

# **Measurement and Design of a Digital Waveguide Slide Guitar Synthesizer**

*David G. Smith*



Music Technology Area, Department of Music Research  
Schulich School of Music  
McGill University  
Montréal, Canada

July 2023

---

A thesis submitted to McGill University in partial fulfillment of the requirements for the degree  
of Master of Arts.

© 2023 David G. Smith

---

# Abstract / Résumé

## English

This thesis introduces a digital waveguide model for simulating the sound of a slide guitar. This model uses a single digital waveguide model for the transverse vibrations of the string as well as an additional module which models the contact sound between the slide and string surfaces. After introducing the background and previous inspiration model, a more well-developed architecture is defined and introduced. Included in this is an examination of its limitations. This model was implemented in MATLAB and facilitates sound synthesis on a sample-by-sample computational basis. The verification techniques used in the developed model are described. As this is a physical model, various measurements were made on an actual acoustic guitar in an attempt to verify the model's accuracy as well as refine aspects missing from previous attempts. After these measurements are made, the strategies used to parameterize the model for sound design and musicality are discussed.

## Français

Ce mémoire présente un modèle de guide d'ondes numérique pour simuler le son d'une guitare slide. Le modèle utilise un seul guide d'ondes numérique pour les vibrations transversales de la corde avec un module additionnel qui modèle le son du contact entre la slide et la surface des cordes. Après une introduction qui présente le contexte et le modèle ayant servi d'inspiration, une architecture plus avancée est présentée. Il s'agit notamment d'examiner ses limites. Le modèle a été mis en œuvre sur MATLAB et facilite la synthèse sonore sur la base de calculs échantillon par échantillon. Les techniques de vérification appliquées au modèle développé sont décrites. Comme il s'agit d'un modèle physique, des mesures ont été effectuées sur une guitare acoustique pour vérifier la précision du modèle et affiner les aspects que les tentatives précédentes auraient négligés. Après avoir traité de ces mesures, le mémoire présente les stratégies de paramétrage du modèle pour la conception sonore et la musicalité.

# Acknowledgements

I would like to thank Prof. Gary Scavone for serving as my thesis advisor. His knowledge of physical modeling, acoustics and acoustic measurement techniques served as a invaluable sounding board during the development process. Many thanks to Prof. Ichiro Fuginaga for agreeing to be my thesis examiner. Laurent Olivier-Lord, Andrew David and John Gilbert deserve credit in helping me edit my French abstract. Dr. Andre Van Schyndel provided valuable feedback to help fine tune the thesis.

Many thanks to both Aybar Adin and Vlad Baran of McGill's Sound Recording Program as well as Yves Méthot of CIRMMT. All of these people were extremely helpful in determining which microphones and other equipment to use for the physical measurements. Song Wang was also extremely helpful in brainstorming ideas when attempting to devise a measurement system to capture the decay parameters associated with longitudinal string motion.

On the DSP side of things, I would like to thank Max Henry as well as inhabitants of the *#dsp* channel on The Audio Programmer Discord Server. Their help in exploring DSP concepts, listening to my results, as well as debugging the nitty-gritty implementation details related to things like time-varying Lagrange interpolation benefited me greatly.

Finally, I would like to thank Nadia Stevens for all her help, support and love throughout this process. This includes picking up the slack around the apartment when I had deadlines to meet as well as when I was operating at a reduced capacity due to a hard to diagnose medical ailment which decided to appear when a healthcare crisis was happening across the country.

---

# Contents

<b>1</b>	<b>Introduction</b>	<b>2</b>
1.1	Motivation . . . . .	2
1.2	Thesis Organization . . . . .	4
<b>2</b>	<b>Background</b>	<b>6</b>
2.1	Physical Modeling and Digital Waveguides . . . . .	6
2.1.1	Fundamental Components . . . . .	6
2.1.2	Applied to String Modeling . . . . .	9
2.2	Development of Slide Guitar Model . . . . .	13
2.2.1	Energy Compensation . . . . .	14
2.2.2	Loop Filter . . . . .	14
2.3	CSG Development . . . . .	16
2.3.1	Contact Sound Analysis . . . . .	16
2.3.2	Adapting From Finger to Slide . . . . .	18
2.3.3	Model Description . . . . .	19
2.4	Recent Developments and Approaches . . . . .	21
2.4.1	Evangelista's DWG Approach . . . . .	21
2.4.2	Belfast Finite-Difference Technique . . . . .	22
<b>3</b>	<b>Description of Slide Guitar Synthesis Model</b>	<b>24</b>
3.1	Introduction of Architecture . . . . .	24
3.1.1	Diagram Conventions . . . . .	24
3.1.2	Single String Slide Synthesizer . . . . .	25
3.1.3	Control Signal Processor . . . . .	25
3.1.4	String DWG . . . . .	26
3.1.5	Contact Sound Generator . . . . .	27
3.2	Limitations of Model . . . . .	32
3.2.1	Minimum Relative String Length . . . . .	32
3.2.2	Non-constant Phase Delay of Filters . . . . .	34
3.2.3	Impulse Train Quantization . . . . .	35
3.2.4	Nyquist Rate Implications . . . . .	37

<b>4 Verification Of Slide Model and Constituent Components</b>	<b>38</b>
4.1 Filters . . . . .	38
4.1.1 Resonator . . . . .	38
4.1.2 DC Blocker . . . . .	39
4.1.3 Smoothing Filter . . . . .	39
4.1.4 Longitudinal Modes . . . . .	41
4.2 Control Signal Processor . . . . .	41
4.2.1 Components of Control Signal Processor . . . . .	42
4.2.2 Control Signal Processor as a Whole . . . . .	43
4.3 Noise Generation Objects . . . . .	44
4.3.1 Impulse Train . . . . .	44
4.3.2 Exponential Decay . . . . .	45
4.3.3 Noise Pulse Train . . . . .	47
4.3.4 Noise Burst Generator . . . . .	49
4.4 Harmonic Accentuators . . . . .	51
4.4.1 Harmonic Resonator Bank . . . . .	51
4.4.2 Reso + Tanh . . . . .	51
4.5 Contact Sound Generator (CSG) . . . . .	52
4.5.1 Wound Variant . . . . .	52
4.5.2 Unwound Variant . . . . .	57
4.6 String DWG and its Components . . . . .	57
4.6.1 Interpolated Delay Line . . . . .	57
4.6.2 Energy Scaler . . . . .	59
4.6.3 Loop Filter . . . . .	63
4.6.4 String Digital Waveguide . . . . .	65
4.7 Slide Synthesizer . . . . .	65
4.8 Real-Time Feasibility . . . . .	68
<b>5 Physical Measurements</b>	<b>70</b>
5.1 Experimental Setup . . . . .	70
5.2 String Winding Density . . . . .	71
5.3 Longitudinal to Transverse Coupling . . . . .	72
5.3.1 Setup and Method . . . . .	72
5.3.2 Results . . . . .	73
5.4 Contact Noise Spectrum . . . . .	74
5.4.1 Method . . . . .	74
5.4.2 Results . . . . .	75
5.4.3 Wound Strings . . . . .	75
5.4.4 Unwound Strings . . . . .	75
5.5 Decay Rate of Single Winding Impact . . . . .	75
5.5.1 Methods . . . . .	76
5.5.2 Results . . . . .	77
5.6 Playing Examples . . . . .	79

<b>6 Sound Design and Control Signals</b>	<b>80</b>
6.1 Wound Contact Sound Generator . . . . .	80
6.1.1 Decay Rate . . . . .	80
6.1.2 Noise Sources and Harmonic Accentuation Combinations . . . . .	81
6.1.3 Matching the Recordings . . . . .	86
6.2 Waveform Initialization . . . . .	88
6.2.1 White vs. Pink Noise . . . . .	89
6.2.2 Removing DC Component . . . . .	89
6.3 Control Signal Parametrization . . . . .	91
6.3.1 $L[n]$ Generation . . . . .	91
6.3.2 Comparison to Observed $L[n]$ . . . . .	92
6.4 Comparing Synthesized Results vs. Recordings . . . . .	93
6.5 Purposeful Instability . . . . .	96
<b>7 Conclusion and Future Work</b>	<b>98</b>
7.1 Conclusion . . . . .	98
7.2 Future Work . . . . .	99
<b>Bibliography</b>	<b>101</b>
<b>A String Winding Measurement Photos</b>	<b>103</b>
<b>B Code and Sound Examples</b>	<b>104</b>

# Chapter 1

## Introduction

### 1.1 Motivation

One of the more unique methods of playing guitar is an approach referred to as “slide guitar.” This consists of using a smooth rigid tube (the slide) to control the length of the string, instead of the frets and fingers. The slide rests on top of the string and does not touch the fingerboard or frets. In this way, it acts as a movable string termination and influences the vibration of the string by creating a new load termination in between the nut and the bridge [Evangelista 2012]. This allows unique articulations and pitch inflections to be generated as the player is no longer constrained to the pitches provided by the fret locations. Vibrato is one of the characteristic articulations of slide playing. Figure 1.1 shows a player using a chrome slide on an acoustic guitar.



**Fig. 1.1** An acoustic guitar played with a chrome slide.

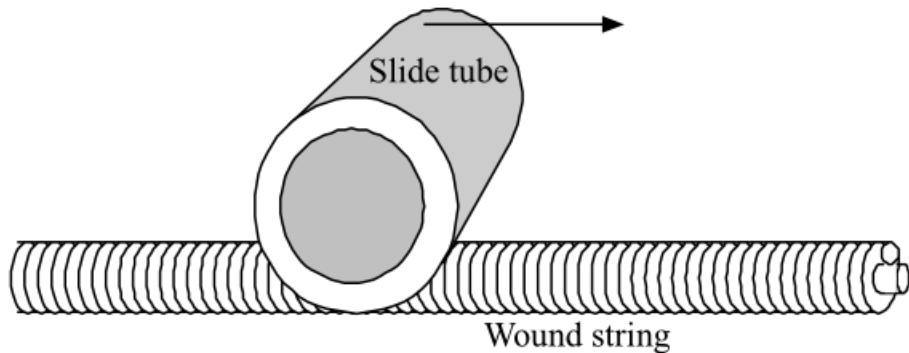
Traditionally, slides are made from ceramic or metal [Bhanuprakash, Stapleton, and Walstijn 2020]. This results in a smooth/polished surface for the slide. The density of the material is

important as this determines the slide's mass and correspondingly the force it applies to the string. Figure 1.2 illustrates different slides made out of a variety of materials. The slide's surface texture and mass are important as they influence how the slide interacts with the surface of the string. This interaction creates a new timbral component which is also influenced by the slide's velocity [Pakarinen, Puputti, and Välimäki 2008].



**Fig. 1.2** A selection of slides made from different materials, all of which are comparatively smooth.

In the case of wound strings, two new sounds are created. The first is a time-varying harmonic component due to the interaction of the slide with the spatially periodic pattern of windings on the string's surface (inherent in a wound string's construction) [Pakarinen, Penttinien, and Bank 2007]. The second component is due to the stimulation of the string's longitudinal modes as the slide introduces disturbances in this direction when it impacts the ridges of the windings. As the slide does not provide sufficient force to change the longitudinal length, the longitudinal mode frequencies are static, regardless of the motion of the slide [Pakarinen, Penttinien, and Bank 2007]. Figure 1.3 shows a close-up illustration of a slide interacting with a wound string.



**Fig. 1.3** A close-up illustration of a slide on wound string [Pakarinen, Puputti, and Välimäki 2008].

Unwound strings have a uniform surface, lacking the ridges created by windings which a slide impacts while traveling the length of the string. Correspondingly, the coefficient of friction

between the unwound string and slide is comparatively much lower than in the wound-string case. This drastically reduces the coupling between the slide and the unwound string from a longitudinal standpoint, with the result that the longitudinal modes are not audible. As a result, the contact sound generated for unwound strings is more akin to white-noise whose amplitude is scaled by the slide velocity and lacks a harmonic component [Pakarinen, Puputti, and Välimäki 2008].

The original sound synthesis model which serves as the impetus for this thesis comes from [Pakarinen, Puputti, and Välimäki 2008]. The goal of this project is to explore this model more fully from a digital signal processing standpoint as compared to the description in the aforementioned paper. This includes aspects related to its implementation, verification and inherent limitations. Possible refinements to the physical modeling will be investigated as well as aspects related to parametrizing the control signals to generate usable and interesting musical signals. The implementation is done in MATLAB to facilitate easier analysis and verification of the model.

This model was mainly chosen due to a match between its complexity and the author's skills at the outset of the thesis. The paper is the only one which emphasized real-time implementations and that was a potential outcome when the thesis commenced. The initial descriptions also contain physical measurements as part of the model's development, which the author had an interest in recreating. Additionally, the details of the implementation are readily available as PD patches in Appendix B of [Puputti 2010].

## **1.2 Thesis Organization**

The thesis is organized into seven different chapters as described below.

### **Chapter 1 - Introduction**

This chapter provides a brief overview of the slide guitar, what the thesis hopes to achieve as well as how the thesis is organized.

### **Chapter 2 - Background**

Chapter 2 will elaborate upon the necessary theoretical knowledge to understand the rest of the thesis. It is not meant to be exhaustive as both digital signal processing and physical modeling are large and broad fields themselves. The aim is to introduce as much theory as is necessary as well as provide references if an interested reader wishes to find more details. Other approaches and techniques will be examined as well.

### **Chapter 3 - Description of Slide Guitar Synthesis Model**

After the necessary theory has been introduced, the architecture of the slide guitar synthesizer will be described as well as its constituent components. Some aspects will have already been introduced in the background, however more details will be provided to fully elucidate the mechanics of the model.

**Chapter 4 - Verification Of Slide Model and Constituent Components**

The techniques used to verify the correctness of the model will be discussed as well as any limitations inherent to it. Audio examples will be provided to help develop an aural intuition for the sounds. However, the synthesis parameters will often not be realistic and physically informed as their goal is to illustrate algorithmic correctness of the model as opposed to usable sonic potential (which will be covered in Chapter 6).

