mathbf{H}\mathbf{C} + \mathbf{F} \\ \boldsymbol{p}[n+1] &= \mathbf{D}\mathbf{H}(b_1 T_s \mathbf{A} - a_1 \mathbf{I})\boldsymbol{x}_d[n] + (b_0 T_s \mathbf{D}\mathbf{H}\mathbf{B} + \mathbf{E})\boldsymbol{u}_d[n+1] \\ &\quad + b_1 T_s \mathbf{D}\mathbf{H}\mathbf{B}\boldsymbol{u}_d[n] + b_1 T_s \mathbf{D}\mathbf{H}\mathbf{C}\boldsymbol{i}[n] \end{aligned} \tag{3.136}$$

in which case, we retrieve the relation

$$\boldsymbol{i}_d[n] = \boldsymbol{f}(\boldsymbol{p}[n] + \mathbf{K}\boldsymbol{i}_d[n]). \tag{3.137}$$

From there, all the subsequent derivations found in Yeh et al. [2010] apply to solve this implicit equation and furthermore the system and update the state variables and the output variables at

every time step.

3.9.5 Nodal discrete K-method

In Yeh et al. [2010], a modification to the nodal K-method is presented which overcomes issues appearing when the conductance matrix $\mathbf{G}$ of the circuit when performing the modified nodal analysis of a circuit of interest happens to be singular. That method was denoted then as *nodal discrete K-method*. The article details how the approach works for the trapezoidal rule, we extend here the formalism to all mappings.

This approach relies on leveraging the use of companion models for reactive elements (capacitor, inductor). For a given Möbius transform, the companion model of a reactive element whose equivalent source variable is denoted x_d is written as

$$x_d[n+1] = -a_1 x_d[n] + T_s[b_0 \dot{x}_d[n+1] + b_1 \dot{x}_d[n]]. \tag{3.138}$$

Thus, considering that the continuous-time equations relating branch voltages and currents are

$$C\dot{v}(t) = i(t) \tag{3.139}$$

for a capacitor of capacitance C and

$$L\dot{i}(t) = v(t) \tag{3.140}$$

for an inductor of inductance L, we get the companion models

$$I_d[n] = \frac{C}{T_s}\left(\frac{1}{b_0} - \frac{a_1}{b_1}\right) v_d[n] - \frac{b_1}{b_0} I_d[n-1] \tag{3.141}$$

with $I_d[n] = i_d[n] - \frac{C}{T_s}\frac{a_1}{b_1} v_d[n]$ (sometimes denoted as *equivalent source current*) for capacitors, and

$$V_d[n] = \left(1 - \frac{a_1 b_0}{b_1}\right) v_d[n] - a_1 V_d[n-1] \tag{3.142}$$

with $V_d[n] = v_d[n] - \frac{L}{T_s}\frac{a_1}{b_1} i_d[n]$ (sometimes denoted as *equivalent source voltage*) for inductors. We should notice here that these expressions are not valid for purely explicit (i.e., $b_0 = 0$) or implicit (i.e., $b_1 = 0$) mappings, so that the framework detailed in Yeh et al. [2010] should be considered only for cases where where $b_0 \neq 0$ and $b_1 \neq 0$. Additionally, the fact that trivial Möbius mappings (i.e., $b_0 a_1 - b_1 \neq 0$) are excluded means that the term in $v_d[n]$ is never zeroed out.

Since both companion models (Eqs. (3.141) and (3.142)) follow the formula showed in Yeh et al. [2010] for state update equations, in the form

$$x[n] = g v_d[n] + s x[n-1], \tag{3.143}$$

when setting x as I_d for capacitors and V_d for inductors, and matching the coefficients g and s to the companion model equations above. As a result, the derivation applied to the companion model equations for the trapezoidal method in Yeh et al. [2010] follows here too, and can be used to put together a discrete K-method system to update the system variables at every time step.

3.9.6 Wave digital filters

Werner [2016] presented the procedure to apply any Möbius transform mapping in context of wave digital filters, we recall here its results for completeness. We will assume here some general fluency by the reader in wave digital filter formalism. For more details regarding that formalism, we refer the reader to the more detailed development given in Ch. 4 as well as Werner [2016].

By nature of their formalism, in the case of linear discretization formulas, wave digital filter formalism only requires to figure out the update equations of reactive elements. In the context of audio circuits, this generally means deriving the equations of the capacitor and the inductor elements. In the case of so-called *voltage waves*, these *scattering* equations describe the formula for converting the so-called *incident wave variables* $a_d[n]$, defined as

$$a_d[n] = v_d[n] + R_p i_d[n] \tag{3.144}$$

(with v_d the voltage across the element, i_d the current through the element and R_p the port resistance) into the *reflected wave variables* $b_d[n]$ written as

$$b_d[n] = v_d[n] - R_p i_d[n]. \tag{3.145}$$

Thus, considering the continuous-time equations describing a capacitor and an inductor (Eqs (3.139) and (3.140)), we can apply the mapping

$$s = \frac{\gamma_1 z + \gamma_2}{\gamma_3 z + \gamma_4} \tag{3.146}$$

and, as shown in Werner [2016], the resulting scattering equations are

$$b_d[n+1] = -\frac{\gamma_4 + R_p C \gamma_2}{\gamma_3 + R_p C \gamma_1} b_d[n] + \frac{\gamma_3 - R_p C \gamma_1}{\gamma_3 + R_p C \gamma_1} a_d[n+1] + \frac{\gamma_4 - R_p C \gamma_2}{\gamma_3 + R_p C \gamma_1} a_d[n] \tag{3.147}$$

for the capacitor and

$$b_d[n+1] = -\frac{L\gamma_4 + R_p \gamma_2}{L\gamma_3 + R_p \gamma_1} b_d[n] + \frac{L\gamma_3 - R_p \gamma_1}{L\gamma_3 + R_p \gamma_1} a_d[n+1] + \frac{L\gamma_4 - R_p \gamma_2}{L\gamma_3 + R_p \gamma_1} a_d[n] \tag{3.148}$$

for the inductor.

One core principle in the wave digital formalism is the concept of *port adapation* which removes

the dependency of $b_d[n+1]$ to $a_d[n+1]$ by appropriately setting R_p. Werner [2016] notes that this means that mappings such that $\gamma_1 = 0$ (i.e., mapping forming explicit methods such as forward Euler) do not allow for that port adaptation and are therefore a priori unsuitable for the wave digital filter formalism. Mappings such that $\gamma_3 = 0$ are also disallowed, but as we discussed earlier, all numerical methods in the literature also rely on that hypothesis. Also, these conditions match the conditions under which the nodal DK-method apply as discussed in Sec. 3.9.5. Adapting the capacitor port resistance is done by setting $R_p = \frac{\gamma_3}{C\gamma_1}$, which adapts Eq. (3.147) to have

$$b_d[n+1] = \frac{\gamma_1\gamma_4 + \gamma_2\gamma_3}{2\gamma_1\gamma_3} b_d[n] + \frac{\gamma_1\gamma_4 - \gamma_2\gamma_3}{2\gamma_1\gamma_3} a_d[n]. \tag{3.149}$$

Adapting the inductor port resistance is done by setting $R_p = \frac{L\gamma_3}{\gamma_1}$, which adapts Eq. (3.148) to have

$$b_d[n+1] = -\frac{\gamma_1\gamma_4 + \gamma_2\gamma_3}{2\gamma_1\gamma_3} b_d[n] + \frac{\gamma_2\gamma_3 - \gamma_1\gamma_4}{2\gamma_1\gamma_3} a_d[n]. \tag{3.150}$$

Here, we want to point out that since the exclusion of non-trivial Möbius mappings, which corresponds to $\gamma_1\gamma_4 - \gamma_2\gamma_3 \neq 0$, guarantees the dependency between $b_d[n+1]$ and $a_d[n]$ for all allowed mappings.

3.9.7 Generalized state-space

In Holters and Zölzer [2015], an alternative to the nodal discrete K-method was proposed to form a system of equations in state-space form as well. They show that many audio circuits of interest result in equations of the form

$$\begin{cases} \boldsymbol{u}(t) = \mathbf{M}_v \boldsymbol{v}(t) + \mathbf{M}_i \boldsymbol{i}(t) + \mathbf{M}_x \boldsymbol{x}(t) + \mathbf{M}_{\dot{x}} \dot{\boldsymbol{x}}(t) + \mathbf{M}_q \boldsymbol{q}(t) & \text{(linear elements)} \\ \mathbf{0} = \boldsymbol{f}(\boldsymbol{q}(t)) & \text{(memoryless nonlinearities)} \\ \mathbf{0} = \mathbf{T}_v \boldsymbol{v}(t) & \text{(Kirchhoff voltage laws)} \\ \mathbf{0} = \mathbf{T}_i \boldsymbol{i}(t) & \text{(Kirchhoff current laws)} \end{cases} \tag{3.151}$$

The article details how the approach works for the trapezoidal rule, we extend here the formalism to all mappings. The first step is to define *canonical states*. For a generic mapping, such a canonical state becomes

$$\overline{\boldsymbol{x}}[n] = \boldsymbol{x}_d[n] - T_s \frac{b_1}{a_1} \dot{\boldsymbol{x}}_d[n] \tag{3.152}$$

so that with

$$\begin{cases} \left(b_0 - \frac{b_1}{a_1}\right) \boldsymbol{x}_d[n] = \frac{1}{T_s}(\overline{\boldsymbol{x}}[n] + a_1 \overline{\boldsymbol{x}}[n-1]) \\ \left(b_0 - \frac{b_1}{a_1}\right) \dot{\boldsymbol{x}}_d[n] = b_0 \overline{\boldsymbol{x}}[n] + b_1 \overline{\boldsymbol{x}}[n-1] \end{cases} \tag{3.153}$$

we can form

$$\begin{cases} \mathbf{M}_{\overline{x}'} = \frac{1}{b_0 - \frac{b_1}{a_1}} \left(\frac{1}{T_s} \mathbf{M}_{\dot{x}} + b_0 \mathbf{M}_x \right) \\ \mathbf{M}_{\overline{x}} = \frac{1}{b_0 - \frac{b_1}{a_1}} \left(\frac{a_1}{T_s} \mathbf{M}_{\dot{x}} + b_1 \mathbf{M}_x \right) \end{cases} \tag{3.154}$$

that is well-defined for any non-trivial mapping (i.e., $a_1 b_0 - b_1 \neq 0$) leading to the system

$$\begin{cases} \mathbf{M}_{\overline{x}} \overline{\boldsymbol{x}}_d[n-1] + \boldsymbol{u}_d[n] = \mathbf{M}_v \boldsymbol{v}_d[n] + \mathbf{M}_i \boldsymbol{i}_d[n] + \mathbf{M}_{\overline{x}'} \overline{\boldsymbol{x}}_d[n] + \mathbf{M}_q \boldsymbol{q}(t) \\ \mathbf{0} = \mathbf{T}_v \boldsymbol{v}(t) \\ \mathbf{0} = \mathbf{T}_i \boldsymbol{i}(t) \\ \mathbf{0} = \boldsymbol{f}(\boldsymbol{q}(t)) \end{cases} \tag{3.155}$$

This system then matches the form in Holters and Zölzer [2015] that can be solved using the same procedure presented in the paper for the trapezoidal method.

3.9.8 Conjugate methods

As we mentioned in Secs. 3.9.1, 3.9.2 and 3.9.3, while the straightforward way numerical methods based on mappings follows the linear one-step discretization form based on Eq. (3.29), it is not the unique way to apply it. In particular, there also exists a class of alternative similar to the approach used in the implicit midpoint method [Germain 2017]. However, this approach present the a priori limitation that the published formalisms available to set up the system of equations and recalled above (e.g., nodal K-method) rely on the linear one-step form.

However, it was shown in [Germain 2017] that, due to the fact that the midpoint method is the conjugate method of the trapezoidal method [Hairer et al. 2006], we can overcome this limitation and use all the methods presented here to generate an implicit midpoint solution sequence if we so desire.

This result can be extended to all mappings. as we show below. For simplicity, we focus on applying these methods to a simple ordinary differential equation $\dot{x}(t) = f(t, x(t))$ with initial condition $x(0) = x_0$. Applying a given mapping in a midpoint-like form, leads to the update equation

$$x_d[n+1] = -a_1 x_d[n] + T_s f(b_0 t_{n+1} + b_1 t_n, b_0 x_d[n+1] + b_1 x_d[n]) \tag{3.156}$$

with initial condition $x_d[0] = x_0$. Let's define the sequence $y_d[n+1] = b_0 x_d[n+1] + b_1 x_d[n]$. Then, we get

$$\begin{cases} x_d[n+1] = -a_1 x_d[n] + T_s f(b_0 t_{n+1} + b_1 t_n, y_d[n+1]) \\ x_d[n] = -a_1 x_d[n-1] + T_s f(b_0 t_n + b_1 t_{n-1}, y_d[n]) \end{cases} \tag{3.157}$$

which we can combine to form

$$\underbrace{b_0 x_d[n+1] + b_1 x_d[n]}_{y_d[n+1]} + \underbrace{b_0 a_1 x_d[n] + b_1 a_1 x_d[n-1]}_{a_1 y_d[n]}$$
$$= T_s[b_0 f(b_0 t_{n+1} + b_1 t_n, y_d[n+1]) + b_1 f(b_0 t_n + b_1 t_{n-1}, y_d[n])] \quad (3.158)$$

so that we have

$$y_d[n+1] = -a_1 y_d[n] + T_s[b_0 f(b_0 t_{n+1} + b_1 t_n, y_d[n+1]) + b_1 f(b_0 t_n + b_1 t_{n-1}, y_d[n])] \quad (3.159)$$

meaning the sequence y_d follows the update equation corresponding to the linear one-step equation associated with that same mapping, setting as initial condition $y_d[0]$ such that it solves the implicit equation

$$b_1 T_s f(-b_1 T_s, y_d[0]) - a_1 y_d[0] = (b_1 - a_1 b_0) x_d[0]. \quad (3.160)$$

Hence, once we set up a system to solve for the sequence y_d, we can get the sequence x_d following the midpoint-like update as solution of the recurrent system

$$\begin{cases} x_d[n] = \frac{1}{b_0} y_d[n] - \frac{b_1}{b_0} x_d[n-1] \\ x_d[0] = x_0 \end{cases} \quad (3.161)$$

This derivation extends trivially to the multidimensional case, as well as to differential algebraic equation systems and state-space systems.

3.10 Case studies

The theory we developed is particularly relevant to the modeling of audio systems due to the ubiquity of circuits using diodes as pseudo-switches in analog audio circuits. The diodes, depending on whether they are conducting or not, activate some sections depending on some feature of the input signal. Such property was found very useful for a class of circuit we will denote as envelope shaping. These circuits are characterized by very simple input voltage signals (typically a rectangular pulse), often delivered by an internal source. The circuit is then composed of various linear components that "shape" a more complex output signal shape in response. In order to generate shapes more complex than what can be achieved solely through linear time-invariant filtering, diodes acting as switches create a nonlinear effect that can be interpreted as some form of time-varying filtering with the filter characteristics changing between phases where the diode conducts (and possibly shorts some part of the circuit) or blocks current (and possibly disconnects some part of the circuit).

Generally, these circuits are meant to be used in scenarios where the changes of state of the diode between conducting and blocking are few over the course of a sonic "event". As a result, the circuit mostly behaves as a linear system except for these changes of state. As a result, oversampling is an unnecessarily costly procedure as the linear segments are generally well modeled at the regular

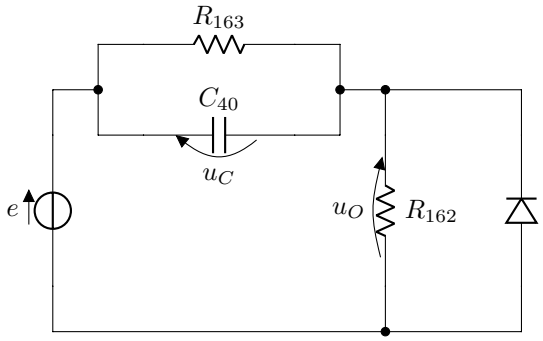


Figure 3.13: TR-808 bass drum pulse shaper circuit.

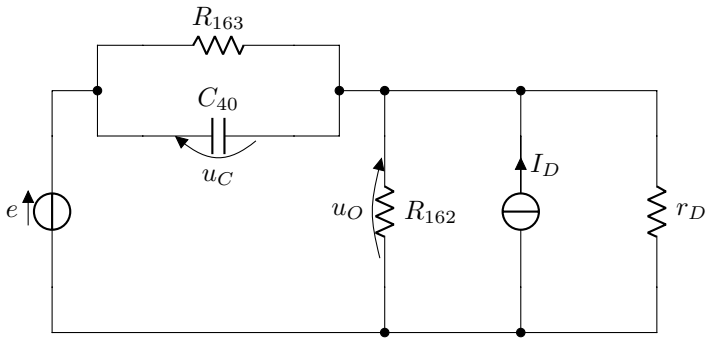


Figure 3.14: TR-808 bass drum pulse shaper circuit with diode companion model.

sampling rate. Hence, a computationally efficient approach to the problem is to use the framework presented here, allowing for fast and accurate simulations of the linear segments, while avoiding issues with the spurious oscillations of the bilinear transform models. Indeed, the instantaneous poles of the system generally become stiff when a diode switches from blocking to conducting, creating such problems.

Below are three case studies of historical audio circuits matching these characteristics. We will show how the theory demonstrated earlier in this chapter can be applied to generate efficient and accurate discretization procedures.

3.10.1 TR-808 bass drum pulse shaper

Circuit description

The diode clipper bass drum pulse shaper analyzed in Werner [2016]. It is a distortion circuit found in the TR-808 drum machine circuit where a saturating diode is adding to a resistor-capacitor high-pass circuit as shown in Fig. 3.13. The circuit is designed to convert an input rectangular pulse into an exponentially-decaying envelope signal. The diode is there to act as a circuit switch, blocking the

negative shape that would be created on the pulse release. The input signal is represented here by the voltage source e and the output signal corresponds to the voltage v_R across the resistor element R_2.

Building on the research presented in this section, Werner [2016] applied the α transform to remove a spurious envelope trigger resulting from the poor behavior of the bilinear transform. However, rather than use our framework, the coefficient α was picked to empirically minimize the root mean-square error on a target response without analysis. Here, we present the full pole analysis of the circuit and selection process for the coefficient α.

Circuit analysis

For the input-output structure given above, the behavior of the system is described through the state-space equation describing the voltage across the capacitor element

$$\dot{u}_C = \frac{e}{R_{162}C_{40}} - \frac{u_C}{(R_{162}||R_{163})C} - \frac{1}{C}f_D(u_C - e). \tag{3.162}$$

To analyze the instantaneous poles of the system, we need to use the diode companion model. The companion model linearizes the diode response around its current voltage point $\tilde{u}_D$. For a general nonlinear diode model $i_D = f_D(u_D)$, the linearization gives us

$$i_D \approx \underbrace{f_D(\tilde{u}_D)}_{I_D} + (u_D - \tilde{u}_D) \cdot \underbrace{\frac{df_D}{du}(\tilde{u}_D)}_{1/r_D}. \tag{3.163}$$

In the case of the Shockley ideal diode model [Shockley 1949], we have

$$i_D = I_s\left(e^{\frac{\tilde{u}_D}{V_T}} - 1\right), \tag{3.164}$$

where I_s and V_T are respectively the saturation current and the thermal voltage as defined in the model. The companion model variables are then given by

$$\begin{cases} I_D = I_s\left(e^{\frac{\tilde{u}_D}{V_T}} - 1\right) \\ r_D = \frac{V_T}{I_s}e^{-\frac{\tilde{u}_D}{V_T}} \end{cases} \tag{3.165}$$

With the companion model, the circuit becomes as shown in Fig. 3.14. In that case, the state-space equation can be rewritten as

$$\dot{u}_C = \frac{e}{(R_{162}||r_D)C_{40}} - \frac{u_C}{(R_{162}||R_{163}||r_D)C_{40}} - \frac{I_D}{C_{40}}. \tag{3.166}$$

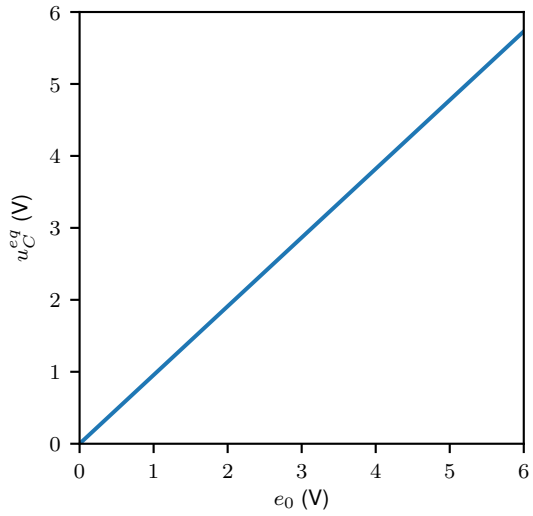


Figure 3.15: Equilibrium capacitor voltage u_C^{eq} in the TR-808 pulse shaper circuit as a function of the constant input voltage e_0 for voltages in the range 0 V to 6 V.

This allows us to see that the only instantaneous pole is given by

$$p = -\frac{1}{(R_{162}||R_{163}||r_D)C_{40}}. \tag{3.167}$$

Hence, we see how on the pulse release, a negative spike in the quantity $e - v_C$ (i.e., any sudden drop of the source voltage e) forces the value of the companion model resistor r_D towards 0, and as a result increases significantly the pole damping. Since the circuit is designed to operate as a pulse shaper, drops of input voltage are an essential part of the functioning of this circuit and must be taken into account in the model design.

Instantaneous pole estimates

In order to design an α-transform discretization that enforces damping monotonicity (as described in Sec. 3.8.2), we need to have some knowledge of the locations of the instantaneous poles with higher damping we expect to encounter in the system. From the empirical study of this system Werner [2016], we know that, in its normal mode of operation:

- The pulse shaper should be fed positive rectangular pulses of about 1 ms,
- While the input positive pulse is on, the system responds generally as a regular linear RC low-pass circuit (due to the fact that the diode is blocking) and creates an exponential pulse decaying fast to an equilibrium as the charge in the capacitor stabilizes,

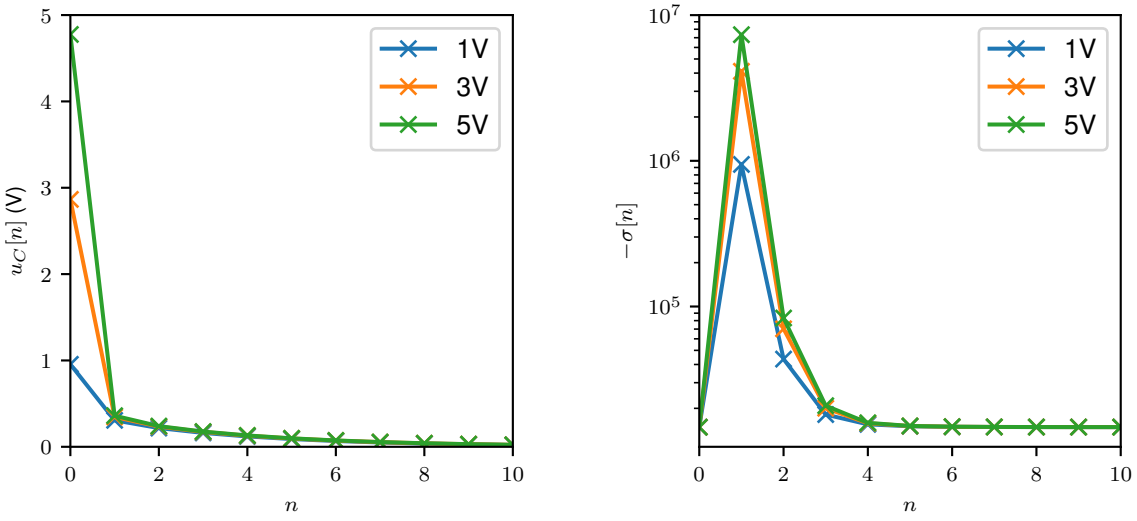


Figure 3.16: Simulated capacitor voltage $u_C[n]$ (left) and corresponding estimated instantaneous continuous-time pole damping (right) for a backward Euler model of the TR-808 pulse shaper. The model starts from the equilibrium voltage for three given input voltages $e_0 = \{1\,\mathrm{V}, 3\,\mathrm{V}, 5\,\mathrm{V}\}$ at time $n = 0$, and the input voltage $e[n]$ is then set to $0\,\mathrm{V}$ for all following samples $n > 0$.

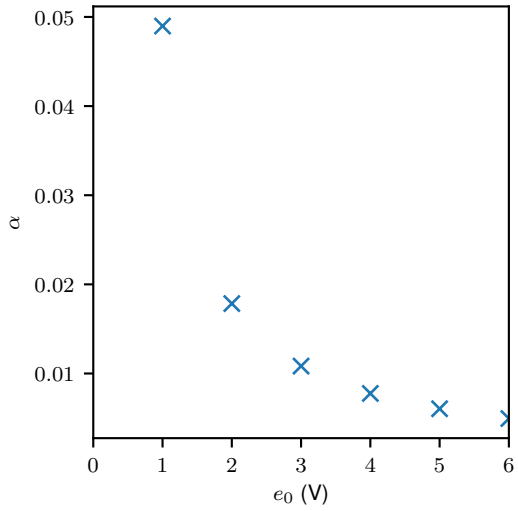


Figure 3.17: Estimated optimal α parameter to preserve damping monotonicity as a function of the starting equilibrium position. The estimates are based on the observed instantaneous poles in the backward Euler simulation described in Fig. 3.16 for an input voltage downward step function going with six different input voltages $e_0 = \{1\,\mathrm{V}, \ldots, 6\,\mathrm{V}\}$.

- When the positive pulse ends, the capacitor start discharging, and the diode temporarily becomes passing ($r_D << 1$), moving the instantaneous poles of the system in the stiff region (i.e., the region of poles with high damping). This causes the bilinear transform model to misbehaves and create an extra pulse as we will see below.

From these empirical observations, it is clear that the problematic case that our method must address are the end of input pulses, generally after the system has (almost) reached his temporary equilibrium state as dictated by the pulse amplitude. As outlined in Sec. 3.8.2, we propose to analyze the locations of the instantaneous pole with higher damping by looking at the response of the backward Euler model of the system in this scenario.

First, we compute the equilibrium voltages for when the circuit is under a constant loading voltage e_0. This equilibrium can be found through root-finding after setting $\dot{u}_C = 0$ in Eq. (3.162). The solution for loading voltages between 0 V and 6 V is shown in Fig. 3.15. We see that the relation between the equilibrium capacitor voltage u_C^{eq} is roughly linear, which matches our intuition that, since the diode is blocking for positive loading voltages, the circuit essentially behaves linearly (with the diode replaced by an open branch). From these equilibria, we then simulate the system when the input voltage is set back to 0 V (the downward step of the pulse) using a backward Euler model, and we estimate the instantaneous pole from the system Jacobian (see Eq. (3.167)) at the capacitor voltages visited by the model, as seen in Fig. 3.16. From there, we can apply our monotonicity criterion (see Sec. 3.8.2) to get an optimal α. The optimal α obtained for various starting equilibria is shown in Fig. 3.17.

Simulation results

We compare simulations of the system for a bilinear transform model, and an α-transform model of the pulse shaper at $f_s = 44.1\,\text{kHz}$ ($T_s \approx 22.67\,\mu\text{s}$). The transform is parametrized following the backward Euler simulation of a downward input voltage step starting at an input voltage 2 V, i.e., $\alpha = 0.0263$. We simulate two pulses, respectively at voltage 1 V (see Fig. 3.18) and 2 V (see Fig. 3.19), both of duration 1 ms (i.e., 44 samples). For comparison, we also show the result of a high resolution simulation obtained using the Matlab numerical solver `ode15s` where the continuous-time input voltage is set as the linear interpolation of the discrete-time pulse.

As we can see, and as has been observed in Werner [2016], the instantaneous poles of the system with higher damping create an additional pulse in the bilinear transform, matched by the inversion of the model damping curve compared with the "analog" system. This pulse becomes stronger and stronger as the intensity of the input pulse increases. On the other hand, the α-transform model optimized for 2 V successfully simulates the system for both input pulses, with a good match in the instantaneous pole trajectories as well, especially in the stiff region. In the non-stiff region, the α-transform pole is slightly under-damped compared to the "analog" reference and the bilinear transform, but the difference appears to have little effect on the model general accuracy. Using

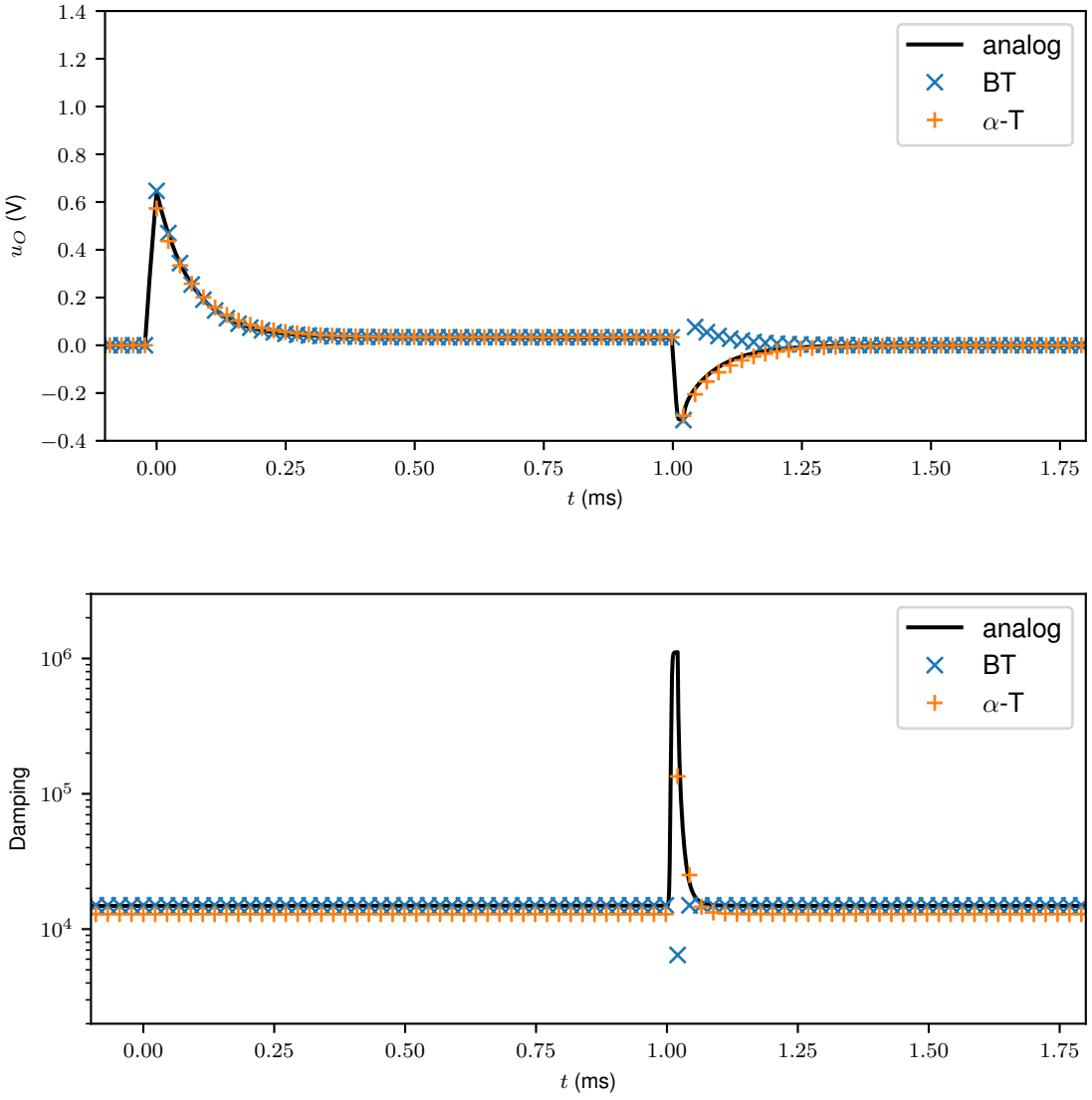


Figure 3.18: Output voltage u_O and corresponding instantaneous pole locations for models of the TR-808 pulse shaper for a 1 ms input pulse at 1 V. We compare a bilinear transform model (blue), an optimized α-transform model for $\alpha = 0.0263$ (blue) at sampling rate $f_s = 44.1\,\text{kHz}$ ($T_s \approx 22.67\,\mu\text{s}$), and a high-resolution "analog" model (black).

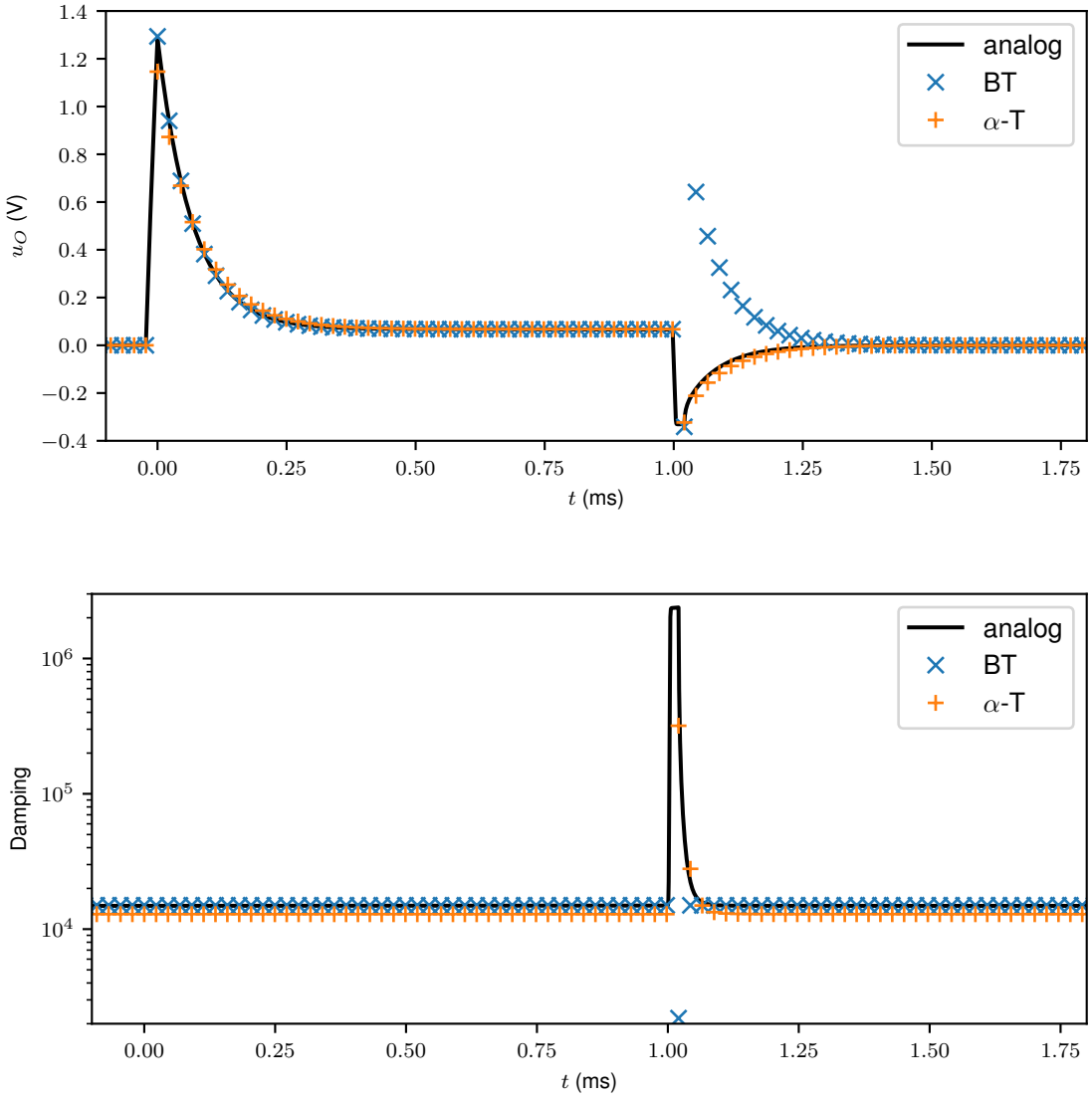


Figure 3.19: Output voltage u_O and corresponding instantaneous pole locations for models of the TR-808 pulse shaper for a 1 ms input pulse at 2 V. We compare a bilinear transform model (blue), an optimized α-transform model for $\alpha = 0.0263$ (blue) at sampling rate $f_s = 44.1\,\text{kHz}$ ($T_s \approx 22.67\,\mu\text{s}$), and a high-resolution "analog" model (black).