**Chapter 5 - Physical Measurements of Slide Sounds**

In this section, the physical experiments which were performed during the development of this model will be described. As will be shown in both Chapters 2 & 3, there is a strong physical basis for the synthesizer model. Many of the model's parameters have a physical correlate, hence why this section comes after the model's description. Some of the experiments will be recreating work from the original paper, while others will attempt to refine the model to make it more physically informed.

**Chapter 6 - Sound Design and Control Signals**

After the model has been described and verified (both algorithmically and physically), the next step is to tune the parameters and control signals. This chapter aims to explore how the parameters were tuned as well as the different architectural and component decisions which were explored in an attempt to create the “best” sounding synthesizer. It will also examine strategies related to generating control signals to achieve different sounds.

**Chapter 7 - Conclusion and Future Work**

The last section provides a brief summary of what has been explored in the thesis as well as expounds upon opportunities for improvement and future research.

# Chapter 2

# Background

In this chapter an introduction will be given to the theoretical framework of digital waveguides, which form the basis for the synthesis model developed in this thesis. After these concepts have been covered, the slide guitar model which inspired this thesis will be outlined. A basic overview will be given with emphasis on the theoretical derivations as a more detailed analysis of the implementation will be covered in subsequent chapters. The chapter will conclude with an overview and comparison of the more recent approaches to slide guitar synthesis.

## 2.1 Physical Modeling and Digital Waveguides

Physical modeling is a discipline which attempts to recreate physical phenomena using computational algorithms [Smith 2023a]. There are many different approaches to this, however one of the most popular and well developed is the technique of digital waveguides (DWG). As the name would imply, this approach uses algorithms and data structures to mimic the method by which waves propagate throughout a medium. In certain cases it can be extremely computationally efficient technique and this had led to its incorporation into many different commercial synthesizers.

### 2.1.1 Fundamental Components

Two of the fundamental components of digital waveguide models are digital filters and digital delay lines [Smith 2023a]. The second component can be broken down into integer and fractional length delay lines. A variety of wave propagation phenomena can be modeled by interconnections of these different components. Through the incorporation of different noise sources and initialization waveforms, a large number of different sound synthesis algorithms can be created.

#### Digital Filters

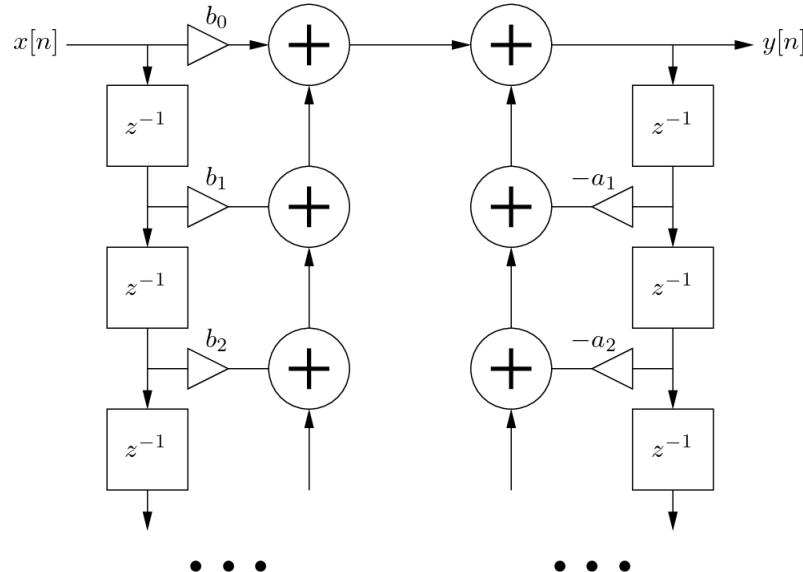
Conceptually digital filters are extremely simple computational structures. They merely add scaled time-delayed versions of their inputs and outputs. Through various combinations of delay and scaling, a range of different frequency domain effects can be achieved. Filters are linear

systems which means that the tools developed in the theory of linear time-invariant (LTI) systems is available to the algorithm designer.

The behaviour of digital filters can be analyzed in the frequency domain in terms of their frequency response. This can further be divided into two components: the magnitude response and the phase response. The magnitude response describes the change in amplitude each frequency component of a signal experiences when the filter is applied to it. The phase response describes the amount of delay experienced by a signal's frequency components, given in radians. The phase delay is a function of frequency indicating the time-delay, in samples, each frequency component experiences. The phase delay can be derived from the phase response.

Digital filters come in two varieties: finite impulse response (FIR) and infinite impulse response (IIR). FIR filters consist of only delayed and scaled copies of the input signal while IIR filters incorporate the output of the filter via feedback connections. This is what gives rise to the “infinite” aspect of their name. Theoretically the feedback causes the filter’s output in response to an impulse signal to never completely decay (ignoring the finite precision effects of digital arithmetic).

All filters exhibit a transient response [Oppenheim, Willsky, and Nawab 1997]. These occur whenever there is a change in the coefficients associated with a filter’s structure as well as when there is a change in the frequency content or phase of the input signal. FIR filters have a transient whose length corresponds to the order of the filter. IIR filters pose more of a problem with regards to transients as their feedback cause the transients to propagate for a much longer time (depending on the attenuation provided by the filter’s coefficients). Figure 2.1 illustrates a generic IIR filter structure.



**Fig. 2.1** A generic digital filter structure [Scavone 2023c]. The  $b_n$  coefficients refer to the feedforward part while the  $a_n$  coefficients refer to the feedback part. An FIR filter would consist of only the feedforward components on the left.

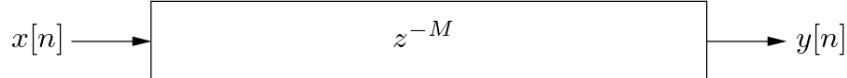
### Digital Delay Lines

The digital delay line is one of the fundamental components of DWG based modeling [Scavone 2023b]. Its main purpose is to provide a computational model of the time delay experienced by waves propagating a particular medium. The values a delay line holds represent physical quantities as they evolve over time at spatially sampled locations of the physical medium. The physical distance between each spatial sample corresponds to the distance a wave travels during one sampling period. Mathematically, this can be expressed in the following equation:

$$X_s = T_s \times c \quad (2.1)$$

where  $T_s$  is the temporal sampling period and  $c$  is the wave propagation speed in the particular media being modeled [Scavone 2023d]. Figure 2.2 illustrates a block diagram representation of a digital delay line (of  $M$  samples) frequently used in signal processing diagrams.

Unfortunately, there is an inherent limitation to digital delay lines. They are limited to lengths of purely integer sample delays. This means that signals can only be delayed/modeled using integer numbers of samples with this structure. In many physical modeling applications, this is problematic due to the fact that the physical world and its associated laws are not spatially quantized. It is often required to know a physical quantity which is at a point that lies between two spatial (or time-domain) samples.



**Fig. 2.2** A block diagram representation of a digital delay component which is  $M$  samples in length [Scavone 2023a].

### One-Pole Impulse Response

Beyond being used to modify the frequency content of signals, filters can also be used as extremely efficient structures to generate signals. This is achieved by tuning their impulse responses to the desired signal characteristics. For example, a one pole-filter<sup>1</sup> can be designed to have the following impulse response:

$$h[n] = \alpha^n \quad (2.2)$$

where  $\alpha$  is the filter's pole location. This means that a one-pole filter can be used to generate an exponential sequence in response to an impulse signal. The shape of the sequence can depends on the value of  $\alpha$ . It is possible to calculate  $\alpha$  to generate a decreasing exponential sequence which decays to a certain value after a particular amount of time. The computational implementation is quite simple as a single-pole filter only requires a single sample delay, single coefficient and single addition.

---

<sup>1</sup>See [https://ccrma.stanford.edu/~jos/filters/Pole\\_Zero\\_Analysis\\_I.html](https://ccrma.stanford.edu/~jos/filters/Pole_Zero_Analysis_I.html) for information on filter poles.

### Delay Line Interpolation

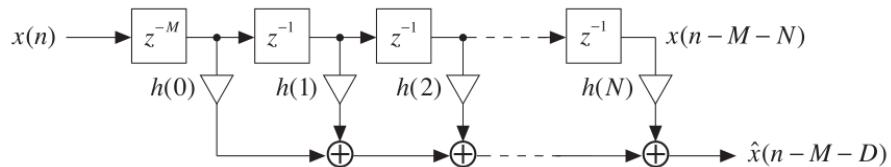
Various approaches have been developed in an attempt to approximate the signal values in between samples of a digital delay line [Laakso et al. 1996]. As a value is being determined which lies in between two known values, the task is fundamentally interpolation. The different interpolation approaches are implemented using filters whose coefficients are calculated in a particular way to achieve the desired degree of fractional/sub-sample approximation. When combined in series at the output of an integer delay line, the overall structure is referred to as a fractional delay line. Given the techniques are realized using filters, the different approaches have various frequency/time domain implications and transients can become an issue (in the case of a time-varying delay line length).

One popular approach to fractional delay line implementations is Lagrange interpolation. In this technique an FIR filter is used where the filter coefficients implement Lagrange interpolation to allow for sub-sample accuracy [Välimäki 1995]. Figure 2.3 illustrates the FIR filter in series with an  $M$  sample delay line. The equation used to generate the coefficients is:

$$h[n] = \prod_{k=0}^N \frac{D - k}{n - k}, \text{ for } n = 0, 1, \dots, N \text{ and } k \neq n \quad (2.3)$$

where  $D$  is the fractional delay,  $N$  is order of the filter and  $k$  is an indexing variable used in the product sequence.

The order of the filter determines the order of the polynomials involved following the theory of Lagrange interpolation. With an order of  $N = 1$ , linear interpolation is achieved. Adjusting the order of the filter allows you to have more control over its frequency response and phase delay. The Lagrange approach can generate a filter structure which has a constant phase delay under certain conditions. Figure 2.4 illustrates how the resulting filter has low-pass magnitude characteristics and can exhibit a constant phase delay under certain conditions.

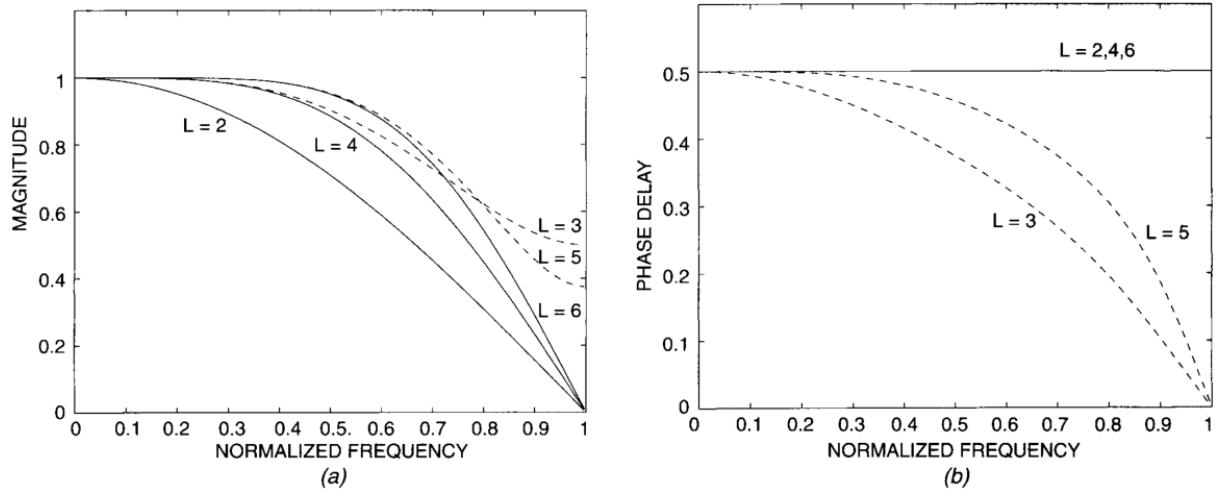


**Fig. 2.3** An FIR structure implementing Lagrange interpolation in series with an  $M$  sample delay line [Välimäki 1995].  $D$  represents the fractional delay and  $N$  represents the filter's order.

#### 2.1.2 Applied to String Modeling

The wave equation can be used to derive the following general solution to wave motion along a one-dimensional string:

$$y(x, t) = Ae^{j(\omega t \pm kx)} \quad (2.4)$$



**Fig. 2.4** Magnitude responses (a) and phase delays (b) for Lagrange interpolating filters of length  $L = 2, 3, 4, 5$ , and  $6$  with  $d = 0.5$  [Laakso et al. 1996].

where  $y(x, t)$  represents the transverse displacement,  $A$  is the complex amplitude,  $\omega$  represents the radian frequency,  $t$  represents time,  $x$  represents spatial location along the string, and  $k$  is the wave number. d'Alembert reformulated this into another expression which can be interpreted as the sum of two identical waves traveling in opposite directions [Smith 2023a]. It is:

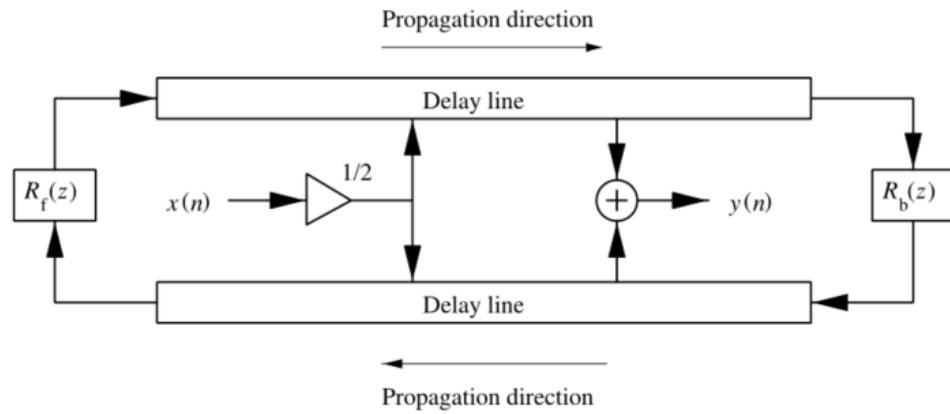
$$y(x, t) = y^+(ct - x) + y^-(ct + x) \quad (2.5)$$

where  $c$  is the wave speed and  $y^+$  and  $y^-$  represent waves traveling in the positive and negative  $x$  directions respectively, with the same speed. In the physical world, strings need to have a finite length and often have a termination at each end which is referred to as a load impedance. These impedances have their own effects and can be considered as filters. When a traveling wave encounters one of the termination impedances, it is reflected back along the string to travel in the opposite direction with the effects of the corresponding load impedance applied to it.

If this was to be implemented via a DWG model, an initial approach would consist of a digital delay line representing each direction of propagation as well as a filter at each end of the string's terminations. Figure 2.5 illustrates a signal-flow diagram for this bi-directional computational model. The traveling wave variables could be any physical quantity related to the string movement (i.e., displacement, acceleration, velocity, etc.) depending on what is desired for the synthesis model, however transverse displacement is commonly used for string models. This is because transverse vibrations are primarily how string instruments generate their characteristic sounds as they are played. In combination with the laws of physics, it is easy to calculate the other physical quantities which might be needed using the transverse displacement. In the case of a guitar the terminations are the nut and the bridge.

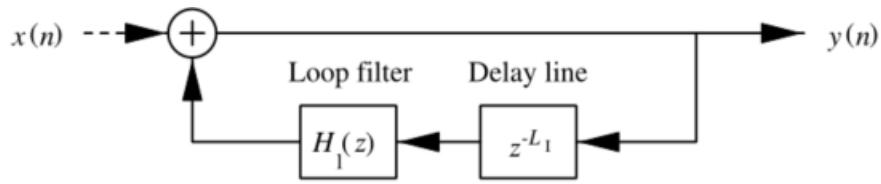
Plucking a string corresponds to imparting an initial displacement along its length [Smith 2023a]. For a DWG model, this corresponds to the waveform which is used to initialize the values of the delay lines when modeling the transverse displacement. The output,  $y[n]$  can be

determined at any spatial location by summing the appropriate spatial samples from each of the different traveling wave simulations (applying interpolation where necessary). Determining where the output will be taken is part of the synthesis model. For the most physically accurate model, this corresponds to where the sound is probed in the system. The bridge is commonly used as the output location for string models as this is where the vibrations of the string are transferred to the body of the instrument. The body's effects often create the distinguishing characteristics of a string instrument.



**Fig. 2.5** A bi-directional DWG structure for a string terminated at both ends.  $R_f(z)$  and  $R_b(z)$  are filters which represent the load impedances at each end of the strings [Karjalainen, Välimäki, and Tolonen 1998].

By exploiting the linearity of the systems involved, as well as the fact that it is generally only necessary to know the output at one specific spatial location, it is possible to commute the different components of the bi-directional system into a substantially more efficient model. Figure 2.6 illustrates this new computational structure, which is referred to as the single-delay loop (SDL) structure [Karjalainen, Välimäki, and Tolonen 1998]. The various traveling losses and other effects from phenomena like reflection and dispersion have been combined into the loop filter  $H_l(z)$ . Both traveling waves are still simulated, however they are not implemented separately. A single delay line is used to model their propagation.



**Fig. 2.6** A single delay line DWG structure for a string [Karjalainen, Välimäki, and Tolonen 1998].

The waveform used to initialize the DWG can have a physical correlate to the variable being modeled (i.e., representing the displacement associated with the plucking position of the string)

[Välimäki and Tolonen 1998]. Noise has also been historically used [Karplus and Strong 1983]. The initialization waveform has a drastic impact on the timbre of the sound synthesized (as will be shown in Sec. 6.2).

### Controlling Pitch

Suppose you have a string of a given length which is vibrating. It will produce a pitch with a corresponding fundamental frequency  $F_0$ . This is referred to as the open string fundamental. If the length of the string is shortened, then the fundamental frequency will be scaled inversely proportional to the relative string length [Välimäki and Tolonen 1998]. If  $L$  represents the relative string length, which exists on the interval  $(0, 1]$ , the fundamental at a specified relative string length can be expressed as:

$$F_L = \frac{F_0}{L} \quad (2.6)$$

This relationship is fundamental to the nature of how the guitar is played, with the frets traditionally determining the different values  $L$  can take.

In terms of the DWG structure, controlling the physical length of the string equates to controlling the number of samples in the delay line. Supposing that  $F_L$  and the sampling rate are known, the DWG length (in samples) can be calculated using the expression:

$$\text{DWG Length} = \frac{F_s}{F_L} \quad (2.7)$$

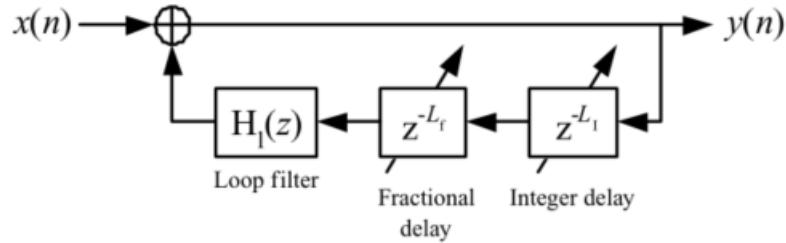
From this we can clearly see why it is necessary to use an interpolation method as described before. If the length of the digital wave guide is limited to purely integer values, then the fundamentals of the synthesized tones will be quantized to integer divisions of the sampling frequency. In order to have the synthesis model be more flexible, expressive and usable, it is necessary to allow the digital waveguide length to take on non-integer values. This issue is well-known in the literature as shown in [Jaffe and Smith 1983].

### Determining DWG Length

In order to express the length of the SDL structure, it is necessary to break down the delay line into its constituent components as shown in Fig. 2.7. The total delay which a waveform propagating through the system will encounter, can be expressed as the sum of the phase delays of the components in the loop:

$$\text{DWG Length} = \tau_{H_L} + L_f + L_I \quad (2.8)$$

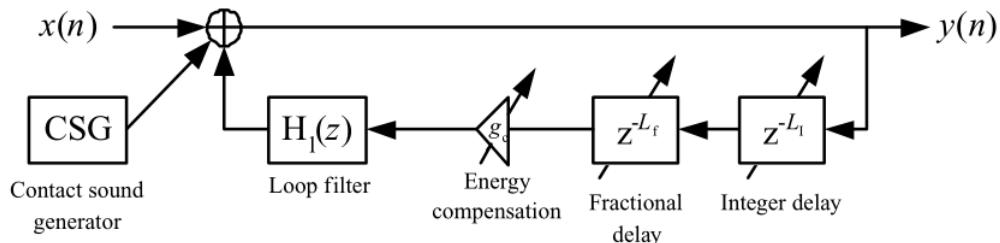
where  $\tau_{H_L}$  represents the phase delay of the loop filter,  $L_f$  represents the phase delay of the filter implementing the fractional interpolation and  $L_I$  represents the integer delay line length. These quantities can be either static or time-varying depending on the type of articulation and sound being synthesized [Karjalainen, Välimäki, and Tolonen 1998]. This also highlights the importance of the loop filter and fractional approximation method used in the model.



**Fig. 2.7** A SDL expanded into its constituent parts [Karjalainen, Välimäki, and Tolonen 1998].

## 2.2 Development of Slide Guitar Model

The slide guitar model which serves as the basis for this thesis was first reported by Pakarinen et al. (2008). The basis for this model is the SDL structure with an additional component to model the slide/string surface interactions and another to scale energy quantities as a function of changes in the string length. The model is shown in Fig. 2.8. The model's loop filter serves the purpose of approximating the losses a vibrating string experiences over time. This model's loop filter was designed by analyzing recordings of a professional guitar player playing notes at the different fret locations along the neck to approximate the losses associated with different string lengths. These measurements at discrete fret-locations were approximated by continuous polynomials which which are a function of relative string length. These polynomials are used to generate the loop filter coefficients so allow for the synthesis of notes in between the frets.



**Fig. 2.8** The slide guitar DWG model from [Pakarinen, Välimäki, and Puputti 2008].

The fundamental control signal for the system is the time-varying relative length of the string. Although not depicted in Fig. 2.8, this is denoted as  $L[n]$ . As shown before, this is what controls the pitch of the synthesized tone. It also controls the sounds generated by the slide/string surface interactions (as will be elaborated upon in Sec. 2.3) as the placement of the slide is ultimately what determines the value  $L[n]$  takes.

### 2.2.1 Energy Compensation

Synthesis of slide guitar sounds differs from that of traditional plucked models in that the pitch is a time-varying quantity that can occur over the course of a single *pluck*. In terms of the DWG model, this translates into a time-varying number of samples for the total DWG length. In order to not have perceptually unnatural shifts in the perceived loudness of the synthesized tone as well as help mitigate aliasing, it is necessary to compensate the energy of the signal based on how the string length changes [Pakarinen, Karjalainen, et al. 2005]. For instance, if you suddenly move the slide so the relative string length is half of its current value, then half of the samples of the DWG will be removed and an unnatural drop in the volume of the sound occurs.

The energy compensation block compensates for the string length changes. It is governed by the following equation:

$$p_c[n] = \sqrt{1 - \Delta x} p[n] = g_c p[n] \quad (2.9)$$

where  $\Delta x$  is the delay-line variation in samples per one time step,  $p[n]$  is the output of the time-varying delay-line,  $g_c$  is the scaling coefficient and  $p_c[n]$  is the energy compensated signal. This is referred to as zero-order energy-preserving interpolation as is explained in [Pakarinen, Karjalainen, et al. 2005]. As the true energetic behavior of a string with a sliding termination is subject to debate, this is a digital approximation which also helps synthesize a more perceptually accurate sound [Pakarinen, Puputti, and Välimäki 2008].

### 2.2.2 Loop Filter

The loop filter is implemented via a single-pole design with the following transfer-function:

$$H_l(z) = g \frac{1 + a}{1 + az^{-1}} \quad (2.10)$$

where  $a$  controls the cut-off frequency and  $g$  controls the gain.  $a$  and  $g$  determine the characteristics of the tone decay and are meant to model the physical losses associated with string vibration. The parameters were derived from recordings of a professional guitar player playing several notes of every fret for each string in an anechoic chamber. [Välimäki and Tolonen 1998] describes this in more detail.

The  $a$  and  $g$  parameters are governed by the following first-order polynomials:

$$g = g_{pol}(0) + g_{pol}(1)m_{fret} \quad (2.11)$$

$$a = a_{pol}(0) + a_{pol}(1)m_{fret} \quad (2.12)$$

$$m_{fret} = -12 \log_2(L) \quad (2.13)$$

where  $g_{pol}(n)$  and  $a_{pol}(n)$  represent the different polynomial coefficients and  $m_{fret}$  represents the fret number.  $m_{fret}$  is derived from the relative string length based on the rules of the 12-tone equal temperament tuning system. These equations illustrate filter parameter values that are interpolated via first-order polynomials. The polynomial coefficients change for each string and are specified in Table 2.1 and Table 2.2. Figure 2.9 shows the polynomial approximation of  $g$  as well as the values extracted from the recordings for the high E string. It is not an exact fit, however

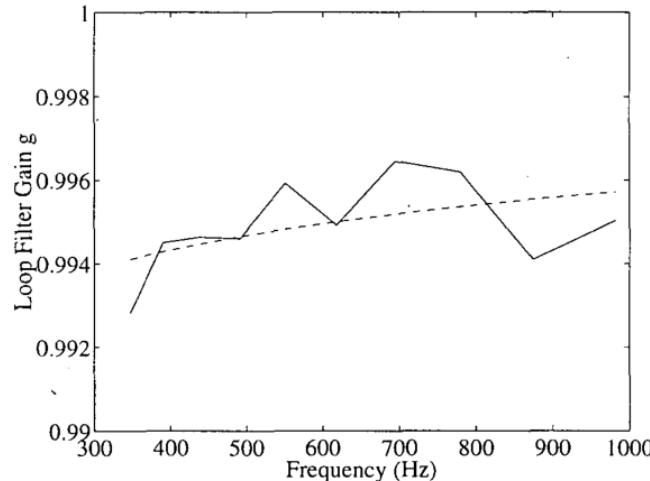
it works well enough. Additionally, the plot is specified across the fundamental frequencies of the notes which are generated at each different fret number.

	$g_{pol}(0)$	$g_{pol}(1)$
String 1	0.99402123928178	0.00008928138142
String 2	0.99247813966550	0.00012644399078
String 3	0.99012478445221	0.00025250158133
String 4	0.98780640700360	0.00037712305083
String 5	0.98347976839019	0.00040239847018
String 6	0.97816203269973	0.00061375406757

**Table 2.1** The coefficients for the first-order polynomial fit to the loop-gain data [Välimäki and Tolonen 1998].

	$a_{pol}(0)$	$a_{pol}(1)$
String 1	-0.02955827361150	0.00134421335136
String 2	-0.03042891937178	0.00113090288951
String 3	-0.03840938807507	0.00081125415233
String 4	-0.06091679973956	0.00298025530804
String 5	-0.05928143968051	0.00171045642780
String 6	-0.08135045114297	-0.00085796015850

**Table 2.2** The coefficients for the first-order polynomial fit to the filter cut-off frequency data [Välimäki and Tolonen 1998].



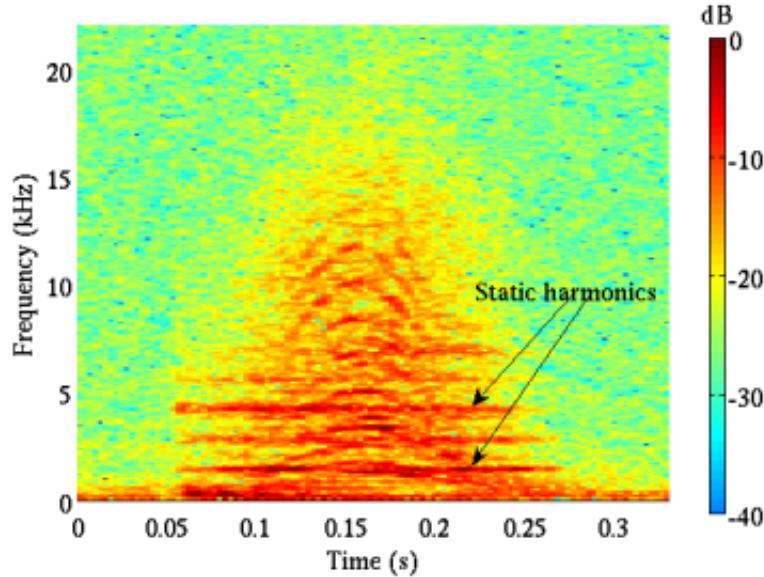
**Fig. 2.9** The loop gain  $g$  for modeling the high E string (solid line) and the first-order polynomial fit (dashed line) as a function of frequency [Välimäki and Tolonen 1998].