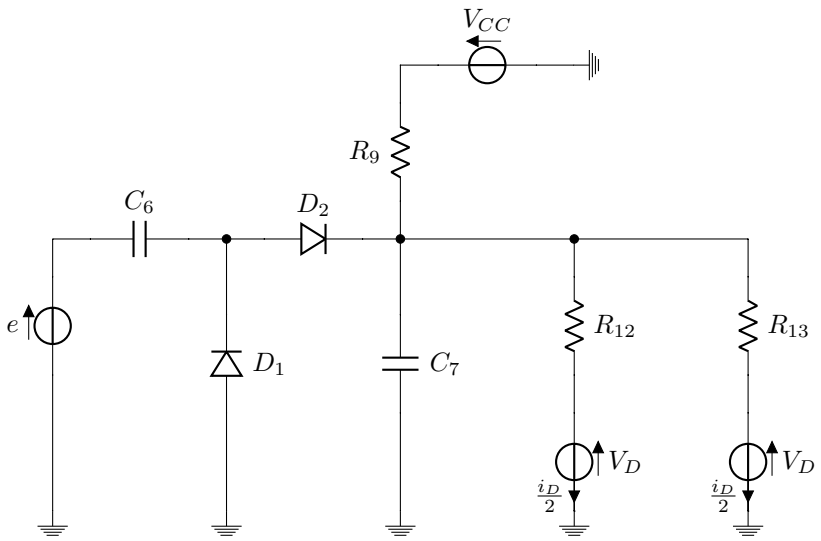


Figure 3.20: Nonlinear section of a DOD FX-25 clone envelope follower circuit.

the parametric α-transform or the $\alpha\beta$-transform (see Sec. 3.7.3) provides the needed degrees of freedom to match that pole location ($p \approx -\frac{1}{(R_{162}||R_{163})C_{40}}$) as well, in addition of controlling damping monotonicity.

3.10.2 DOD FX-25 envelope follower

Circuit description

The next circuit we study is the envelope follower circuit of a DOD FX-25 guitar pedal clone as analyzed in [Bogason 2018]. This circuit is particularly interesting as it can be decomposed into a linear section followed by a nonlinear section due to the presence of an operational transconductance amplifier set up as voltage follower between the two. We present here an extended analysis based on the results in Bogason [2018]. The qualitative behavior of the circuit is as follows. Any input signal is first equalized by the linear input filter which is the combination of a 1st-order high-pass filter and a 1st-order high-shelf filter. The equalized signal is then use as excitation signal for the nonlinear circuit. The two-diode circuit creates a two-stage decaying exponential envelope response as the diode switch from blocking to passing, or from passing to blocking. This creates two instants where the instantaneous poles of the system wander into the stiff region.

The transfer function of the linear section can be found in Bogason [2018], we focus here on the extended analysis of the nonlinear section.

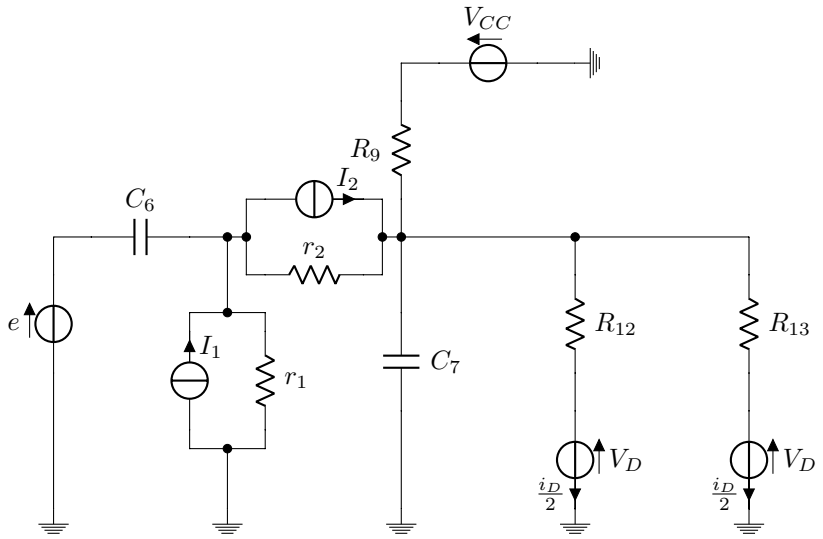


Figure 3.21: Nonlinear section of a DOD FX-25 clone envelope follower circuit with diode companion model.

Circuit analysis

The nonlinear section of that circuit corresponds to the circuit displayed in Fig. 3.20, where the input voltage e corresponds to the output measured at the output of the operational transconductance amplifier in the linear section as described in Bogason [2018]. Its output corresponds to the total current i_D flowing through the voltage sources V_D, which is split equally as $R_{12} = R_{13}$. Note that these sources are not actual voltage sources but simplified equivalents of the circuitry associated with two operational transconductance amplifiers in the circuit. The output current i_O is expressed as

$$i_O = i_D = (u_{C_7} - V_D)(1/R_{12} + 1/R_{13}). \tag{3.168}$$

The two diodes allow for three different regimes in the circuit. When a positive transient is applied to the circuit, the diode D_2 becomes conducting, allowing for the rapid charging of capacitor C_6 and C_7 which then become vanishing voltage sources once the input transient is done. Similarly, the diode D_1 becomes conducting for negative transients, but in this case, only the capacitor C_6 gets charged during that transient. In the absence of transients, both diodes are blocking, so that the dynamics of the circuit are essentially driven by the capacitor energy being released and dissipated through the circuit resistors.

For the input-output structure given above and using the notation from Fig. 3.20, the behavior

of the system is described through the state-space system

$$\begin{aligned}\dot{u}_{C_6} &= \frac{1}{C_6} f_D(e - v_{C_6} - v_{C_7}) - \frac{1}{C_6} f_D(v_{C_6} - e),\\ \dot{u}_{C_7} &= \frac{1}{C_7} f_D(e - v_{C_6} - v_{C_7}) + \frac{V_{CC}}{R_9 C_7} + \frac{V_D}{(R_{12}||R_{13})C_7} - \frac{v_{C_7}}{(R_9||R_{12}||R_{13})C_7}.\end{aligned} \tag{3.169}$$

To analyze the instantaneous poles of the system, we use the diode companion model as shown in Fig. 3.14. This leads to the linearized circuit displayed in Fig. 3.21. In that case, the state-space system can be rewritten as the matrix system

$$\begin{aligned}\begin{bmatrix}\dot{v}_{C_1}\\ \dot{v}_{C_2}\end{bmatrix} = &\begin{bmatrix} -\dfrac{1}{(r_1||r_2)C_6} & -\dfrac{1}{r_2 C_6}\\ -\dfrac{1}{r_2 C_7} & -\dfrac{1}{(R_9||R_{12}||R_{13}||r_2)C_7}\end{bmatrix}\begin{bmatrix}v_{C_6}\\ v_{C_6}\end{bmatrix}\\ &+\begin{bmatrix}\dfrac{1}{(r_1||r_2)C_6} & 0 & 0\\ \dfrac{1}{r_2 C_7} & \dfrac{1}{R_9 C_7} & \dfrac{1}{(R_{12}||R_{13})C_7}\end{bmatrix}\begin{bmatrix}e\\ V_{cc}\\ V_D\end{bmatrix} + \begin{bmatrix}-\dfrac{1}{C_6} & \dfrac{1}{C_6}\\ 0 & \dfrac{1}{C_7}\end{bmatrix}\begin{bmatrix}I_1\\ I_2\end{bmatrix}.\end{aligned} \tag{3.170}$$

Analytical pole analysis

We know that the behavior of the system is strongly tied to the instantaneous poles of the system given as the eigenvalues of the matrix

$$A = \begin{bmatrix} -\dfrac{1}{(r_1||r_2)C_6} & -\dfrac{1}{r_2 C_6}\\ -\dfrac{1}{r_2 C_7} & -\dfrac{1}{(R_p||r_2)C_7}\end{bmatrix} \tag{3.171}$$

with $R_p = (R_9||R_{12}||R_{13})$.

The determinant of that matrix is given as

$$|A| = \frac{r_1 + r_2 + R_p}{r_1 r_2 R_p C_6 C_7} > 0. \tag{3.172}$$

The eigenvalues λ are given as roots of the equation

$$\lambda^2 + \left(\frac{1}{(r_1||r_2)C_6} + \frac{1}{(R_p||r_2)C_7}\right)\lambda + \frac{r_1 + r_2 + R_p}{r_1 r_2 R_p C_6 C_7} = 0. \tag{3.173}$$

The discriminant of the polynomial is given by

$$\Delta = \left(\frac{1}{(r_1||r_2)C_6} + \frac{1}{(R_p||r_2)C_7}\right)^2 - 4\frac{r_1 + r_2 + R_p}{r_1 r_2 R_p C_6 C_7}. \tag{3.174}$$

One thing to notice is that the discriminant can also be written as

$$\Delta = \left(\frac{1}{(r_1||r_2)C_6} - \frac{1}{(R_p||r_2)C_7}\right)^2 + \frac{4}{r_2^2 C_6 C_7} > 0 \tag{3.175}$$

showing that the instantaneous poles of the system are always real. Finally, we can use the fact that the discriminant is such that

$$\sqrt{\Delta} < \frac{1}{(r_1||r_2)C_6} + \frac{1}{(R_p||r_2)C_7} \tag{3.176}$$

to prove that the instantaneous poles of the system are always negative, which is expected since the system is composed of passive elements. The instantaneous poles of the system are then given as

$$p_1 = -\frac{1}{2}\left(\frac{1}{(r_1||r_2)C_6} + \frac{1}{(R_p||r_2)C_7}\right) + \frac{1}{2}\sqrt{\left(\frac{1}{(r_1||r_2)C_6} + \frac{1}{(R_p||r_2)C_7}\right)^2 - 4\frac{r_1 + r_2 + R_p}{r_1 r_2 R_p C_6 C_7}} \tag{3.177}$$

for the pole with lower damping and

$$p_2 = -\frac{1}{2}\left(\frac{1}{(r_1||r_2)C_6} + \frac{1}{(R_p||r_2)C_7}\right) - \frac{1}{2}\sqrt{\left(\frac{1}{(r_1||r_2)C_6} + \frac{1}{(R_p||r_2)C_7}\right)^2 - 4\frac{r_1 + r_2 + R_p}{r_1 r_2 R_p C_6 C_7}} \tag{3.178}$$

for the pole with higher damping.

Empirical analysis

In this section, we simulate the full system, including the linear stage described in Bogason [2018]. However, we focus the analysis of the system instantaneous poles on the nonlinear section alone, as the two stages are decoupled by an operational amplifier. Additionally, the highly damped instantaneous poles of the system generally occurs in the nonlinear stage as the two diodes switch from blocking to passing in the presence of transients.

We run a rising step function of amplitude 100 mV through a backward Euler model of the system, and find an optimized $\alpha = 0.0424$ for a sampling rate of $f_s = 44.1\,\text{kHz}$ ($T_s \approx 22.67\,\mu\text{s}$). We then simulate the response of a bilinear transform model and the α-transform model to a 10 ms-long rectangular pulse, as suggested in Bogason [2018]. The intensity of the pulse is set at 100 mV. For comparison, we also show the result of a high resolution simulation obtained using the Matlab numerical solver `ode15s` where the continuous-time input voltage is set as the linear interpolation of the discrete-time pulse. The output current i_O of these models is shown in Fig. 3.22.

From the plotted responses, we see that the bilinear transform model widely overshoots the response of the "analog" model on the upward step of the pulse. That error actually creates a second issue on the downward step of the pulse, where due to the distorted state of the model variables, the increase in damping of the poles creates a second spurious increase in the output current that is not qualitatively present in the "analog" model response. A better understanding of

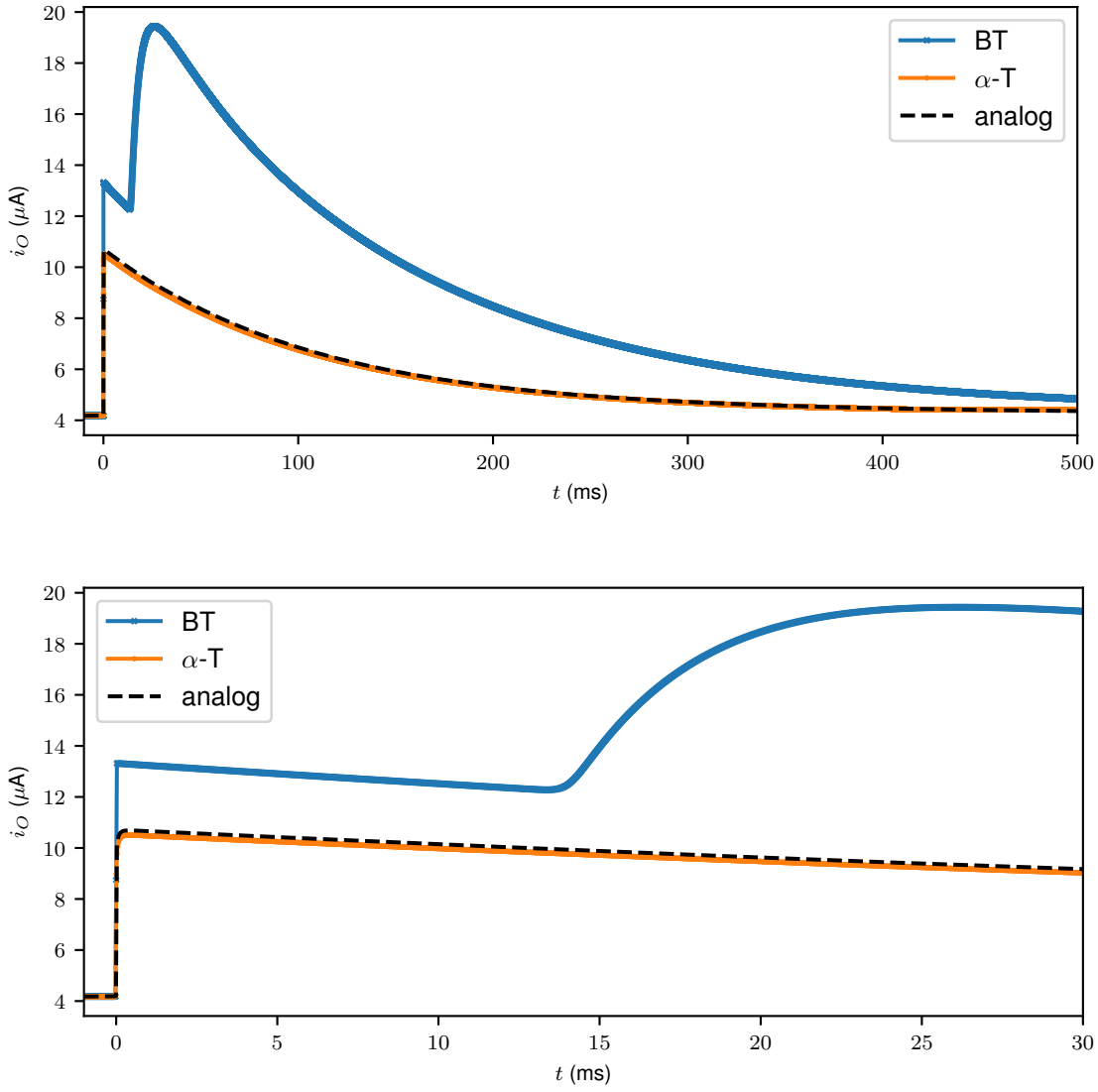


Figure 3.22: Output current i_O for models of the DOD FX-25 envelope follower for a 10 ms input pulse at 100 mV. We compare a bilinear transform model (blue), an optimized α-transform model for $\alpha \approx 0.0424$ (blue) at sampling rate $f_s = 44.1$ kHz ($T_s \approx 22.67$ µs), and a high-resolution "analog" model (black) at two different time scales.

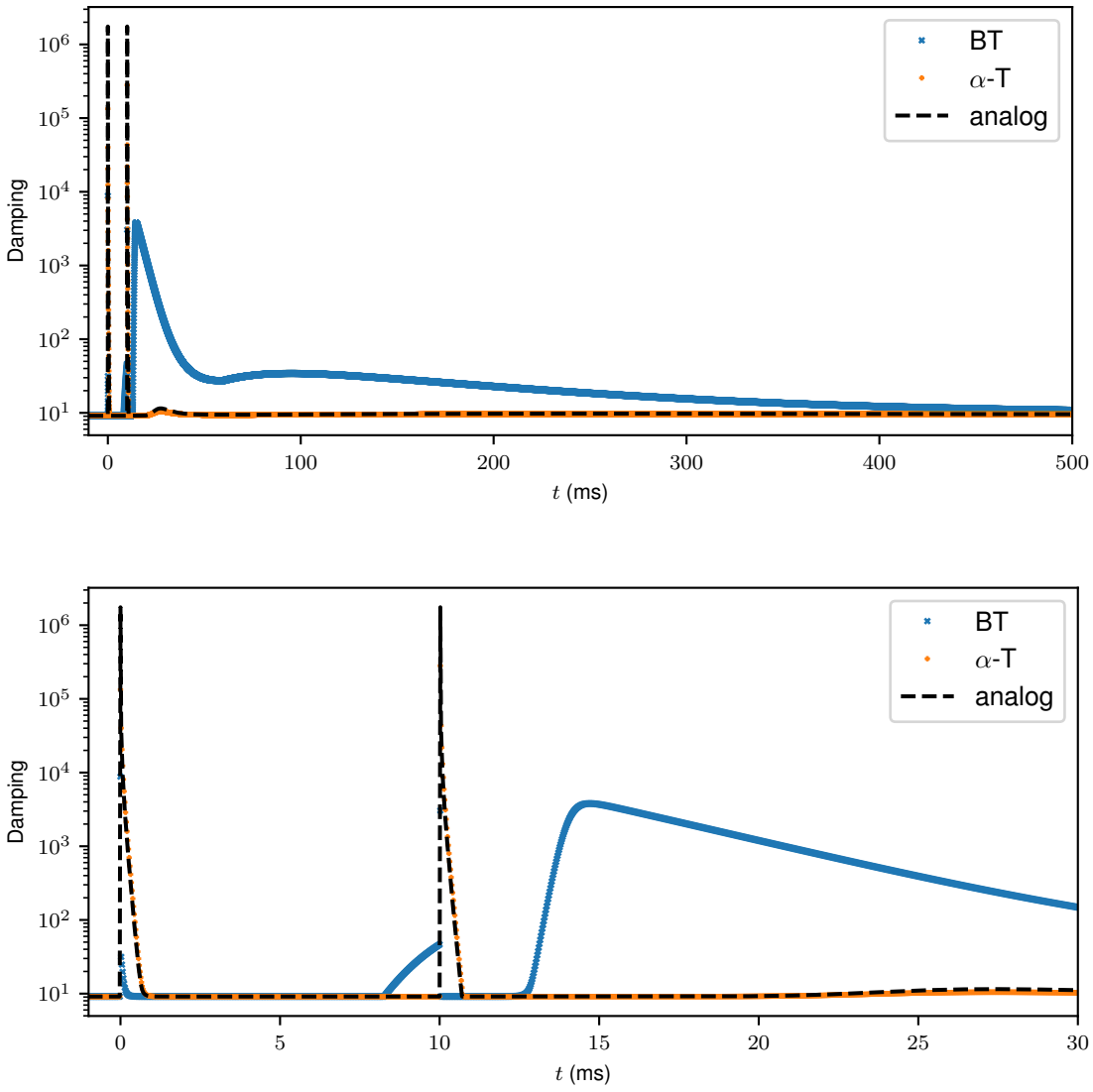


Figure 3.23: Damping estimate of the instantaneous pole with higher damping for models of the DOD FX-25 envelope follower for a 10 ms input pulse at 100 mV. We compare a bilinear transform model (blue), an optimized α-transform model for $\alpha \approx 0.0424$ (blue) at sampling rate $f_s = 44.1$ kHz ($T_s \approx 22.67$ µs), and a high-resolution "analog" model (black) at two different time scales.

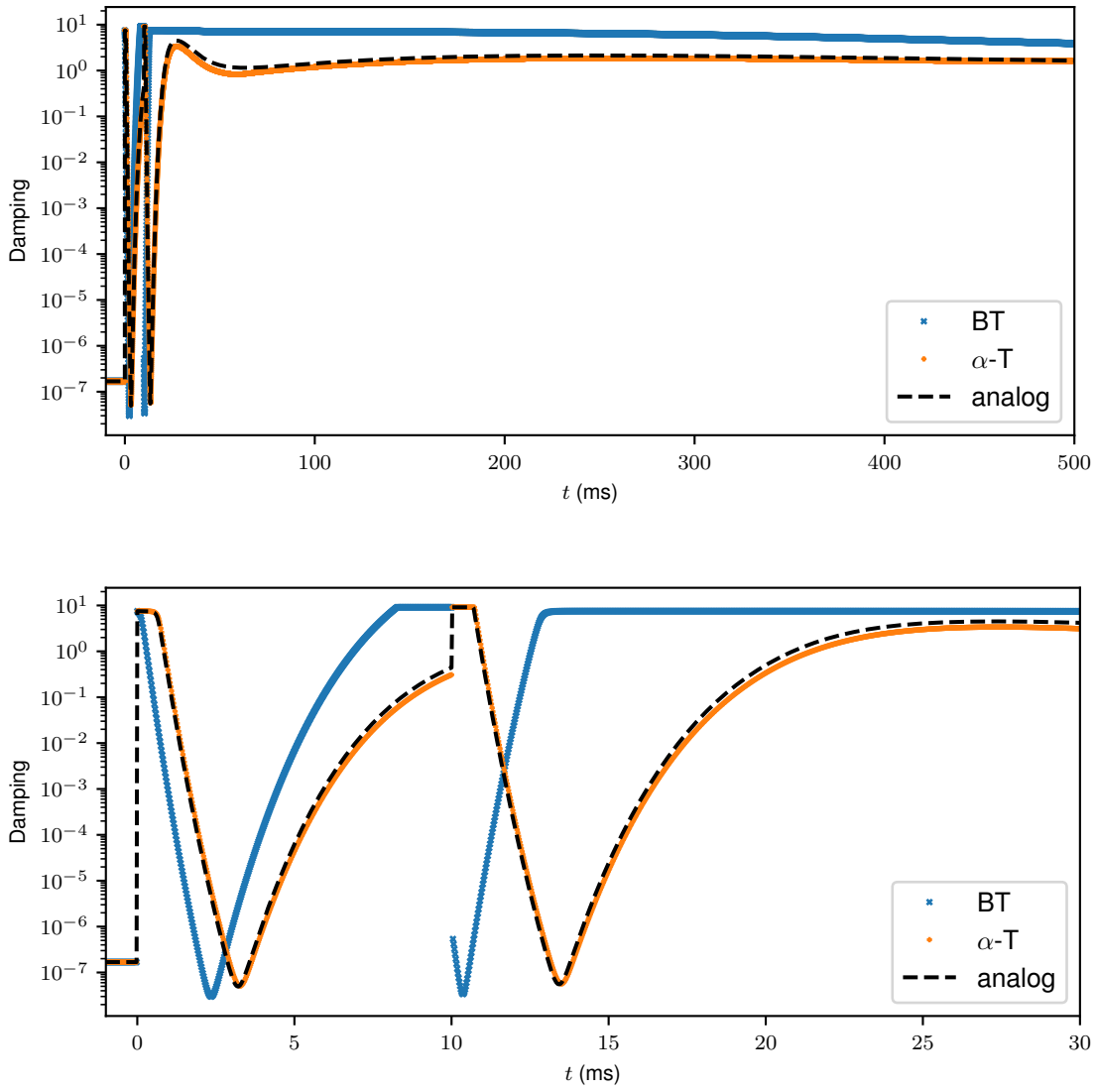


Figure 3.24: Second instantaneous pole estimate for models of the DOD FX-25 envelope follower for a 10 ms input pulse at 100 mV. We compare a bilinear transform model (blue), an optimized α-transform model for $\alpha \approx 0.0424$ (blue) at sampling rate $f_s = 44.1\,\text{kHz}$ ($T_s \approx 22.67\,\mu\text{s}$), and a high-resolution "analog" model (black) at two different time scales.

the source of that distortion can be observed in the location estimates of the instantaneous poles with higher damping as shown in Fig. 3.23. We see how the instantaneous pole for the bilinear transform fails to follow the "analog" one, resulting in the spurious behavior in the output. As for the second pole shown in Fig. 3.24, its general trend is qualitatively correct, but its timing is distorted by the amplitude distortion in the model. On the other hand, the α-transform approach manages to closely follow the target pole trajectories and qualitatively match the expected system output. Again, as in the case of the pulse shaper, the poles are generally slightly under-damped but without significant consequences on the model accuracy. Some of that distortion could be mitigated through the use of the free parameter in the parametric α-transform. On the other hand, using the $\alpha\beta$-transform would be much riskier, as we see that the second pole has very small damping at its rest equilibrium (see samples before $t = 0$) so that setting β any higher than 1 would likely result in an unstable model.

3.10.3 Keio Mini Pops 7 bass drum voice circuit

Circuit description

The Mini Pops 7 (MP-7) [Keio Electronic Laboratory Corporation 1966a,b] is another early analog drum machine from the 1960s, as it was released in 1966 by the Keio Electronic Laboratory Corporation (now Korg Inc.). It features 20 different drum sounds. Here we focus on the bass drum voice circuit shown in Fig. 3.25. This circuit converts a short positive pulse into a oscillating waveform with a decaying envelope.

The pulse is delivered through a diode. The diode becomes conducting on the positive transient of the pulse, allowing for the quasi-instantaneous charging of capacitor C_1 close to the pulse voltage value. Once the pulse ends, the diode becomes blocking and disconnects the source branch from the circuit as long as the capacitor C_1 has a charge. The capacitor C_1 then slowly discharges its energy in the rest of the circuit, acting as a vanishing source. The LC tank formed by the inductor L and the capacitor C_2 creates an oscillation close to the LC resonance frequency at $1/\sqrt{LC_2}$ (here about 65 Hz) whose amplitude decays as the energy in capacitors C_1 dissipates in the various circuit resistors. The capacitor C_3 and series resistor R_3 and R_4 create a (coupled) output low-pass filtering effect which shift only slightly the resonant frequency of the circuit. Additionally, the small residual voltage created across the diode due to the voltage changes in the circuit creates a small oscillation in the circuit resonant frequency and introducing a small timbral coloration.

In the actual circuit, the resistor R_4 is a potentiometer acting as a voltage divider. Since it only changes the measured output voltage by a multiplicative gain, we will only consider the output voltage to be across the full resistor R_4.

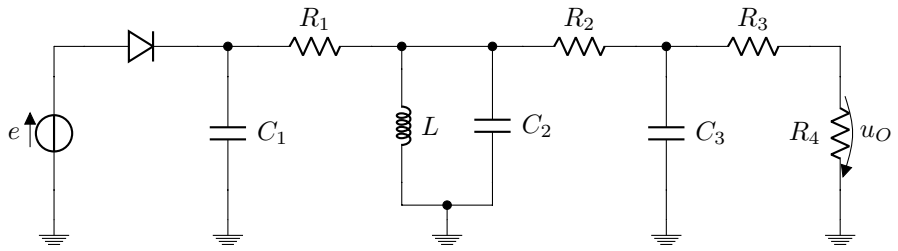


Figure 3.25: Keio MP-7 bass drum voice circuit.

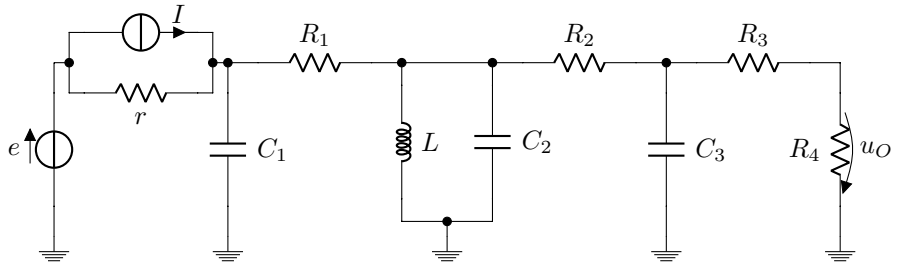


Figure 3.26: Keio MP-7 bass drum voice circuit with diode companion model.

Circuit analysis

For numbers, elements are labeled from left to right and from top to bottom. Except for diodes (for which we follow the diode direction), all voltages are taken from left node to right node and from top node to bottom node

The circuit state equations for the four state variables (branch voltages v_{C_1}, v_{C_2}, v_{C_3} across respectively capacitors C_1, C_2, C_3 and branch current i_L through inductor L) can be derived quickly by simple analysis as

$$\dot{u}_{C_1} = -\frac{v_{C_1} - v_{C_2}}{C_1 R_1} + \frac{1}{C_1} f_D(e(t) - v_{C_1}), \tag{3.179}$$

$$\dot{u}_{C_2} = \frac{v_{C_1}}{R_1 C_2} - \frac{v_{C_2}}{(R_1||R_2)C_2} + \frac{v_{C_3}}{R_2 C_2} - \frac{i_L}{C_2}, \tag{3.180}$$

$$\dot{u}_{C_3} = \frac{v_{C_2}}{R_2 C_3} - \frac{v_{C_3}}{(R_2||(R_3+R_4))C_3}, \tag{3.181}$$

$$\dot{i}_L = \frac{v_{C_2}}{L}. \tag{3.182}$$

Here again, we use the diode companion model to form the linearized circuit equations and analyze the instantaneous poles of the system. The companion model leads to the circuit displayed

in Fig. 3.26. The linearized equations are then given by

$$\begin{bmatrix} \dot{u}_{C_1} \\ \dot{u}_{C_2} \\ \dot{u}_{C_3} \\ \dot{i}_L \end{bmatrix} = \begin{bmatrix} -\frac{1}{(R1||r)C_1} & \frac{1}{R_1C_1} & 0 & 0 \\ \frac{1}{R_1C_2} & -\frac{1}{(R_1||R_2)C_2} & \frac{1}{R_2C_2} & -\frac{1}{C_2} \\ 0 & \frac{1}{R_2C_3} & -\frac{1}{(R_2||(R_3+R_4))C_3} & 0 \\ 0 & \frac{1}{L} & 0 & 0 \end{bmatrix} \begin{bmatrix} v_{C_1} \\ v_{C_2} \\ v_{C_3} \\ i_L \end{bmatrix} + \begin{bmatrix} \frac{1}{rC_1} & \frac{1}{C_1} \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} e \\ I \end{bmatrix}. \quad (3.183)$$

The instantaneous poles are then found as the eigenvalues of the matrix

$$\begin{bmatrix} -\frac{1}{(R1||r)C_1} & \frac{1}{R_1C_1} & 0 & 0 \\ \frac{1}{R_1C_2} & -\frac{1}{(R_1||R_2)C_2} & \frac{1}{R_2C_2} & -\frac{1}{C_2} \\ 0 & \frac{1}{R_2C_3} & -\frac{1}{(R_2||(R_3+R_4))C_3} & 0 \\ 0 & \frac{1}{L} & 0 & 0 \end{bmatrix}. \quad (3.184)$$

Empirical instantaneous pole analysis

Unfortunately, the eigenvalues of the matrix in Eq. (3.184) cannot be simply expressed analytically. Upon empirical examination, we find that, out of the 4 poles, we can roughly make the following segregation: 2 (complex) poles correspond to the system decaying oscillations (the bass drum "pitch") mostly through the exchange of energy between the capacitor C_2 and the inductor L, 2 real pole controls the loading and unloading of energy in the circuit mainly through the charging and discharging of capacitors C_1 on the input side, and capacitor C_3 on the output side.

We run a rising step function of amplitude 2 V through a backward Euler model of the system, and find an optimized $\alpha = 0.0162$ for a sampling rate of $f_s = 44.1$ kHz ($T_s \approx 22.67$ µs). We then simulate the response of a bilinear transform model and the α-transform model to a 1 ms-long rectangular pulse. The intensity of the pulse is set at 2 V. For comparison, we also show the result of a high resolution simulation obtained using the Matlab numerical solver `ode15s` where the continuous-time input voltage is set as the linear interpolation of the discrete-time pulse. The output current u_O of these models is shown in Fig. 3.27.