### 2.3 CSG Development

The handling noises based on wound string and finger interactions have previous been investigated in [Pakarinen, Penttinen, and Bank 2007] and [Penttinen et al. 2006]. The slide and string interactions are slightly different, however the underlying framework developed by the finger-noise analysis is valid as the slide synthesis algorithm was adapted from it. In this subsection, the finger-noise analysis will be introduced first followed by the adaptations necessary for the slide scenario.

#### 2.3.1 Contact Sound Analysis

Figure 2.10 shows the spectrogram of the noise generated by dragging a finger tip along the surface of a wound guitar string [Pakarinen, Penttinen, and Bank 2007]. As can clearly be seen, there are two spectral components to the sound. The first is a time-varying harmonic component which corresponds to the interactions of the finger tip with the spatially periodic windings on the string's surface. The second is a static component which is due to the longitudinal modes being stimulated by the finger tip's impacts with the windings. Similar results have been shown in [Penttinen et al. 2006] where the guqin, a fretless Chinese stringed instrument using wound strings, was examined.



**Fig. 2.10** The spectrogram of handling noise generated by sliding a finger on a wound guitar string from [Pakarinen, Penttinen, and Bank 2007].

#### Harmonic Component

An object moving along a wound string creates a repeated force excitation to the string based on its velocity as well as the texture of the string's surface [Pakarinen, Penttinen, and Bank 2007].

The time-domain force signal generated when a finger moves along a wound string might therefore be approximated by:

$$F_{harm}(t) = \left[ \sum_k \delta(t - t_k) \right] * f(t) \quad (2.14)$$

where  $f(t)$  is the impulse response from a single finger-to-winding collision,  $t_k$  is the time instant where the  $k$ th impulse response is generated and  $\delta$  is Dirac's delta function. A finger-to-winding collision is defined as when a finger tip slides from one winding and impacts the next. This equation can also be interpreted as a repeated sequence of impulses which is filtered by the transfer function of a single finger-to-winding collision.

To understand how the  $t_k$  values are established, let's assume the finger motion is described by a constant velocity. The rate at which winding collisions will occur is a function of the finger velocity as well as linear string winding density. Given that this rate is inversely proportional to  $t_k$ , the following mathematical expression holds:

$$t_k = \frac{k}{n_w |v_f|} \quad (2.15)$$

where  $n_w$  represents the linear winding density of the string (winds per meter) and  $v_f$  represents the finger tip velocity (meters per second), and  $k$  is an indexing variable to distinguish the order of the impulses. This means the quantity  $n_w |v_f|$  has units of windings per second and represents the rate of collisions (as alluded to before). By either increasing the linear winding density or finger speed, the frequency at which the impulse responses are generated can be controlled. As the  $n_w$  parameter is constant for a string, the instantaneous fundamental frequency of the generated harmonic component is controlled by the speed at which the finger moves.

This theory can be verified empirically by observing the spectrum in Fig. 2.10. In the figure, the finger starts at rest and accelerates to reach its maximum velocity halfway through the slide. At this point it begins to decelerate eventually returning to rest once the movement comes to an end. The minimum and maximum values of harmonic frequency trajectories illustrate the same behavior. The harmonics follow a differentiable trajectory. It can be concluded that the finger velocity must also be differentiable, which is useful information when generating control signals for the synthesizer.

As shown experimentally in [Pakarinen, Penttinens, and Bank 2007], the amplitude of the harmonic noise component is linearly related to the slide velocity. This can also be intuited from a physical standpoint given that the higher the finger velocity, the more momentum is transferred to the string during the collision. Assuming linearity, this would manifest itself as a velocity dependent scaling component associated with each  $t_k$  value in Eq. 2.14.

### Static Component

The other component from the contact sound, which has been explicitly labeled in Fig. 2.10, is a static component due to the longitudinal modes of the string. To understand this, it is necessary to examine the partial differential equation which describes the longitudinal vibrations in a string.

As derived in [Bank 2006], the longitudinal wave equation for a string can be expressed as:

$$\mu \frac{\partial^2 \xi}{\partial t^2} = ES \frac{\partial^2 \xi}{\partial x^2} - 2R(f)\mu \frac{\partial \xi}{\partial t} + d(x, t) \quad (2.16)$$

where:

- $\xi(x, t)$  is the longitudinal displacement
- $E$  is Young's modulus, which is related to the string's stiffness
- $S$  is the string's cross sectional area
- $\mu$  is the linear mass density
- $R(f)$  is a frequency dependent frictional resistance
- $d(x, t)$  is the excitation force density

For a longitudinal string wave, the propagation speed is  $c_L = \sqrt{\frac{ES}{\mu}} = \sqrt{E\rho}$ , where  $\rho$  is the density of the material. Contrary to transverse modes, the longitudinal propagation speed does not depend on the tension in the string. Due to this constant longitudinal wave speed and the fact that the slide does not change the length of the string from a longitudinal perspective, the longitudinal modes are constant in frequency.

As shown in [Pakarinen, Penttinen, and Bank 2007] and [Morse 1981], for a given  $F(t)$  applied at  $x_{exc}$ , the bridge force can be expressed as:

$$F_b(t) = \frac{ES}{\mu L^2} \sum_{k=1}^{\infty} \left[ \frac{k}{f_k} e^{-tR(f_k)} \sin(2\pi f_k t) \right] * \left[ \sin\left(\frac{k\pi x_{exc}}{L}\right) F_{exc}(t) \right] \quad (2.17)$$

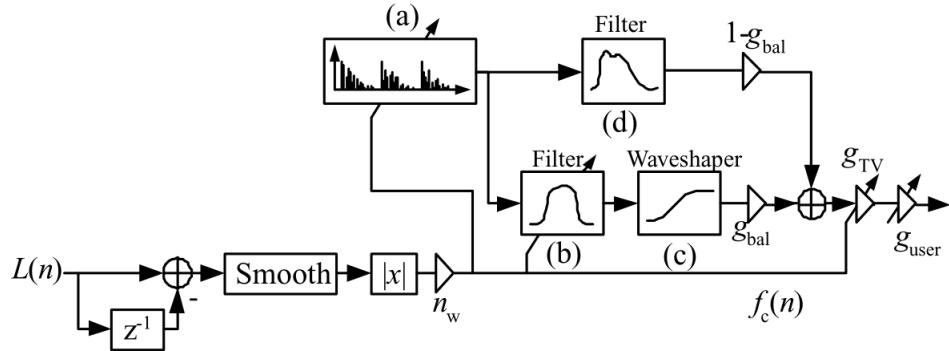
where the longitudinal modal frequencies are  $f_k = k \frac{c_L}{2L}$ ,  $x_{exc}$  is the point of excitation and  $F_{exc}(t)$  is the excitation force applied. Equation 2.17 shows that the force signal excites a set of parallel resonances where the excitation amplitude depends on  $x_{exc}$ . In general,  $x_{exc}$  has a strong shape over the spectrum and in the cases where a mode has a node at that point, the harmonic will be eliminated. This theory was experimentally verified in [Pakarinen, Penttinen, and Bank 2007]. This equation also gives rise to the exciter-resonator perspective as the longitudinal modes can be interpreted as a set of parallel resonances stimulated by an excitation,  $F_{exc}(t)$ , whose amplitude depends on  $x_{exc}$  [Pakarinen, Penttinen, and Bank 2007].

### 2.3.2 Adapting From Finger to Slide

Extrapolation from a finger tip to a rigid slide object is straightforward. The physical properties of the string do not change, only the surface of the object interacting with the string. The only part of the equations which would change is the impulse response function  $f(t)$  in Eq. 2.14. This would now represent the impulse response of a single slide-to-winding collision. Additionally, the finger tip velocity should be replaced by the slide velocity.

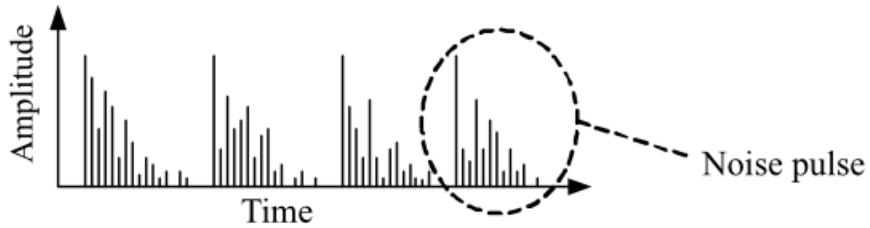
### 2.3.3 Model Description

The CSG of the slide synthesis model can be seen as a discretization of the exciter-resonator model developed in the previous section. Figure 2.11 shows a high-level signal-flow diagram of the original CSG presented in [Pakarinen, Puputti, and Välimäki 2008]. A noise pulse train is chosen as the



**Fig. 2.11** The CSG model from [Pakarinen, Välimäki, and Puputti 2008].

excitation signal (labeled as block (a) in Fig. 2.11). This choice is based on the assumption that the impulse response from a single slide-to-winding collision can be computationally modeled as an exponentially decaying noise burst [Pakarinen, Puputti, and Välimäki 2008]. Example noise bursts are illustrated in Fig. 2.12. The time-interval between noise pulses is controlled by the slide velocity as indicated in Eq. 2.15. In a certain regard, the CSG can be viewed more as a periodic impact synthesis model. The faster the slide moves, the denser the pulse train becomes, with some of the IRs overlapping depending on the decay rate associated with the string-to-winding collision. The decay rate and correspondingly duration of the noise pulses are parameters associated with each different string thickness. This model is meant to perceptually achieve a similar sound so some theoretical aspects are missing. Contrary to the theory, the location of stimulation does not scale the amplitude of the stimuli. The internals of this module will be elaborated upon in a later chapter.



**Fig. 2.12** Noise pulses from [Pakarinen, Välimäki, and Puputti 2008].

At the lower-left branch of the CSG, the control signal  $L(n)$  is input. This represents the relative length of the string. The first-order derivative of  $L(n)$  is approximated to generate the relative slide-velocity. From there the signal runs through a block labeled “Smooth” which

performs two different operations. The first action is to smooth the input signal to help handle any discontinuities which arise during the differentiation process. A discontinuity in slide-velocity would not be physically possible due to the human controlled nature of the slide motion. The second operation the block performs is interpolation in the case that the control signal runs at a different sampling frequency than the audio signals (as is often the case). The interpolation allows the constituent synthesis model processing blocks to adjust their parameters in a more gradual way as well and mitigates the introduction of transients in digital filters. In this scenario, ultimately a noise-like signal is being filtered so this is not as much of a concern but good to be aware of in general.

After the slide velocity has been up-sampled to the audio rate, the absolute value is then taken to produce the slide speed. Given that the impulse response generated from the impact of the slide with a winding is agnostic to the direction the slide is traveling, this is a valid operation. Subsequently, the relative slide speed is multiplied by  $n_w$ , the linear winding density, to generate the control signal labeled  $f_c(n)$ . This signal mimics the relationships expressed in Eq. 2.15. Accordingly, it controls the firing rate of the noise pulse generator as well as scales the output via the gain block  $g_{TV}$  to match the amplitude measurements made in [Pakarinen, Penttinen, and Bank 2007].

The output of the noise pulse generator goes to two different branches, each approximating a different aspect of the contact sound. The lower branch is a 2nd-order resonator filter followed by a waveshaper implemented via a hyperbolic tangent function. The 2nd-order resonator has its center frequency controlled by the aforementioned  $f_c(n)$  in order to extract the time-varying fundamental from the harmonic pulse train output. The series connection with the waveshaper creates and accentuates the higher harmonics in a computationally efficient manner. The number of harmonics can be controlled via a scaling factor to the input of the hyperbolic tangent. This is meant to be tuned to match the number of harmonics that have been observed in measurements.

The upper branch serves to emulate the static longitudinal modes using the filter labeled (d). It is a 4th-order IIR which approximates the two most prominent longitudinal modes observed. The coefficients of the filter are dependent on the different string/slide combinations as the different slide materials interact with the windings in a different manner. The filter's responses have been approximated via linear predictive coding (LPC) as described in [Pakarinen, Puputti, and Välimäki 2008].  $g_{bal}$  is a scaling coefficient which controls the balance between the longitudinal modes and the harmonic contact sound components.

## Unwound Strings

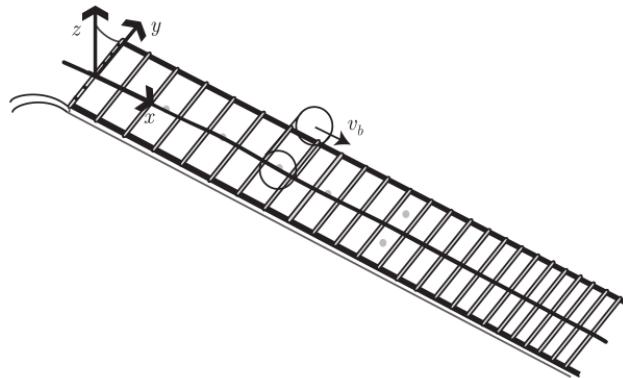
Not depicted in Fig. 2.11 is the model which generates the sounds for unwound strings. In this scenario the smooth surface of the slide interacts with the smooth surface of the string, resulting in a pure friction model for the synthesis with no longitudinal component. This can be achieved by replacing block (a) with a white noise generator whose output is scaled by the slide velocity [Pakarinen, Puputti, and Välimäki 2008].

## 2.4 Recent Developments and Approaches

[Pakarinen, Puputti, and Välimäki 2008] is one of the first articles published regarding slide guitar synthesis. There have been refinements and advancements since it was released in 2008. These approaches tend to take a more theoretical approach as compared to the empirical one described in [Pakarinen, Puputti, and Välimäki 2008]. The theoretical refinements add computational complexity at the potential cost of a real-time implementation.

### 2.4.1 Evangelista's DWG Approach

In 2012, Gianpaolo Evangelista published a paper as part of an AES convention which proposes a physically inspired DWG-based model [Evangelista 2012]. His model is a theoretical advancement in many ways, one of which is an expansion in the number of dimensions used to model the vibrations as indicated in Fig. 2.13. Two DWG structures model the transverse vibrations, one for each of the orthogonal polarizations. Additionally, a DWG is also used to model the longitudinal vibrations as opposed to an IIR filter. This would more accurately model the system and makes the slide's direction important in addition to its speed. The transverse and longitudinal DWGs are weakly coupled in a non-linear fashion at the point where the slide has been placed.

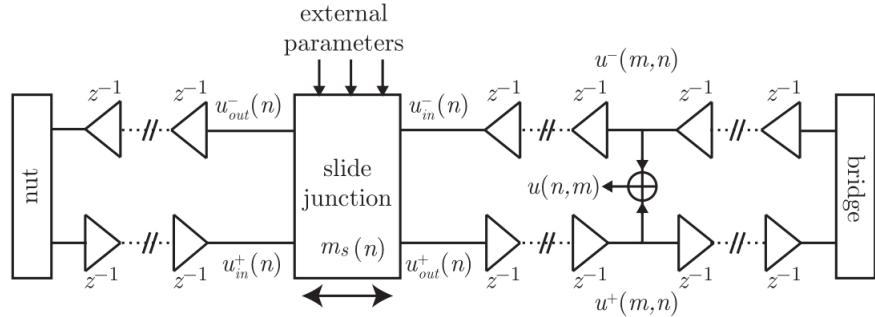


**Fig. 2.13** The coordinate system used to describe string and slide motion in Evangelista's approach [Evangelista 2012].

As a basis for the friction sounds, Evangelista uses the Lu-Gre model. The Lu-Gre model only considers flat surfaces interacting and the wound string surface is a violation of this assumption. A new noise term has been added to the model to compensate and this addition is supported by experimental evidence. A non-linear state-space model is used to describe the string and Lu-Gre bristle interactions. This produces a single equation which is solved via the Newton-Raphson method. Additionally, the transverse vibrations are taken into account when generating the friction sounds as there is minor interaction between the string and slide at the point of contact.

A position-dependent slide junction has also been introduced to refine the model further. This is placed at integer string samples and uses fractional delay methods when required. The slide junction models the interactions between the slide and string based on the respective materials of each component. A single parameter is used to control elasticity of the string-winding coefficients

on the range from being completely inelastic to elastic. The equations governing the slide/string interactions at each node are non-linear. Figure 2.14 depicts the slide junction.



**Fig. 2.14** The slide junction model introduced in [Evangelista 2012].

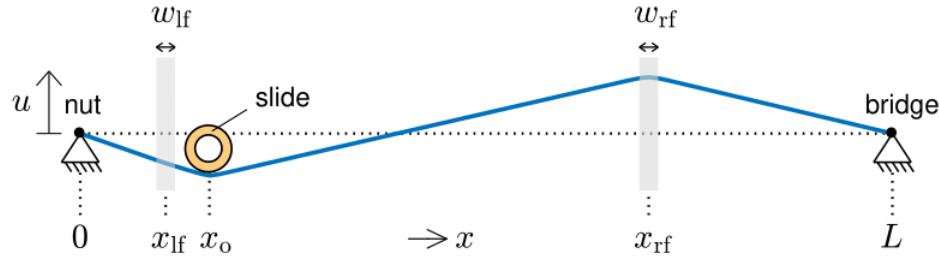
A less substantial change is added support for either round-wound or flat-wound strings in the synthesis model by changing the geometry of the string surfaces. There is no mention of the control signal and its characteristics for optimum realism. He makes a claim that this could be run as a real-time model however no evidence is provided to support this. Given that the complexity of the theoretical advancements substantially increase the amount of computation necessary to render a sound, it is questionable if a real-time implementation is possible. Unfortunately sound samples are not included to assess the quality of the results.

#### 2.4.2 Belfast Finite-Difference Technique

The most recent approach to slide guitar synthesis is a finite-difference approach published by a group of researchers at Queen's University Belfast [Bhanuprakash, Stapleton, and Walstijn 2020]. The fundamental approach itself is also a departure from the previous models, as it attempts to numerically simulate the differential equations used to describe the string motion as opposed to simulating the traveling wave variables. By contrast, this is a much more mathematically and computationally intense and does not lend itself easily to real-time synthesis applications. There are also various stability criteria which must be satisfied. The extra complexity is worth it however as the sounds produced are substantially more expressive and realistic. The audio examples can be found at <https://abhiram1989.wixsite.com/slidesstringfdmodel>.

In terms of the physical model, advancements have been made to factor in the left and right hand actions. The damping provided by the part of the left-hand behind the slide is explicitly modeled as this is an integral part of the technique involved on a real instrument. The slide attachment/de-attachment sounds are taken into account as these are included within the hand motions. Comparatively speaking the vibration modeling is simpler than Evangelista's or the Finnish approach described in [Pakarinen, Puputti, and Välimäki 2008]. Only one polarization for transverse vibrations is included and the longitudinal vibrations are neglected entirely. Figure 2.15 illustrates the spatial layout of the model.

From a friction standpoint the model used is also lacking as friction is not explicitly modeled (in contrast to the other approaches). However, in the synthesized sounds a noise component can



**Fig. 2.15** This shows the spatial layout for finite difference approach. The gray areas represent finger force regions [Bhanuprakash, Stapleton, and Walstijn 2020].

be heard. This is especially true during vibrato sections of phrases. This implies that the sliding noise is not purely based on friction and is likely a mix of friction as well as restoring force-based phenomena which would represent an advancement in the overall knowledge of physics of slide guitar in general.

The control of the model to properly generate realistic and musical sounding results is also a crucial focus of the paper. Four control signals are used in the examples they have provided. These include the slide position, the left hand height, the force applied by the left hand and the force applied by the right hand when plucking. Video footage of expert players was analyzed in an attempt to understand the motions better. Additionally, the researchers attempted to gain an embodied understanding of the motions by learning to play the instrument themselves. The results have paid off as is illustrated by the expressiveness and realism in the synthesized sounds.

## Chapter 3

# Description of Slide Guitar Synthesis Model

In this section, we introduce the synthesis model which was developed based on the theory from Chapter 2. After the model has been introduced, there will be an exploration of the limitations of the design and the theoretical reasons behind them.

### 3.1 Introduction of Architecture

The slide guitar synthesizer developed in this thesis is heavily influenced by the model introduced in [Pakarinen, Puputti, and Välimäki 2008] and [Puputti 2010]. Modifications were made during development and they will be explained as they are introduced. The model consists of three main components: the Control Signal Processor (CSP), the Contact Sound Generator (CSG) and a String Digital Waveguide (DWG) model. The Control Signal Processor's main job is to compute derived parameters from the control signal which are used to control the other two components. The Contact Sound Generator produces the sound corresponding to the slide and string surface interactions. The String Digital Waveguide synthesizes the sounds associated with the transverse vibrations of a plucked string of time-varying length. Audio examples will also be provided in an effort to facilitate an aurally intuitive understanding of the model by bridging the gap between the theoretical design and the perceptual/experiential end result.

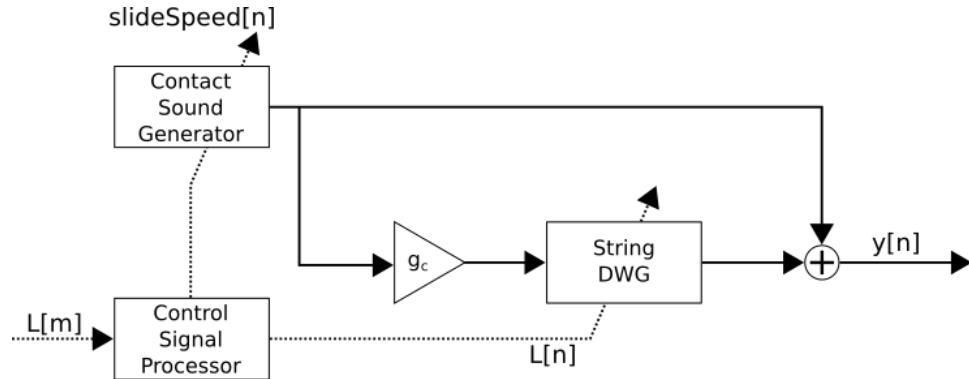
#### 3.1.1 Diagram Conventions

The following conventions will be adhered to in this section's diagrams. This was done to improve clarity and reduce ambiguity as compared to the diagrams from the original papers [Pakarinen, Puputti, and Välimäki 2008] and [Puputti 2010]. Figure 3.1 serves an illustrative example.

In synthesis systems, signals can functionally divided into two different categories: control signals and audio signals. This is illustrated by the use of dashed lines for control signals, and the use solid lines for audio signals. It is also common that the control signals are run at a sampling rate which is lower than or equal to the audio rate. This has been represented by the use of different indices for the time-index.  $[m]$  represents a signal at the control rate while  $[n]$  represents

a signal at the audio rate. It is possible to have a control signal at the audio rate as is illustrated by  $L[n]$  in Fig. 3.1.

### 3.1.2 Single String Slide Synthesizer



**Fig. 3.1** The high-level architecture for a single string slide synthesizer.

The highest-level component of the synthesis system is depicted in Fig. 3.1. This is a synthesizer for a single string where the pitch is controlled by a slide. Similar to the model introduced in Chapter 2, this consists of a module which represents a variable length string digital waveguide as well as a contact sound generator for the string/slide surface interactions.

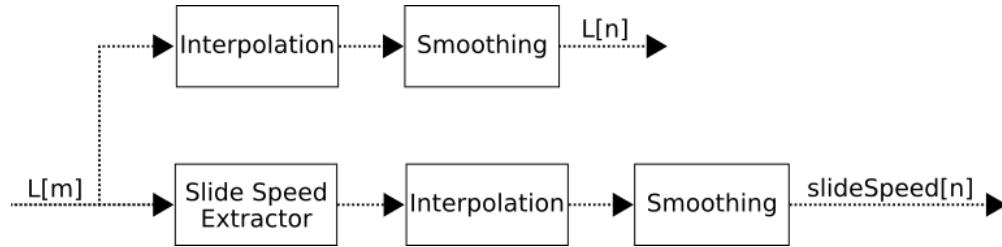
The first new addition is a gain block which controls the coupling between the longitudinal motion of the slide and the transverse vibrations of the string. The CSG model only considers longitudinal slide motion in its algorithm. The coupling phenomena was experimentally observed on a per string basis, as will be shown in Sec. 5.3.

The Control Signal Processor is another new block which will be more carefully detailed in the next section. It was placed at the highest possible layer in the architecture. This placement improves computational efficiency by removing redundant computations in lower-levels as well as ensures all the constituent processing and synthesis objects are operating on a common set of control signal values.

### 3.1.3 Control Signal Processor

Figure 3.2 illustrates the internals of how the Control Signal Processor operates. Its input signal is the relative length control signal at control rate. Its output signals are the slide speed as well as relative length signal at audio rate. The purpose of this block is to extract the slide speed signal as well as change the control rate signals to audio rate.

Linear interpolation is used to upsample the control rate signals to the audio rate. This allows more gradual changes between control signal values to be seen by the audio rate objects as this is beneficial for preventing unwanted artifacts (e.g, transients). Supposing that  $R$  represents the ratio between the control rate and audio rate, then for each one control-sample,  $R - 1$  audio

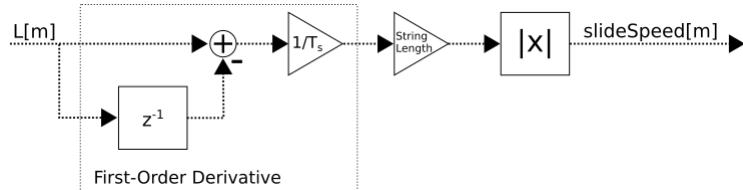


**Fig. 3.2** The signal-flow diagram for control signal processor block.

samples are calculated via the interpolation. In the case where the audio rate is 48,000 kHz and the control rate is 1 kHz then  $R = \frac{48,000}{1,000} = 48$  and 47 additional samples would be calculated.

The smoothing helps eliminate any discontinuities which may be present in the interpolated signal. It is implemented via a 10-point moving window averager.

### Slide Speed Extractor

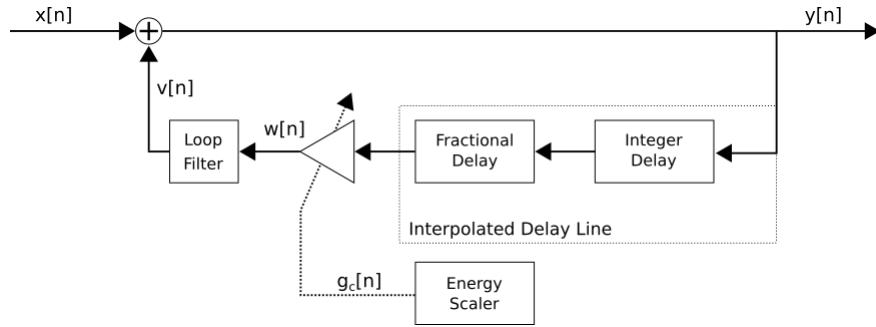


**Fig. 3.3** The signal-flow diagram for slide speed extractor block.

Figure 3.3 shows a signal-flow diagram for the slide speed extractor. This is similar to the model introduced in [Pakarinen, Puputti, and Välimäki 2008] with some refinements for precision and clarity. Functionally it operates in the following manner. The first step is to take a difference between two consecutive samples and divide this by the sampling period. This is an approximation of the first derivative and has units of  $\frac{\Delta \text{relative length}}{\text{sec}}$ . The next step is to convert this from a relative length to an absolute length through multiplication by the length of a string in meters. This produces the absolute slide velocity in  $\frac{\text{meters}}{\text{sec}}$ . From there, the absolute value is taken to convert the velocity to a speed. Speed is used here as the Contact Sound Generator is agnostic to the direction the slide moves and is the only synthesis object which is concerned with slide speed.