From the plotted responses, we see that all models produce the expected decaying oscillation corresponding to the bass drum tone. However, we also see that the bilinear transform model widely overshoots the response of the "analog" model on the upward step of the pulse, so that the waveform has a much higher amplitude than expected in the real-world circuit. A better understanding of the source of that distortion can be observed in the damping estimates of the instantaneous pole dampings with highest damping as shown in Fig. 3.28. We see how the instantaneous pole for the bilinear transform fails to follow the "analog" one, resulting in the spurious behavior in the output. On the other hand, the α-transform approach manages to more closely follow the expected system output, thanks to the improved control of the stiffer pole. Again, as in the case of the pulse shaper, the other poles are slightly distorted, creating some small error in the response, but

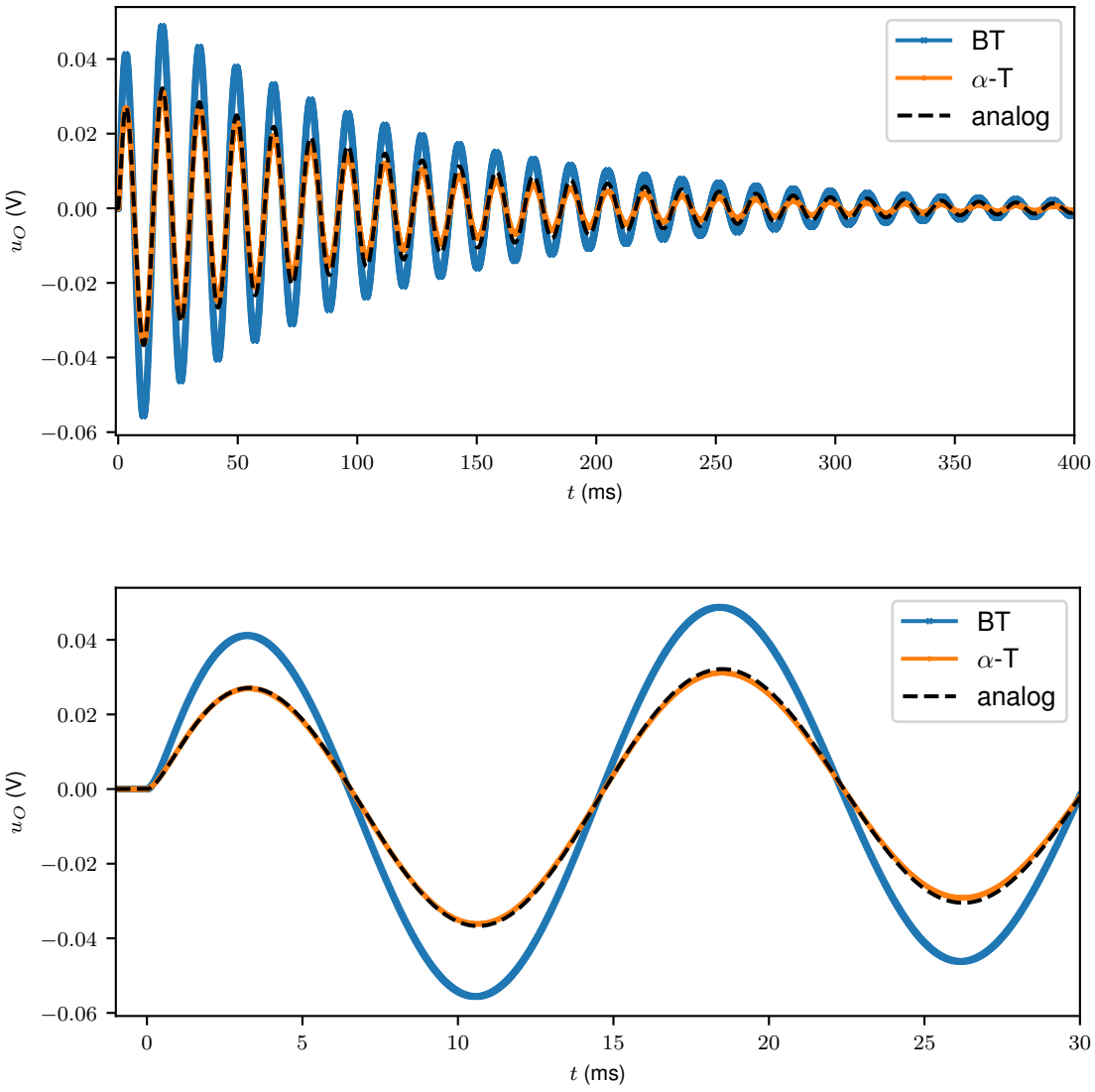


Figure 3.27: Output current u_O for models of the Keio MP-7 bass drum voice for a 10 ms input pulse at 100 mV. We compare a bilinear transform model (blue), an optimized α-transform model for $\alpha = 0.0162$ (blue) at sampling rate $f_s = 44.1$ kHz ($T_s \approx 22.67$ µs), and a high-resolution "analog" model (black) at two different time scales.

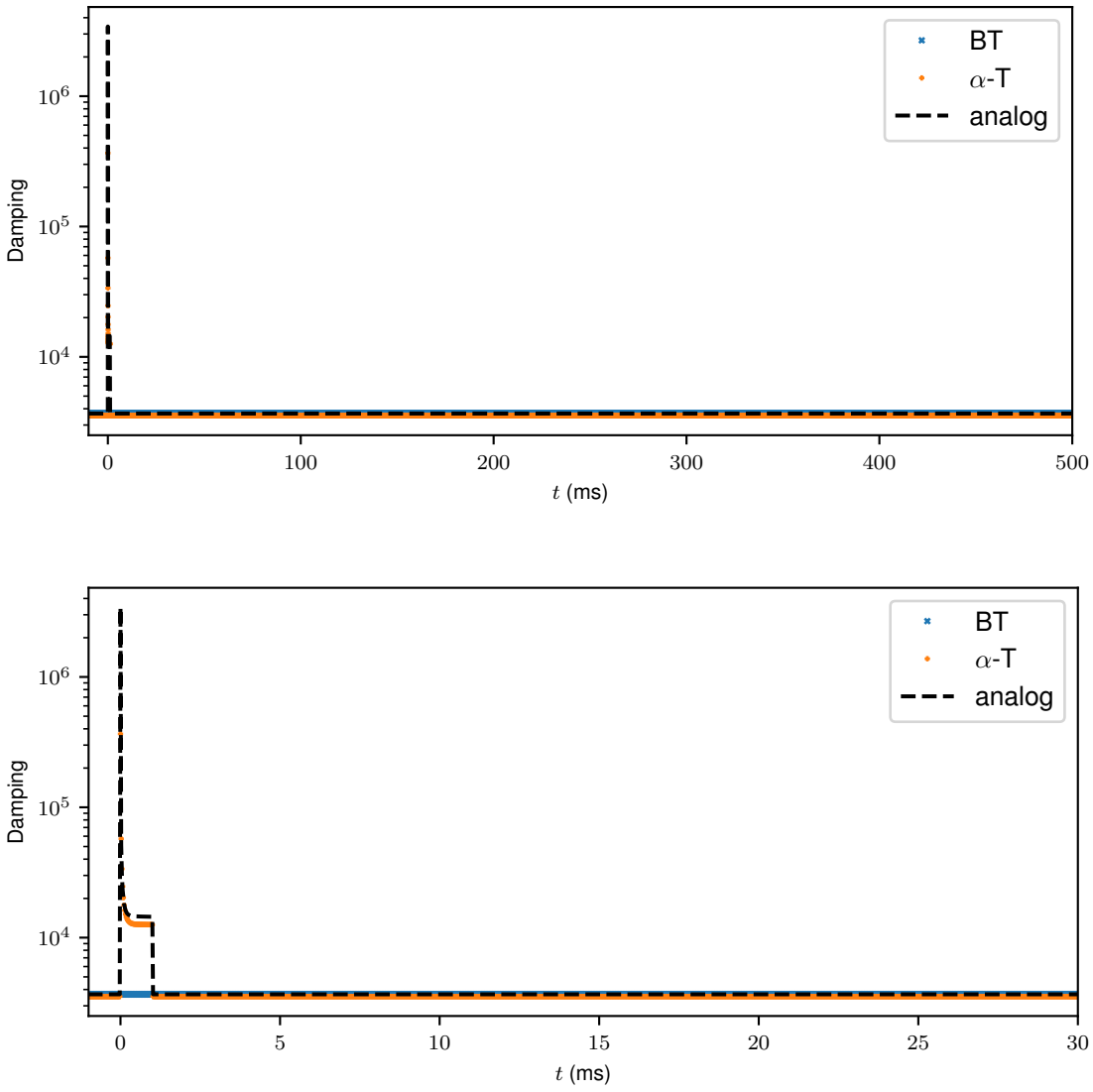


Figure 3.28: Damping estimate of the instantaneous pole with highest damping for models of the Keio MP-7 bass drum voice for a 10 ms input pulse at 100 mV. We compare a bilinear transform model (blue), an optimized α-transform model for $\alpha \approx 0.0162$ (blue) at sampling rate $f_s = 44.1$ kHz ($T_s \approx 22.67$ µs), and a high-resolution "analog" model (black) at two different time scales.

without significant consequences on the model accuracy. In particular, we see how the resonating poles of the circuit are somewhat over-damped, and shifted a bit lower in frequency. Some of that distortion could be mitigated through the use of the free parameter in the parametric α-transform or the $\alpha\beta$-transform to improve the match around these complex poles.

3.11 Conclusion

In this chapter, we discussed the theory and application of discretization methods based on Möbius transformations. In particular, we discussed the implications of the transformation parameters in terms of pole mapping between the s-plane and the z-plane and the qualitative way it could affect the resulting behavior of models based on such transform. Despite the limited amount of control these methods offer, as allowed by their 3 free parameters, they provide a straightforward way to design compact models, often with reasonable accuracy. As a practical example, we discussed how such methods could be used to mitigate the poor behavior observed when using the typical standard and parametric bilinear transform-based methods. One example is the use of the so-called α-transform where a free parameter can be set to improve such behavior, for example using the proposed damping-monotonicity preserving condition. This general approach is applied to three different nonlinear lumped audio circuits successfully building models with much improved behavior and accuracy. In particular, this approach successfully addresses the issue arising in bilinear transform models of circuits with diodes with abrupt changes between the passing and blocking regimes.

Chapter 4

Elementwise numerical methods for linear lumped system modeling

In this chapter, we discuss a novel framework for the design of optimized elementwise numerical methods for the case of linear lumped system modeling. Such systems are ubiquitous in the field of audio and so a variety of methods have been designed depending on the level of knowledge regarding the internal structure of the original system (see Sec. 0.2 for references). Our method is based on the principles of numerical discretization in the context of physical modeling. This presents the benefit of preserving the structure of the underlying system by manipulating its physical equations. At its core, our method relies on applying elementwise numerical methods[1] across the system, i.e., have a different discretization applied to the local equations relative to each element. As such, it differs from the more typical approach where a system-wide method is applied. In particular, we examine the case where all the elements are discretized using a prototype method with one or more free parameters (e.g., the parametric bilinear transform, the α-transform), and where the free parameters are set individually for each element. Our method can be set to preserve the computational cost of the initial discretization while resulting in a more accurate model by measures to be described. Many of the results described here have been previously presented in [Germain and Werner 2017a] and [Germain and Werner 2017b].

[1] Note that in Germain and Werner [2017a] and Germain and Werner [2017b], we used the term *differentiated numerical methods* to refer to this method. We introduce this new terminology of *elementwise numerical methods* here due to the potential ambiguity of that original terminology.

4.1 Linear lumped system modeling

4.1.1 General system description

Linear lumped systems can generally be described through an ordinary differential equation governing the relationship between input $e(t)$ and output $o(t)$ quantities following Eq. (3.3), i.e.,

$$\sum_{k=0}^{K} a_k \frac{d^k}{dt^k} o(t) = \sum_{l=0}^{L} b_l \frac{d^l}{dt^l} e(t), \ \forall t \in \mathbb{R}, \tag{4.1}$$

which can be equivalently expressed in the frequency domain using the Laplace transform (see Eq. (3.3)) as

$$\left[\sum_{k=0}^{K} a_k s^k\right] o(s) = \left[\sum_{l=0}^{L} b_l s^l\right] e(s), \ \forall s \in \mathbb{C}. \tag{4.2}$$

Linear systems are then traditionally described through their input-output transfer function (see Eq. (3.5)), as

$$H(s) = \frac{o(s)}{e(s)} = \frac{\sum_{l=0}^{L} b_l s^l}{\sum_{k=0}^{K} a_k s^k}. \tag{4.3}$$

Such a definition can easily be extended to the case where multiple output quantities o_m ($m \in \{1, \dots, M\}$) are considered and grouped in a vector $\boldsymbol{o} = [o_1 \cdots o_M]^\mathsf{T}$. We then have a system of ordinary differential equations with an equation per output quantity, leading to a vector transfer function $\mathbf{H} = [H_1 \cdots H_M]^\mathsf{T}$, with $H_m(s) = o_m(s)/e(s)$ ($m \in \{1, \dots, M\}$).

4.1.2 Electrical network equivalence

In this chapter, we will solely discuss linear system descriptions based on electrical circuit theory. The reason why is that lumped systems in other contexts (e.g., mechanical systems) can be equivalently described in identical terms [Firestone 1933, Olson 1943, Werner 2016] by swapping elementary mechanical components (e.g., mass, spring, damper) for their equivalent elementary electrical components (e.g., resistor, capacitor, inductor).

4.1.3 Physical modeling and gray-box modeling

As mentioned in Sec. 0.2, there are three main avenues for modeling in the available literature: black-box, white-box and gray-box modeling. Thanks to the ubiquity of linear systems, all three methods have received considerable attention in the literature. We refer the reader to the introduction for a general discussion on the topic and references

For the modeling of linear systems whose internal structure is known (e.g., electrical circuits), the equations coefficients b_l and a_k in Eqs. (4.1), (4.2) and (4.3) can generally be expressed as

complicated functions of the various physical parameters in the system (e.g., component resistance and/or capacitance). It is also generally possible to capture the system structure using other formalisms than the transfer function as we will see later in this chapter. The remaining problem is to devise a process generating a computer model from that representation in continuous time to a representation in discrete-time.

White-box modeling essentially relies on s-to-z mapping approaches. First-order mappings of the kind we described in Ch. 3 are a popular approach due to their ability to match the order of complexity of the discrete-time model and the continuous-time system. The standard bilinear transform is by far the most common method, due to its property of mapping the imaginary axis in the s-plane to the unit circle in the z-plane. Without ancillary information regarding the system, the mappings are generally chosen to be equivalent to a numerical method (e.g., the trapezoidal method for the standard bilinear transform) as alternative mappings do not provide as many blind guarantees to be well-behaved at all sampling rates. However, the distortion introduced in the frequency response by these methods generally requires oversampling the discrete-time system to achieve reasonable wideband accuracy. More complex approaches also exist to generate candidate mappings (e.g., the fractional bilinear transform [Pei and Hsu 2008]) generally associated with the need to formulate a "stabilization" approach as these often result in discrete-time systems with unstable discrete-time poles.

For linear systems more than for nonlinear systems, gray-box modeling approaches (see Sec. 0.2) are a popular alternative to design more accurate systems without oversampling. Methods such as the impulse invariant method, the step invariant method or the matched Z-transform [Rabiner and Gold 1975, Parks and Burrus 1987, Oppenheim and Schafer 2009] leverage the knowledge of the pole locations of the system to fit a discrete-time prototype with the same order. In the case of the invariant methods, their goal is to match as well as possible the time-domain response of the system to a particular input signal (e.g., impulse, step) by optimizing the discrete-time zero locations of the model when the pole locations are set to have identical properties (i.e., damping and frequency, see Sec. 3.2) for the continuous-time system and its model. The matched Z-transform matched the properties of both the zero and the pole locations between model and system, with only the model gain left as a free parameter to optimize, which often leads to a less accurate system than the invariant methods [Parks and Burrus 1987]. Another thread of research has focused on ways to modify the discretization of the Laplace ideal differentiator s, the Laplace ideal integrator $1/s$ and their powers (i.e., s^n and $1/s^n$) by introducing free parameters to be optimized [Šekara and Stojić 2005, Šekara 2006, Al Alaoui 2008] and better approximate the integrator (or differentiator) while conserving its order, often by following an error criterion based on rectangular integration, i.e., of the form

$$\text{error} = \sum_{n=0}^{N} \left(\int_{nT_s}^{(n+1)T_s} (h(t) - h_d[n]) \right)^2 \tag{4.4}$$

where h is the impulse response of the continuous-time system (i.e., the inverse Laplace transform of the transfer function in Eq. (4.3)) and h_d the impulse response of its model at sampling period T_s. Other designs can be attempted using the general literature on higher-order (e.g., Simpson's rule) digital integrator and differentiator design [Le Bihan 1993, Papamarkos and Chamzas 1996, Tseng 2006, Al Alaoui 2011], but these generally offer no guarantee of stability when applied to arbitrary systems rather than the pure integrator and differentiator operator they were designed for. As a result, designs based on first-order methods (e.g., standard bilinear transform, backward Euler method and any combination of the two) remain the main approach to obtain ready-to-use stable models. In the context of audio, optimized approaches also exist in the case of equalization filter design [Välimäki and Reiss 2016], but they generally only target simple linear systems (e.g., biquad filters) and/or transfer function types (e.g., low-pass filters, high-pass filters, shelf filters, one-resonance filters). Simpler heuristics, such as warping compensation using the parametric bilinear transform (see Secs. 3.7.3 and 3.8.2) are also popular in the audio field for simple systems with a single salient feature of interest in their frequency response, but they generalize poorly to systems with more complex features.

In the novel approach described here, we aim at directly optimizing the transfer function of the discrete-time system, while staying as close as possible to the fundamental physical structure of the continuous-time device we are modeling, in particular by explicitly isolating the individual contribution of all the system elements to the transfer function. In particular, this approach leverages the property that each element[2] of our system is generally described by a *local* equation between the element's local quantities (e.g., Ohm's law which describes the relation between the branch voltage across and current through a given electrical resistor). The physics of the system then generally dictates how each element's local quantities interacts with other variables in the rest of the circuit (e.g., Kirchhoff laws [Vlach 2002]) in order to form the *global* behavior equations of the system. This implies that we can look at the modeling of the system from the perspective of these two scales which will be important in our approach in this chapter.

4.2 Electrical network conventions

4.2.1 Frequency domain description

In this chapter, all the systems of interest are linear. As such, they can equivalently be described in the time domain or in the frequency domain. When dealing with dynamic linear elements (e.g., capacitor, inductor), the circuit analysis methods used below require using a frequency domain formulation so that all the equations describing the electrical component behavior are linear equations

[2]Note that, in this chapter, we will generally associate *elements* with individual electric components for conciseness. However, the framework straightforwardly extends to the case where elements correspond to subsystems, i.e., grouping of two or more components, in the system. For example, we could imagine grouping the mass and the spring forming a resonance in a mechanical system.

in terms of branch voltages and currents. To do so, we represent each quantity by its Laplace transform, and consider all linear ordinary differential equations in the Laplace domain.

4.2.2 Electrical variables

Branch/node variables

The typical way to describe quantities in an electrical system is using *Kirchhoff variables* (i.e., *voltage* and *current* variables) associated with circuit branches and nodes. More precisely, these variables can be categorized as:

- *node voltages* (denoted here as v), i.e., electrical potential at a given node in the circuit *with reference to* an arbitrary common reference electrical potential shared between all node voltages (often associated with a given circuit node, the *datum* node), or
- *branch voltages* (denoted here as u), i.e., voltage (or potential difference) across a branch measured as the difference between the node voltages at each terminal node of the branch, or
- *branch currents* (denoted here as i), i.e., a measure of the flow rate of electrical charge through a given branch.

In that context, the behavior of a given circuit is governed by the behavior of the individual elements (generally described through *constitutive equations* between the voltages and currents of the branches linked to that particular element) and their connections (generally described through the *Kirchhoff laws*).

Port variables and wave variable formalism

An alternative representation of circuits is based on the decomposition of the circuit in filters (that we'll denote "wave-domain filters") processing so-called *wave variables*. That theory emerged from the traveling wave formulation of lumped electrical elements introduced and described in Belevitch [1962]. Each filter is characterized by a mathematical relation between the wave variables associated with the one or more *ports*. These wave variables are divided into "input" *incident wave* variables (denoted as a) and "output" *reflected wave* variables (denoted as b). These quantities are mostly used in the context of wave digital filter models [Fettweis 1986, Werner et al. 2015c, Werner 2016].

In that context, the behavior of a given circuit is governed by the behavior of the individual filters (described through *scattering equations* between incident and reflected wave variables for the port of that particular filter) and the connections between ports belonging to different filters (described through *port connection equations*).

Associated with the wave variable formalism is the concept of *port resistance*. The port resistance $\overline{R}$ is usually found as a strictly positive quantity associated with each given port in the literature,

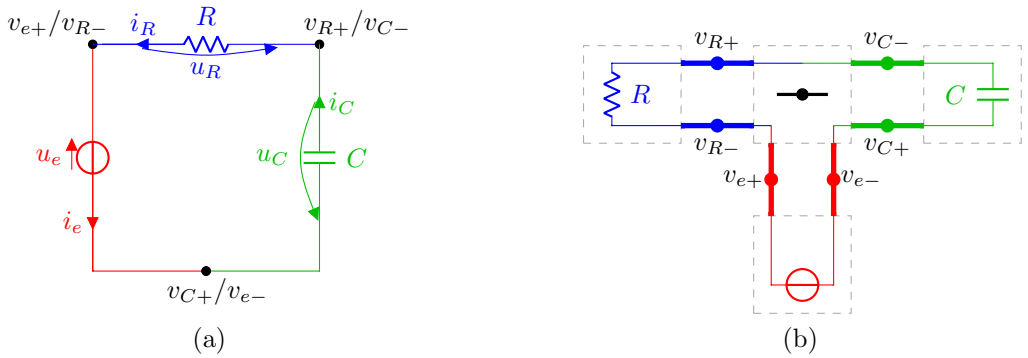


Figure 4.1: Illustration of branch/node and port variable equivalency for an RC series system with (a) the circuit schematic and (b) the equivalent connection tree (see [Werner 2016]) where the circuit elements and adapters are connected with the ports highlighted as bold lines. The colors (respectively red, blue and green) illustrate the correspondence between each circuit branch (respectively the voltage source, resistor and capacitor branches) and their equivalent ports in the connection tree. We also see how each port relates to the network terminal nodes of each branch. The dashed boxes in the connection tree corresponds to elements and adapters, which are connected at their respective ports. The center box in the connection tree corresponds to a series adaptor (see [Werner 2016]).

though the formalism only requires a non-zero quantity. Among other things, the port resistance is used to associated the wave variables a and b of the port to a *port voltage* u and a *port current* i. For the sake of simplicity, but without loss of generality, we will here only discuss *voltage waves* (so called because they are dimensionally equivalent to voltages) defined as [Werner 2016]

$$a = u + \overline{R}i \quad \text{and} \quad b = u - \overline{R}i, \tag{4.5}$$

or alternatively

$$u = \frac{a+b}{2} \quad \text{and} \quad i = \frac{a-b}{2\overline{R}}. \tag{4.6}$$

The port resistance also has an influence on forming the equations describing an electrical circuit network which will be discussed in Sec. 4.4.2.

4.2.3 Branch/node and port variable equivalency

In the context of circuit modeling, it is generally possible to associate every port to an equivalent circuit branch, so that each branch voltage (respectively current) corresponds to one or more port voltages (resp. currents). Consequently, a port also corresponds to the two network terminal nodes forming that branch. In Fig. 4.1, we see how that equivalency can be observed in a simple RC series circuit. For more information regarding the formalism of the connection tree, we refer the reader to

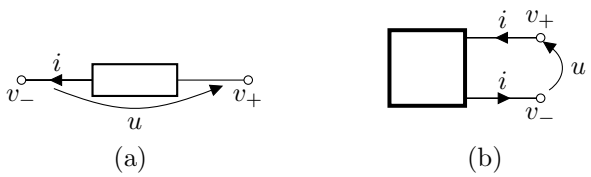


Figure 4.2: Current and voltage polarity definition for (a) circuit branches and (b) circuit ports.

Werner [2016].

4.2.4 Branch/port current and voltage polarity

For branch current and voltage polarity definitions, we follow the convention illustrated in Fig. 4.2, meaning that we measure the current flow through the branch (respectively port) in the opposite direction than the one we measure the voltage across the branch (resp. port).

4.3 Network description

To describe the network forming an electrical circuit, we need equations linking the circuit variables across different locations in the circuit. As for the variables, we can derive this description either from the perspective of the branch/node formalism, or the port formalism.

4.3.1 Branch/node equations

In the branch/node formalism, the network is formed through the connection of various branches at the level of each node. Then, the Kirchhoff laws [Vlach 2002] describe how the network structure links the various branch voltages, node voltages and branch currents across that network.

Kirchhoff's current law (KCL)

The Kirchhoff's current law (KCL) is derived from the principle of conservation of electric charge at every network node. It corresponds to the fact that the algebraic sum of currents in an electric network flowing into a node is equal to zero. If $\mathbf{A}$ is the incidence matrix describing the topology of that network, that condition can be written as [Vlach 2002]

$$\mathbf{A}\boldsymbol{i}(s) = \mathbf{0}_N \tag{4.7}$$

where N is the number of distinct nodes in the circuit.

As such, the Kirchhoff's current law gets us a comprehensive description of the interaction between the branch currents across the entire circuit.

Kirchhoff's voltage law (KVL)

The Kirchhoff's voltage law (KVL) is derived from the principle of conservation of energy at every network branch. It corresponds to the fact that the directed sum of electrical voltages around a closed loop is zero. If $\mathbf{A}$ is the incidence matrix describing the topology of that network, that condition can be written as [Vlach 2002]

$$\boldsymbol{u}(s) - \mathbf{A}^{\mathsf{T}}\boldsymbol{v}(s) = \mathbf{0}_B \tag{4.8}$$

where B is the number of distinct branches in the circuit.

As such, the Kirchhoff's voltage law gets us a comprehensive description of the interaction between the branch and node voltages across the entire circuit.

4.3.2 Port connection equations

In the port formalism, the network is formed by connecting pairs of port together. These connections form equations following the rule that, for two connected ports, the reflected wave quantity from each port becomes the incident wave of the other. Additionally, a well-formed electrical network generally means that all the ports for all the wave-domain filters describing the circuit are connected to another one. In mathematical terms, if we gather the vectors of all incident wave variables $\boldsymbol{a}$ and reflected wave variables $\boldsymbol{b}$ across the entire network (with corresponding elements on each vector corresponding to the same port, the port connection equations are written as

$$\boldsymbol{a}(s) = \mathbf{V}\boldsymbol{b}(s), \tag{4.9}$$

where $\mathbf{V}$ is a permutation matrix (i.e., a matrix such that strictly one element on each row and on each column is equal to one, and all other entries are uniformly zero) and a symmetrical matrix (i.e., $\mathbf{V} = \mathbf{V}^{\mathsf{T}}$). Having the entry $\mathbf{V}_{k,l} = \mathbf{V}_{l,k} = 1$ means that the pair of ports labeled k and l in the network are connected to each other. Additionally, $\mathbf{V}$ is by definition orthogonal (i.e., $\mathbf{V}^{-1} = \mathbf{V}^{\mathsf{T}}$).

Another requirement of the various port connections is that each connected pair of ports is set with the same port resistance[3], i.e.,

$$\mathbf{V}_{k,l} = 1 \Rightarrow \overline{R}_k = \overline{R}_l. \tag{4.10}$$

[3]Note that, in alternative definitions of the port formalism found in the literature [D'Angelo and Välimäki 2012, Werner 2016, Werner et al. 2016a], the port resistance matching requirement is absent, but is replaced by the need to add scattering equations in order to convert the wave variables between any two ports with mismatched port resistance. However, that operation can be interpreted as the addition of an two-port adapter in-between these two ports to perform that scattering operation. The two ports of that intermediary adapter are then set so that their respective port resistance matches the port resistance of their connected port, becoming mathematically equivalent to the port resistance matching equivalent. Subsequently, that scattering adapter can also be absorbed into either of the two elements and/or adapters it is connected to. As a result, our definition requiring that port resistances match between two ports is completely equivalent to these alternative definitions and results in no loss of generality.

This requirement is equivalent to requiring that each connected pair of ports is associated to identical port voltages and currents following the relation between port wave and Kirchhoff variables as described in Eqs. (4.5) and (4.6).

4.4 Linear circuit description

For a given network, the remaining equations needed to describe an electrical circuit behavior express the influence of the different elements composing the circuit on the various electrical variables with the proper equations. Here too, the approach differs depending on if a branch/node or a port formalism is chosen.

4.4.1 Branch formalism

Branch constitutive equations

In branch formalism, the final block of equations needed to analyze the circuit is the branch constitutive equations of the various elements in the circuit. Since we are focused on the case of circuits with linear elements and one-port resistive sources, the constitutive equations of all elements can be grouped to form a system of linear equations between branch voltage vector $\boldsymbol{u}$ and current vector $\boldsymbol{i}$ following the form

$$\mathbf{P}(s)\boldsymbol{u}(s) - \mathbf{Q}(s)\boldsymbol{i}(s) = \boldsymbol{e}(s), \tag{4.11}$$

where $\boldsymbol{e}(s)$ is an electrical *source* vector.

Sparse tableau formulation

Sparse tableau formulation forms a linear equation system describing the behavior of the various branch currents, branch voltages and node voltages in an electrical circuit described in the branch formalism. This is done by essentially grouping the equations as described in the KCL equations (Eq. (4.7)), KVL equations (Eq. (4.8)) and the element constitutive equations (Eq. (4.11)).

As mentioned earlier in Sec. 4.2.2, the node voltages are always defined up to an arbitrary constant offset (common to all node voltages in the system), so that one of the node voltages is always a *dependent variable* (i.e., it can always be expressed as a linear combinations of other node voltages in the circuit). Then, we select one node as *datum node* and arbitrarily set to 0. As a consequence, we can omit it from the system, and thus remove the corresponding column from the incidence matrix $\mathbf{A}$ in order to obtain a description with only *independent* (Kirchhoff) circuit variables.

The resulting system of equations is then written as

$$\left[\begin{array}{c|c|c} \mathbf{I}_B & \mathbf{0}_{B,B} & -\mathbf{A}^\mathsf{T} \\ \hline \mathbf{P}(s) & -\mathbf{Q}(s) & \mathbf{0}_{B,N} \\ \hline \mathbf{0}_{N,B} & \mathbf{A} & \mathbf{0}_{N,N} \end{array}\right] \left[\begin{array}{c} \boldsymbol{u}(s) \\ \hline \boldsymbol{i}(s) \\ \hline \boldsymbol{v}(s) \end{array}\right] = \left[\begin{array}{c} \mathbf{0}_B \\ \hline \boldsymbol{e}(s) \\ \hline \mathbf{0}_N \end{array}\right]. \tag{4.12}$$

We arrange the branch order so that branches related to a given element are grouped together, so that the constitutive equation matrices $\mathbf{P}$ and $\mathbf{Q}$ are diagonal by block, with the size of each block associated with the number of ports of the corresponding underlying element.

Reduced tableau formulation

The KVL equations as given in Eq. (4.8) allow to solve the sparse tableau formulation for node voltages $\boldsymbol{v}$ and branch currents $\boldsymbol{i}$ first and then deduce the corresponding branch voltages $\boldsymbol{u}$ as

$$\boldsymbol{u}(s) = \mathbf{A}^\mathsf{T}\boldsymbol{v}(s). \tag{4.13}$$

Using Eq. (4.13), we can then formulate a reduced formulation of the system in Eq. (4.12) as

$$\left[\begin{array}{c|c} -\mathbf{Q}(s) & \mathbf{P}(s)\mathbf{A}^\mathsf{T} \\ \hline \mathbf{A} & \mathbf{0}_{N,N} \end{array}\right] \left[\begin{array}{c} \boldsymbol{i}(s) \\ \hline \boldsymbol{v}(s) \end{array}\right] = \left[\begin{array}{c} \boldsymbol{e}(s) \\ \hline \mathbf{0}_N \end{array}\right]. \tag{4.14}$$

4.4.2 Port formalism

As mentioned earlier in Sec. 4.2.2, when using a formulation based on wave variables, a linear electrical network described using the port formalism is composed of various wave-domain filters with a certain number of ports each. A common distinction among wave-domain filters is between *elements* which correspond to electrical components (e.g., resistor, capacitor, inductor, diode, transformer), and *adaptors* which correspond to circuit connections in the circuit (e.g., series and parallel connections), though hybrid elements are also encountered. Typical elements have one or two ports, while adaptors have 2 or more. The most common type of adaptors are the 3-port series and the 3-port parallel adaptor, due to the fact that an adaptor with larger number of ports (also called *macroadaptors* [Sarti and De Sanctis 2009]) can often be factored into inter-connected networks of these 2 types of adaptor.

Filter scattering equations

The Kth element is associated with an incident wave vector $\boldsymbol{a}_K$ and a reflected vector $\boldsymbol{b}_K$ whose dimension is equal to the number of ports for that element. The element behavior is then described by a set of scattering wave equations describing the relation between the incident wave vector $\boldsymbol{a}_K$

and the reflected wave vector $\boldsymbol{b}_K$. In the case of a linear network, such relation is expressed by the linear relation

$$\boldsymbol{b}_K(s) = \mathbf{S}_K(s;\overline{\mathbf{R}}_K)\boldsymbol{a}_K(s) + \mathbf{T}_K(s;\overline{\mathbf{R}}_K)\boldsymbol{e}_K(s) \tag{4.15}$$

where $\boldsymbol{e}_K$ is a source term of arbitrary dimension, $\mathbf{S}_K$ is the element scattering matrix, and $\mathbf{T}_K$ is the element source scattering matrix. As notated, in wave variable formalism, the matrices $\mathbf{S}_K$ and $\mathbf{T}_K$ should be described as a function of a set of port resistances (one port resistance per port) gathered in the diagonal matrix $\overline{\mathbf{R}}_K$. These port resistances are the same used to relate port wave variables and Kirchhoff quantities as described in Eq. (4.5).

We can concatenate the behavior of all the elements to form the scattering wave equations of the full network as

$$\boldsymbol{b}(s) = \mathbf{S}(s;\overline{\mathbf{R}})\boldsymbol{a}(s) + \mathbf{T}(s;\overline{\mathbf{R}})\boldsymbol{e}(s). \tag{4.16}$$

Port-based system description

The complete system can then be described fully using the port connection equations (see Eq. (4.9)) and the wave-domain filter scattering equations (see Eq. (4.16)), so that the equations describing the system are expressed as

$$\left[\begin{array}{c|c} \mathbf{I}_P & -\mathbf{V} \\ \hline -\mathbf{S}(s;\mathbf{R}) & \mathbf{I}_P \end{array}\right] \left[\begin{array}{c} \boldsymbol{a}(s) \\ \hline \boldsymbol{b}(s) \end{array}\right] = \left[\begin{array}{c} \mathbf{0}_P \\ \hline \mathbf{T}(s;\mathbf{R})\boldsymbol{e}(s) \end{array}\right]. \tag{4.17}$$

Reduced port-based system description

In the typical case where all ports in the circuit have been connected to another one, the system described in Eq. (4.17) is overdetermined since every single incident wave quantity is necessarily equal to another reflected wave quantity through the port connection equations (see Eq. (4.9)). The system can then be reduced to solve either solely for the incident wave quantities as

$$\big(\mathbf{V} - \mathbf{S}(s;\overline{\mathbf{R}})\big)\boldsymbol{a}(s) = \mathbf{T}(s;\overline{\mathbf{R}})\boldsymbol{e}(s) \tag{4.18}$$

or solely for the reflected wave quantities as

$$\big(\mathbf{I}_B - \mathbf{S}(s;\overline{\mathbf{R}})\mathbf{V}\big)\boldsymbol{b}(s) = \mathbf{T}(s;\overline{\mathbf{R}})\boldsymbol{e}(s). \tag{4.19}$$

Branch-to-node voltage conversion

As indicated in Sec. 4.2.3, the typical way of representing circuits through the branch or the port formalism provides a straightforward relation between branch voltages (respectively currents) and port voltages (resp. currents). However, in many applications, the output of the system is better

expressed as a function of various node voltages in the circuit (e.g., differences between various node voltages and the datum node).