#### 3.1.4 String DWG

Figure 3.4 illustrates the string digital wave guide model. The model itself does not differ from the original as described in [Pakarinen, Puputti, and Välimäki 2008]. However, the diagram here differs in an attempt to improve clarity as compared to the original.  $L[n]$  is not depicted as all the signal processing blocks shown rely on this in some manner. Additionally, there have been intermediate signals introduced ( $v[n]$  and  $w[n]$ ) as they were beneficial in developing the implementation.



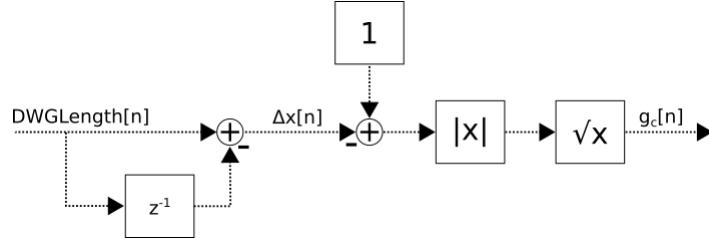
**Fig. 3.4** The signal-flow diagram for the time-varying string DWG.  $L[n]$  is not depicted as every object consumes it in some fashion.

### Energy Scaler

It was necessary to slightly rearrange the original energy scaling equation (Eq. 2.9) from Chapter 2 to indicate the time-varying aspect of  $g_c[n]$  and help facilitate its computation. The implementation computes the gain coefficient using the following expression:

$$g_c[n] = \sqrt{1 - \Delta x[n]} \quad (3.1)$$

where  $\Delta x[n]$  is the difference in the length of the digital waveguide between two successive samples and  $g_c[n]$  is the coefficient which performs the scaling. Figure 3.5 illustrates the signal-flow diagram of the energy scaler object which computes this gain value.



**Fig. 3.5** The signal-flow diagram for the energy scaler block implementing Eq. 3.1.

### Loop Filter

The loop filter is single-pole and simulates the losses associated with string vibration. It has been described previously in Sec. 2.2.2. There is no novelty in its implementation due to the standardization of the single-pole design.

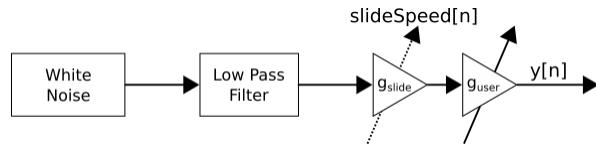
#### 3.1.5 Contact Sound Generator

Two varieties of the Contact Sound Generator exist, corresponding to the two different varieties of strings which exist. First, the unwound approach will be described. It is the simpler of the two

models and describes the sound produced when the slide interacts with the smooth surface of an unwound string. Subsequently, the more complex wound string variant will be examined in detail.

### Unwound Strings

Figure 3.6 shows the signal-flow diagram for the unwound Contact Sound Generator. The unwound strings have a substantially simpler algorithm as the sound is generated from two smooth surfaces interacting with each other. It is more akin to a friction sound generator as opposed to an impact sound, which matches the interaction between the surfaces. This contact noise can easily be modeled by low-pass filtered white noise which has its amplitude scaled by the slide's speed. A user-tunable parameter for the overall contact sound level is placed at the end of the chain. This does not differ from the original design described in [Pakarinen, Puputti, and Välimäki 2008] and implemented in [Puputti 2010].



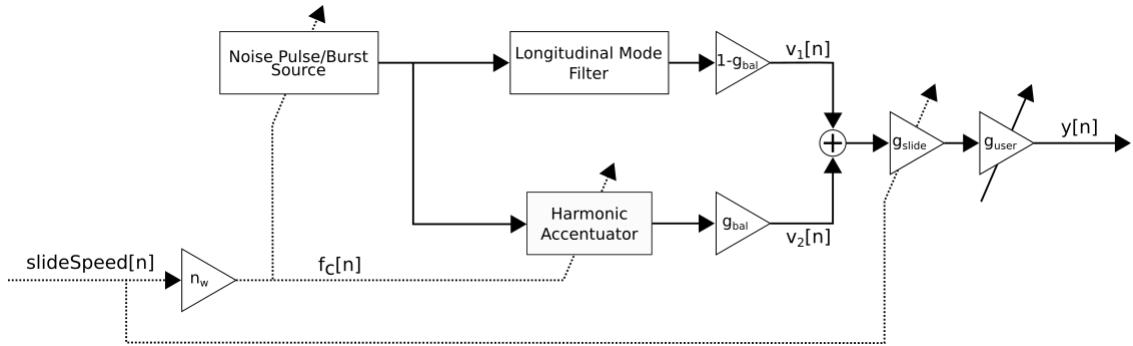
**Fig. 3.6** The signal-flow diagram for unwound Contact Sound Generator block.

### Wound Strings

Figure 3.7 illustrates the signal-flow diagram for the wound string Contact Sound Generator. The core functionality does not differ from the module which was suggested in [Pakarinen, Puputti, and Välimäki 2008] in that it uses the speed of the slide to generate a sound containing a time-varying harmonic component ( $v_2[n]$ ) and a static component due to the longitudinal modes ( $v_1[n]$ ). It does, however, differ substantially from the implementation in [Puputti 2010]. Additionally, variations of different components have been implemented to experiment with different timbres as indicated by the more general “Noise Pulse/Burst Source” and “Harmonic Accentuator” blocks.

The longitudinal mode filter remains a 4th-order IIR using the same coefficients as specified in the original paper [Pakarinen, Puputti, and Välimäki 2008]. There is also the linked pair of gain blocks which allows the balance between the static and harmonic components to be varied. The last gain block,  $g_{user}$ , allows the overall sound level to be specified (as in the unwound implementation).

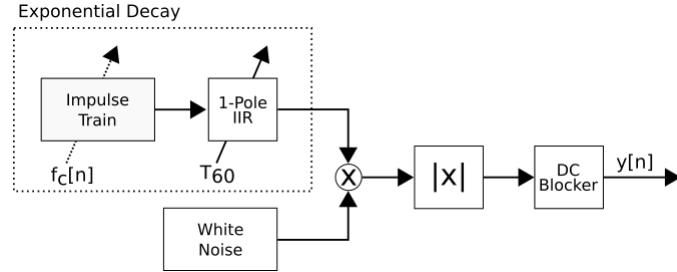
The first step in the wound Contact Sound Generator is to convert the incoming  $slideSpeed[n]$  to a frequency based on the linear density of string windings associated with the string. This  $f_c[n]$  represents the rate at which the slide-to-winding collisions occur. Its name comes from the fact that the control signal is used to control the centre frequency of the resonator filter in the harmonic accentuator which is tuned to the fundamental frequency of the generated contact sound. The  $n_w$  parameter is stored here to keep all the information specific to the string’s physical properties in a single location.



**Fig. 3.7** The signal-flow diagram for wound CSG block.

**Noise Source** Two variations for noise sources were developed through the course of this thesis. The first is conceptually not different from what was described in [Pakarinen, Puputti, and Välimäki 2008], however in implementation it differs significantly from the version in [Puputti 2010]. The second is more akin to what was introduced in the guqin model in [Penttinen et al. 2006].

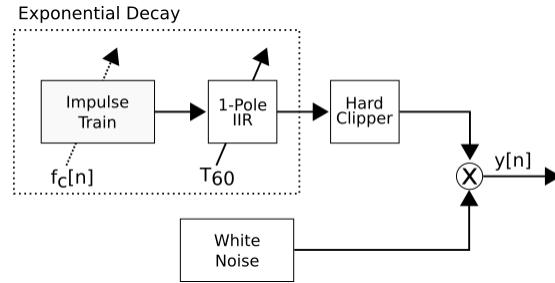
**Noise Pulse Train** Figure 3.8 illustrates the first variation on a noise source. It consists of absolute-valued white noise which has an amplitude envelope applied to it. This amplitude envelope is generated by an impulse train which is fed into a one-pole filter. The firing rate of the impulse train is controlled by the  $f_c[n]$  signal which mimics the generation of impulses from the slide hitting windings as it moves. The decay rate of the impulse response of the one-pole filter is controlled by the  $T_{60}$  value measured for each string (which will be elaborated upon in Sec. 5.5). The use of a one-pole filter allows the generated impulses to stack on top of each other and also benefits from being extremely computationally efficient. A DC blocker was added to remove the DC component introduced by the absolute-value function. Without applying an absolute value, the signal remains too noise-like to be harmonic.



**Fig. 3.8** The signal-flow diagram for the Noise Pulse Train.

**Noise Burst Generator** Figure 3.9 illustrates the second variation of a noise source. This attempts to combine the method used in the guqin model [Penttinen et al. 2006] with more string

specific characteristics as the slide guitar model [Pakarinen, Puputti, and Välimäki 2008]. White noise is multiplied by an amplitude envelope as before. However, in this variation the output of the one-pole filter is hard-clipped to a value of 1. In areas of a slow slide movement the output is similar to the noise pulse train, however as the slide speed increases and more windings are struck, the signal transitions into white noise and the harmonic component is lost (subject to the value of  $T_{60}$ ). The one-pole also ensures that the starting and stopping of the noise will be more “natural” with the addition of the decay rate. Otherwise, the envelope would be a pure step-function and not allow more gradual changes. This component was developed as the more noise-like characteristics of its output would perform better with the Harmonic Resonator Bank (described next) and to provide a second approach for comparison with the Noise Pulse Train.



**Fig. 3.9** The signal-flow diagram for the Noise Burst Generator.

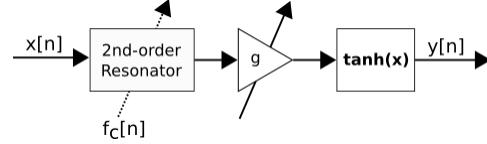
**Impulse Train** Both the Noise Pulse Train and Noise Burst Generator have a common component in the Impulse Train. The Impulse Train object outputs a discrete-impulse signal at a specified period in samples. Internally, it counts the number of samples which have passed until it is time to generate a new impulse. The period, in samples, is computed using the following equation:

$$\text{period} = \text{round} \left( \frac{F_s}{f_c[n]} \right) \quad (3.2)$$

**Harmonic Accentuation** Two variations for accentuating and generating the harmonics of the wound contact sounds were investigated. The first method is the same as what is described in [Pakarinen, Puputti, and Välimäki 2008], while the second is more akin to the method proposed for the guquin model [Penttinen et al. 2006]. These methods provide more control over the strength and number of harmonics as compared to what is available based purely on the parameters of the noise sources.

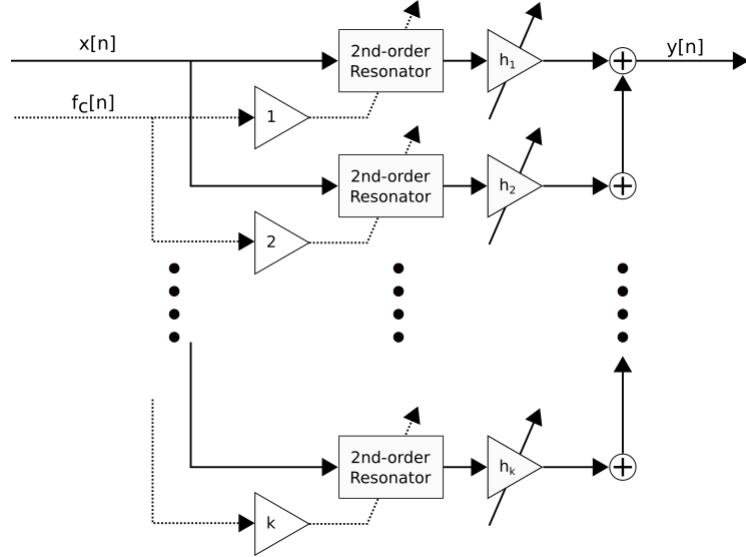
**Reso + Tanh** The first method is a second-order resonator in series with a hyperbolic tangent function as illustrated in Fig. 3.10. The second-order resonator has its center frequency controlled by  $f_c[n]$  and its  $r = .99$ . This configuration allows the filter to isolate the fundamental of the input signal. Assuming the input signal has a fundamental, the tanh function will introduce harmonics. The number of harmonics is controlled by the scaling factor  $g$  ahead of it in the

signal chain. This provides an extremely computationally efficient approach to generating the harmonics, at the expense of more fine-tuned control over the number and strength of each individual harmonic.



**Fig. 3.10** The signal-flow diagram for the Reso + Tanh waveshaper.

**Harmonic Resonator Bank** The second approach is illustrated in Fig. 3.11. This method is more computationally expensive, but provides much more control over the strength, number and location of the different harmonics. Additionally, it works with input sources that are both noisy and harmonic. It consists of a set of parallel second-order resonators whose centre frequencies are all harmonically linked to each other. At the output of each resonator is a tuneable gain coefficient to control the strength of the isolated harmonic. Six harmonics were chosen based on the spectrograms in [Pakarinen, Puputti, and Välimäki 2008], however the object itself supports an arbitrary number.



**Fig. 3.11** The signal-flow diagram for the Harmonic Resonator Bank.

## 3.2 Limitations of Model

### 3.2.1 Minimum Relative String Length

Although not explicitly mentioned in the source literature, there is a limit imposed on the minimum value  $L[n]$  can take based on the magnitude response of the Loop Filter. The limits of this can be deduced from the original figures which describe the loop coefficients in [Välimäki and Tolonen 1998].