The Kirchhoff voltage law as expressed in Eq. (4.8) provides us with a straightforward way to convert node voltages into branch voltages. However, the inverse transformation that we need, from branch voltages $\boldsymbol{u}$ to node voltages $\boldsymbol{v}$, is less well-defined. Indeed, as the system is under-determined by design, and we have to choose between an infinity of pseudoinverse matrices of the incidence matrix $\mathbf{A}$. An obvious solution is to use the Moore–Penrose inverse [Ben-Israel and Greville 2003] defined as

$$\mathbf{A}^{+} = \mathbf{A}^{\mathsf{T}}(\mathbf{A}\mathbf{A}^{\mathsf{T}})^{-1} \tag{4.20}$$

with $\boldsymbol{v} = \mathbf{A}^{+}\boldsymbol{u}$. One downside from this approach is that the matrix $\mathbf{A}^{+}$ is dense and hard to relate to the circuit structure.

In more traditional circuit analysis, node voltages are generally found as integer sums of branch voltages through path searches between the datum node and the node voltages of interest. This approach can be described more systematically through the search of integer generalized inverses for incidence matrices (see Bevis et al. [1981] and Cederbaum [1984]). Finding such generalized inverses relies on well-known graph single-source path-finding algorithms—e.g., Bellman–Ford algorithm [Cormen et al. 2009], Djikstra algorithm [Cormen et al. 2009], A* algorithm [Hart et al. 1968], Floyd–Warshall algorithm [Cormen et al. 2009]—as it rely on exploiting the structure of a spanning tree over the network graph described by the incidence matrix $\mathbf{A}$ and with all graph edge weights equal to 1.

4.4.3 Formalism equivalency

In the typical context where port voltages and currents correspond identically to branch voltages and currents, we can form the equivalency between the branch and the port formalism expressing the scattering equations (see Eq. (4.16)) using quantities from the constitutive equations (see Eq. (4.11)) as

$$\boldsymbol{b}(s) = (\mathbf{Q} + \mathbf{P}\overline{\mathbf{R}})^{-1}(\mathbf{Q} - \mathbf{P}\overline{\mathbf{R}})\boldsymbol{a}(s) + 2(\mathbf{Q} + \mathbf{P}\overline{\mathbf{R}})^{-1}\overline{\mathbf{R}}\boldsymbol{e}(s) \tag{4.21}$$

so that we have the relationships

$$\mathbf{S}(s;\overline{\mathbf{R}}) = (\mathbf{Q} + \mathbf{P}\overline{\mathbf{R}})^{-1}(\mathbf{Q} - \mathbf{P}\overline{\mathbf{R}}) \tag{4.22}$$

$$\mathbf{T}(s;\overline{\mathbf{R}}) = 2(\mathbf{Q} + \mathbf{P}\overline{\mathbf{R}})^{-1}\overline{\mathbf{R}} \tag{4.23}$$

assuming $\mathbf{Q} + \mathbf{P}\overline{\mathbf{R}}$ invertible.

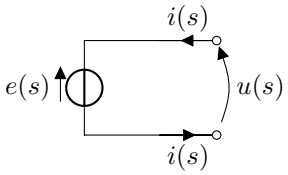
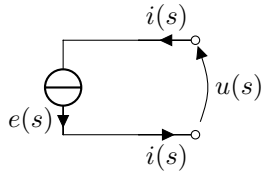
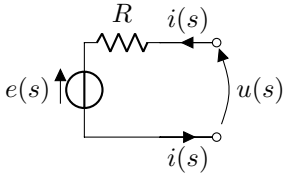
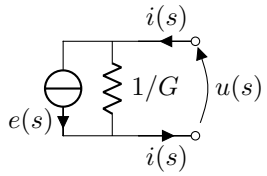
Source type	Element	Constitutive equation	Scattering equation
Ideal voltage		$u(s) = e(s)$	$b(s) = -a(s) + 2\,e(s)$
Ideal current		$i(s) = e(s)$	$b(s) = a(s) - 2\overline{R}\,e(s)$
Resistive voltage		$u(s) - Ri(s) = e(s)$	$b(s) = \frac{R-\overline{R}}{R+\overline{R}}\,a(s) + \frac{2\overline{R}}{R+\overline{R}}\,e(s)$
Resistive current		$-Gu(s) + i(s) = e(s)$	$b(s) = \frac{1-G\overline{R}}{1+G\overline{R}}\,a(s) - \frac{2\overline{R}}{1+G\overline{R}}\,e(s)$

Table 4.1: Constitutive and scattering equations of one-port resistive sources.

4.5 Components of interest

For all types of elements we expect to find in audio circuits of interest, we recall here:

- their constitutive equations, meaning the relationships among the Kirchhoff (i.e., voltage and current) quantities associated with the circuit branches they involve, and
- their scattering equations, meaning the relationship between the incident and the reflected wave quantities associated with the circuit ports they involve.

4.5.1 One-port resistive sources

Electrical networks describing a linear system allow for a single independent resistive source. In this document, we limit ourselves to the case of one-port sources as it covers all of our cases of interest. With a single one-port resistive source and linear electrical components otherwise, the constitutive

equations (see Eq. (4.11)) can be split as

$$\mathbf{P}_L(s)\boldsymbol{u}_L(s) - \mathbf{Q}_L(s)\boldsymbol{i}_L(s) = \mathbf{0}_L \qquad \text{(linear components)} \tag{4.24a}$$

$$P_e u_e(s) - Q_e i_e(s) = e(s) \qquad \text{(one-port resistive source)} \tag{4.24b}$$

where the voltage–current pair (u_e, i_e) corresponds the pair associated with the single source branch. The source element also serves the purpose of input when considering the input-output transfer function of the system. We can similarly isolate the port corresponding to the source in the circuit scattering equations (see Eq. (4.16)) as

$$\boldsymbol{b}_N(s) = \mathbf{S}_N(s;\overline{\mathbf{R}}_N)\boldsymbol{a}_N(s) \qquad \text{(linear passive ports)} \tag{4.25a}$$

$$b_e(s) = S_e(\overline{R}_e)a_e(s) + T_e(\overline{R}_e)e(s) \qquad \text{(one-port resistive source)} \tag{4.25b}$$

Eq. (4.25b) can be expressed equivalently using Eq. (4.24b) as

$$b_e(s) = \underbrace{\frac{Q_e - P_e\overline{R}_e}{Q_e + P_e\overline{R}_e}}_{S_e(\overline{R}_e)}\, a(s) + \underbrace{\frac{2\overline{R}_e}{Q_e + P_e\overline{R}_e}}_{T_e(\overline{R}_e)}\, e(s) \tag{4.26}$$

assuming $Q_e + P_e\overline{R}_e \neq 0$.

Typical types of sources found in audio circuits are the following:

- *Ideal voltage source*: An ideal voltage source is characterized by the constitutive equation

$$u_e(s) = e(s) \tag{4.27}$$

  and the scattering equation

$$b_e(s) = -a_e(s) + 2\,e(s). \tag{4.28}$$

- *Ideal current source*: An ideal current source is characterized by the constitutive equation

$$i_e(s) = e(s) \tag{4.29}$$

  and the scattering equation

$$b_e(s) = a_e(s) - 2\overline{R}_e\,e(s). \tag{4.30}$$

- *Resistive voltage source*: A resistive voltage source, corresponding to an ideal voltage source in series with a resistor of resistance $R > 0$ (and conductance $G = 1/R$), is characterized by the constitutive equation

$$u_e(s) - Ri_e(s) = e(s) \tag{4.31}$$

and the scattering equation

$$b_e(s) = \frac{R - \overline{R}_e}{R + \overline{R}_e}\, a_e(s) + \frac{2\overline{R}_e}{R + \overline{R}_e}\, e(s). \tag{4.32}$$

- *Resistive current source*: A resistive current source, corresponding to an ideal voltage source in parallel with a resistor with conductance $G > 0$ (and resistance $R = 1/G$) is characterized by the constitutive equation

$$-Gu_e(s) + i_e(s) = e(s) \tag{4.33}$$

and the scattering equation

$$b_e(s) = \frac{1 - G\overline{R}_e}{1 + G\overline{R}_e}\, a_e(s) - \frac{2\overline{R}_e}{1 + G\overline{R}_e}\, e(s). \tag{4.34}$$

The equations for these sources are summarized in Tab. 4.1. With a single one-port resistive source in the circuit, we can simplify the various systems of circuit equations, with the sparse tableau formulation (Eq. (4.12)) as

$$\left[\begin{array}{cc|cc|c} \mathbf{I}_L & \mathbf{0}_L & \mathbf{0}_{L,L} & \mathbf{0}_L & -\mathbf{A}_L^\mathsf{T} \\ \mathbf{0}_L^\mathsf{T} & 1 & \mathbf{0}_L^\mathsf{T} & 0 & -\mathbf{A}_e^\mathsf{T} \\ \hline \mathbf{P}_L(s) & \mathbf{0}_L & -\mathbf{Q}_L(s) & \mathbf{0}_L & \mathbf{0}_{L,N} \\ \mathbf{0}_L^\mathsf{T} & P_e & \mathbf{0}_L^\mathsf{T} & -Q_e & \mathbf{0}_N^\mathsf{T} \\ \hline \mathbf{0}_{N,L} & \mathbf{0}_N & \mathbf{A}_L & \mathbf{A}_e & \mathbf{0}_{N,N} \end{array}\right] \left[\begin{array}{c} \boldsymbol{u}_L(s) \\ \boldsymbol{u}_e(s) \\ \hline \boldsymbol{i}_L(s) \\ \boldsymbol{i}_e(s) \\ \hline \boldsymbol{v}(s) \end{array}\right] = \left[\begin{array}{c} \mathbf{0}_L \\ 0 \\ \hline \mathbf{0}_L \\ e(s) \\ \hline \mathbf{0}_N \end{array}\right], \tag{4.35}$$

the reduced tableau formulation (Eq. (4.14)) as

$$\left[\begin{array}{cc|c} -\mathbf{Q}_L(s) & \mathbf{0}_L & \mathbf{P}_L(s)\mathbf{A}_L^\mathsf{T} \\ \mathbf{0}_L^\mathsf{T} & -Q_e & P_e\mathbf{A}_e^\mathsf{T} \\ \hline \mathbf{A}_L & \mathbf{A}_e & \mathbf{0}_{N,N} \end{array}\right] \left[\begin{array}{c} \boldsymbol{i}_L(s) \\ \boldsymbol{i}_e(s) \\ \hline \boldsymbol{v}(s) \end{array}\right] = \left[\begin{array}{c} \mathbf{0}_L \\ e(s) \\ \hline \mathbf{0}_N \end{array}\right], \tag{4.36}$$

the port-based description (Eq. (4.17)) as

$$\left[\begin{array}{cc|cc} \mathbf{I}_N & \mathbf{0}_N & -\mathbf{V}_{N,N} & -\mathbf{V}_{N,e} \\ \mathbf{0}_N^\mathsf{T} & 1 & -\mathbf{V}_{e,N} & 0 \\ \hline -\mathbf{S}_N(s;\overline{\mathbf{R}}_N) & \mathbf{0}_N & \mathbf{I}_N & \mathbf{0}_N \\ \mathbf{0}_N^\mathsf{T} & -S_e(s;\overline{R}_e) & \mathbf{0}_N^\mathsf{T} & 1 \end{array}\right] \left[\begin{array}{c} \boldsymbol{a}_N(s) \\ a_e(s) \\ \hline \boldsymbol{b}_N(s) \\ b_e(s) \end{array}\right] = \left[\begin{array}{c} \mathbf{0}_N \\ 0 \\ \hline \mathbf{0}_N \\ T_e(s;\overline{R}_e)e(s) \end{array}\right] \tag{4.37}$$

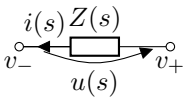
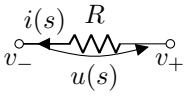
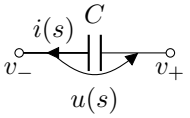
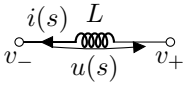
Type	Symbol	Impedance $Z(s)$	Admittance $Y(s)$	Scattering $S(s)$
Generic		-	-	$\dfrac{Z(s)-\overline{R}}{Z(s)+\overline{R}}$
Resistor		R	G	$\dfrac{R-\overline{R}}{R+\overline{R}}$
Capacitor		$\dfrac{1}{Cs}$	Cs	$\dfrac{1-\overline{R}Cs}{1+\overline{R}Cs}$
Inductor		Ls	$\dfrac{1}{Ls}$	$\dfrac{Ls-\overline{R}}{Ls+\overline{R}}$

Table 4.2: Characteristics of typical one-port linear elements.

and the reduced port-based description (Eqs. (4.18) and (4.19)), as

$$\begin{bmatrix} \big(\mathbf{V}_{N,N} - \mathbf{S}_N(s;\overline{\mathbf{R}}_N)\big) & \mathbf{V}_{N,e} \\ \mathbf{V}_{e,N} & -S_e(s;\overline{R}_e) \end{bmatrix} \begin{bmatrix} \boldsymbol{a}_N(s) \\ a_e(s) \end{bmatrix} = \begin{bmatrix} \mathbf{0}_P \\ T_e(s;\overline{R}_e)e(s) \end{bmatrix} \tag{4.38}$$

for the incident wave quantities, or as

$$\begin{bmatrix} \big(\mathbf{I}_N - \mathbf{S}_N(s;\overline{\mathbf{R}}_N)\mathbf{V}_{N,N}\big) & -\mathbf{S}_N(s;\overline{\mathbf{R}}_N)\mathbf{V}_{N,e} \\ S_e(s;\overline{R}_e)\mathbf{V}_{e,N} & 1 \end{bmatrix} \begin{bmatrix} \boldsymbol{b}_N(s) \\ b_e(s) \end{bmatrix} = \begin{bmatrix} \mathbf{0}_P \\ T_e(s;\overline{R}_e)e(s) \end{bmatrix} \tag{4.39}$$

for the reflected wave quantities.

4.5.2 One-port linear elements

One-port elements correspond to one branch voltage–current pair (u, i). In the case of linear elements, the constitutive equations correspond to the linear relation

$$u(s) - Z(s)i(s) = 0 \tag{4.40}$$

where Z corresponds to the *impedance* of the one-port linear element. An alternative formulation is as

$$Y(s)u(s) - i(s) = 0 \tag{4.41}$$

where Y corresponds to the *admittance* of the element.

These elements can also be described by their wave-domain scattering equation as

$$b(s) = S(s)a(s) \tag{4.42}$$

where S can be expressed as a function of Z as

$$S(s) = \frac{Z(s) - \overline{R}}{Z(s) + \overline{R}} \tag{4.43}$$

or equivalently as a function of Y as

$$S(s) = \frac{1 - \overline{R}Y(s)}{1 + \overline{R}Y(s)}. \tag{4.44}$$

Below we describe the three typical one-port linear elements found in audio circuits:

- *Resistor*: A resistor with *resistance* $R > 0$ (and *conductance* $G = 1/R > 0$) correspond to the constitutive equation

  $$u(s) - Ri(s) = 0 \tag{4.45}$$

  or equivalently

  $$Gu(s) - i(s) = 0 \tag{4.46}$$

  and

  $$b(s) = \frac{R - \overline{R}}{R + \overline{R}}\, a(s) = \frac{1 - \overline{R}G}{1 + \overline{R}G}\, a(s) \tag{4.47}$$

  so that $Z(s) = R$, $Y(s) = G$ and $S(s) = \frac{R-\overline{R}}{R+\overline{R}} = \frac{1-\overline{R}G}{1+\overline{R}G}$.

- *Capacitor*: A capacitor with *capacitance* $C > 0$ correspond to the constitutive equation

  $$Cs\, u(s) - i(s) = 0 \tag{4.48}$$

  and

  $$b(s) = \frac{1 - \overline{R}Cs}{1 + \overline{R}Cs} a(s) \tag{4.49}$$

  so that $Z(s) = 1/(Cs)$, $Y(s) = Cs$ and $S(s) = \frac{1-\overline{R}Cs}{1+\overline{R}Cs}$.

- *Inductor*: An inductor with *inductance* $L > 0$ correspond to the constitutive equation

  $$u(s) - Ls\, i(s) = 0 \tag{4.50}$$

  and

  $$b(s) = \frac{Ls - \overline{R}}{Ls + \overline{R}} a(s) \tag{4.51}$$

so that $Z(s) = Ls$, $Y(s) = 1/(Ls)$ and $S(s) = \frac{Ls-\overline{R}}{Ls+\overline{R}}$.

4.6 Element discretization

For linear systems, the process of discretizing the entire system can often be equated to discretizing the individual elements as discrete-time elements and then interfacing them in the same structure as their continuous-time counterparts. In particular, this is generally the case when using systematic discretization procedure such as numerical discretization or integration methods (e.g., bilinear transform, trapezoidal integration).

4.6.1 Discrete-time constitutive equations

As such, elements are characterized by a discrete-time constitutive equations expressed in the z domain as

$$\mathbf{P}_d(z)\boldsymbol{u}_d(z) - \mathbf{Q}_d(z)\boldsymbol{i}_d(z) = 0 \tag{4.52}$$

where the matrices $\mathbf{P}_d$ and $\mathbf{Q}_d$ are derived using a systematic procedure from the matrices $\mathbf{P}$ and $\mathbf{Q}$ from Eq. (4.24a).

Discrete-time scattering equations

Similarly, the element discretization produces a discrete-time scattering equation between discrete-time wave variables $\boldsymbol{b}_d$ and $\boldsymbol{a}_d$ in the z domain as

$$\boldsymbol{b}_d(z) = \mathbf{S}_d(z;\overline{R})\boldsymbol{a}_d(z) \tag{4.53}$$

where the scattering is expressed as

$$\mathbf{S}_d(z) = \left(\mathbf{P}_d(z) - \mathbf{Q}_d(z)\overline{\mathbf{R}}^{-1}\right)^{-1}\left(\mathbf{P}_d(z) + \mathbf{Q}_d(z)\overline{\mathbf{R}}^{-1}\right) \tag{4.54}$$

assuming $\left(\mathbf{P}_d(z) - \mathbf{Q}_d(z)\overline{\mathbf{R}}^{-1}\right)$ is invertible.

4.6.2 Discretization using s-to-z mappings

A lot of systematic discretization methods can be expressed as mapping from the s domain to the z domain $s = \mathcal{T}(z;\theta)$ parametrized by some set of free parameters θ (see Sec. 4.7.8 for more details).

In that case, the element equations can be easily found through the substitutions

$$\begin{aligned} \mathbf{P}_d(z) &= \mathbf{P}(\mathcal{T}(z;\theta)) \\ \mathbf{Q}_d(z) &= \mathbf{Q}(\mathcal{T}(z;\theta)) \\ \mathbf{S}_d(z) &= \mathbf{S}(\mathcal{T}(z;\theta)) \end{aligned} \tag{4.55}$$

Generic one-port linear element

For a generic one-port linear element of impedance $Z(s)$ (Eq. (4.40)), the discretized element impedance is given by $Z_d(z) = Z(\mathcal{T}(z))$ and the scattering coefficient is given by

$$S_d(z) = S(\mathcal{T}(z;\theta)) = \frac{Z(\mathcal{T}(z;\theta)) - \overline{R}}{Z(\mathcal{T}(z;\theta)) + \overline{R}}, \tag{4.56}$$

such that we have the incident-reflected wave relationship at the port given by

$$b_d(z) = S_d(z)a_d(z). \tag{4.57}$$

Resistor

In the case of a resistor, the transfer function and scattering coefficient of the discretized element stay the same, as the impedance of a resistor does not depend on the variable s. Consequently, $Z_d(z) = R$ and

$$S_d(z) = \frac{R - \overline{R}}{R + \overline{R}}. \tag{4.58}$$

Capacitor

For a capacitor, the discretized impedance becomes $Z_d(z) = 1/C\mathcal{T}(z;\theta)$ and the discretized scattering coefficient becomes

$$S_d(z) = \frac{1 - \overline{R}C\mathcal{T}(z;\theta)}{1 + \overline{R}C\mathcal{T}(z;\theta)}. \tag{4.59}$$

Inductor

For an inductor, the discretized impedance becomes $Z_d(z) = 1/C\mathcal{T}(z;\theta)$ and the discretized scattering coefficient becomes

$$S_d(z) = \frac{L\mathcal{T}(z;\theta) - \overline{R}}{L\mathcal{T}(z;\theta) + \overline{R}}. \tag{4.60}$$

4.6.3 Element port adaptation

Element port adaptation is an important concept in wave digital filter formalism as a way to break delay-free loops in the computation in order to form *explicit* scattering equations for a maximum

of wave variables [Werner 2016]. In mathematical terms, this requires setting the diagonal positive definite matrix $\overline{\mathbf{R}}$ such that all elements of matrix $\mathbf{S}_d(z)$ correspond to a filter which does not have a zero-delay feedforward path. Equivalently, this requires setting $\overline{\mathbf{R}}$ such that equation Eq. (4.53) describes a strictly causal system, i.e., that the wave samples $\boldsymbol{b}_d[n]$ depends on the wave samples $\boldsymbol{a}_d[n-m]$ only for $m > 0$.

The result of that adaptation process is to make an adapted element *computable*. Indeed, as wave-domain filter structure form essentially a loop relationship between incident and reflected waves, a strictly causal relation ensures the realizability of the update equations describing the loop.

Generic one-port linear element

Adaptation for a generic one-port linear element is done by setting the port resistance as

$$\overline{R} = \lim_{z\to\infty} Z(\mathcal{T}(z;\theta)) = Z(\mathcal{T}(\infty;\theta)) \tag{4.61}$$

so that the scattering coefficient becomes

$$S_d(z) = \frac{Z(\mathcal{T}(z;\theta)) - Z(\mathcal{T}(\infty;\theta))}{Z(\mathcal{T}(z;\theta)) + Z(\mathcal{T}(\infty;\theta))}. \tag{4.62}$$

Resistor

An adapted resistor element is given by setting the port resistance $\overline{R}$ to the resistor value R, so that the scattering coefficient S_d is zero.

Capacitor

An adapted capacitor element is given by setting the port resistance $\overline{R}$ to the resistor value $1/C\mathcal{T}(\infty;\theta)$, so that the scattering coefficient S_d is given by

$$S_d(z) = \frac{\mathcal{T}(\infty;\theta) - \mathcal{T}(z;\theta)}{\mathcal{T}(\infty;\theta) + \mathcal{T}(z;\theta)}. \tag{4.63}$$

Inductor

An adapted inductor element is given by setting the port resistance $\overline{R}$ to the resistor value $L\mathcal{T}(\infty;\theta)$, so that the scattering coefficient S_d is given by

$$S_d(z) = \frac{\mathcal{T}(z;\theta) - \mathcal{T}(\infty;\theta)}{\mathcal{T}(z;\theta) + \mathcal{T}(\infty;\theta)}. \tag{4.64}$$

We can make the observation that for this discretization process, we will always have the scattering coefficient of an inductor is the opposite of the scattering coefficient of a capacitor.

4.7 Optimization formulation for RLC networks

In this section, we describe how to formulate a transfer function optimization problem in the case of an RLC network, which corresponds to the most frequent scenario of interest in audio applications. RLC networks correspond to electrical circuits composed only of connected resistors, capacitors and inductors.

Our main objective is to build a model whose transfer function maximally approximates the transfer function of the original system in the frequency range $\left[0, \frac{f_s}{2}\right]$ or a subset of it for a given error measure, as discussed below.

4.7.1 Transfer function from voltage–current description

RLC networks can be described by their voltage–current characteristic using the Kirchhoff domain constitutive equations of the different elements, i.e., the resistor (Eq. (4.45)), the capacitor (Eq. (4.48)) and the inductor (Eq. (4.50)). As shown in Eq. (4.41), the voltage–current characteristic of these three elements is summarized by their admittance Y as

$$i(s) = Y(s)u(s). \tag{4.65}$$

Then, as described in Eq. (4.36), our system with source input term $e(s)$ follows the reduced tableau system

$$\left[\begin{array}{cc|c} \mathbf{Y}^{-1}(s) & \mathbf{0}_L & -\mathbf{A}_L^\mathsf{T} \\ \mathbf{0}_L^\mathsf{T} & -Q_e & P_e\mathbf{A}_e^\mathsf{T} \\ \hline \mathbf{A}_L & \mathbf{A}_e & \mathbf{0}_{N,N} \end{array}\right] \left[\begin{array}{c} \boldsymbol{i}_L(s) \\ \boldsymbol{i}_e(s) \\ \hline \boldsymbol{v}(s) \end{array}\right] = \left[\begin{array}{c} \mathbf{0}_L \\ e(s) \\ \hline \mathbf{0}_N \end{array}\right] \tag{4.66}$$

with $\mathbf{Y}$ a diagonal matrix where each diagonal term contains the admittance of one given one-port linear element, so that $\mathbf{Z}^{-1} = \mathbf{Y}$.

For RLC circuits, this equation is generally invertible so that we can solve for all the branch currents and node voltages as

$$\left[\begin{array}{c} \boldsymbol{i}_L(s) \\ \boldsymbol{i}_e(s) \\ \hline \boldsymbol{v}(s) \end{array}\right] = \left[\begin{array}{cc|c} \mathbf{Y}^{-1}(s) & \mathbf{0}_L & -\mathbf{A}_L^\mathsf{T} \\ \mathbf{0}_L^\mathsf{T} & -Q_e & P_e\mathbf{A}_e^\mathsf{T} \\ \hline \mathbf{A}_L & \mathbf{A}_e & \mathbf{0}_{N,N} \end{array}\right]^{-1} \left[\begin{array}{c} \mathbf{0}_L \\ e(s) \\ \hline \mathbf{0}_N \end{array}\right]. \tag{4.67}$$

If necessary, branch voltages can then be deduced using the KVL equations from Eq. (4.13) as explained in Sec. 4.4.1.

In our discussion, we only consider output quantities that are linear combinations of node voltage and branch current quantities, as it corresponds to the most typical case in linear system simulation.

As such, the outputs should be written as

$$\boldsymbol{o}(s) = \mathbf{M}\boldsymbol{v}(s) - \mathbf{N}\boldsymbol{i}(s) = \left[-\mathbf{N}_L \ -N_e \,\middle|\, \mathbf{M}\right] \left[\begin{array}{c} \boldsymbol{i}_L(s) \\ \boldsymbol{i}_e(s) \\ \hline \boldsymbol{v}(s) \end{array}\right]. \tag{4.68}$$

$\mathbf{M}$ and $\mathbf{N}$ describe the relative contributions of the various circuit node voltages and branch currents to each input. For example, the coefficient $M_{k,l}$ indicates how much the voltage of the lth node contributes to the kth output, and the coefficient $N_{k,l}$ how much the current of the lth branch (negatively) contributes to the kth output.

Following that expression, the transfer function of the system (as defined in Eq. (4.3)) is given by

$$\mathbf{H}(s) = \frac{\boldsymbol{o}(s)}{e(s)} = \left[-\mathbf{N}_L \ -N_e \,\middle|\, \mathbf{M}\right] \left[\begin{array}{cc|c} \mathbf{Y}^{-1}(s) & \mathbf{0}_L & -\mathbf{A}_L^\mathsf{T} \\ \mathbf{0}_L^\mathsf{T} & -Q_e & P_e\mathbf{A}_e^\mathsf{T} \\ \hline \mathbf{A}_L & \mathbf{A}_e & \mathbf{0}_{N,N} \end{array}\right]^{-1} \left[\begin{array}{c} \mathbf{0}_L \\ 1 \\ \hline \mathbf{0}_N \end{array}\right] \tag{4.69}$$

where we have split the output contribution of the input branch N_e and the linear element branches $\mathbf{N}_L$.

By using block matrix inversion formulas [Zhang 2005], we get

$$\mathbf{H}(s) = \frac{1}{Q_e + P_e\mathbf{A}_e^\mathsf{T}\mathbf{X}^{-1}(s)\mathbf{A}_e} \left[-\mathbf{N}_L \ -N_e \,\middle|\, \mathbf{M}\right] \begin{bmatrix} \mathbf{Y}(s)\mathbf{A}_L^\mathsf{T}\mathbf{X}^{-1}(s)\mathbf{A}_e \\ -1 \\ \mathbf{X}^{-1}(s)\mathbf{A}_e \end{bmatrix} \tag{4.70}$$

with $\mathbf{X} = \mathbf{A}_L\mathbf{Y}\mathbf{A}_L^\mathsf{T}$, and we obtain the closed-form expression

$$\mathbf{H}(s) = \frac{N_e + \left(\mathbf{M} - \mathbf{N}_L\mathbf{Y}(s)\mathbf{A}_L^\mathsf{T}\right)\left(\mathbf{X}^{-1}(s)\mathbf{A}_e\right)}{Q_e + P_e\mathbf{A}_e^\mathsf{T}\mathbf{X}^{-1}(s)\mathbf{A}_e}. \tag{4.71}$$

Typical types of transfer functions can be solved using this framework:

- *voltage–current driving point transfer function:* In this case, we have $P_e = 1$, $Q_e = 0$, $\mathbf{N} = \mathbf{0}_L$, $N_e = -1$ and $\mathbf{M} = \mathbf{0}_N$, so that the transfer function becomes

$$H(s) = -\frac{1}{\mathbf{A}_e^\mathsf{T}\mathbf{X}^{-1}(s)\mathbf{A}_e} \tag{4.72}$$

- *Current-voltage driving point transfer function:* In this case, we have $P_e = 0$, $Q_e = -1$, $\mathbf{N} = \mathbf{0}_L$, $N_e = 0$ and $\mathbf{M} = \mathbf{A}_e$, so that the transfer function becomes

$$H(s) = -\mathbf{A}_e^\mathsf{T}\mathbf{X}^{-1}(s)\mathbf{A}_e. \tag{4.73}$$

- *Incident-reflected wave driving point transfer function:* If we denote $\overline{R}_e$ the port resistance of the driving/input port, we have $P_e = 1$ and $Q_e = -\overline{R}_e$ (so that $a(s) = e(s)$), as well as $\mathbf{N} = \mathbf{0}_L$, $N_e = \overline{R}_e$ and $\mathbf{M} = \mathbf{A}_e$. Then, the transfer function becomes

$$H(s) = \frac{\mathbf{A}_e^\mathsf{T}\mathbf{X}^{-1}(s)\mathbf{A}_e + \overline{R}_e}{\mathbf{A}_e^\mathsf{T}\mathbf{X}^{-1}(s)\mathbf{A}_e - \overline{R}_e}. \tag{4.74}$$

- *Voltage-to-voltage transfer function on branch l:* In this case, we have $P_e = 1$, $Q_e = 0$, $\mathbf{N} = \mathbf{0}_L$, $N_e = 0$ and $\mathbf{M} = \mathbf{A}_l$, so that the transfer function becomes

$$H(s) = \frac{\mathbf{A}_l^\mathsf{T}\mathbf{X}^{-1}(s)\mathbf{A}_e}{\mathbf{A}_e^\mathsf{T}\mathbf{X}^{-1}(s)\mathbf{A}_e}. \tag{4.75}$$

4.7.2 Transfer function from wave-domain description

Another way of describing the circuit behavior is through the wave-domain description of the various elements composing the circuit. Wave-domain modeling aims at computing the various quantities of the circuit based on a description in terms of incident and reflected wave components (defined following the wave definitions in Eq. (4.5)) computed at the port of various elements/group of elements in the circuit, using scattering relationships following the generic template given in Eq. (4.15).

Network adaptor scattering matrix

One important aspect of the process of defining the behavior of a system expressed in the wave domain is to describe the scattering between the wave quantities at various ports due to the network topology, summarized as a *network adaptor* with ports to which the ports of all the individual elements of the circuit are connected. To find the scattering relationship of this adaptor, we perform the following calculation inspired by the derivation in Werner et al. [2015c]. By definition, the incident wave quantities $\boldsymbol{a}$ and the reflected wave quantities $\boldsymbol{b}$ at the ports of the general network adaptor are expressed as a function of the port voltages $\boldsymbol{u}$ across and port currents $\boldsymbol{i}$ through these ports following

$$\boldsymbol{a} = \boldsymbol{u} + \overline{\mathbf{R}}\boldsymbol{i} \tag{4.76a}$$

$$\boldsymbol{b} = \boldsymbol{u} - \overline{\mathbf{R}}\boldsymbol{i} \tag{4.76b}$$

where $\overline{\mathbf{R}}$ is a diagonal matrix whose diagonal elements correspond to the port resistances of the adaptor ports.