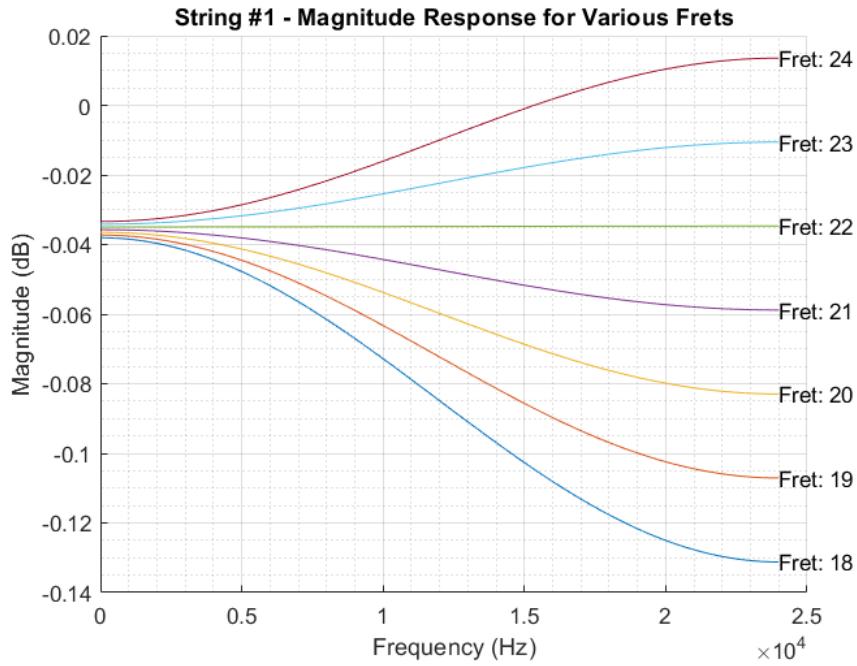
[Välimäki and Tolonen 1998] describe the method by which the polynomials used to generate the Loop Filter's  $a$  and  $g$  values are derived from recordings of a professional guitar player playing several tones on all frets of every string in an anechoic chamber. Unfortunately, there is no standardization between guitar manufacturers as to the appropriate number of frets for a guitar. Common values are 21, 22 and 24 [Erlewine 2001]. With knowledge of the model used by the player in the aforementioned paper, one could look up the number of frets on a manufacturer's website. However, this information is not available in the source material. By observing the Loop Filter's magnitude response as it changes based on  $L[n]$  and reverse-engineering the values in the original graphs it is possible to reasonably conclude the recordings used to derive the polynomials stopped at the 19th fret.

Figure 2.9 from the Sec. 2.2.2 illustrates the loop gain  $g$  across different frequencies for the high E string. By combining Eq. 2.6 and Eq. 2.13, it is possible to generate the fundamental frequencies associated with each fret on the string. Table 3.1 illustrates this for a selection of frets on the high E string. From this table we can see that the most likely candidate for the upper-most fret used by the player in the recordings is the 19th fret based on the upper limit for the x-axis in Fig. 2.9.

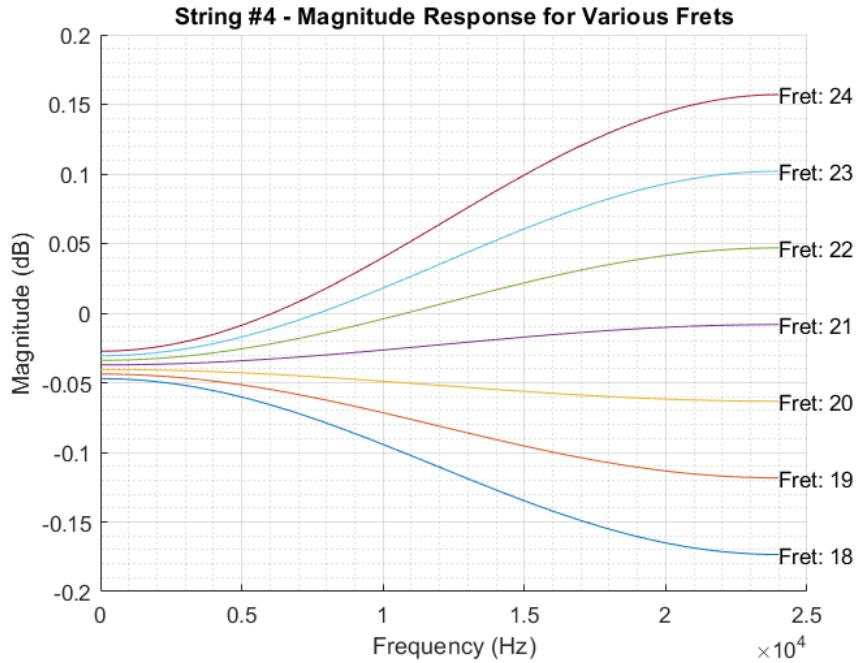
Fret #	Fundamental (Hz)
0	329.63
18	932.33
19	987.77
20	1046.5

**Table 3.1** The calculated fundamental frequencies for a selection of frets on the high E string.

Figure 3.12 and Fig. 3.13 illustrate the magnitude response for string #1 (the high E) and string #4 (the D string) across a variety of upper frets to show how they transition in this range. The general pattern is that each filter gradually shifts from a low-pass type response to more of a high-pass type response. Eventually the gain goes above 0 dB as well. This occurs at fret 24 for the high E string and fret 22 for the D string. Given the placement of the filter in the feedback loop of the digital waveguide, a positive gain value results in an unstable state. Based on the data presented in Fig. 2.9 and Table 3.1, the Loop Filter has clearly gone beyond the range of its original design under these conditions. However, this would be considered a limitation as in slide playing the slide often goes beyond the 24th fret. Even before reaching the point of positive amplification, the deviation from a low-pass filter causes the harmonics to decay in a manner which is not consistent with physical reality and is considered a limitation.



**Fig. 3.12** The loop filter magnitude response for the high E string at a variety of frets to illustrate its transition to a positive gain in the upper frequencies.



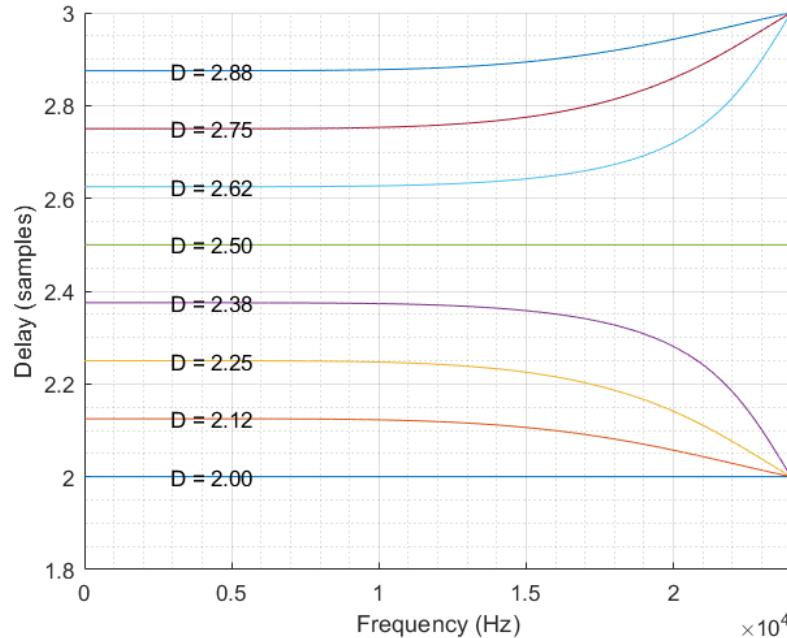
**Fig. 3.13** The loop filter magnitude response for the D string at a variety of frets to illustrate its transition to a positive gain in the upper frequencies.

### 3.2.2 Non-constant Phase Delay of Filters

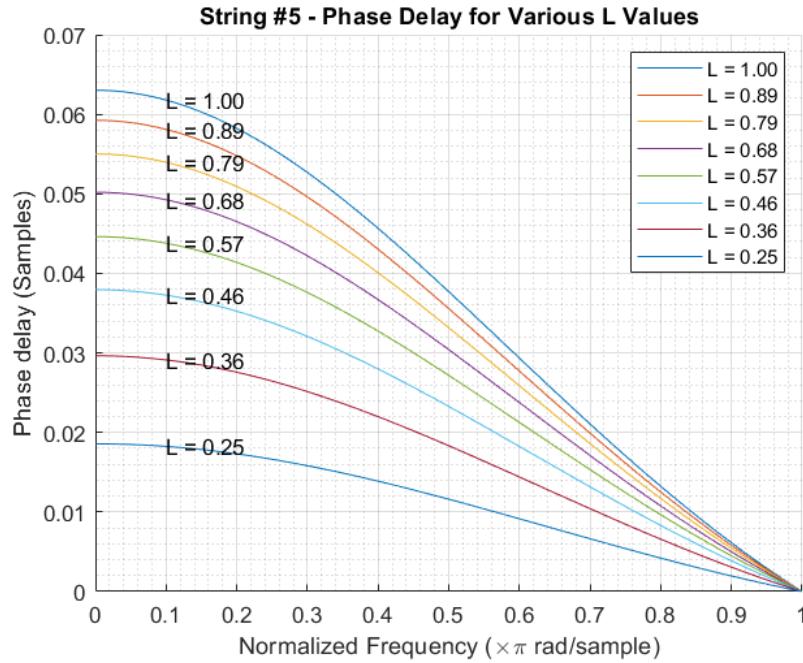
As mentioned in Sec. 2.1.1, the phase delay is a function of frequency indicating the time-delay, in samples, each frequency component experiences. The phase delay is physically correlated with the wave speed and in order to have the most accurate tuning, both these functions would be constant across the spectrum. However, a constant function is not always the most physically accurate.

Both the loop filter as well as the interpolation filter illustrate non-constant phase delays. This is shown in Fig. 3.14 and Fig. 3.15. The loop filter illustrates this as it is an IIR filter. This is inherent in their design [Oppenheim and Schafer 2010]. The interpolation filter is an FIR and under certain circumstances it can actually illustrate a constant delay (when the order is even and the fractional delay is 0.5) [Laakso et al. 1996].

Many strings in reality exhibit some form of stiffness which results in different wave speeds for different frequencies. This dispersion results in the observed overtones being slightly different than what an idealized string model would predict. The nature of a non-constant phase delay is similar to this, however it is more of an uncontrolled artifact here as opposed to intentional modeling. Although the values are small here and likely do not have an impact, they clearly vary with the relative length signal and ultimately will affect the accuracy of the tuning from a computational standpoint. Given that this model was originally designed to be played in real-time, this can easily be compensated for via “on-the-fly” tuning by ear if need be.



**Fig. 3.14** The phase delays for Lagrange interpolating filters with order = 5 at various delay values.



**Fig. 3.15** The phase delay for A string's loop filter at various relative lengths.

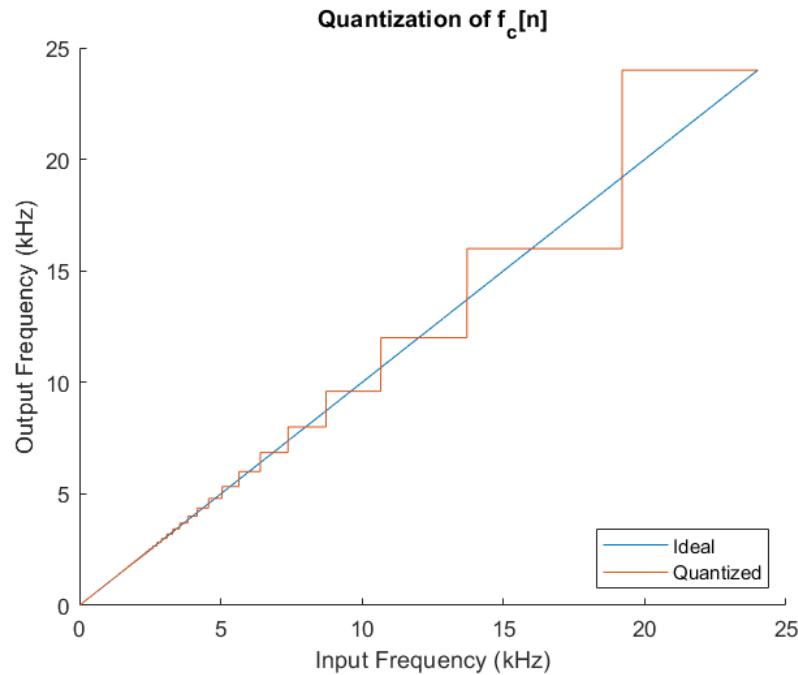
### 3.2.3 Impulse Train Quantization

The Impulse Train is limited to integer period values as its implementation is based on counting individual samples. Given the reciprocal nature between frequency and period, this ends up quantizing the frequencies which the Impulse Train effectively operates at. The quantization is worse at higher frequencies as compared to lower ones for a selected  $F_s$  and can be mitigated by increasing  $F_s$  as this correspondingly decreases  $T_s$  and increases the time-resolution of the system. Similar effects have been noted in [Jaffe and Smith 1983] in regards to tuning. One possible approach to mitigating this is by interpolating the impulse train signal.

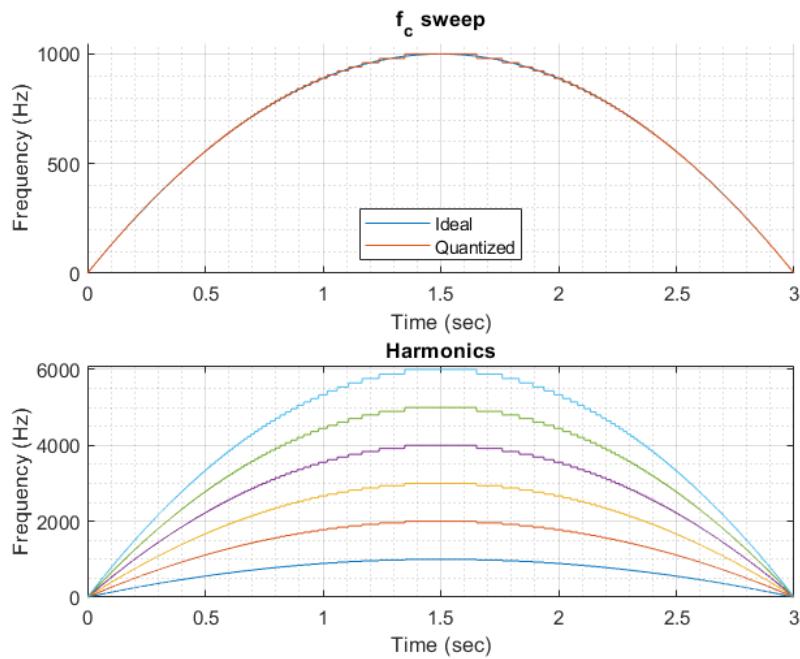
Figure 3.16 illustrates the effects of the rounding on the range of possible frequencies for a sampling rate of 48 kHz. The effects on an  $f_c[n]$  sweep matching the trajectory a slide might generate as well as the corresponding harmonics are shown in Fig. 3.17. The effects can be seen in the output of the Noise Pulse Train (see Sec. 3.1.5) object to the parabolic  $f_c[n]$  sweep shown in Fig. 3.18 and can be heard in *f\_cSweepQuantization-NPT.wav*<sup>1</sup>. The Noise Pulse Train object uses a  $T_{60}$  value tuned to generate harmonics as opposed to noise. The effects of this parameter will be detailed more fully in the Sound Design and Parametrization chapter.