The equations complement the other circuit equations describing the relationships between the branch voltages across, the branch currents through, and the node voltages $\boldsymbol{v}$ at the network adaptor

ports to form the following system of equations

$$\begin{aligned} \boldsymbol{u} - \mathbf{A}^\mathsf{T}\boldsymbol{v} &= 0 \qquad \text{(KVL)} \\ \boldsymbol{u} + \overline{\mathbf{R}}\boldsymbol{i} &= \boldsymbol{a} \qquad \text{(incident wave definition)} \\ \mathbf{A}\boldsymbol{i} &= 0 \qquad \text{(KCL)} \end{aligned} \tag{4.77}$$

or in matrix form

$$\begin{bmatrix} \mathbf{I}_B & \mathbf{0}_{B,N} & -\mathbf{A}^\mathsf{T} \\ \mathbf{I}_B & \overline{\mathbf{R}} & \mathbf{0}_{B,N} \\ \mathbf{0}_{N,B} & \mathbf{A} & \mathbf{0}_{N,N} \end{bmatrix} \begin{bmatrix} \boldsymbol{u} \\ \boldsymbol{i} \\ \boldsymbol{v} \end{bmatrix} = \begin{bmatrix} \mathbf{0}_{B,1} \\ \boldsymbol{a} \\ \mathbf{0}_{N,1} \end{bmatrix}. \tag{4.78}$$

The system can then be reduced by eliminating the node voltages as

$$\begin{bmatrix} \overline{\mathbf{R}} & -\mathbf{A}^\mathsf{T} \\ \mathbf{A} & \mathbf{0}_{N,N} \end{bmatrix} \begin{bmatrix} \boldsymbol{i} \\ \boldsymbol{v} \end{bmatrix} = \begin{bmatrix} \boldsymbol{a} \\ \mathbf{0}_{N,1} \end{bmatrix}. \tag{4.79}$$

We can then solve for the branch voltages and currents as

$$\begin{bmatrix} \boldsymbol{i} \\ \boldsymbol{v} \end{bmatrix} = \begin{bmatrix} \overline{\mathbf{R}} & -\mathbf{A}^\mathsf{T} \\ \mathbf{A} & \mathbf{0}_{N,N} \end{bmatrix}^{-1} \begin{bmatrix} \boldsymbol{a} \\ \mathbf{0}_{N,1} \end{bmatrix}. \tag{4.80}$$

By using block matrix inversion formulas [Zhang 2005], we obtain the closed-form expression

$$\begin{bmatrix} \boldsymbol{i} \\ \boldsymbol{v} \end{bmatrix} = \begin{bmatrix} \overline{\mathbf{G}}\Big(\mathbf{I}_B - \mathbf{A}^\mathsf{T}\big(\mathbf{A}\overline{\mathbf{G}}\mathbf{A}^\mathsf{T}\big)^{-1}\mathbf{A}\overline{\mathbf{G}}\Big) \\ -\big(\mathbf{A}\overline{\mathbf{G}}\mathbf{A}^\mathsf{T}\big)^{-1}\mathbf{A}\overline{\mathbf{G}} \end{bmatrix} \boldsymbol{a} \tag{4.81}$$

where $\overline{\mathbf{G}}$ is a diagonal matrix whose diagonal elements are the port conductances of the adaptor. This can then be combined with Eq. (4.76b) to finally form the network adaptor scattering equation (reflected waves as function of incident waves) as

$$\boldsymbol{b} = \Big(2\mathbf{A}^\mathsf{T}\big(\mathbf{A}\overline{\mathbf{G}}\mathbf{A}^\mathsf{T}\big)^{-1}\mathbf{A}\overline{\mathbf{G}} - \mathbf{I}_B\Big)\boldsymbol{a} \tag{4.82}$$

so that the network adaptor scattering matrix $\mathbf{C}$, defined as $\boldsymbol{b} = \mathbf{C}\boldsymbol{a}$, can finally be expressed as a function of the circuit topology following

$$\mathbf{C} = 2\mathbf{A}^\mathsf{T}\big(\mathbf{A}\overline{\mathbf{G}}\mathbf{A}^\mathsf{T}\big)^{-1}\mathbf{A}\overline{\mathbf{G}} - \mathbf{I}_B \tag{4.83}$$

Note that this expression verifies the known properties that the scattering matrix of a connection adapter is orthogonal (i.e., represents a lossless system), symmetric (i.e., represents a reciprocal

system) and involutary (i.e., represents a self-inverse system) meaning

$$\mathbf{C} = (\mathbf{C}^{\mathsf{T}})^{-1} \tag{4.84}$$

and

$$\mathbf{C} = \mathbf{C}^{\mathsf{T}} \tag{4.85}$$

and (redundantly)

$$\mathbf{C} = \mathbf{C}^{-1}. \tag{4.86}$$

Network port adaptation

When designing a wave digital filter model, a typical operation is called *port adaptation* for an adaptor. The operation consists of setting the port resistance at a given port of the adaptor as a function of the other port resistances in the adaptor to remove a delay-free loop at that port and preserve explicit computability of the model. In mathematical terms, that corresponds to setting the port resistances such that one of the diagonal terms in the scattering matrix $\mathbf{C}$ is zero. In our case, a typical operation would be to perform port adaptation at the source port e, i.e.,

$$\mathbf{C}_{e,e} = 0. \tag{4.87}$$

Using the Sherman-Morrison formula [Press et al. 2007], we get that

$$\left(\mathbf{A}\overline{\mathbf{G}}\mathbf{A}^{\mathsf{T}}\right)^{-1} = \left(\mathbf{A}_L\overline{\mathbf{G}}_L\mathbf{A}_L^{\mathsf{T}}\right)^{-1} - \frac{\left(\mathbf{A}_L\overline{\mathbf{G}}_L\mathbf{A}_L^{\mathsf{T}}\right)^{-1}\mathbf{A}_e\mathbf{A}_e^{\mathsf{T}}\left(\mathbf{A}_L\overline{\mathbf{G}}_L\mathbf{A}_L^{\mathsf{T}}\right)^{-1}}{2\mathbf{A}_e^{\mathsf{T}}\left(\mathbf{A}_L\overline{\mathbf{G}}_L\mathbf{A}_L^{\mathsf{T}}\right)^{-1}\mathbf{A}_e}. \tag{4.88}$$

From that equation and Eq. (4.83), we can isolate $C_{e,e}$ and get that

$$C_{e,e} = 2\frac{\mathbf{A}_e^{\mathsf{T}}\left(\mathbf{A}_L\overline{\mathbf{G}}_L\mathbf{A}_L^{\mathsf{T}}\right)^{-1}\mathbf{A}_e}{\overline{R}_e} - 1. \tag{4.89}$$

Canceling that term then leads to the closed-form expression of the port resistance $\overline{R}_e$ as

$$\overline{R}_e = \mathbf{A}_e^{\mathsf{T}}\left(\mathbf{A}_L\overline{\mathbf{G}}_L\mathbf{A}_L^{\mathsf{T}}\right)^{-1}\mathbf{A}_e. \tag{4.90}$$

When we adapt the port resistance $\overline{R}_e$, the (adapted) scattering matrix is finally given as

$$\mathbf{C} = \frac{1}{\overline{R}_e}\begin{bmatrix} -\mathbf{A}_L^{\mathsf{T}}\left(\mathbf{A}_L\overline{\mathbf{G}}_L\mathbf{A}_L^{\mathsf{T}}\right)^{-1}\mathbf{A}_e\mathbf{A}_e^{\mathsf{T}}\left(\mathbf{A}_L\overline{\mathbf{G}}_L\mathbf{A}_L^{\mathsf{T}}\right)^{-1}\mathbf{A}_L\overline{\mathbf{G}}_L & \mathbf{A}_L^{\mathsf{T}}\left(\mathbf{A}_L\overline{\mathbf{G}}_L\mathbf{A}_L^{\mathsf{T}}\right)^{-1}\mathbf{A}_e \\ \mathbf{A}_e^{\mathsf{T}}\left(\mathbf{A}_L\overline{\mathbf{G}}_L\mathbf{A}_L^{\mathsf{T}}\right)^{-1}\mathbf{A}_L & 0 \end{bmatrix} + \begin{bmatrix} 2\mathbf{A}_L^{\mathsf{T}}\left(\mathbf{A}_L\overline{\mathbf{G}}_L\mathbf{A}_L^{\mathsf{T}}\right)^{-1}\mathbf{A}_L\overline{\mathbf{G}}_L - \mathbf{I}_L & 0 \\ 0 & 0 \end{bmatrix}. \tag{4.91}$$

Transfer function expression

We can finally combine the various scattering expressions describing the circuit behavior, with the scattering from the linear elements given by

$$\boldsymbol{b}_L(s) = \mathbf{S}_L(s)\boldsymbol{a}_L(s) \tag{4.92}$$

and the scattering from the source element given by

$$b_e(s) = S_e a_e(s) + 2T_e\, e(s) \tag{4.93}$$

to form the system

$$\underbrace{\begin{bmatrix}\boldsymbol{b}_L(s)\\ b_e(s)\end{bmatrix}}_{\boldsymbol{b}(s)} = \underbrace{\begin{bmatrix}\mathbf{S}_L(s) & \mathbf{0}_L\\ \mathbf{0}_L^\mathsf{T} & S_e\end{bmatrix}}_{\mathbf{B}(s)}\underbrace{\begin{bmatrix}\boldsymbol{a}_L(s)\\ a_e(s)\end{bmatrix}}_{\boldsymbol{a}(s)} + 2T_e e(s)\begin{bmatrix}\mathbf{0}_L\\ 1\end{bmatrix}. \tag{4.94}$$

Along with the network adaptor as described in Eq. (4.91), we get the relation

$$\begin{bmatrix}\boldsymbol{b}(s)\\ \boldsymbol{a}(s)\end{bmatrix} = 2T_e\, e(s)\begin{bmatrix}(\mathbf{I}_B - \mathbf{B}(s)\mathbf{C})^{-1}\\ (\mathbf{C} - \mathbf{B}(s))^{-1}\end{bmatrix}\begin{bmatrix}\mathbf{0}_L\\ 1\end{bmatrix}. \tag{4.95}$$

If the output variable $\boldsymbol{o}$ is given as a linear combination $\mathbf{K}$ of various wave variables through the relation

$$\boldsymbol{o}(s) = \mathbf{K}\begin{bmatrix}\boldsymbol{b}(s)\\ \boldsymbol{a}(s)\end{bmatrix} \tag{4.96}$$

which leads to the general transfer function expression as

$$\mathbf{H}(s) = \frac{\boldsymbol{o}(s)}{e(s)} = 2T_e\,\mathbf{K}\begin{bmatrix}(\mathbf{I}_B - \mathbf{B}(s)\mathbf{C})^{-1}\\ (\mathbf{C} - \mathbf{B}(s))^{-1}\end{bmatrix}\begin{bmatrix}\mathbf{0}_L\\ 1\end{bmatrix}. \tag{4.97}$$

4.7.3 Transfer function discretization with s-to-z mappings

In Sec. 4.6.2, we saw that a discretization that can be expressed as s-to-z mapping essentially swaps the reactive elements of the circuit for their discretized version. As such, this process generates a discrete-time model whose transfer function $H_d(z)$ follows the same matrix equations except for the swapping of the admittance response (or the scattering response in a wave-domain formulation) at their respective location, replacing $\mathbf{Y}(s)$ with $\mathbf{Y}_d(z) = \mathbf{Y}(\mathcal{T}(z))$ in Eq. (4.71), and $\mathbf{B}(s)$ with $\mathbf{B}_d(z) = \mathbf{B}(\mathcal{T}(z))$ in Eq. (4.97) to form $\mathbf{H}_d(z) = \mathbf{H}(\mathcal{T}(s))$.

Standard bilinear transform and frequency warping

The most typical approach to discretize audio linear lumped systems using first-order mappings is the standard bilinear transform, which corresponds to the trapezoidal rule in numerical integration (see Sec. 3.7.3). As we mentioned before, the mapping corresponds to

$$s \mapsto \frac{2}{T_s}\frac{1-z^{-1}}{1+z^{-1}} \tag{4.98}$$

where T_s refers to the sampling period of the discretized system. A well-known distortion we mentioned introduced by this method is the so-called "frequency warping" we mentioned in Sec. 3.8.2. As a reminder, while ideally, we would like the original and discretized systems to have identical responses at each frequency (i.e., $\mathbf{H}(j\Omega) = \mathbf{H}_d(e^{-j\Omega T_s})$ for all $\Omega \in [-\pi/T_s, \pi/T_s]$), the system obtained using the standard bilinear transform verifies instead $\mathbf{H}(j\Omega) = \mathbf{H}_d(e^{-j\omega T_s})$ with the "warping" relation

$$\omega = \frac{2}{T_s}\tan^{-1}\left(\frac{\Omega T_s}{2}\right). \tag{4.99}$$

Parametric bilinear transform and frequency warping compensation

The parametric bilinear transform is a common way to modify the standard bilinear transform in order to alter the warping distortion [Smith III 2007b]. By replacing the quantity T_s in Eq. (4.98) with an appropriately chosen coefficient T, we can enforce the property that $\mathbf{H}(j\Omega) = \mathbf{H}_d(e^{-j\Omega T_s})$ for a *single* "matched" radian frequency Ω using

$$\frac{2}{\Omega}\tan\left(\frac{\Omega T_s}{2}\right). \tag{4.100}$$

The resulting system then exhibits a different frequency warping, as $\mathbf{H}(j\Omega) = \mathbf{H}_d(e^{-j\omega' T})$ for

$$\omega' = \frac{2}{T}\tan^{-1}\left(\frac{\Omega T_s}{2}\right). \tag{4.101}$$

The matched radian frequency is typically chosen to match the frequency of a salient property of the original system frequency response (e.g., resonant frequency, cutoff frequency) in the discretized system frequency response. However, only a single frequency can be matched, leaving no additional control over the error at other frequencies. This can become problematic if the system frequency response exhibits salient features spread over a wide frequency range [Werner et al. 2016a].

Elementwise discretization

In typical applications, the discretization mapping is applied globally to the system. Our proposed approach is to instead apply a different mapping to each linear element in the circuit.

By construction, the coefficients of the branch equations (associated with the L linear elements of the circuits) are distributed on the diagonal of the admittance matrix $\mathbf{Y}$ (and the impedance matrix $\mathbf{Z}$). The lth diagonal coefficient then contains at most one instance of s that we "attach" to that coefficient (and denote s_l). Applying elementwise transforms to each element then corresponds to selecting L coefficient sets θ_l ($l \in \{1 \dots L\}$) so that we form the matrices $\mathbf{Z}$ and $\mathbf{Y}$ by mapping any instance of s_l (associated with the branch equation of the lth element) as

$$s_l \mapsto \mathcal{T}(z; \theta_l). \tag{4.102}$$

4.7.4 Transfer function gradient

Numerous iterative optimization algorithms require the calculation of various derivative quantities of their objective function to find a solution. Using a finite difference approximation of these derivatives is often possible, but an exact derivative can improve convergence and lower computational costs. Thanks to the closed-form expressions of the transfer function (e.g., Eqs. (4.71), (4.97)), a close form of its derivatives can also be derived. Discretization based on s-to-z mappings such at the first-order mappings described in Sec. 3.7.3 can be described as parametrized by a set of free mapping/discretization parameters θ. We can then use the chain rule to express these quantities in a relatively simplified closed-form.

voltage–current description

The gradient in voltage–current description can be found by differentiating the matrix expression in Eq. (4.70). As we can see, the only dependent element is the admittance matrix $\mathbf{Y}$ in the center matrix, so that

$$\frac{\partial \mathbf{H}_d}{\partial \theta} = \frac{\left[-\mathbf{N}_L \; -N_e \,\middle|\, \mathbf{M}\right]}{Q_e + P_e \mathbf{A}_e^\mathsf{T} \mathbf{X}_d^{-1} \mathbf{A}_e} \left(\begin{bmatrix} \mathbf{I}_L - \mathbf{Y}\mathbf{A}_L^\mathsf{T}\mathbf{X}^{-1}\mathbf{A}_L \\ 0 \\ -\mathbf{X}^{-1}\mathbf{A}_L \end{bmatrix} \frac{\partial \mathbf{Y}}{\partial \theta} \mathbf{A}_L^\mathsf{T}\mathbf{X}^{-1}\mathbf{A}_e \right.$$
$$\left. - P_e \frac{\mathbf{A}_e^\mathsf{T}\mathbf{X}^{-1}\mathbf{A}_L \frac{\partial \mathbf{Y}}{\partial \theta} \mathbf{A}_L^\mathsf{T}\mathbf{X}^{-1}\mathbf{A}_e}{Q_e + P_e \mathbf{A}_e^\mathsf{T}\mathbf{X}^{-1}\mathbf{A}_e} \begin{bmatrix} \mathbf{Y}\mathbf{A}_L^\mathsf{T}\mathbf{X}^{-1}\mathbf{A}_e \\ -1 \\ \mathbf{X}^{-1}\mathbf{A}_e \end{bmatrix} \right). \tag{4.103}$$

In the case where the parameter θ is attached to a specific element in the circuit, only one diagonal element of the admittance matrix $\mathbf{Y}$ actually depends on θ. In the case where the parameter corresponds to a parameter for the mth element, we then get

$$\frac{\partial \mathbf{H}}{\partial \theta} = \frac{\left[-\mathbf{N}_L \; -N_e \,\middle|\, \mathbf{M}\right]}{Q_e + P_e \mathbf{A}_e^\mathsf{T} \mathbf{X}^{-1} \mathbf{A}_e} \left(\mathbf{A}_l^\mathsf{T} \mathbf{X}^{-1} \mathbf{A}_e\right) \frac{\partial Y_l}{\partial \theta} \left(\begin{bmatrix} \boldsymbol{\delta}_l - \mathbf{Y}\mathbf{A}_L^\mathsf{T} \mathbf{X}^{-1} \mathbf{A}_l \\ 0 \\ -\mathbf{X}^{-1} \mathbf{A}_l \end{bmatrix} - P_e \frac{\mathbf{A}_l^\mathsf{T} \mathbf{X}^{-1} \mathbf{A}_e}{Q_e + P_e \mathbf{A}_e^\mathsf{T} \mathbf{X}^{-1} \mathbf{A}_e} \begin{bmatrix} \mathbf{Y}\mathbf{A}_L^\mathsf{T} \mathbf{X}^{-1} \mathbf{A}_e \\ -1 \\ \mathbf{X}^{-1} \mathbf{A}_e \end{bmatrix} \right). \quad (4.104)$$

Note however that setting a different coefficient for every single element in the system is not absolutely necessary and the approach also allows for the sharing of a coefficient across multiple elements. In its most extreme case, we can share the coefficients system-wide to find a system-wide optimal discretization. This is what we do in Sec. 4.8.2 when we compute a system-wide optimal parametric bilinear transform for comparison purposes.

Wave-domain description

Similarly, we can find the gradient of the wave-domain expression in Eq. (4.97) as

$$\mathbf{H}(s) = 2T_e \, \mathbf{K} \begin{bmatrix} (\mathbf{I}_B - \mathbf{B}\mathbf{C})^{-1} \\ (\mathbf{C} - \mathbf{B})^{-1} \end{bmatrix} \frac{\partial \mathbf{B}}{\partial \theta} (\mathbf{C} - \mathbf{B})^{-1} \begin{bmatrix} \mathbf{0}_L \\ 1 \end{bmatrix}. \quad (4.105)$$

Here too, in cases where the parameter θ only affect a single element, it means only one diagonal element of $\mathbf{B}$ depends on it. In this case, the gradient becomes

$$\mathbf{H}(s) = 2T_e \, \mathbf{K} \, \frac{\partial S_l}{\partial \theta} \begin{bmatrix} (\mathbf{I}_B - \mathbf{B}\mathbf{C})^{-1} \\ (\mathbf{C} - \mathbf{B})^{-1} \end{bmatrix} \boldsymbol{\delta}_l^\mathsf{T} \boldsymbol{\delta}_l (\mathbf{C} - \mathbf{B})^{-1} \begin{bmatrix} \mathbf{0}_L \\ 1 \end{bmatrix}. \quad (4.106)$$

4.7.5 Objective function

In the audio context, we generally focus on preserving the transfer function of a linear system so that we choose as objective function ϵ (or error function) the function

$$\begin{aligned} \epsilon &= ||\mathbf{H} - \mathbf{H}_d|| \\ &= \int_{\omega_1}^{\omega_2} \left\| \mathbf{H}(j\omega T_s) - \mathbf{H}_d\big(e^{j\omega T_s}\big) \right\| d\omega \\ &= \int_{\omega_1}^{\omega_2} \left\| \mathbf{H}\big(\mathcal{T}\big(e^{j\omega T_s}\big)\big) - \mathbf{H}_d\big(e^{j\omega T_s}\big) \right\| d\omega \end{aligned} \quad (4.107)$$

with $0 \leq \omega_1 \leq \omega_2 \leq f_s/2$. In this case, the notation $\|\cdot\|$ corresponds to some form of metric of discrepancy between the two transfer functions, referred to as the *loss function*.

Loss function

A typical choice for the loss function is the ℓ^2 loss or square loss, for which

$$\begin{aligned}\left\|\mathbf{H}\big(\mathcal{T}\big(e^{j\omega T_s}\big)\big) - \mathbf{H}_d\big(e^{j\omega T_s}\big)\right\|_{\ell^2} &= \left\|\mathbf{H}\big(\mathcal{T}\big(e^{j\omega T_s}\big)\big) - \mathbf{H}_d\big(e^{j\omega T_s}\big)\right\|^2 \\ &= \sum_{m=1}^{M} \left|H_m\big(\mathcal{T}\big(e^{j\omega T_s}\big)\big) - H_{d,m}\big(e^{j\omega T_s}\big)\right|^2.\end{aligned} \tag{4.108}$$

As mentioned before, optimization algorithms benefit from knowledge of the analytical expression of the gradient. In this case, we can derive the gradient of the error with respect to the discretization mapping parameters as

$$\frac{\partial \epsilon_{\ell^2}}{\partial \theta} = 2\int_{\omega_1}^{\omega_2} \Re\left[\left(\frac{\partial \mathbf{H}_d^\star}{\partial \theta}\right)^{\mathsf{T}} \big(\mathbf{H}\big(\mathcal{T}\big(e^{j\omega T_s}\big)\big) - \mathbf{H}_d\big(e^{j\omega T_s}\big)\big)\right] d\omega. \tag{4.109}$$

Another relatively common choice is the ℓ^1 loss or absolute value loss, for which

$$\left\|\mathbf{H}\big(\mathcal{T}\big(e^{j\omega T_s}\big)\big) - \mathbf{H}_d\big(e^{j\omega T_s}\big)\right\|_{\ell^1} = \sum_{m=1}^{M} \left|H_m\big(\mathcal{T}\big(e^{j\omega T_s}\big)\big) - H_{d,m}\big(e^{j\omega T_s}\big)\right| \tag{4.110}$$

and the gradient is then given by

$$\frac{\partial \epsilon_{\ell^2}}{\partial \theta} = 2\int_{\omega_1}^{\omega_2} \Re\left[\sum_{m=1}^{M}\left(\frac{\partial H_{d,m}}{\partial \theta}\right)^\star \frac{H_m\big(\mathcal{T}\big(e^{j\omega T_s}\big)\big) - H_{d,m}\big(e^{j\omega T_s}\big)}{|H_m(\mathcal{T}(e^{j\omega T_s})) - H_{d,m}(e^{j\omega T_s})|}\right] d\omega. \tag{4.111}$$

We will illustrate the results obtained with both these loss functions in Sec. 4.8. However, our framework here is quite flexible and and allows for more complex loss functions that are more tailored to audio applications by incorporating prior knowledge on human audio perception. For example, we could add frequency-dependent weightings to the error (e.g., a relative loss like $\frac{\|H_d - H\|}{\|H\|}$) and/or consider logarithmic-type errors to better match the perceptual error of our system. In this case, we must keep in mind that more complex loss might necessitate a more elaborate design of the optimization process (e.g., to deal with problems such as non-convexity of the loss function). Another possible approach to design a loss with stronger perceptual meaning would be to use conformal maps such as the Bark or the ERB bilinear transform [Strube 1980, Smith III and Abel 1999] in order to achieve a frequency warping similar to that of the human auditory system.

4.7.6 Regularizations

We want to impose penalties on the discretization parameters moving away from prescribed default parameters (e.g., the bilinear transform parameters, the usual parametric bilinear transform parameters) to ensure we get reasonable parameter values. We could also limit the spread of values by

penalizing the distance between the found values so that they all hover around similar values (to mimic the case of the parametric bilinear transform with a common parameter, however far of the standard bilinear transform. This leads to defining an objective function of the form

$$\epsilon = ||\mathbf{H} - \mathbf{H}_d|| + \sum_{l=0}^{L} \lambda_l |T_l - T_s|^2 \tag{4.112}$$

to penalize departing from the standard bilinear transform coefficients and

$$\epsilon = ||\mathbf{H} - \mathbf{H}_d|| + \sum_{l=0}^{L} \sum_{l'=0}^{L} \lambda_{l,l'} |T_l - T_{l'}|^2 \tag{4.113}$$

to penalize the spread among elements. These types of regularizations are simple to implement and the required modifications of the analytical is straightforward.

4.7.7 Initialization

Optimization algorithms require an initial guess. In the absence of indication, setting the parameters as the parameters for the standard bilinear transform would generally be considered the better guess due to the guaranteed good fit in the lower frequency range where the frequency warping distortion is minimal. In cases where a single frequency seems more significant, it is also possible to start from the parameters of the parametric bilinear transform that matches that particular frequency between the continuous-time and the discrete-time domains.

4.7.8 Examples of mappings

Discretization with parametric bilinear transforms

Recalling the type of mappings we described in Sec. 3.7.3, we can identify these free parameters. For example, in the case of the parametric bilinear transform, which is of the form

$$s = \frac{2}{T} \frac{1 - z^{-1}}{1 + z^{-1}}, \tag{4.114}$$

the free parameters correspond to the time-like coefficient T, meaning $\theta_{\text{PBT}} = \{T\}$. In that case, the typical one-port linear elements have the admittance and scattering functions as described in Tab. 4.3, and the gradient of these quantities with respect to the mapping parameter is shown in Tab. 4.4.

Element type	Admittance	Scattering	Adapted scattering
Resistor	$Y_d(z) = R$	$S_d(z) = \dfrac{R - \overline{R}}{R + \overline{R}}$	$S_d(z) = 0$
Capacitor	$Y_d(z) = \dfrac{2C}{T}\dfrac{1 - z^{-1}}{1 + z^{-1}}$	$S_d(z) = \dfrac{\left(1 - \frac{2\overline{R}C}{T}\right) + \left(1 + \frac{2\overline{R}C}{T}\right)z^{-1}}{\left(1 + \frac{2\overline{R}C}{T}\right) + \left(1 - \frac{2\overline{R}C}{T}\right)z^{-1}}$	$S_d(z) = z^{-1}$
Inductor	$Y_d(z) = \dfrac{T}{2L}\dfrac{1 + z^{-1}}{1 - z^{-1}}$	$S_d(z) = \dfrac{\left(\frac{2L}{\overline{R}T} - 1\right) - \left(\frac{2L}{\overline{R}T} + 1\right)z^{-1}}{\left(\frac{2L}{\overline{R}T} + 1\right) - \left(\frac{2L}{\overline{R}T} - 1\right)z^{-1}}$	$S_d(z) = -z^{-1}$

Table 4.3: Discretized admittance and scattering values of typical one-port linear elements for parametric bilinear mappings.

Element type	Admittance	Scattering
Resistor	$\dfrac{\partial Y_d}{\partial T}(z) = 0$	$\dfrac{\partial S_d}{\partial T}(z) = 0$
Capacitor	$\dfrac{\partial Y_d}{\partial T}(z) = \dfrac{2C}{T}\dfrac{z^{-1} - 1}{z^{-1} + 1}$	$\dfrac{\partial S_d}{\partial T}(z) = \dfrac{2}{T}\dfrac{RC\mathcal{T}_{PBT}(z)}{(1 + RC\mathcal{T}_{PBT}(z))^2}$
Inductor	$\dfrac{\partial Y_d}{\partial T}(z) = \dfrac{1}{2L}\dfrac{1 + z^{-1}}{1 - z^{-1}}$	$\dfrac{\partial Y_d}{\partial T}(z) = -\dfrac{2}{T}\dfrac{LR\,\mathcal{T}_{PBT}(z)}{(L\mathcal{T}_{PBT}(z) + R)^2}$

Table 4.4: Derivative of the discretized admittance and scattering values of typical one-port linear elements with respect to mapping parameter T for parametric bilinear mappings.

Element type	Admittance	Scattering	Adapted scattering
Resistor	$Y_d(z) = R$	$S_d(z) = \dfrac{R - \overline{R}}{R + \overline{R}}$	$S_d(z) = 0$
Capacitor	$Y_d(z) = \dfrac{C(1+\alpha)}{T_s}\dfrac{1 - z^{-1}}{1 + \alpha z^{-1}}$	$S_d(z) = \dfrac{\left(\frac{1}{1+\alpha} - \frac{\overline{R}C}{T_s}\right) + \left(\frac{\alpha}{1+\alpha} + \frac{\overline{R}C}{T_s}\right)z^{-1}}{\left(\frac{1}{1+\alpha} + \frac{\overline{R}C}{T_s}\right) + \left(\frac{\alpha}{1+\alpha} - \frac{\overline{R}C}{T_s}\right)z^{-1}}$	$S_d(z) = \dfrac{(1+\alpha)z^{-1}}{2 - (1-\alpha)z^{-1}}$
Inductor	$Y_d(z) = \dfrac{T_s}{L(1+\alpha)}\dfrac{1 + \alpha z^{-1}}{1 - z^{-1}}$	$S_d(z) = \dfrac{\left(\frac{L}{\overline{R}T_s} - \frac{1}{1+\alpha}\right) - \left(\frac{L}{\overline{R}T_s} + \frac{\alpha}{1+\alpha}\right)z^{-1}}{\left(\frac{L}{\overline{R}T_s} + \frac{1}{1+\alpha}\right) - \left(\frac{L}{\overline{R}T_s} - \frac{\alpha}{1+\alpha}\right)z^{-1}}$	$S_d(z) = \dfrac{-(1+\alpha)z^{-1}}{2 - (1-\alpha)z^{-1}}$

Table 4.5: Discretized admittance and scattering values of typical one-port linear elements for α-transform mappings.

Element type	Admittance	Scattering
Resistor	$\frac{\partial Y_d}{\partial \alpha}(z) = 0$	$\frac{\partial S_d}{\partial \alpha}(z) = 0$
Capacitor	$\frac{\partial Y_d}{\partial \alpha}(z) = \frac{C}{T}\left(\frac{1-z^{-1}}{1+\alpha z^{-1}}\right)^2$	$\frac{\partial S_d}{\partial \alpha}(z) = -\frac{2\overline{R}C}{T}\left(\frac{\frac{1-z^{-1}}{1+\alpha}}{\left(\frac{1}{1+\alpha}+\frac{\overline{R}C}{T}\right)+\left(\frac{\alpha}{1+\alpha}-\frac{\overline{R}C}{T}\right)z^{-1}}\right)^2$
Inductor	$\frac{\partial Y_d}{\partial \alpha}(z) = -\frac{T}{L(1+\alpha)^2}$	$\frac{\partial S_d}{\partial \alpha}(z) = \frac{2L}{\overline{R}T}\left(\frac{\frac{1-z^{-1}}{1+\alpha}}{\left(\frac{L}{\overline{R}T_s}+\frac{1}{1+\alpha}\right)-\left(\frac{L}{\overline{R}T_s}-\frac{\alpha}{1+\alpha}\right)z^{-1}}\right)^2$

Table 4.6: Derivative of the discretized admittance and scattering values of typical one-port linear elements with respect to mapping parameter α for α-transform mappings.

Element type	Admittance	Scattering	Adapted scattering
Resistor	$Y_d(z) = R$	$S_d(z) = \frac{R-\overline{R}}{R+\overline{R}}$	$S_d(z) = 0$
Capacitor	$Y_d(z) = \frac{C(1+\alpha)}{T}\frac{1-z^{-1}}{1+\alpha z^{-1}}$	$S_d(z) = \frac{\left(\frac{1}{1+\alpha}-\frac{\overline{R}C}{T}\right)+\left(\frac{\alpha}{1+\alpha}+\frac{\overline{R}C}{T}\right)z^{-1}}{\left(\frac{1}{1+\alpha}+\frac{\overline{R}C}{T}\right)+\left(\frac{\alpha}{1+\alpha}-\frac{\overline{R}C}{T}\right)z^{-1}}$	$S_d(z) = \frac{(1+\alpha)z^{-1}}{2-(1-\alpha)z^{-1}}$
Inductor	$Y_d(z) = \frac{T}{L(1+\alpha)}\frac{1+\alpha z^{-1}}{1-z^{-1}}$	$S_d(z) = \frac{\left(\frac{L}{\overline{R}T}-\frac{1}{1+\alpha}\right)-\left(\frac{L}{\overline{R}T}+\frac{\alpha}{1+\alpha}\right)z^{-1}}{\left(\frac{L}{\overline{R}T}+\frac{1}{1+\alpha}\right)-\left(\frac{L}{\overline{R}T}-\frac{\alpha}{1+\alpha}\right)z^{-1}}$	$S_d(z) = \frac{-(1+\alpha)z^{-1}}{2-(1-\alpha)z^{-1}}$

Table 4.7: Discretized admittance and scattering values of typical one-port linear elements for parametric α-transform mappings.

Discretization with α-transforms

In the case of the α-transform, which is of the form

$$s = \frac{1+\alpha}{T_s}\frac{1-z^{-1}}{1+\alpha z^{-1}}, \tag{4.115}$$

the free parameters correspond to the weighting coefficient α, meaning $\theta_{\mathrm{AT}} = \{\alpha\}$. In that case, the typical one-port linear elements have the admittance and scattering functions as described in Tab. 4.5, and the gradient of these quantities with respect to the mapping parameter is shown in Tab. 4.6.

Element type	Admittance	Scattering
Resistor	$\frac{\partial Y_d}{\partial \alpha}(z) = 0$	$\frac{\partial S_d}{\partial \alpha}(z) = 0$
	$\frac{\partial Y_d}{\partial T}(z) = 0$	$\frac{\partial S_d}{\partial T}(z) = 0$
Capacitor	$\frac{\partial Y_d}{\partial T}(z) = \frac{C}{T}\left(\frac{1-z^{-1}}{1+\alpha z^{-1}}\right)^2$	$\frac{\partial S_d}{\partial \alpha}(z) = -\frac{\frac{2\bar{R}C}{T}\left(\frac{1-z^{-1}}{1+\alpha}\right)^2}{\left(\left(\frac{1}{1+\alpha}+\frac{\bar{R}C}{T}\right)+\left(\frac{\alpha}{1+\alpha}-\frac{\bar{R}C}{T}\right)z^{-1}\right)^2}$
	$\frac{\partial Y_d}{\partial T}(z) = \frac{C(1+\alpha)}{T^2}\frac{1-z^{-1}}{1+\alpha z^{-1}}$	$\frac{\partial S_d}{\partial T}(z) = \frac{\frac{2\bar{R}C}{T^2}\frac{1+\alpha z^{-1}}{1+\alpha}(1-z^{-1})}{\left(\left(\frac{1}{1+\alpha}+\frac{\bar{R}C}{T}\right)+\left(\frac{\alpha}{1+\alpha}-\frac{\bar{R}C}{T}\right)z^{-1}\right)^2}$
Inductor	$\frac{\partial Y_d}{\partial \alpha}(z) = -\frac{T}{L(1+\alpha)^2}$	$\frac{\partial S_d}{\partial \alpha}(z) = \frac{\frac{2L}{\bar{R}T}\left(\frac{1-z^{-1}}{1+\alpha}\right)^2}{\left(\left(\frac{L}{\bar{R}T}+\frac{1}{1+\alpha}\right)-\left(\frac{L}{\bar{R}T}-\frac{\alpha}{1+\alpha}\right)z^{-1}\right)^2}$
	$\frac{\partial Y_d}{\partial T}(z) = \frac{1}{L(1+\alpha)}\frac{1+\alpha z^{-1}}{1-z^{-1}}$	$\frac{\partial S_d}{\partial T}(z) = \frac{\frac{2L}{\bar{R}T^2}\frac{1+\alpha z^{-1}}{1+\alpha}(z^{-1}-1)}{\left(\left(\frac{L}{\bar{R}T}+\frac{1}{1+\alpha}\right)-\left(\frac{L}{\bar{R}T}-\frac{\alpha}{1+\alpha}\right)z^{-1}\right)^2}$

Table 4.8: Derivative of the discretized admittance and scattering values of typical one-port linear elements with respect to mapping parameters α and T for parametric α-transform mappings.

Discretization with parametric α-transforms

In the case of the more general parametric α-transform, which is of the form

$$s = \frac{1+\alpha}{T}\frac{1-z^{-1}}{1+\alpha z^{-1}}, \tag{4.116}$$

the free parameters correspond to the coefficients α and T, meaning $\theta_{\text{PAT}} = \{\alpha, T\}$. In that case, the typical one-port linear elements have the admittance and scattering functions as described in Tab. 4.7, and the gradient of these quantities with respect to the mapping parameters is shown in Tab. 4.8.

4.8 Case studies

We apply our approach to three different circuits to validate it. In these, we limit ourselves to the case of applying an elementwise parametric bilinear mapping for simplicity. First, we look at a resonant RLC series circuit with a single resonance. Then we look at a Helmholtz resonator tree like the ones presented in [Paiva and Välimäki 2012] which exhibits multiple resonances. In both cases, we look at the frequency response error introduced by the typical standard bilinear transform (BT), the parametric bilinear transform using the same T coefficient system-wide (PBT), and our approach (*Elem.*) using the jointly optimized elementwise T coefficients among linear circuit elements compared to the response of the original continuous-time system ($Analog$). Finally, we look at the case of the Hammond vibrato circuit [Werner et al. 2016a] which is a multi-output system with a complex transfer function. In this section, we present comparison of the transfer functions between continuous-time system (H) and various discrete-time models (H_d). For simplicity and compactness, we overload the notation relative to the values of the transfer functions at frequency f as $H(f)$ and $H_d(f)$. With respect to the transfer function denoted as functions of s in continuous-time and z in discrete-time, these two quantities correspond to

$$H(f) \equiv H(s_f) \text{ for } s_f = 2\pi j f, \text{ and } H_d(f) \equiv H_d(z_f) \text{ for } z_f = \exp(2\pi j f). \tag{4.117}$$

4.8.1 Resonant RLC series circuit

The results of this case study were originally presented in Germain and Werner [2017a]. Here, we study a resonant RLC series circuit such as the one in Fig. 4.3, with the lowest node as datum node. For this system, we then have

$$\mathbf{A}_L = \begin{bmatrix} 1 & -1 & 0 \\ 0 & 1 & -1 \\ 0 & 0 & 1 \end{bmatrix}, \quad \mathbf{A}_e = \begin{bmatrix} 0 \\ 0 \\ -1 \end{bmatrix}$$

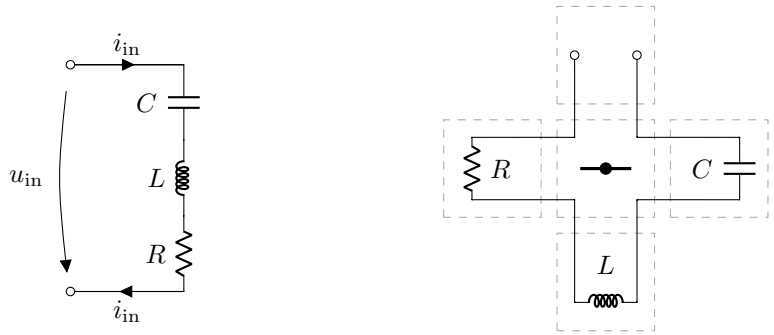


Figure 4.3: (a) RLC series circuit and (b) equivalent connection tree with one free port.

$$\text{and} \quad \mathbf{Z} = \begin{bmatrix} 1/(sC) & 0 & 0 \\ 0 & sL & 0 \\ 0 & 0 & R \end{bmatrix}. \tag{4.118}$$

We know that such a circuit presents a distinctive feature in its voltage–current transfer function in the form of a large resonant peak. As such, it would be generally considered a perfect candidate for the typical application of a system-wide parametric bilinear transform matching the resonance frequency. However, our study below will show how our optimization can still uncover a non-trivial elementwise discretization formula that significantly lowers the model error.

We optimize the voltage–current transfer $i_{\text{in}}/u_{\text{in}}$ function of the system using elementwise parametric bilinear transforms with a different parameter T for each linear element. The component values for the circuit are set as $R = 25\,\Omega$, $L = 2\,\text{mH}$, $C = 0.2\,\mu\text{F}$. As such, the RLC circuit has a resonance at 7.958 kHz, with a quality factor Q of 4. In the case of a system-wide parametric bilinear transform, the typical approach is to pick T to match the frequency response at the resonant peak ($T = 25.46\,\mu\text{s}$).

Implementation considerations

In order to compute the different integral quantities (error function and its gradient as in Sec. 4.7.5), we use the MATLAB implementation[4] of the adaptive Simpson quadrature for the two first studies [Gander and Gautschi 2000]. For the optimization, we use the MATLAB implementation[5] of a subspace trust-region algorithm based on the interior-reflective Newton method from [Coleman and Li 1996], to which we supply the analytic gradient expression (see Secs. 4.7.4, 4.7.5 and 4.7.8). All the algorithms are applied using the default parameters. The system is sampled at 44.1 kHz (or $T_s = 22.68\,\mu\text{s}$). The optimization is performed over the frequency range $[20\,\text{Hz}, 20\,\text{kHz}]$, which represents

[4] https://www.mathworks.com/help/matlab/ref/quad.html
[5] https://www.mathworks.com/help/optim/ug/fminunc.html

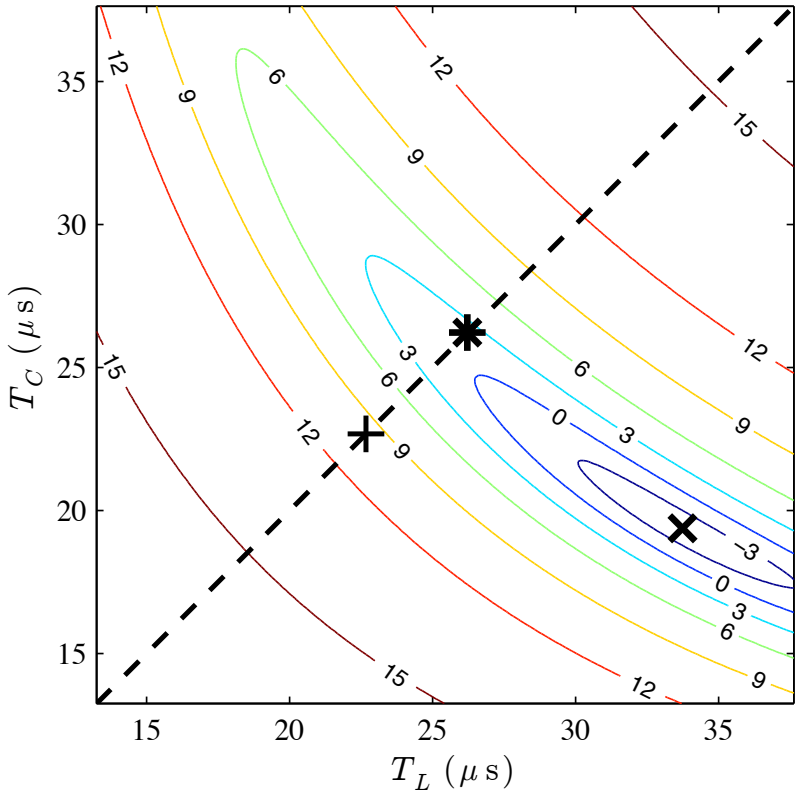


Figure 4.4: Contour plot of the ℓ^2 error ϵ (in dB scale) for an RLC series circuit for the elementwise parametrizations (T_C, T_L). Error locations for the different approaches are indicated (BT: $+$, PBT: $*$, Elementwise: $\times$). The dashed line indicates the space of possible parametrization of the bilinear transforms using a system-wide T coefficient (from Germain and Werner [2017a]).

the frequency range of interest for audio applications. For the initial point of the optimization process, we set both coefficients T_L and T_C at T_s.

Discussion

The ℓ^2 error values ϵ for various combinations of T_C and T_L are shown in Fig. 4.4. The error for the coefficients corresponding to the system-wide standard bilinear transform, the system-wide parametric bilinear transform matching the resonant peak and the jointly optimized elementwise discretization are also indicated. The plot shows that the system-wide standard bilinear transform introduces significant frequency response error compared to the elementwise method. We also see that using any version of the parametric bilinear transform system-wide (i.e., moving along the dashed line) clearly cannot lower the error as much as it can be using elementwise T coefficients for the capacitor and the inductor. Finally, we see how the heuristic of matching the resonant peak is not exactly equivalent to minimizing the ℓ^2 error ϵ as function of T.

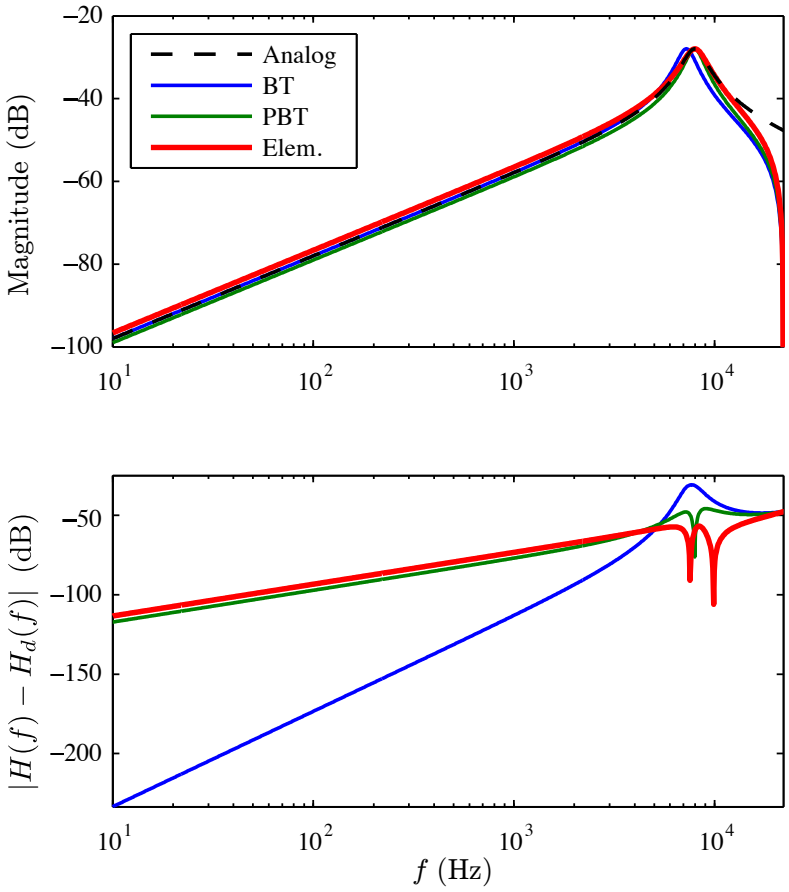


Figure 4.5: Magnitude response (top) and error $|H(f) - H_d(f)|$ (bottom) for the system-wide standard bilinear transform (BT), the system-wide parametric bilinear transform (PBT), and our elementwise approach (Elem.) applied to the RLC series circuit (from Germain and Werner [2017a]). The response of the continuous-time circuit (Analog) is also shown for comparison.

name	value	name	value
T_C	19.38 µs	T_L	33.74 µs

Table 4.9: Jointly optimized elementwise T coefficients for the RLC series circuit.

	BT	PBT	Elem.
ϵ	9.8884	1.2120	0.3448

Table 4.10: ℓ^2 error ϵ for the system-wide standard bilinear transform (BT), the system-wide parametric bilinear transform (PBT) and the elementwise approach (Elem.) applied to the RLC series circuit.

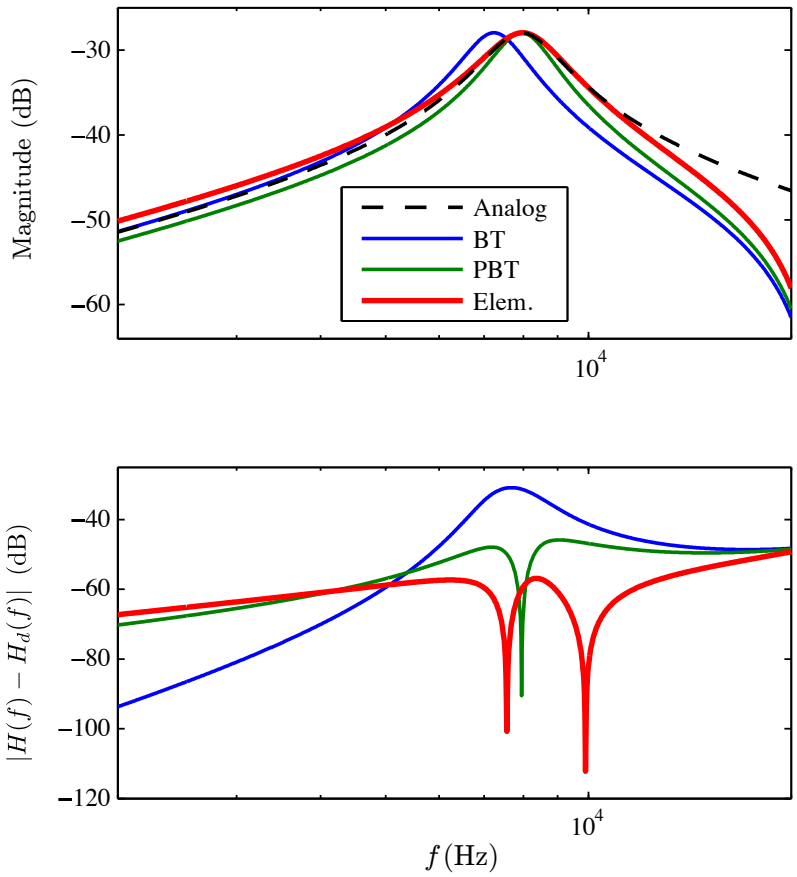


Figure 4.6: Zoomed-in magnitude response (top) and error $|H(f) - H_d(f)|$ (bottom) for the system-wide standard bilinear transform (BT), the system-wide parametric bilinear transform (PBT), and our elementwise approach (Elem.) applied to the RLC series circuit (from Germain and Werner [2017a]). The response of the continuous-time circuit (Analog) is also shown for comparison.

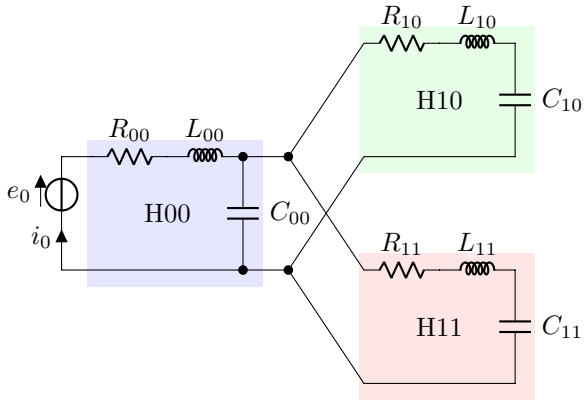


Figure 4.7: Circuit schematic of the Helmholtz resonator tree. The colors refers to locations of identical colors in Fig. 4.8.

H00		H10		H11	
name	value	name	value	name	value
R_{00}	$25\,\Omega$	R_{10}	$25\,\Omega$	R_{11}	$25\,\Omega$
L_{00}	$10\,\text{mH}$	L_{10}	$50\,\text{mH}$	L_{11}	$250\,\text{mH}$
C_{00}	$1\,\mu\text{F}$	C_{10}	$5\,\mu\text{F}$	C_{11}	$25\,\mu\text{F}$

Table 4.11: Electrical component values for the Helmholtz resonator tree circuit.

Tab. 4.9 shows the T coefficients found after the joint optimization using an ℓ^2 error function. Tab. 4.10 shows how much the elementwise approach lowers the error compared to the system-wide standard bilinear transform and the system-wide parametric bilinear transform (matching the resonant peak). Figs. 4.5 and 4.6 show full-range and zoomed-in frequency responses and error $|H(f) - H_d(f)|$ for the standard bilinear transform, the parametric bilinear transform and the elementwise discretization. For the standard bilinear transform, the frequency warping introduces significant error around the resonant peak. The parametric bilinear transform cancels the error at the peak location but a lot of error remains around it because the peak width of the original system and the discretized system do not match [Stilson 2006]. The jointly optimized elementwise discretization lowers the error, distributes it more uniformly across the frequency range, and matches much better the resonant peak in frequency and width.

4.8.2 Helmholtz resonator tree

The results of this case study were originally presented in Germain and Werner [2017a]. Here, we study a Helmholtz resonator tree circuit [Paiva and Välimäki 2012] such as the one in Figs. 4.7 and

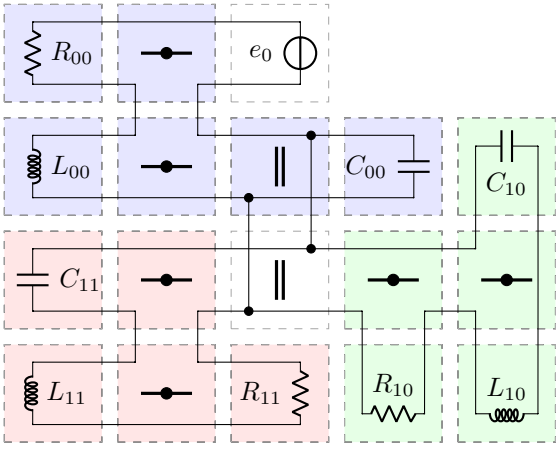


Figure 4.8: Connection tree of the Helmholtz resonator tree. The colors refers to locations of identical colors in Fig. 4.7.

4.8, with the lower-right node as datum node. For this system, we have

$$\mathbf{A}_L = \begin{bmatrix} 1 & -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & -1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}^{\mathsf{T}},$$

$$\mathbf{A}_e = \begin{bmatrix} -1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \text{ and } \mathbf{Z}_L = \operatorname{diag}\left(\begin{bmatrix} R_{00} \\ sL_{00} \\ 1/sC_{00} \\ R_{10} \\ sL_{10} \\ 1/sC_{10} \\ R_{11} \\ sL_{11} \\ 1/sC_{11} \end{bmatrix} \right). \tag{4.119}$$

By design, such a circuit presents three distinctive features in its voltage–current transfer function

in the form of three large resonant peaks. As such, the typical approach of a system-wide parametric bilinear transform to match some frequency cannot be applied in any straightforward manner. Indeed, only one of the three resonant frequencies can be matched with a given parametrization of the transform, and that matching will generally worsen the model error around the other two resonant peaks. Furthermore, the literature does not provide us any heuristic to select manually either system-wide or elementwise discretizations for the elements of the circuit. On the other hand, our study below will show how our optimization uncovers a non-trivial elementwise discretization that successfully matches all the transfer function features as well as significantly lowering the model error.

We optimize the voltage–current response (i_0/e_0) of this system with the component values from Tab. 4.8.2 using elementwise parametric bilinear transforms with a different parameter T for each linear element. As designed, this system presents three distinct resonant peaks of similar intensity and quality factor, meaning three salient elements widely spread over the frequency range. Also, while the circuit may appear as three RLC series sub-circuits (highlighted with three different colors in Figs. 4.7 and 4.8), the parallel connections (denoted || in Fig. 4.8) generate a load between those sub-circuits that prevents treating them independently, since the properties of each resonant peak are functions of all the components. As mentioned above, there is no longer a typical heuristic for the selection of a system-wide T coefficient for the parametric bilinear transform as it can only be tuned to a single frequency and the system presents three resonant peaks. Instead, for a fair comparison to the case of the system-wide parametric bilinear transform, we select the parametrization coefficient $T = 26.22\,\mu\text{s}$, which we find by minimizing the error ϵ using our optimization procedure but with a shared system-wide coefficient T across all elements instead of independent elementwise coefficients.

Implementation considerations

The implementation is identical to the one used in Sec. 4.8.1 above. In order to compute the different integral quantities (error function and its gradient as in Sec. 4.7.5), we use the MATLAB implementation[6] of the adaptive Simpson quadrature for the two first studies [Gander and Gautschi 2000]. For the optimization, we use the MATLAB implementation[7] of a subspace trust-region algorithm based on the interior-reflective Newton method from [Coleman and Li 1996], to which we supply the analytic gradient expression (see Secs. 4.7.4, 4.7.5 and 4.7.8). All the algorithms are applied using the default parameters. The system is sampled at 44.1 kHz (or $T_s = 22.68\,\mu\text{s}$). The optimization is performed over the frequency range $[20\,\text{Hz}, 20\,\text{kHz}]$, which, again, represents the frequency range of interest for audio applications. For the initial point of the optimization process, we set all six elementwise coefficients T equal to T_s.

[6] https://www.mathworks.com/help/matlab/ref/quad.html
[7] https://www.mathworks.com/help/optim/ug/fminunc.html

name	value	name	value	name	value
$T_{C_{00}}$	21.20 µs	$T_{C_{10}}$	19.95 µs	$T_{C_{11}}$	20.35 µs
$T_{L_{00}}$	34.80 µs	$T_{L_{10}}$	24.79 µs	$T_{L_{11}}$	25.10 µs

Table 4.12: Jointly optimized elementwise coefficients T for the Helmholtz resonator tree circuit.

	BT	PBT	Elem.
ϵ	14.7981	1.9538	0.3372

Table 4.13: ℓ^2 error ϵ for the system-wide standard bilinear transform (BT), the system-wide parametric bilinear transform (PBT) and the elementwise approach (Elem.) applied to the Helmholtz resonator tree circuit.

Discussion

Tab. 4.12 shows the elementwise T coefficients found after the joint optimization for an ℓ^2 error function. Tab. 4.13 shows how the elementwise approach lowers the error by two orders of magnitude compared to the standard bilinear transform and one order of magnitude compared to the optimized parametric bilinear transform. Fig. 4.9 shows the frequency responses and error $|H(f) - H_d(f)|$ for the standard bilinear transform, optimized parametric bilinear transform and the jointly optimized elementwise approach. The warping distortion of the standard bilinear transform introduces error, most significantly around the resonant peak with the highest frequency. Optimizing a system-wide parametrization of the parametric bilinear transform matches better that peak in frequency, but does not match its width or the frequency and width of the peaks at lower frequencies. By jointly optimizing the elementwise coefficients T, the error is distributed much more uniformly across the entire frequency range. We also get a very large error improvement around the resonant peak with the highest frequency and a discretized system frequency response that exhibits three resonant peaks with the correct frequency and width.

Name	value	units	Name	value	units	Name	value	units
R_c	22	kΩ	R_{2-}, R_{3-}	0.15	MΩ	$L_1 \dots L_{18}$	500	mH
R_{1+}	27	kΩ	R_{4+}	33	kΩ	$C_1 \dots C_{17}$	0.004	µF
R_{1-}	68	kΩ	R_{5+}	18	kΩ	C_{18}	0.001	µF
R_{2+}	56	kΩ	R_{6+}	12	kΩ	R_t	15	kΩ
R_{3+}	39	kΩ	$R_{4-} \dots R_{6-}$	0.18	MΩ			

Table 4.14: Circuit component values for the Hammond chorus/vibrato circuit shown in Fig. 4.10.

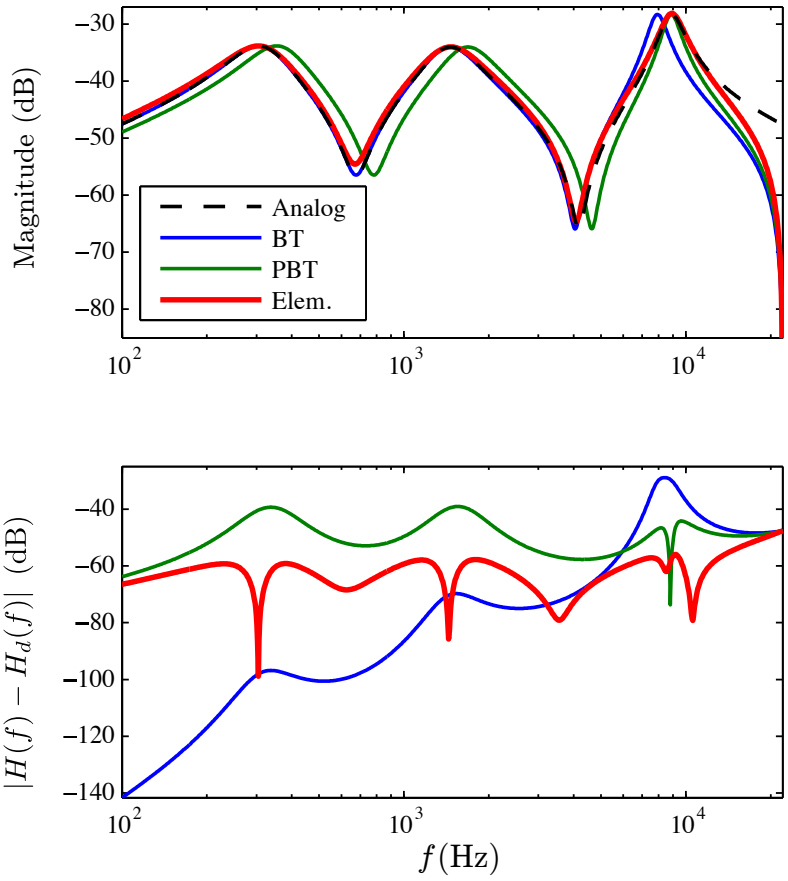


Figure 4.9: Magnitude response (top) and error $|H(f) - H_d(f)|$ (bottom) for the system-wide standard bilinear transform (BT), the system-wide parametric bilinear transform (PBT), and our elementwise approach (Elem.) applied to the Helmholtz resonator tree circuit (from Germain and Werner [2017a]). The response of the continuous-time circuit (Analog) is also shown for comparison.

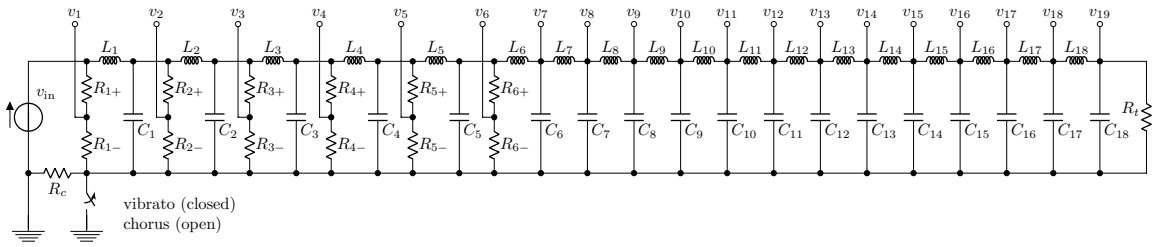


Figure 4.10: Circuit schematic of the Hammond vibrato/chorus circuit.

4.8.3 Hammond vibrato circuit

The results of this case study were originally presented in Germain and Werner [2017b]. Here, we study the LC ladder section of the Hammond organ vibrato/chorus effect circuit [Werner et al. 2016a], i.e., a 19-stage ladder structure terminated by a resistor as seen in Fig. 4.10, and with component values as shown in Tab. 4.14. Our study only investigates the vibrato setting of the circuit, meaning the switch is closed so that the resistor R_c is shorted and can be ignored. The input of the system corresponds to the voltage v_{in} represented here as a voltage source, while the output variables $o^{(m)}$ $(m = 1 \dots 19)$ correspond to the 19 node voltages v_m $(m = 1 \dots 19)$ as labeled in Fig. 4.10. In the full system, these outputs feed into a second section of the circuit, the "scanner" device [Werner et al. 2016a] which consists of a moving rotor attached to a stack of keystone-shaped output plates. This device "scans" successively a set of 16 stacks as it overlaps (fully or partially) with 1 or 2 of these stacks while rotating. The 16 stacks are connected to various subsets of the taps v_m $(m = 1 \dots 19)$, and the subset of connected taps depends on a setting knob with 3 different options [Werner et al. 2016a]. As such, the scanner section essentially performs an interpolation between two of the node voltages v_m $(m = 1 \dots 19)$. It is generally an acceptable approximation to neglect the loading between the LC ladder and the scanner device and model the two separately [Werner et al. 2016a], so that the scanner need not be part of this analysis.

As designed, we learn in Werner et al. [2016a] that the transfer function associated with these outputs corresponds generally to a low-pass filter with a cutoff at roughly 7075 Hz, but with a passband containing a complex pattern of ripples. This pattern of ripples is an essential part of the sonic signature of the device, as, combined with the scanner device, it creates a complex frequency-dependent amplitude modulation that is well-known to users of the device. In Werner et al. [2016a], we also learn that neither the response associated with the system discretized using the standard bilinear transform (see Fig. 4.11a) nor the one associated with the system discretized using the parametric bilinear transform for which T was chosen to match the responses at the cutoff frequency 7075 Hz (see Fig. 4.11b) properly match the transfer functions in the passband for a model sampled at $T_s = 48\,\text{kHz}$, with both methods distorting the intended pattern of frequency-dependent modulation. In order to reduce that distortion without resorting to oversampling, we show how applying our approach allows to better match the system transfer functions through the use of jointly optimized elementwise parametric bilinear transforms.

Implementation considerations

The implementation of the circuit is coded in MATLAB 2016b. In order to compute the different integral quantities (error functions and their gradients as in Sec. 4.7.5), we use the Matlab function `integral` [8] [Shampine 2008] with default options. For the optimization, we use the Matlab function

[8] https://www.mathworks.com/help/matlab/ref/integral.html

k	1	2	3	4	5	6	7	8	9
L_ℓ	24.852	22.298	22.575	20.400	21.033	21.926	22.075	21.723	21.661
C_k	18.107	20.962	26.161	21.771	20.974	21.551	22.082	22.011	21.713
k	10	11	12	13	14	15	16	17	18
L_k	22.153	22.376	21.979	21.625	21.567	21.968	22.534	23.631	18.128
C_k	21.990	22.486	22.340	21.960	22.326	22.437	22.387	22.447	20.365

Table 4.15: Elementwise T coefficients assigned to each reactance of the Hammond vibrato circuit after joint optimization for the ℓ^2 error function as found in Germain and Werner [2017b]. The coefficients are displayed in µs, rounded to five significant figures.

k	1	2	3	4	5	6	7	8	9
L_k	22.023	22.227	22.401	22.441	22.466	22.634	22.911	24.393	20.084
C_k	20.863	20.917	20.959	20.917	20.990	20.939	20.848	21.114	20.912
k	10	11	12	13	14	15	16	17	18
L_k	23.412	24.757	20.555	21.230	22.601	22.228	21.696	21.649	21.840
C_k	13.870	21.222	22.792	20.150	21.658	21.972	21.205	20.819	20.801

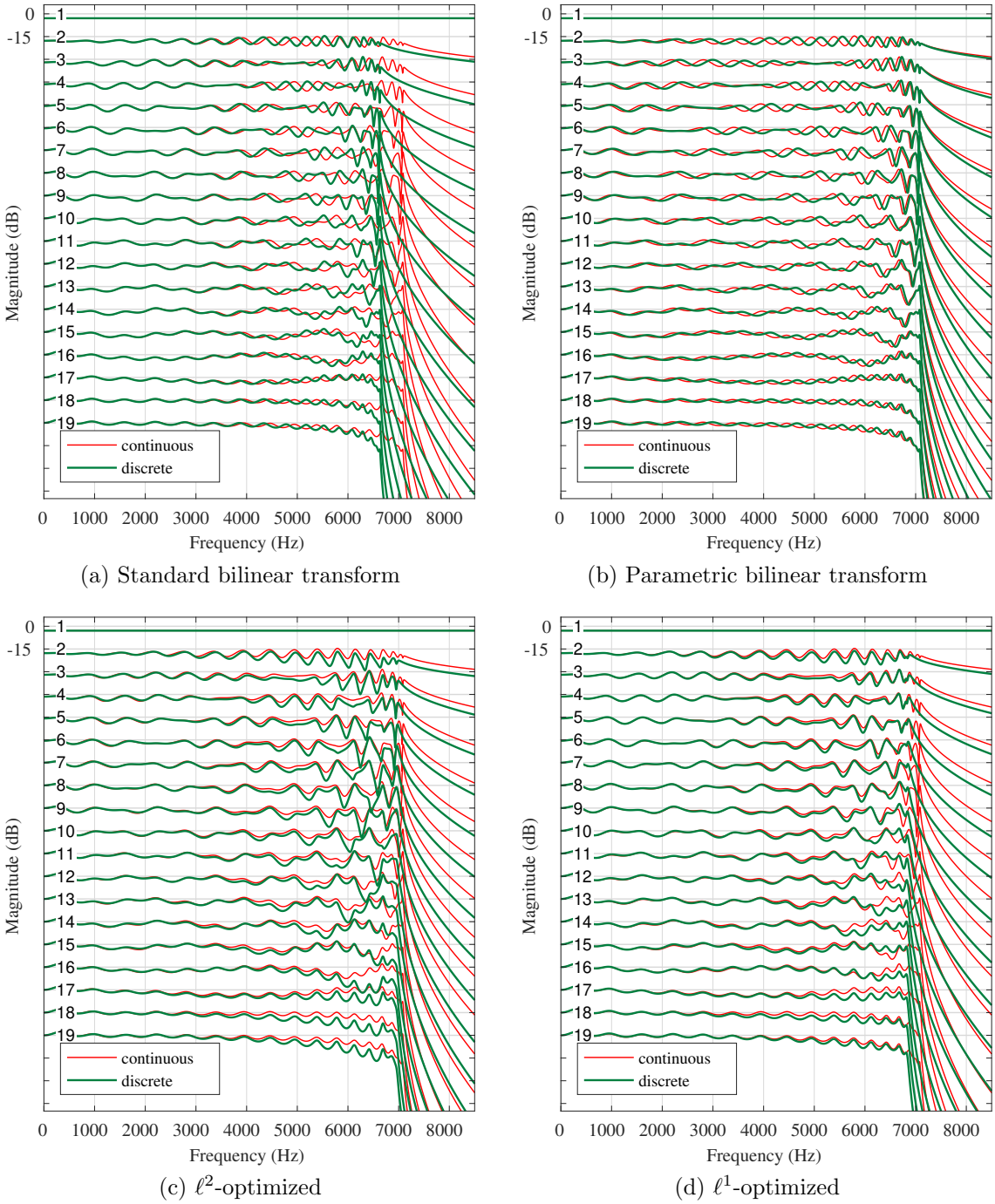
Table 4.16: Elementwise T coefficients assigned to each reactance of the Hammond vibrato circuit after joint optimization for the ℓ^1 error function, as found in Germain and Werner [2017b]. The coefficients are displayed in µs, rounded to five significant figures.

`fmincon` [9] [Byrd et al. 2000, Waltz et al. 2006] to perform the iterative error minimization, with default options, using the known analytical gradient expression, and setting the constraints so that the elementwise coefficients T of the parametric bilinear transforms are strictly positive for all elements. We set the sampling frequency at 48 kHz as in Werner et al. [2016a]. The optimization is performed over the frequency range $[20\,\text{Hz}, 20\,\text{kHz}]$, which, again, represents the frequency range of interest for audio applications. For the initial point of the optimization process, we set all the elementwise coefficients T as equal to the standard bilinear transform solution $T_s = 20.833\,\mu\text{s}$.

Discussion

The elementwise coefficients T obtained optimizing for (respectively) the ℓ^2 and ℓ^1 error functions are shown in (respectively) Tabs. 4.15 and 4.16. We can see that some of these coefficients differ substantially from the solution obtained for the system-wide standard bilinear transform (i.e., with the elementwise coefficients T all equal to $T_s = 20.833\,\mu\text{s}$) and the solution for the system-wide parametric bilinear transform for which the cutoff frequency response $f = 7075\,\text{Hz}$ is matched (i.e., with the elementwise coefficients T all equal to $22.462\,\mu\text{s}$). The magnitude response corresponding to the 19 outputs for each of the 4 methods are shown in Fig. 4.11 alongside the target continuous-time

[9] https://www.mathworks.com/help/matlab/ref/fmincon.html



(a) Standard bilinear transform

(b) Parametric bilinear transform

(c) ℓ^2-optimized

(d) ℓ^1-optimized

Figure 4.11: Responses of the LC ladder at the 19 tap indices, using (a) the standard bilinear transform, (b) the parametric bilinear transform set to match the circuit's cutoff frequency at $f = 7075\,\text{Hz}$, and using elementwise parametric bilinear transforms optimized for (c) the ℓ^2 and (d) the ℓ^1 error functions. For readability, the taps are offset in 15dB increments.

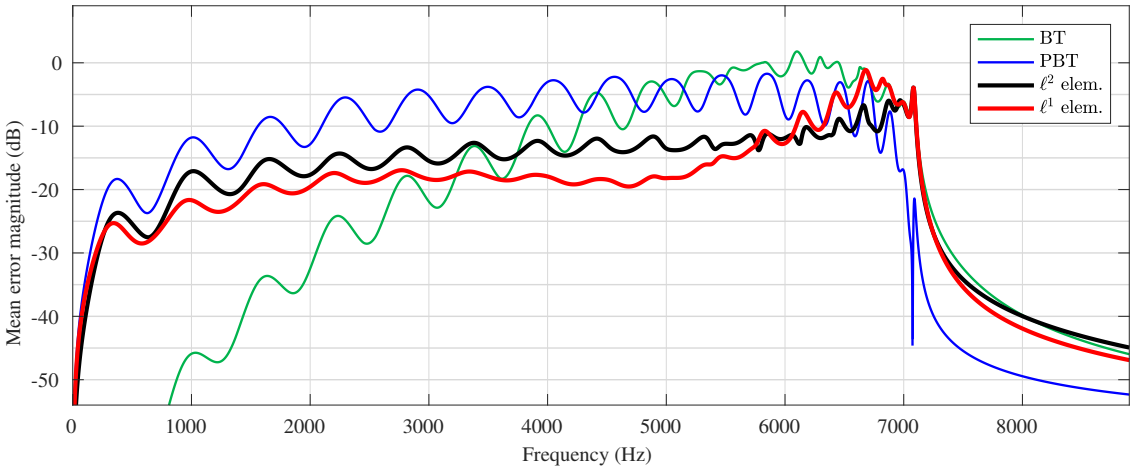


Figure 4.12: Mean value of the error magnitudes $|H^{(m)}(f) - H_d^{(m)}(f)|$ across all 19 output voltage nodes of the Hammond vibrato/chorus circuit at each frequency f for the standard bilinear transform (BT), the parametric bilinear transform (PBT), and for elementwise parametric bilinear transforms optimized for the ℓ^2 and ℓ^1 error functions.

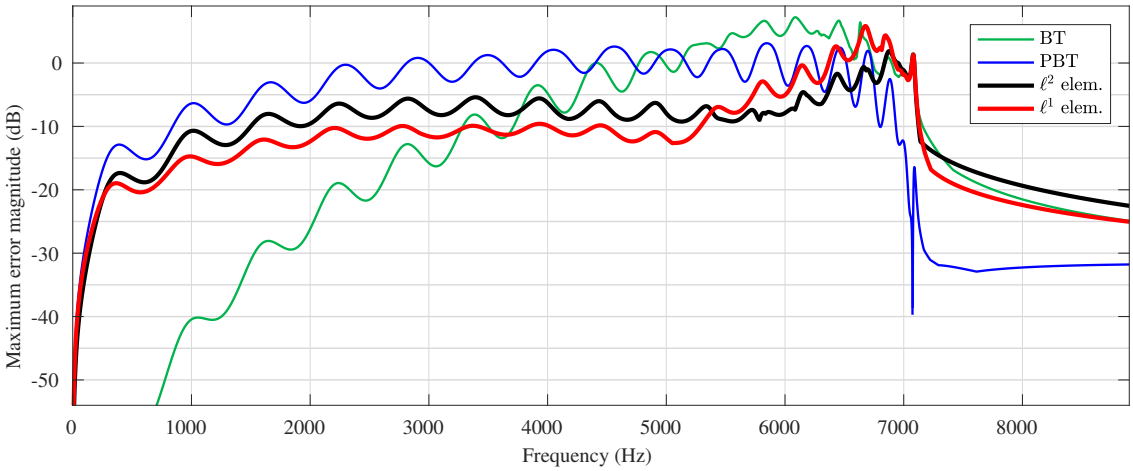


Figure 4.13: Maximum value of the error magnitudes $|H^{(m)}(f) - H_d^{(m)}(f)|$ across all 19 output voltage nodes of the Hammond vibrato/chorus circuit at each frequency f for the standard bilinear transform (BT), the parametric bilinear transform (PBT), and for elementwise parametric bilinear transforms optimized for the ℓ^2 and ℓ^1 error functions.

system response. To better visualize the error, we also show (respectively) the mean and maximum value of the error $|H^{(m)}(f) - H_d^{(m)}(f)|$ across frequencies for all 4 methods in (respectively) Figs. 4.12 and 4.13. As expected, for the standard bilinear transform, the error starts at zero and grows with frequency due to the warping distortion, with a rather large error on the upper half of the passband. By design, the parametric bilinear transform matches perfectly the response at the cutoff frequency, but the error in the passband is much greater due to increased warping effect. Finally, we see that both optimized methods significantly lower and spread the error across the passband, with the exception of the region around the cutoff frequency where the residual error is most significant. In more detail, the ℓ^2 optimization appears to spread the error more evenly up to the cutoff frequency, while the ℓ^1 optimization lowers more significantly the error across most of the passband, while leaving more error around the cutoff frequency. As such, both optimized approach manage to achieve our objective of significantly lowering the response error without oversampling and without added computation cost at runtime. The user only need to select the preferred error function depending on their context, i.e., depending on the desired trade-off for the discretized system.

4.9 Mappings, equivalent elements and stability

4.9.1 Equivalent bilinear elements

It is known that applying discretizations following the form of the α-transform to a circuit can be interpreted as adding "numerical" resistors around the reactances of the original circuit and then discretize it using the trapezoidal method, i.e., the standard bilinear transform [Gao et al. 2003]. A similar observation was made in the wave digital filter literature for various classes of discretization methods (i.e., implicit Runge-Kutta methods) in order to prove the passivity, and equivalently the passivity of numerical methods. That information was leveraged to generate equivalent wave digital filter structures for these methods using the basic wave digital filter elements (e.g., delays, series and parallel connections, gyrators, etc...) [Fränken and Ochs 2001a,b, 2002]. Similarly, we can show that all three mappings of interest can be interpreted similarly, which has consequences in understanding the stability properties of our approach.

Identifying the "bilinear" equivalent of a discretized element goes as follows:

1. Express the discretized constitutive equation of the element according to the chosen mapping,
2. Factor out the standard bilinear transform mapping,
3. Swap the standard bilinear transform mapping for a continuous-time differentiation to form an equivalent continuous-time constitutive equation, and
4. Interpret the resulting equation in terms of passive electrical component values (i.e., resistance, capacitance, inductance).

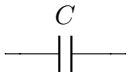
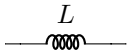
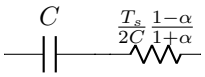
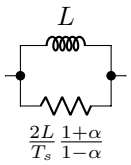
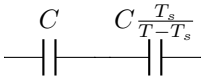
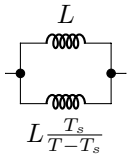
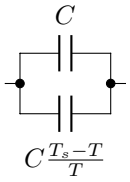
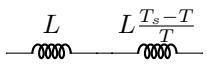
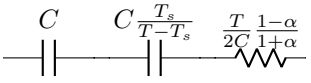
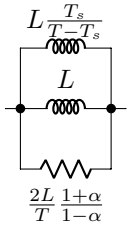
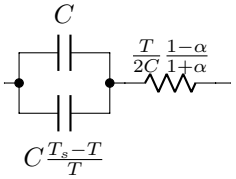
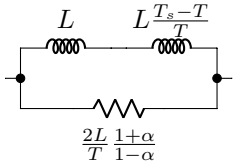
Discretization		Capacitor (Capacitance C)	Inductor (Inductance C)
Standard bilinear transform			
α-transform			
Parametric bilinear transform	$T > T_s$		
	$T < T_s$		
Parametric α-transform	$T > T_s$		
	$T < T_s$		

Table 4.17: Equivalent bilinear elements of a capacitor of capacitance C and an inductor of inductance L for the standard bilinear transform, the α-transform, the parametric bilinear transform and the parametric α-transform.

In this section, we apply that procedure to typical electrical impedances. The resulting bilinear equivalents are summarized in Tab. 4.17.

Resistor

Since the resistor is a static impedance, we have already seen that its constitutive equation does not involve the mapping at all. As such, isolated resistors are unaffected by the mapping, meaning they are equivalent to themselves for all mappings.

Capacitor

When we use the parametric bilinear transform on a capacitor, its discretized constitutive equation becomes

$$\frac{2C}{T}\frac{1-z^{-1}}{1+z^{-1}}V_d(z) = I_d(z) \tag{4.120}$$

where V_d and I_d are the z-transform of the discrete-time voltage across and discrete-time current through the element. This equation can straightforwardly be rewritten as

$$\frac{V_d(z)}{I_d(z)} = \frac{T}{CT_s}\underbrace{\left(\frac{T_s}{C}\frac{1+z^{-1}}{1-z^{-1}}\right)}_{\substack{\text{standard bilinear}\\ \text{mapping}}} \tag{4.121}$$

so that we can swap the standard bilinear mapping for the continuous-time differentiation s to form an equivalent continuous-time constitutive equation

$$\frac{V(s)}{I(s)} = \frac{T}{T_s}\frac{1}{sC} \tag{4.122}$$

where V and I are the Laplace transform of the continuous-time voltage across and continuous-time current through the element. Here, we finally see that for the parametric bilinear transform, a capacitor of capacitance C has for bilinear equivalent a capacitor of capacitance CT_s/T.

Furthermore, we can distinguish two distinct cases: $T > T_s$ and $T < T_s$, since for the case T_s, the element is trivially equivalent to itself. For the case $T > T_s$, we can rewrite the constitutive equation as

$$\frac{V(s)}{I(s)} = \frac{1}{sC} + \frac{1}{sC}\underbrace{\frac{T-T_s}{T_s}}_{\geq 0}. \tag{4.123}$$

As a result, we can interpret the equivalent element for the parametric bilinear transform mapping as adding a "numerical" capacitor of capacitance $C\frac{T_s}{T-T_s}$ in series with the original capacitor. For

the case $T < T_s$, we get instead

$$\frac{I(s)}{V(s)} = sC\frac{T_s}{T} = sC + sC\underbrace{\frac{T_s - T}{T}}_{\geq 0} \tag{4.124}$$

so that we can interpret the equivalent element for the parametric bilinear transform mapping as adding a "numerical" capacitor of capacitance $C\frac{T_s-T}{T}$ in parallel with the original capacitor.

In the case of the α-transform, its discretized constitutive equation becomes

$$\frac{C(1+\alpha)}{T_s}\frac{1-z^{-1}}{1+\alpha z^{-1}}V_d(z) = I_d(z) \tag{4.125}$$

which we can write as

$$\frac{V_d(z)}{I_d(z)} = \frac{T_s}{2C}\frac{1+z^{-1}}{1-z^{-1}} + \frac{T_s}{2C}\frac{1-\alpha}{1+\alpha} \tag{4.126}$$

leading to the bilinear equivalent constitutive equation

$$\frac{V(s)}{I(s)} = \frac{1}{sC} + \frac{T_s}{2C}\frac{1-\alpha}{1+\alpha}. \tag{4.127}$$

As such, we see that, for $-1 < \alpha \leq 1$, the capacitor of capacitance C has for bilinear equivalent the same capacitor in series with a "numerical" resistor of resistance $\frac{T_s}{2C}\frac{1-\alpha}{1+\alpha}$, which is consistent with the observations in Gao et al. [2003]. For other values of α, we have $\frac{1-\alpha}{1+\alpha} < 0$. Since we cannot have negative resistances, we can't find a satisfactory way of formulating a bilinear equivalent from only passive electrical elements in this case, we'll see later how that observation ties into the issue of stability.

Finally, for the case of the parametric α-transform, the capacitor of capacitance C has for bilinear equivalent the capacitor of capacitance CT_s/T in series with a resistor of resistance $\frac{T}{2C}\frac{1-\alpha}{1+\alpha}$. Here again, we find that the equivalence only functions when $-1 < \alpha \leq 1$ to ensure that the resistance is positive. We can also express these equivalent as follows:

- for $T > T_s$, we get the original capacitor in series with a "numerical" capacitor of capacitance $C\frac{T_s}{T-T_s}$ and a "numerical" resistor of resistance $\frac{T}{2C}\frac{1-\alpha}{1+\alpha}$, and
- for $T_s > T$, we get the original capacitor in parallel with a "numerical" capacitor of capacitance $C\frac{T}{T_s-T}$ and then that group in series a "numerical" resistor of resistance $\frac{T}{2C}\frac{1-\alpha}{1+\alpha}$.

Inductor

Using similar calculations as for the capacitor, we get the following:

- For the parametric bilinear transform, the bilinear equivalent of the inductor of inductance L is an inductor of inductance LT_s/T, or equivalently:

  - for $T > T_s$, the inductor itself in parallel with a "numerical" inductor of inductance $L\frac{T_s}{T-T_s}$, and
  - for $T < T_s$, the inductor itself in series with a "numerical" inductor of inductance $L\frac{T_s-T}{T}$.

- For the α-transform, and for $-1 < \alpha \leq 1$, the bilinear equivalent of the inductor is the inductor itself in parallel with a "numerical" resistor of resistance $\frac{2L}{T_s}\frac{1+\alpha}{1-\alpha}$, no equivalent exists for other values of α.
- For the parametric α-transform, and for $-1 < \alpha \leq 1$, the bilinear equivalent of the inductor is the inductor of inductance LT_s/T in parallel with a "numerical" resistor of resistance $\frac{2L}{T}\frac{1+\alpha}{1-\alpha}$, no equivalent exists for other values of α. Here again, we can also express it as:
  - for $T > T_s$, the inductor itself in parallel with a "numerical" inductor of inductance $L\frac{T_s}{T-T_s}$ and a "numerical" resistor of resistance $\frac{2L}{T}\frac{1+\alpha}{1-\alpha}$, and
  - for $T < T_s$, the inductor itself in series with a "numerical" inductor of inductance $L\frac{T_s-T}{T}$, the both of them in parallel with a "numerical" resistor of resistance $\frac{2L}{T}\frac{1+\alpha}{1-\alpha}$.

4.9.2 Bilinear equivalent and stability of elementwise discretization

The research in Fränken and Ochs [2001a,b, 2002] discusses how typical discretization methods such as implicit Runge-Kutta methods can be shown to be A-stable if and only if they are guaranteed to discretize any passive electrical circuit into a bilinear equivalent circuit of passive elements as well. We can extend that analysis here to find that all parametric α-transforms for which $-1 < \alpha \leq 1$ will be A-stable, which confirms what we already found when discussing first-order s-to-z mappings in Ch. 3. More importantly, we see that we can assert that our elementwise discretization approach is guaranteed to be an A-stable approach as long as each individual element is discretized using an A-stable mapping, even though they differ between elements. As such, we see that the passivity criterion actually allows us to study the *global* stability of the system through the *local* stability of the discretizations of individual elements, in the following sense: Having all electrical elements of a passive circuit be individually discretized using A-stable elementwise discretization methods is a sufficient condition to guarantee the stability of the resulting discrete-time circuit model.

4.10 Circuit component value optimization

4.10.1 Computational circuit design

The method outlined in this chapter presents some similarities with the thread of research in the literature around the problem of linear circuit design and sensitivity analysis [Vlach and Singhal 1993]. The problem of circuit design corresponds to finding the component values of a circuit, given a chosen circuit topology. The component values are aimed at achieving an arbitrary target frequency response as closely as possible.

4.10.2 Sensitivity analysis

The problem of sensitivity analysis corresponds to the analysis of the variation of the circuit response in case the physical component values in a real-world implementation of a circuit design. Indeed, electrical components are known to generally have non-trivial tolerances with respect to their nominal physical characteristics (e.g., capacitance, inductance, resistance), meaning it is highly beneficial to favor circuit designs that result in a low sensitivity to component value error. Both problems require to formalize the dependency of the circuit's response to its individual component values, similarly as here where we show how the computer model's response depends on the individual components' discretization formulas.

4.10.3 Component value inference

Finally, our approach also shares some conceptual similarities with recent research that aims at modifying the reported component values (e.g., transistor characteristics) in an audio circuit schematic, when the ideal response obtained from the direct implementation of that schematic does not match the expected response from the circuit [Holmes and van Walstijn 2016]. The objective is then to optimize the parameters of these components in order to obtain empirically the real-world component characteristics, and, as a result, to allow for an accurate modeling of the effect.

4.11 Conclusion

In this chapter, we presented a framework to design accurate and computationally efficient models of linear lumped audio systems. While typical physical modeling approaches rely on applying a discretization method globally to the system equations, we propose here to apply elementwise discretization methods to different elements of the original system. This approach is meant to expose parameters (i.e., degrees of freedom) in the respective discretization of each element, allowing for the joint optimization of these parameters in order to improve the response of the system model. We apply that principle to the case of *RLC* audio circuits, and present an approach to efficiently identify the contribution of each reactive element and its discretized version to the response of the original system and its discrete-time model based on the sparse tableau analysis of that system. That expression can then be fed to numerical solvers to tune the discretization of each reactive element and achieve a better response. Through case studies, we show how using this approach through the use of elementwise parametric bilinear transforms among the various reactances of three circuits of interest allows the generation of models with significantly improved accuracy over the global application of any parametric bilinear transform discretization, and identical computational cost at runtime.

Chapter 5

Conclusions

In this document, we presented tools for designing and analyzing physical models of lumped audio systems at a fixed rate, with the general objective of avoiding the use of oversampling while maintaining or improving the model accuracy compared to typical approaches.

In the first chapter, we discussed a framework outlining a well-defined process to compare continuous-time systems and their associated variables to their discrete-time models and their associated variables. In particular, it emphasizes a generalized description of aliasing as the by-product of the intrinsic ambiguity of converting from a discrete-time sequence to a continuous-time signal (interpolation) and back (projection). In the second chapter, we described mathematical tools aimed at characterizing the amount of aliasing generated by different discretization models both analytically and empirically. The analytical analysis was an extension of the mathematical work presented in Thornburg [1999] extended to arbitrary periodic waveforms and some classes of dynamic systems. The empirical analysis is based on the principles of the "harmonic balance" methods [Urabe 1965]. It allows for the efficient comparison of the harmonic components of the expected response of a continuous-time system and the simulated response of its model(s). In the third chapter, we presented a class of discretization methods based on Möbius transformations. This class generalizes well-known discretization methods used in the context of lumped audio system modeling such as the bilinear transform or the backward Euler method. The chapter describes the desirable properties of such methods and outlines design strategies for nonlinear systems. Finally, in the fourth chapter, we presented an approach to the design of optimized models of linear lumped audio systems. This approach applies different discretization schemes to the various dynamic elements (e.g., reactances) of the system. The free parameters of these schemes are then jointly optimized by explicitly minimizing the transfer function error using traditional optimization methods.

Altogether, these results and the future work around them provide useful insight on the proper approach to design efficient and accurate models of lumped audio systems of interest, and generally advance the field of virtual analog modeling.

First chapter contributions In chapter 1, we formalized the relationship between continuous-time and discrete-time signals flowing in and out of (respectively) continuous-time systems and their discrete-time model(s). To do so, we presented a strict definition of the continuous-to-discrete and the discrete-to-continuous conversion based (respectively) on the projection-sampling operation and on interpolation operation. We illustrated these operations for typical classes of continuous-time signals (i.e., band-limited, piecewise constant, piecewise linear). We showed how that theoretical framework can be related to the general practice in analog-to-digital and digital-to-analog conversion. We also showed how the framework ends up aligning with basic concepts of perceptual transparency for audio when we set the simulation rate of the discrete-time models to typical values such as 44.1 kHz or 48 kHz. We showed how the framework can be applied to form well-posed comparisons between continuous-time systems and their discrete-time models by properly pre- and post-processing the quantities in and out of them. Finally, we illustrated the framework by examining its implications in the simple cases of a linear time-invariant system and or a static square distortion, and how the choice of the projection space for the discrete-time sequence can affect the modeling strategy.

Second chapter contributions In chapter 2, we re-derived the unpublished proof mentioned in Thornburg [1999] to analytically express the weights (amplitude and phase) of the harmonics in the output of a static nonlinearity for a cosinusoidal input, and we proposed an alternative proof. We derived a similar analytical expression for the weights of the harmonics in the case of an arbitrary band-limited periodic input, and an arbitrary periodic input. We then showed how a similar analysis could be applied to the analysis of the weights of the harmonics of the recent anti-derivative anti-aliasing methods [Parker et al. 2016] using the first-order method as an example. We also showed how find an approximate estimate of the harmonic weights for a discrete-time sequence by approximating it as a truncated Fourier series. We followed by expanding to the case of (non-autonomous) dynamical nonlinear systems (i.e., systems described by ordinary differential equations or differential algebraic equations), and showed how the analytical analysis can be extended to the particular case of systems in the form described in Yeh et al. [2010]. Finally, the main contribution of the chapter was to show how the "harmonic balance" methods [Urabe 1965, Gilmore and Steer 1991a,b] can be re-purposed to perform an efficient empirical estimation of the weights of the output signal harmonics of many audio dynamical systems of interest whose input signal is periodic, approximating the output by explicitly projecting it onto a truncated Fourier series. We also demonstrated how the same approach can be applied to the case of discrete-time models, where the output discrete-time sequence can similarly be projected on a truncated Fourier series. As proof of concept, we showed how to derive the equations to solve in the case of the bilinear transform, and we presented a case study where the output harmonic structure of a simple audio diode clipper for the continuous-time system and three typical discretization approaches were computed and compared for a cosinusoidal input.

Third chapter contributions In chapter 3, we presented the class of discretization based on Möbius transformations. We described how a given transformation can be related to a linear multistep discretization method. We showed how a linear transfer function would get converted under any arbitrary transformation. We described how typical constraints on the mapping between the s-plane and the z-plane (real-axis symmetry, frequency sign conservation, DC mapping, zero damping mapping) translates into constraints on the coefficients. We derived the closed-form expression of the iso-contours for frequency and damping after they get mapped from the s-plane to the z-plane, or from the z-plane to the s-plane. We described the conditions for bounded-input bounded-output stability, A-stability and L-stability for a given transformation. We also described how to determine consistency and order of convergence, and we extended that definition to the case of transformations with dependent coefficients, to accommodate for approaches such as the parametric bilinear transform where the coefficients are dependent on the sampling rate and any other relevant quantity of the system (e.g., a particular pole location). We presented several subclasses of mapping parametrizations that reduce the number of free parameters from three in the general formulation to one or two, and we explained how each of these subclasses was a generalization of well-known methods such as the bilinear transform. We described qualitatively the influence of the free parameters in these subclasses on the mapping of the iso-contours for frequency from the s-plane to the z-plane. We added to previously proposed design criteria for transformations with dependent coefficients and we presented our novel criterion based on damping monotonicity conservation. On the contrary to these other criteria, it focuses on verifying a qualitative property (the damping monotonicity) over an entire area of interest in the s- and z-planes. In particular, we showed how this criterion dictated the choice of the one free parameter in the case of the α-transform subclass. We also proved all the known criteria for designing transformations with dependent coefficients are second-order accurate discretization methods, since all of them converge asymptotically to the bilinear transform as the sampling period vanishes. We described a strategy to apply the criteria to the case of nonlinear systems by observing their instantaneous poles. In particular, we proposed a strategy how to efficiently parametrize the α-transform under the damping monotonicity condition by performing simple backward Euler simulations of the system of interest under specific regimes. We described how any linear multistep method based on Möbius transformations translated when being applied to the following typical formulation of the equations describing lumped audio systems: a general system of ordinary differential equations, a general system of differential algebraic equations, a system in state-space form, a system derived using the nodal K-method [Yeh et al. 2010], a system derived using the nodal discrete K-method [Yeh et al. 2010] and a system derived using the generalized state-space approach [Holters and Zölzer 2015]. We showed how, echoing the results found in Werner [2016] for wave digital filters, the nodal discrete K-method does not allow for purely implicit or explicit Möbius transformations. We also showed how the condition excluding trivial Möbius transformations (i.e., transformations reduced to a point in space) translates to preventing degenerated cases

in the system equations in the nodal discrete-K method and wave digital filters. We showed how any linear multistep formula based on a Möbius transformation, and any midpoint-like method based on the same transformation are conjugate methods the same way the trapezoidal method and the implicit midpoint method are [Hairer et al. 2006]. This implies that the output of any midpoint-like discrete-time model can be calculated using the discrete-time model derived for the linear multistep formula using the same transformation. We detailed three cases studies. We completed the case studies of the TR-808 pulse shaper and the DOD FX-25 envelope follower (respectively from Werner [2016] and Bogason [2018]) with the addition of an analytical analysis of the instantaneous poles and the application of our empirical approach to the design of an appropriate α-transform. We also applied our methods to the Keio MP-7 bass voice circuit [Keio Electronic Laboratory Corporation 1966a,b]. In all three cases, our approach successfully eliminated the spurious behavior exhibited by methods such as the standard bilinear transform and reduced simulation error.

Fourth chapter contributions In Chapter 4, we formulated an optimization problem aimed at improving the response of discrete-time models of linear time-invariant systems using the particular example of systems that can be expressed in the form of an RLC electrical circuit. The method leverages the use of discretization methods that possess one or more free parameters such as the one presented in Ch. 3. To allow for significant improvements of the model's response, we proposed setting the discretization parameters independently for each reactance (i.e., dynamical element) in the target system. We showed how to set the optimization problem in closed form from the sparse Tableau analysis [Vlach 2002] of the system, for both the single-output and multiple-output cases, allowing explicit access to the contribution of the response of each reactance and its discretized form to the system's and the models' transfer functions, for any arbitrary input or output variables (branch current, branch or node voltage, wave variable and any linear combination of these). This property allows for a direct manipulation of the discretization free parameters and an easy derivation of the optimization gradients. We showed how the optimization problem and its gradient can also be derived in the context of wave variables, when the circuit elements are described as one-port wave-variable elements (i.e., filters) rather than as voltage-current filters, with the same explicit access to each element's individual influence on the transfer function of the system. We derived the expression for the scattering of any arbitrary "network adaptor" (i.e., the adaptor exchanging the wave quantities between the circuit's one-port elements), showing how a closed-form expression (function of the port resistances of all its ports and the circuit's incidence matrix) can be found from a modified sparse Tableau analysis. We also found the closed-form expression for the adaptation of the one constrained port resistance as a function of the circuit incidence matrix and the rest of the port resistances of the network adaptor. We showed how to express the optimization function and its gradient in the context of typical error metrics (ℓ^2 and ℓ^1), and how to add typical regularization terms to impose constraints on the model free parameters. To illustrate and apply the optimization framework, we applied the approach to the case of circuits discretized using parametric bilinear

transforms with elementwise parameters among reactances. We described three case studies: a simple RLC series resonator, a three-resonance Helmholtz tree resonator circuit [Paiva and Välimäki 2012], and finally the full 36-pole 19-output Hammond vibrato circuit, showing significant improvement in the optimized discrete-time model response compared to all baselines. And we concluded the chapter by showing how, using equivalent bilinear elements, we were able to prove that using stable methods for each individual elements was a sufficient condition for the stability of the entire model, due to the fact that any model formed using that method verifies the sufficient passivity criterion described in Fränken and Ochs [2001a].

5.1 Final comments

Lumped audio system modeling is one of the core fields of music technology as a true blend of music practice and science. A part of the music practice has always been the understanding, emulation and improvement on the techniques and aesthetics of their predecessors. Virtual analog research has carried that torch, bringing that practice to the digital age. Physical modeling in particular has allowed for the most intuitive approach to that idea, by embedding at its core the understanding of the sound-generating principles of historical systems. As for all fields dealing with numerical methods, lumped audio modeling has had to work around stringent computational limitations. And while all fields have naturally striven to improve speed, the audio field always had a singular focus on achieving real-time and low latency modeling, a unique philosophy that has inspired decades of innovations.

White-box modeling research has usually revolved around three topics: model formalism (e.g., state-space models, wave digital filters), efficient root finding/inversion and discretization methods. This text presents a contribution to this last one, joining other recent and decisive developments from an always expanding body of research. Combined with the never-ending progress in computational power (e.g., graphics processing units) and its portability (e.g., tablets and mobile phones), virtual analog is making strides towards its goals: understanding historical music systems, preserving and democratizing their sound and aesthetics, and laying out the tools for re-inventing and expanding them, following in the steps of instrument makers and circuit-bending adepts. The present findings are here to help better understand the challenges of lumped audio model discretization, and offer novel tools and solutions to analyze and overcome them. I hope it inspires present and future researchers to push always further the limits of this wonderful field.

Appendix A

Fourier transform of the Bessel functions of the first kind

The nth-order Bessel function of the first kind J_n can be defined through the integral representation [Olver et al. 2010]

$$J_n(x) = \frac{1}{\pi}\int_0^\pi \cos(n\tau - x\sin\tau)d\tau. \tag{A.1}$$

This definition can be leveraged to find the Fourier transform of J_n. To do so, we need to perform the change of variable $\omega = \sin\tau$, or

$$\tau = \begin{cases} \arcsin\omega & \text{for } \tau \in [0, \pi/2], \\ \pi - \arcsin\omega & \text{for } \tau \in]\pi/2, \pi], \end{cases} \tag{A.2}$$

so that

$$\frac{d\tau}{d\omega} = \begin{cases} \frac{1}{\sqrt{1-\omega^2}} & \text{for } \tau \in [0, \pi/2], \\ \frac{-1}{\sqrt{1-\omega^2}} & \text{for } \tau \in]\pi/2, \pi]. \end{cases} \tag{A.3}$$

Then we have

$$\begin{aligned} J_n(x) &= \frac{1}{\pi}\int_0^\pi \cos(n\tau - x\sin\tau)d\tau \\ &= \frac{1}{\pi}\left[\int_0^{\pi/2} \cos(n\tau - x\sin\tau)d\tau + \int_{\pi/2}^\pi \cos(n\tau - x\sin\tau)d\tau\right] \\ &= \frac{1}{\pi}\left[\int_0^1 \frac{\cos(n\arcsin\omega - x\omega)}{\sqrt{1-\omega^2}}d\omega + \int_0^1 \frac{\cos(n\pi - n\arcsin\omega - x\omega)}{\sqrt{1-\omega^2}}d\omega\right] \\ &= \frac{1}{\pi}\left[\int_0^1 \frac{\cos(n\arcsin\omega - x\omega) + (-1)^n\cos(n\arcsin\omega + x\omega)}{\sqrt{1-\omega^2}}d\omega\right]. \end{aligned} \tag{A.4}$$

Using trigonometric identities, for $n = 2m$ even, we have

$$\begin{aligned} J_n(x) &= \frac{2}{\pi}\left[\int_0^1 \frac{\cos(n \arcsin\omega)\cos(x\omega)}{\sqrt{1-\omega^2}} d\omega\right] \\ &= \frac{1}{\pi}\left[\int_{-1}^1 (-j)^n \frac{T_n(\omega)}{\sqrt{1-\omega^2}} e^{jx\omega} d\omega\right] \end{aligned} \tag{A.5}$$

where T_n is the nth-order Chebyshev polynomial of the first kind, and for $n = 2m+1$ odd, we have

$$\begin{aligned} J_n(x) &= \frac{2}{\pi}\left[\int_0^1 \frac{\sin(n \arcsin\omega)\sin(x\omega)}{\sqrt{1-\omega^2}} d\omega\right] \\ &= \frac{1}{\pi}\left[\int_{-1}^1 (-j)^n \frac{T_n(\omega)}{\sqrt{1-\omega^2}} e^{jx\omega} d\omega\right]. \end{aligned} \tag{A.6}$$

From Eqs. (A.5) and (A.6), we conclude that the Fourier transform $\mathrm{FT}\{J_n\}$ of the nth-order Bessel function of the first kind is given by

$$\mathrm{FT}\{J_n\}(\omega) = \mathbf{1}_{[-1,1]}(\omega) \frac{(-j)^n}{\pi} \frac{T_n(\omega)}{\sqrt{1-\omega^2}}. \tag{A.7}$$

Bibliography

Abel, Jonathan S., Sean Coffin, and Kyle S. Spratt. A modal architecture for artificial reverberation. *The Journal of the Acoustical Society of America (JASA)*, 134(5):4220, 2013.

Al Alaoui, Mohamad Adnan. Novel digital integrator and differentiator. *Electronics Letters*, 29(4): 376–378, February 1993.

Al Alaoui, Mohamad Adnan. Al-Alaoui operator and the α-approximation for discretization of analog systems. *Facta Universitatis, Series: Electronics and Energetics*, 19(1):143–146, April 2006.

Al Alaoui, Mohamad Adnan. Al-Alaoui operator and the new transformation polynomials for discretization of analogue systems. *Electrical Engineering*, 90(6):455–467, June 2008.

Al Alaoui, Mohamad Adnan. Class of digital integrators and differentiators. *IET Signal Processing*, 5(2):251–260, April 2011.

Belevitch, Vitold. Summary of the history of circuit theory. *Proceedings of the IRE*, 50(5):848–855, May 1962.

Ben-Israel, Adi and Thomas N. E. Greville. *Generalized inverses: theory and applications*, volume 15. Springer-Verlag, New York, NY, 2nd edition, 2003.

Bevis, Jean H., Frank J. Hall, and Irving J. Katz. Integer generalized inverses of incidence matrices. *Linear Algebra and its Applications*, 39:247–258, 1981.

Bilbao, Stefan, Fabián Esqueda, Julian D. Parker, and Vesa Välimäki. Antiderivative antialiasing for memoryless nonlinearities. *IEEE Signal Processing Letters*, 24(7):1049–1053, July 2017a.

Bilbao, Stefan, Fabian Esqueda, and Vesa Välimäki. Antiderivative antialiasing, lagrange interpolation and spectral flatness. In *Proceedings of the 2017 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*, pages 141–145, 2017b.

Billings, Stephen A. Identification of nonlinear systems - a survey. *IEE Proceedings D—Control Theory and Applications*, 127(6):272–285, November 1980.

Biolek, Dalibor and Viera Biolkova. Algorithmic s-z transformations for continuous-time to discrete-time filter conversion. In *Proceedings of the 2001 IEEE International Symposium on Circuits and Systems (ISCAS)*, volume 1, pages 588–590, May 2001.

Biolkova, Viera and Dalibor Biolek. Generalized Pascal matrix of first order s-z transforms. In *Proceedings of the 6th IEEE International Conference on Electronics, Circuits and Systems (ICECS)*, volume 2, pages 929–931, Paphos, Cyprus, 5–8 September 1999.

Blagouchine, Iaroslav V. and Eric Moreau. Analytic method for the computation of the total harmonic distortion by the Cauchy method of residues. *IEEE Transactions on Communications*, 59(9):2478–2491, September 2011.

Bogason, Ólafur. Modeling auto circuits containing typical nonlinear components with wave digital filters. Master's Thesis, McGill University, 2018.

Bosi, Marina and Richard E. Goldberg. *Introduction to Digital Audio Coding and Standards*. Springer, 2002.

Byrd, Richard H., Jean Charles Gilbert, and Jorge Nocedal. A trust region method based on interior point techniques for nonlinear programming. *Mathematical Programming*, 89(1):149–85, 2000.

Cederbaum, Israel. Some applications of graph theory to network analysis and synthesis. *IEEE Transactions on Circuits and Systems*, 31(1):64–68, January 1984.

Chen, Jen-Yi, Yu-Chun Hsu, Shu-Sheng Lee, Tamal Mukherjee, and Gary K. Fedder. Modeling and simulation of a condenser microphone. In *Proceedings of the 2007 IEEE International Solid-State Sensors, Actuators and Microsystems Conference*, pages 1299–1302, Lyon, France, 10–14 June 2007.

Coleman, Thomas F. and Yuying Li. An interior, trust region approach for nonlinear minimization subject to bounds. *SIAM Journal on Optimization*, 6:418–445, 1996.

Cormen, Thomas H., Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein. *Introduction to Algorithms*. The MIT Press, 3rd edition, 2009.

Dahlquist, Germund G. A special stability problem for linear multistep methods. *BIT Numerical Mathematics*, 3(1):27–43, 1963.

Daly, Paul. A comparison of virtual analogue Moog VCF models. Master's Thesis, University of Edinburgh, Edinburgh, Scotland, August 2012.

D'Angelo, Stefano and Vesa Välimäki. Wave-digital polarity and current inverters and their application to virtual analog audio processing. In *Proceedings of the 2012 IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pages 469–472, Kyoto, Japan, 25–30 March 2012.

D'Angelo, Stefano and Vesa Välimäki. An improved virtual analog model of the Moog ladder filter. In *Proceedings of the 2013 IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pages 729–733, Vancouver, Canada, 26–31 May 2013. IEEE.

D'Angelo, Stefano and Vesa Välimäki. Generalized Moog ladder filter: Part I—linear analysis and parameterization. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 22(12): 1825–1832, December 2014a.

D'Angelo, Stefano and Vesa Välimäki. Generalized Moog ladder filter: Part II—explicit nonlinear model through a novel delay-free loop implementation method. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 22(12):1873–1883, December 2014b.

Dattoli, Giuseppe, Cesare Chiccoli, Silveria Lorenzutta, Giuseppe Maino, Maria Richetta, and Amalia Torre. Fourier expansions and multivariable bessel functions concerning radiation problems. *Radiation Physics and Chemistry*, 47(2):183–189, 1996.

Dempwolf, Kristjan, Martin Holters, Stephan Möller, and Udo Zölzer. The influence of small variations in a simplified guitar amplifier model. In *Proceedings of the 12th International Conference on Digital Audio Effects (DAFx-09)*, Como, Italy, 1–4 September 2009.

Deuflhard, Peter. *Newton Methods for Nonlinear Problems: Affine Invariance and Adaptive Algorithms*. Springer-Verlag, Berlin, Germany, 2011.

Dostál, Tomáš and Jiří Pospíšil. Switched circuits I. Research report FE-58, Department of Radioelectronics, Brno University of Technology, 1985.

Dunkel, W. Ross, Maximilian Rest, Kurt James Werner, Michael Jørgen Olsen, and Julius O. Smith III. The Fender Bassman 5F6-A family of preamplifier circuits—a wave digital filter case study. In *Proceedings of the 19th International Conference on Digital Audio Effects (DAFx-16)*, pages 263–270, Brno, Czech Republic, 5–9 September 2016.

Eichas, Felix and Udo Zölzer. Black-box modeling of distortion circuits with block-oriented models. In *Proceedings of the 19th International Conference on Digital Audio Effects (DAFx-16)*, Brno, Czech Republic, 5–9 September 2016.

Eichas, Felix and Udo Zölzer. Gray-box modeling of guitar amplifiers. *Journal of the Audio Engineering Society (JAES)*, 66(12):1006–1015, 2018.

Eichas, Felix, Stephan Möller, and Udo Zölzer. Block-oriented modeling of distortion audio effects using iterative minimization. In *Proceedings of the 18th International Conference on Digital Audio Effects (DAFx-15)*, pages 243–248, Trondheim, Norway, 30 November–3 December 2015.

Esqueda, Fabián, Henry Pöntynen, Julian D. Parker, and Stefan Bilbao. Virtual analog models of the Lockhart and Serge wavefolders. *Applied Sciences*, 7(12):1328, 2017.

Falaize, Antoine. *Modélisation, simulation, génération de code et correction de systèmes multi-physiques audios: approche par réseau de composants et formulation Hamiltonienne à Ports*. Ph.D. Dissertation, Université Pierre & Marie Curie-Paris 6, January 2017.

Falaize, Antoine and Thomas Hélie. Passive simulation of electrodynamic loudspeaker for guitar amplifiers: a port-Hamiltonian approach. In *Proceedings of the International Symposium on Musical Acoustics (ISMA)*, Le Mans, France, 7–12 July 2014.

Falaize, Antoine and Thomas Hélie. Guaranteed-passive simulation of an electro-mechanical piano: a port-Hamiltonian approach. In *Proceedings of the 18th International Conference on Digital Audio Effects (DAFx-15)*, pages 57–64, Trondheim, Norway, 30 November–3 December 2015.

Falaize, Antoine and Thomas Hélie. Passive guaranteed simulation of analog audio circuits: A port-Hamiltonian approach. *Applied Sciences*, 6(10), 2016. Article #273.

Falaize, Antoine, Nicholas Lopes, Denis Matignon, and Bernhard Maschke. Energy-balanced models for acoustic and audio systems: a port-Hamiltonian approach. In *Proceedings of the Unfold Mechanics for Sounds and Music Colloquium*, Paris, France, 11–12 September 2014.

Falaize, Antoine, Nicolas Papazoglou, Thomas Hélie, and Nicholas Lopes. Compensation of loudspeaker's nonlinearities based on flatness and port-Hamiltonian approach. In *Proceedings of the 22ème Congreès Français de Méchanique*, Lyon, France, 24–28 August 2015.

Falaize-Skrzek, Antoine and Thomas Hélie. Simulation of an analog circuit of a wah pedal: a port-Hamiltonian approach. In *Proceedings of the 135th Convention of the Audio Engineering Society (AES)*, New York, NY, 17–20 October 2013. convention paper #8981.

Farina, Angelo. Advancements in impulse response measurements by sine sweeps. In *Proceedings of the 122th Convention of the Audio Engineering Society (AES)*, May 2007. convention paper #7121.

Fettweis, Alfred. Some principles of designing digital filters imitating classical filter structures. *IEEE Transactions on Circuit Theory*, 18(2):314–316, 1971.

Fettweis, Alfred. Wave digital filters: Theory and practice. *Proceedings of the IEEE*, 74(2):270–327, 1986.

Firestone, Floyd A. A new analogy between mechanical and electrical systems. *The Journal of the Acoustical Society of America (JASA)*, 4:249–267, January 1933.

Fontana, Federico. Preserving the structure of the Moog VCF in the digital domain. In *Proceedings of the International Computer Music Conference (ICMC)*, pages 291–294, Copenhagen, Denmark, 27–31 August 2007.

Foster, Scott. Impulse response measurement using Golay codes. In *Proceedings of the 1986 IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, volume 11, pages 929–932, Tokyo, Japan, 7–11 April 1986.

Fränken, Dietrich and Karlheinz Ochs. Numerical stability properties of passive Runge-Kutta methods. In *Proceedings of the 2001 IEEE International Symposium on Circuits and Systems (ISCAS)*, volume 3, pages 473–476, Sydney, Australia, 6–9 May 2001a.

Fränken, Dietrich and Karlheinz Ochs. Synthesis and design of passive Runge-Kutta methods. *International Journal of Electronics and Communications (AEÜ)*, 55(6):417–425, 2001b.

Fränken, Dietrich and Karlheinz Ochs. Improving wave digital simulation by extrapolation techniques. *AEU International Journal of Electronics and Communications*, 56(5):327–336, 2002.

Fränken, Dietrich, Klaus Meerkötter, and Joachim Waßmuth. Passive parametric modeling of dynamic loudspeakers. *IEEE Transactions on Speech and Audio Processing*, 9(8):885–891, November 2001.

Gander, Walter and Walter Gautschi. Adaptive quadrature—revisited. *BIT Numerical Mathematics*, 40:84–101, 2000.

Gao, Wenzhong, Eugene Solodovnik, Roger A. Dougal, George Cokkinides, and Athanasios P. Sakis Meliopoulos. Elimination of numerical oscillations in power system dynamic simulation. In *Proceeding of the 18th Annual IEEE Applied Power Electronics Conference and Exposition (APEC'03)*, volume 2, pages 790–794, Miami Beach, FL, 9–13 February 2003.

Germain, François G. A nonlinear analysis framework for electronic synthesizer circuits. Master's Thesis, McGill University, Montréal, Canada, October 2011.

Germain, François G. Fixed-rate modeling of audio lumped systems: A comparison between trapezoidal and implicit midpoint methods. In *Proceedings of the 20th International Conference of Digital Audio Effects (DAFx-17)*, pages 168–175, Edinburgh, UK, 5–9 September 2017.

Germain, François G. and Kurt James Werner. Design principles for lumped model discretization using Möbius transforms. In *Proceedings of the 18th International Conference of Digital Audio Effects (DAFx-15)*, pages 371–378, Trondheim, Norway, 30 November–3 December 2015.

Germain, François G. and Kurt James Werner. Joint parameter optimization of differentiated discretization schemes for audio circuits. In *Proceedings of the 142nd Convention of the Audio Engineering Society (AES)*, volume 105, San Francisco, CA, May 2017a. convention paper #9751.

Germain, François G. and Kurt James Werner. Optimizing differentiated discretization for audio circuits beyond driving point transfer functions. In *Proceedings of the 2017 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*, pages 384–388, New Paltz, NY, 15–18 October 2017b.

Gilmore, Rowan J. and Michael B. Steer. Nonlinear circuit analysis using the method of harmonic balance—a review of the art. part i. introductory concepts. *International Journal of Microwave and Millimeter-Wave Computer-Aided Engineering*, 1(1):22–37, 1991a.

Gilmore, Rowan J. and Michael B. Steer. Nonlinear circuit analysis using the method of harmonic balance—a review of the art. ii. advanced concepts. *International Journal of Microwave and Millimeter-Wave Computer-Aided Engineering*, 1(2):159–180, 1991b.

Golden, Roger M. Digital filter synthesis by sampled-data transformation. *IEEE Transactions on Audio and Electroacoustics*, 16(3):321–329, September 1968.

Hairer, Ernst and Gerhard Wanner. *Solving ordinary differential equations II: Stiff and differential-algebraic problems*, volume 14. Springer-Verlag, 2nd edition, 1996.

Hairer, Ernst, Syvert P. Nørsett, and Gerhard Wanner. *Solving ordinary differential equations I: Nonstiff problems*. Springer-Verlag, Berlin, 1993.

Hairer, Ernst, Christian Lubich, and Gerhard Wanner. *Geometric numerical integration: structure-preserving algorithms for ordinary differential equations*, volume 31. Springer Science & Business Media, 2006.

Hart, Peter E., Nils J. Nilsson, and Bertram Raphael. A formal basis for the heuristic determination of minimum cost paths. *IEEE Transactions on Systems Science and Cybernetics*, 4(2):100–107, July 1968.

Hélie, Thomas. On the use of Volterra series for real-time simulations of weakly nonlinear analog audio devices: Application to the Moog ladder filter. In *Proceedings of the 9th International Conference on Digital Audio Effects (DAFx-06)*, volume 9, pages 7–12, Montréal, Canada, 18–20 September 2006.

Hélie, Thomas. Volterra series and state transformation for real-time simulations of audio circuits including saturations: Application to the Moog ladder filter. *IEEE Transactions on Audio, Speech, and Language Processing*, 18(4):747–759, 2010.

Hélie, Thomas. Lyapunov stability analysis of the Moog ladder filter and dissipativity aspects in numerical solutions. In *Proceedings of the 14th International Conference on Digital Audio Effects (DAFx-11)*, pages 19–23, Paris, France, 2011.

Herrmann, Otto E. Design of nonrecursive digital filters with linear phase. *Electronics Letters*, 6 (11):328–329, May 1970.

Herrmann, Otto E. and Hans W. Schuessler. Design of nonrecursive digital filters with minimum phase. *Electronics Letters*, 6(11):329–330, May 1970.

Hochbruck, Marlis and Alexander Ostermann. Exponential integrators. *Acta Numerica*, 19:209–286, 2010.

Hofstetter, Ed, Alan V. Oppenheim, and J. Siegel. On optimum nonrecursive digital filters. In *Proceedings of the 9th Allerton Conference on Circuits and System Theory*, pages 789–798, Monticello, IL, 6–8 October 1971.

Holmes, Ben and Maarten van Walstijn. Improving the robustness of the iterative solver in state-space modelling of guitar distortion circuitry. In *Proceedings of the 18th International Conference on Digital Audio Effects (DAFx-15)*, pages 49–56, Trondheim, Norway, 30 November–3 December 2015.

Holmes, Ben and Maarten van Walstijn. Physical model parameter optimisation for calibrated emulation of the Dallas Rangemaster tremble booster guitar pedal. In *Proceedings of the 19th International Conference on Digital Audio Effects (DAFx-16)*, pages 47–54, Brno, Czech Republic, 5–9 September 2016.

Holters, Martin and Udo Zölzer. A generalized method for the derivation of non-linear state-space models from circuit schematics. In *Proceedings of the 23rd European Signal processing Conference (EUSIPCO)*, pages 1073–1077, Nice, France, 31 August–4 September 2015.

Holters, Martin and Udo Zölzer. A k-d tree based solution cache for the non-linear equation of circuit simulations. In *Proceedings of the 24th European Signal Processing Conference (EUSIPCO)*, pages 1028–1032, Budapest, Hungary, 29 August–2 September 2016.

Hörmander, Lars. *Linear Partial Differential Operators*. Springer-Verlag, Berlin, Germany, 1963.

Huovilainen, Antti. Nonlinear digital implementation of the Moog ladder filter. In *Proceedings of the 7th International Conference on Digital Audio Effects (DAFx-04)*, Naples, Italy, 5–8 October 2004.

Janczak, Andrzej. *Identification of nonlinear systems using neural networks and polynomial models: a block-oriented approach*, volume 310. Springer Science & Business Media, 2004.

Kartofelev, Dmitri, Anatoli Stulov, Heidi-Maria Lehtonen, and Vesa. Välimäki. Modeling a vibrating string terminated against a bridge with arbitrary geometry. In *Proceedings of the Stockholm Music Acoustics Conference*, pages 626–632, Stockholm, Sweden, 30 July–3 August 2013.

Keio Electronic Laboratory Corporation. MP-7 over-all circuit diagram, 1966a.

Keio Electronic Laboratory Corporation. *Mini Pops-7 Owner's Manual*, 1966b.

Kiiski, Roope, Fabián Esqueda, and Vesa Välimäki. Time-variant gray-box modeling of a phaser pedal. In *Proceedings of the 19th International Conference on Digital Audio Effects (DAFx-16)*, pages 31–38, Brno, Czech Republic, 5–9 September 2016.

Laroche, Jean. On the stability of time-varying recursive filters. *Journal of the Audio Engineering Society (JAES)*, 55(6):460–471, 2007.

Le Bihan, Jean. Novel class of digital integrators and differentiators. *Electronics Letters*, 29(11): 971–973, May 1993.

Lehtonen, Heidi-Maria, Jussi Pekonen, and Vesa Välimäki. Audibility of aliasing distortion in sawtooth signals and its implications for oscillator algorithm design. *The Journal of the Acoustical Society of America (JASA)*, 132(4):2721–2733, 2012.

Liniger, Werner. Global accuracy and a-stability of one- and two-step integration formulae for stiff ordinary differential equations. In Morris, John L., editor, *Conference on the Numerical Solution of Differential Equations*, volume 109 of *Lecture Notes in Mathematics*, pages 188–193, Dundee, Scotland, 23–27 June 1969. Springer-Verlag, Berlin, Heidelberg.

Liniger, Werner and Ralph A. Willoughby. Efficient integration methods for stiff systems of ordinary differential equations. *SIAM Journal on Numerical Analysis*, 7(1):47–66, 1970.

Lopes, Nicholas, Thomas Hélie, and Antoine Falaize. Explicit second-order accurate method for the passive guaranteed simulation of port-Hamiltonian system. *IFAC-PapersOnLine*, 48(13):223–228, 2015.

Mačák, Jaromír. *Real-Time Digital Simulation of Guitar Amplifiers as Audio Effects*. Ph.D. Dissertation, Brno University of Technology, Brno, Czech Republic, 2012.

Mačák, Jaromír and Jiri Schimmel. Real-time guitar tube amplifier simulation using an approximation of differential equations. In *Proceedings of the 13th International Conference on Digital Audio Effects (DAFx-10)*, Granz, Austria, 6–10 September 2010.

Mačák, Jaromír and Jiri Schimmel. Simulation of a vaccuum-tube push-pull guitar power amplifier. In *Proceedings of the 14th International Conference on Digital Audio Effects (DAFx-11)*, volume 14, pages 59–62, Paris, France, 19–23 September 2011.

Maestre, Esteban, Gary P. Scavone, and Julius O. Smith III. Modeling a vibrating string terminated against a bridge with arbitrary geometry. In *Proceedings of the Stockholm Music Acoustics Conference*, pages 626–632, Stockholm, Sweden, 30 July–3 August 2013.

Markel, John D. and Augustine H. Gray Jr. *Linear Prediction of Speech*. Springer-Verlag, Berlin, Germany, 1976.

McClellan, James H. and Thomas W. Parks. A unified approach to the design of optimum FIR linear-phase digital filters. *IEEE Transactions on Circuit Theory*, 20(6):697–701, November 1973.

Moin, Parviz. *Fundamentals of Engineering Numerical Analysis*. Cambridge University Press, 2010.

Muller, Rémy and Thomas Hélie. Trajectory anti-aliasing on guaranteed-passive simulation of nonlinear physical systems. In *Proceedings of the 20th International Conference on Digital Audio Effects (DAFx-17)*, Edinburgh, UK, 5–9 September 2017.

Nocedal, Jorge and Stephen J. Wright. *Numerical Optimization*. Springer-Verlag, New York, NY, 2nd edition, 2006.

Novak, Antonin, Laurent Simon, Pierrick Lotton, and Frantisek Kadlec. Modeling of nonlinear audio systems using swept-sine signals: Application to audio effects. In *Proceedings of the 12th International Conference on Digital Audio Effects (DAFx-09)*, pages 1–4, Como, Italy, 1–4 September 2009.

Novak, Antonin, Laurent Simon, Frantisek Kadlec, and Pierrick Lotton. Nonlinear system identification using exponential swept-sine signal. *IEEE Transactions on Instrumentation and Measurement*, 59(8):2220–2229, August 2010a.

Novak, Antonin, Laurent Simon, and Pierrick Lotton. Analysis, synthesis, and classification of nonlinear systems using synchronized swept-sine method for audio effects. *EURASIP Journal on Advances in Signal Processing*, 2010(1):793816, July 2010b.

Novak, Antonin, Laurent Simon, Pierrick Lotton, and Joël Gilbert. Chebyshev model and synchronized swept sine method in nonlinear audio effect modeling. In *Proceedings of the 13th International Conference on Digital Audio Effects (DAFx-10)*, Graz, Austria, 6–10 September 2010c.

Olsen, Michael Jørgen, Kurt James Werner, and Julius O. Smith III. Resolving grouped nonlinearities in wave digital filters using iterative techniques. In *Proceedings of the 19th International Conference on Digital Audio Effects (DAFx-16)*, pages 279–286, Brno, Czech Republic, 5–9 September 2016.

Olson, Harry F. *Dynamical Analogies*. D. Van Nostrand Company, New York, NY, 1943.

Olver, Frank W. J., Daniel W. Lozier, Ronald F. Boisvert, and Charles W. Clark, editors. *NIST Handbook of Mathematical Functions*. Cambridge University Press, 2010.

Oppenheim, Alan V. and Ronald W. Schafer. *Discrete-Time Signal Processing*. Prentice Hall, 2009.

Paiva, Rafael C. D. and Vesa Välimäki. The Helmholtz resonator tree. In *Proceedings of the 15th International Conference on Digital Audio Effects (DAFx-12)*, York, UK, 17–21 September 2012.

Paiva, Rafael C. D., Jyri Pakarinen, Vesa Välimäki, and Miikka Tikander. Real-time audio transformer emulation for virtual tube amplifiers. *EURASIP Journal on Advances in Signal Processing*, 2011. Article ID 347645.

Pakarinen, Jyri and Matti Karjalainen. Enhanced wave digital triode model for real-time tube amplifier emulation. *IEEE Transactions on Audio, Speech, and Language Processing*, 18(4):738–746, May 2010.

Pakarinen, Jyri and David Te-Mao Yeh. A review of digital techniques for modeling vacuum-tube guitar amplifiers. *Computer Music Journal*, 33(2):85–100, 2009.

Pakarinen, Jyri, Miikka Tikander, and Matti Karjalainen. Wave digital modeling of the output chain of a vacuum-tube amplifier. In *Proceedings of the 12th International Conference on Digital Audio Effects (DAFx-09)*, pages 55–59, Como, Italy, 1–4 September 2009.

Papamarkos, Nikos and Christodoulos Chamzas. A new approach for the design of digital integrators. *IEEE Transactions on Circuits and Systems—I: Fundamental Theory and Applications*, 43(9): 785–791, September 1996.

Parker, Julian D., Vadim Zavalishin, and Efflam Le Bivic. Reducing the aliasing of nonlinear waveshaping using continuous-time convolution. In *Proceedings of the 19th International Conference on Digital Audio Effects (DAFx-16)*, pages 137–144, Brno, Czech Republic, 5–9 September 2016.

Parks, Thomas W. and C. Sidney Burrus. *Digital filter design*. Wiley-Interscience, New York, 1987.

Parks, Thomas W. and James H. McClellan. Chebyshev approximation for nonrecursive digital filters with linear phase. *IEEE Transactions on Circuit Theory*, 19(2):189–194, March 1972a.

Parks, Thomas W. and James H. McClellan. A program for the design of linear phase finite impulse response digital filters. *IEEE Transactions on Audio and Electroacoustics*, 20(3):195–199, August 1972b.

Paschou, Effrosyni, Fabiàn Esqueda, Vesa Välimäki, and John Mourjopoulos. Modeling and measuring a Moog voltage-controlled filter. In *Proceedings of the 2017 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)*, pages 1641–1647, Kuala Lumpur, 12–15 December 2017.

Pei, Soo-Chang and Hong-Jie Hsu. Fractional bilinear transform for analog-to-digital conversion. *IEEE Transactions on Signal Processing*, 56(5):2122–2127, May 2008.

Pelgrom, Marcel J. M. *Analog-to-digital conversion*. Springer-Verlag, New York, NY, 2nd edition, 2013.

Press, William H., Saul A. Teukolsky, William T. Vetterling, and Brian P. Flannery. *Numerical recipes: The art of scientific computing*. Cambridge University Press, New York, NY, 3rd edition, 2007.

Rabiner, Lawrence R. The design of finite impulse response digital filters using linear programming techniques. *The Bell System Technical Journal*, 51(6):1177–1198, July 1972a.

Rabiner, Lawrence R. Linear program design of finite impulse response (FIR) digital filters. *IEEE Transactions on Audio and Electroacoustics*, 20(4):280–288, October 1972b.

Rabiner, Lawrence R. and Bernard Gold. *Theory and application of digital signal processing*. Prentice-Hall, Englewodd Cliffs, NJ, 1975.

Sarti, Augusto and Giovanni De Sanctis. Systematic methods for the implementation of nonlinear wave-digital structures. *IEEE Transactions on Circuits and Systems—I: Regular Papers*, 56(2): 460–472, February 2009.

Shampine, Lawrence F. Vectorized adaptive quadrature in MATLAB. *Journal of Computational and Applied Mathematics*, 211(2):131–140, 2008.

Shockley, William. The theory of p–n junctions in semiconductors and p–n junction transistors. *The Bell System Technical Journal*, 28(3):435–489, July 1949.

Smith, Steven W. *The scientist and engineer's guide to digital signal processing*. California Technical Publishing, 1997.

Smith III, Julius O. *Mathematics of the Discrete Fourier Transform (DFT)*. W3K Publishing, second edition, 2007a.

Smith III, Julius O. *Introduction to Digital Filters with Audio Applications*. W3K Publishing, 2007b.

Smith III, Julius O. *Physical Audio Signal Processing*. W3K Publishing, 2010. online book.

Smith III, Julius O. *Spectral Audio Signal Processing*. W3K Publishing, 2011.

Smith III, Julius O. and Jonathan S. Abel. Bark and ERB bilinear transforms. *IEEE Transactions on Speech and Audio Processing*, 7(6):697–708, Nov 1999.

Stilson, Tim S. *Efficiently-Variable Non-Oversampled Algorithms in Virtual-Analog Music Synthesis—A Root-Locus Perspective*. Ph.D. Dissertation, Stanford University, Stanford, California, 2006.

Stilson, Tim S. and Julius O. Smith III. Analyzing the Moog VCF with considerations for digital implementation. In *Proceedings of the International Computer Music Conference (ICMC)*, pages 398–401, Hong Kong, 19–24 August 1996.

Strube, Hans Werner. Linear prediction on a warped frequency scale. *The Journal of the Acoustical Society of America (JASA)*, 68(4):1071–1076, 1980.

Thornburg, Harvey. Antialiasing for nonlinearities: Acoustic modeling and synthesis applications. In *Proceedings of the International Computer Music Conference (ICMC)*, Beijing, China, 22–27 October 1999.

Tseng, Chien-Cheng. Digital integrator design using Simpson rule and fractional delay filter. *IEE Proceedings - Vision, Image and Signal Processing*, 153(1):79–86, February 2006.

Urabe, Minoru. Galerkin's procedure for nonlinear periodic systems. *Archive for Rational Mechanics and Analysis*, 20(2):120–152, January 1965.

Välimäki, Vesa. Discrete-time synthesis of the sawtooth waveform with reduced aliasing. *IEEE Signal Processing Letters*, 12(3):214–217, March 2005.

Välimäki, Vesa and Joshua D. Reiss. All about audio equalization: Solutions and frontiers. *Applied Sciences*, 6(5), 2016. Article #129.

Vlach, Jiří. Tableau and modified nodal formulations. In Chen, Wai-Kai, editor, *The Circuits and Filters Handbook*, pages 663–684. CRC Press, Boca Raton, FL, 2nd edition, 2002.

Vlach, Jiří and Kishore Singhal. *Computer methods for circuit analysis and design*. Van Nostrand Reinhold Company, New York, 2nd edition, 1993.

Šekara, Tomislav B. New transformation polynomials for discretization of analogue systems. *Electrical Engineering*, 89(2):137–147, 2006.

Šekara, Tomislav B. and Milić R. Stojić. Application of the α-approximation for discretization of analogue systems. *Facta Universitatis (NIŠ), Ser.: Elec. Energ.*, 18(3):571–586, December 2005.

Waltz, Richard A., José Luis Morales, Jorge Nocedal, and Dominique Orban. An interior algorithm for nonlinear optimization that combines line search and trust region steps. *Mathematical Programming*, 107(3):391–408, 2006.

Wang, Weijun, Rongming Lin, Quanbo Zou, and Xinxin Li. Modeling and characterization of a silicon condenser microphone. *Journal of Micromechanics and Microengineering*, 14:403–409, March 2004.

Werner, Kurt James. *Virtual Analog Modeling of Audio Circuitry Using Wave Digital Filters*. Ph.D. Dissertation, Stanford University, 2016.

Werner, Kurt James and Julius O. Smith III. An energetic interpretation of nonlinear wave digital filter lookup table error. In *Proccedings of the 2015 IEEE International Symposium on Signals, Circuits and Systems (ISSCS)*, Iași, Romania, 9–10 July 2015.

Werner, Kurt James, Jonathan S. Abel, and Julius O. Smith III. A physically-informed, circuit-bendable, digital model of the Roland TR-808 bass drum circuit. In *Proceedings of the 17th International Conference on Digital Audio Effects (DAFx-14)*, pages 159–166, Erlangen, Germany, 1–5 September 2014a.

Werner, Kurt James, Jonathan S. Abel, and Julius O. Smith III. Three models of circuit-bent TR-808 voices: the bass drum, cymbal, and cowbell. In *Triple CCRMAlite Science Showcase*, Stanford, CA, 26–27 October 2014b.

Werner, Kurt James, Jonathan S. Abel, and Julius O. Smith III. More cowbell: a physically-informed, circuit-bendable, digital model of the TR-808 cowbell. In *Proceedings of the 137th Convention of the Audio Engineering Society (AES)*, volume 137, Los Angeles, CA, 9–12 October 2014c. convention paper #9207.

Werner, Kurt James, Jonathan S. Abel, and Julius O. Smith III. The TR-808 cymbal: a physically-informed, circuit-bendable, digital model. In *Proceedings of the Joint Session of the International Computer Music Conference (ICMC) and the Sound and Music Computing Conference (SMC)*, volume 2014, pages 1453–1460, Athens, Greece, 14–20 September 2014d.

Werner, Kurt James, Vaibhav Nangia, Alberto Bernardini, Julius O. Smith III, and Augusto Sarti. An improved and generalized diode clipper model for wave digital filters. In *Proceedings of the 139th Convention of the Audio Engineering Society (AES)*, New York, NY, October 2015a. convention paper #9360.

Werner, Kurt James, Vaibhav Nangia, Julius O. Smith III, and Jonathan S. Abel. Resolving wave digital filters with multiple/multiport nonlinearities. In *Proceedings of the 18th International Conference on Digital Audio Effects (DAFx-15)*, pages 387–394, Trondheim, Norway, 30 November–3 December 2015b.

Werner, Kurt James, Vaibhav Nangia, Julius O. Smith III, and Jonathan S. Abel. A general and explicit formulation for wave digital filters with multiple/multiport nonlinearities and complicated topologies. In *Proceedings of the 2015 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*, New Paltz, NY, 18–21 Octobter 2015c.

Werner, Kurt James, Julius O. Smith III, and Jonathan S. Abel. Wave digital filter adaptors for arbitrary topologies and multiport linear elements. In *Proceedings of the 18th International Conference on Digital Audio Effects (DAFx-15)*, pages 379–386, Trondheim, Norway, 30 November–3 December 2015d.

Werner, Kurt James, W. Ross Dunkel, and François G. Germain. A computational model of the Hammond organ vibrato/chorus using wave digital filters. In *Proceedings of the 19th International Conference on Digital Audio Effects (DAFx-16)*, pages 271–278, Brno, Czech Republic, 5–9 September 2016a.

Werner, Kurt James, W. Ross Dunkel, Maximilian Rest, Michael Jørgen Olsen, and Julius O. Smith III. Wave digital filter modeling of circuits with operational amplifiers. In *Proceedings of the 24th European Signal Processing Conference (EUSIPCO)*, pages 1033–1037, Budapest, Hungary, 28 August–2 September 2016b.

Werner, Kurt James, Alberto Bernardini, Julius O. Smith III, and Augusto Sarti. Modeling circuits with arbitrary topologies and active linear multiports using wave digital filters. *IEEE Transactions on Circuits and Systems—I: Regular Papers*, 65(12):4233–4246, December 2018.

Yeh, David T. *Digital Implementation of Musical Distortion Circuits by Analysis and Simulation*. Ph.D. Dissertation, Stanford University, Stanford, CA, 2009.

Yeh, David T. Automated physical modeling of nonlinear audio circuits for real-time audio effects—part II: BJT and vacuum tube examples. *IEEE Transactions on Audio, Speech, and Language Processing*, 20(4):1207–1216, 2012.

Yeh, David T. and Julius O. Smith III. Simulating guitar distortion circuits using wave digital and nonlinear state-space formulations. In *Proceedings of the 11th International Conference on Digital Audio Effects (DAFx-08)*, pages 19–26, Espoo, Finland, 1–4 September 2008.

Yeh, David T., Jonathan S. Abel, and Julius O. Smith III. Simplified, physically-informed models of distortion and overdrive guitar effects pedals. In *Proceedings of the 10th International Conference on Digital Audio Effects (DAFx-07)*, pages 189–196, Bordeaux, France, 10–15 September 2007a.

Yeh, David T., Jonathan S. Abel, and Julius O. Smith III. Simulation of the diode limiter in guitar distortion circuits by numerical solution of'ordinary differential equations. In *Proceedings of the 10th International Conference on Digital Audio Effects (DAFx-07)*, pages 197–204, Bordeaux, France, 10–15 September 2007b.

Yeh, David T., Jonathan S. Abel, Andrei Vladimirescu, and Julius O. Smith III. Numerical methods for simulation of guitar distortion circuits. *Computer Music Journal*, 32(2):23–42, 2008.

Yeh, David T., Jonathan S. Abel, and Julius O. Smith III. Automated physical modeling of nonlinear audio circuits for real-time audio effects—part i: Theoretical development. *IEEE Transactions on Audio, Speech, and Language Processing*, 18(4):728–737, May 2010.

Zavalishin, Vadim. The art of VA filter design. Online, 28 October 2018. Revision 2.1.0.

Zhang, Fuzhen, editor. *The Schur complement and its applications*. Springer Springer Science+Business Media, 2005.