---

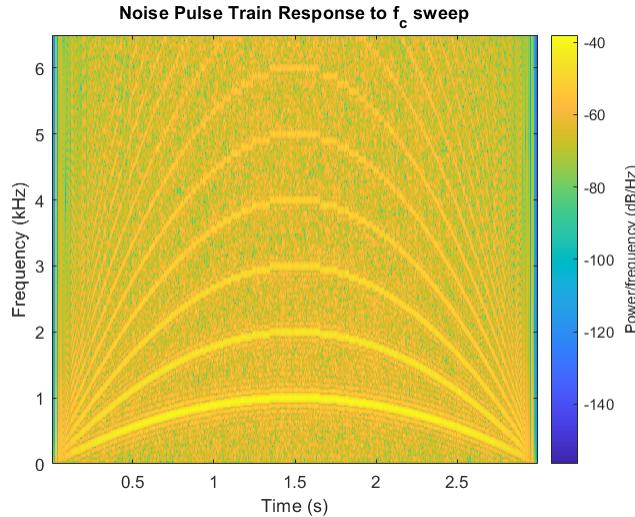
<sup>1</sup>Sound examples can be found: <https://github.com/dgsmith1988/Masters-Thesis/tree/main/Sound%20Examples>



**Fig. 3.16** The quantization effects from Eq. 3.2 for a sampling rate of 48 kHz.



**Fig. 3.17** The quantization effects for a parabolic sweep and the corresponding impact on the first six harmonics. Note how the effects are amplified in the higher harmonics.



**Fig. 3.18** Spectrogram of the output of the Noise Pulse Train in response to the parabolic  $f_c[n]$  sweep from Fig. 3.17. The plot has been zoomed in to emphasize the effects on the first six harmonics. The output can be heard in *f\_cSweepQuantization-NPT.wav*<sup>2</sup>.

### 3.2.4 Nyquist Rate Implications

A limitation in all digital systems exists as an upper limit on the frequencies which can be represented. This is referred to as the Nyquist Rate and is equal to  $\frac{F_s}{2}$ . With regard to the slide guitar synthesis model, the largest implication of the Nyquist Rate is an upper limit on the harmonics which can be represented in the system. From a physical standpoint, these harmonics above this Nyquist rate would exist. In actual practice, these upper harmonics are very weakly stimulated and contribute very little to the actual perception of the sound as the human range of hearing is limited to approximately 20 kHz [Plack 2018]. Using a sampling rate of 48 kHz results in a Nyquist Rate of 24 kHz, and taking into account the limits of human hearing the physical inaccuracies are acceptable from the standpoint of a sound synthesis algorithm.

---

<sup>2</sup>Sound examples can be found: <https://github.com/dgsmith1988/Masters-Thesis/tree/main/Sound%20Examples>

## Chapter 4

# Verification Of Slide Model and Constituent Components

In the following chapter, the testing methods used to verify the correctness of the synthesis algorithm will be detailed. As an overall strategy, the various function blocks (e.g., integer delay lines and filters) were verified first before being integrated into the larger constituent components (e.g., noise generators and string digital waveguides) which make up the synthesis model itself. The lower-level components will be introduced first as they are what the higher-level objects are built from. The objects are also roughly organized according to where they appear in the model. For instance, the constituent Control Signal Processor objects are grouped together. Filters are an exception to this rule. Full details of the testing scenarios described can be found in the link found in Appendix B.

These testing and verification methods are meant to be purely functional and as objective as possible. The subjective aspects (e.g., usability, musicality) will be considered more thoroughly in Chapter 6 which covers sound design and parametrization of the synthesizer.

### 4.1 Filters

All the filters were implemented using a class built around MATLAB's *filter()* function. This class holds the current filter state and the filter's coefficients as well as provides functions for computing the next sample of output and generating frequency responses. The general strategy for verifying correct operation of the filters is ensuring their frequency response or impulse response is correct. An exception to this rule is the Loop Filter, which will be explained in more detail in Sec. 4.6.3.

#### 4.1.1 Resonator

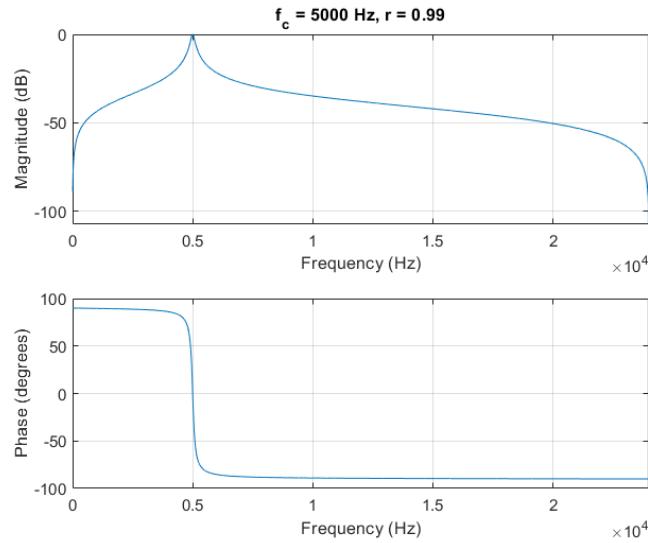
As a second order filter, the resonator has the difference equation:

$$y[n] = b_0x[n] + b_1x[n - 1] + b_2x[n - 2] + a_1y[n - 1] + a_2y[n - 2] \quad (4.1)$$

The coefficients are defined as:

$$\begin{aligned} b_0 &= \frac{1-r^2}{2} \\ b_1 &= 0 \quad a_1 = -2r \cos(2\pi f_c T_s) \\ b_2 &= -b_0 \quad a_2 = r^2 \end{aligned} \tag{4.2}$$

As a test, the class was configured with the following parameters:  $F_s = 48,000$  Hz,  $f_c = 5,000$  Hz and  $r = .99$ . This is illustrated below in Fig. 4.1.



**Fig. 4.1** The measured frequency response for 2nd-order resonator test.

#### 4.1.2 DC Blocker

The DC blocker is implemented with the following difference equation:

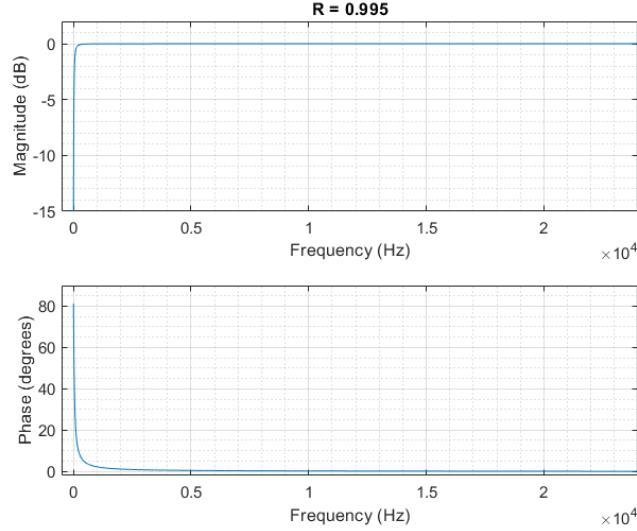
$$y[n] = g(x[n] - x[n - 1]) + Ry[n - 1] \tag{4.3}$$

where  $g = \frac{1+R}{2}$ . As there is a feedback term ( $y[n - 1]$ ), this is an IIR filter. Figure 4.2 illustrates the DC blocker's frequency response for a value of  $R = .995$ .

#### 4.1.3 Smoothing Filter

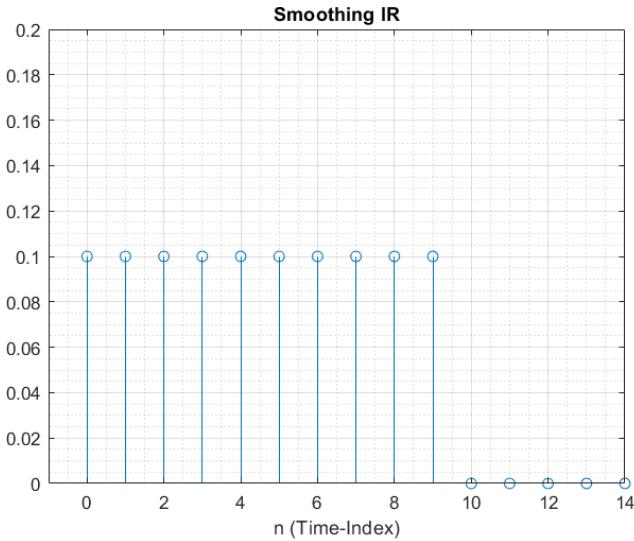
The smoothing filter is implemented via a 10-point moving average where every sample is weighted evenly. Its difference equation is:

$$y[n] = \frac{1}{10} \sum_{m=0}^9 x[n - m] \tag{4.4}$$



**Fig. 4.2** The measured frequency response for the DC Blocker test.

This is verified by the output shown in Fig. 4.3. Extra elements are shown to indicate the filter outputs zeros after the 10th iteration (corresponding to  $n = 9$ ). Given that it is easier to understand this filter from its impulse response, and this is paired to the frequency response via the Fourier Transform, verification for this was done in the time-domain.

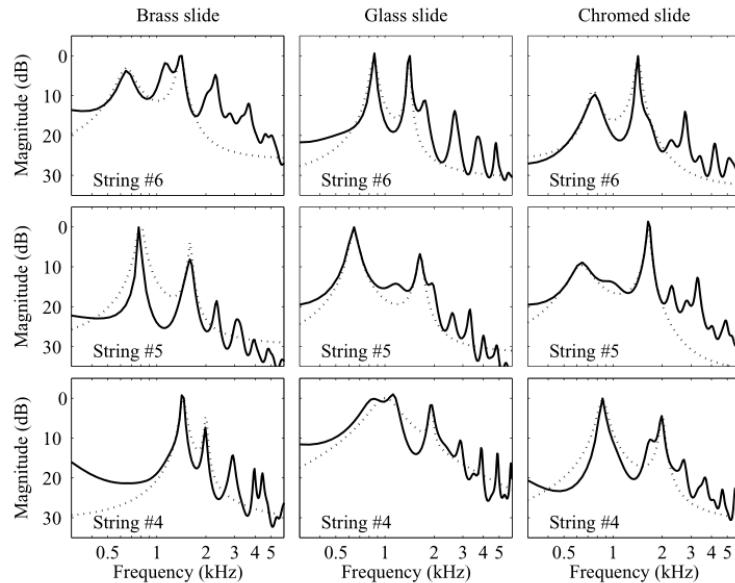


**Fig. 4.3** The impulse response of the implemented smoothing filter.

#### 4.1.4 Longitudinal Modes

The precise method by which the longitudinal mode filters were derived and designed is not completely specified in either [Puputti 2010] or [Pakarinen, Puputti, and Välimäki 2008]. What is explained is that a linear-prediction filter of order 100 was used to estimate the spectrum of the different modes of each slide/wound string interaction. From this, a 4th-order IIR filter was derived based on the most prominent resonances. The pole/zero locations for these 4th-order filters are what is provided for an implementation. Only plots of the magnitude responses for the different filters as well as the linear-prediction estimates are provided. These plots are shown in Fig. 4.4. These 4th-order approximations were recreated for the implemented longitudinal mode filters as shown in Fig. 4.5. Verification was done through visual comparison of the plots as there is no data to easily facilitate computing an error function.

In summary, all the filters work according to their expected behavior.

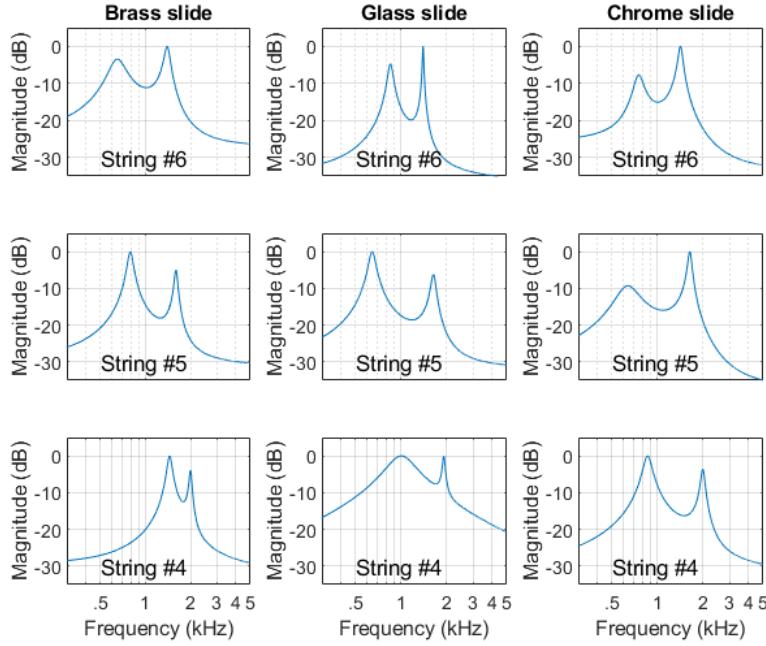


**Fig. 4.4** Original figure from [Pakarinen, Puputti, and Välimäki 2008] for comparison purposes. Solid lines represent spectral estimates using linear-prediction filter of order 100. Dotted lines indicate modal filter magnitude responses.

## 4.2 Control Signal Processor

The Control Signal Processor (see Sec. 3.1.3) consists of three components:

1. **Slide Speed Extractor** - This block performs numerical differentiation on the relative length signal followed by an absolute-value to generate the slide speed control signal.
2. **Interpolator** - This block performs linear interpolation to upsample control rate signals to the audio rate.



**Fig. 4.5** The implemented longitudinal mode filters' magnitude responses. These correspond to the dotted lines of Fig. 4.4.

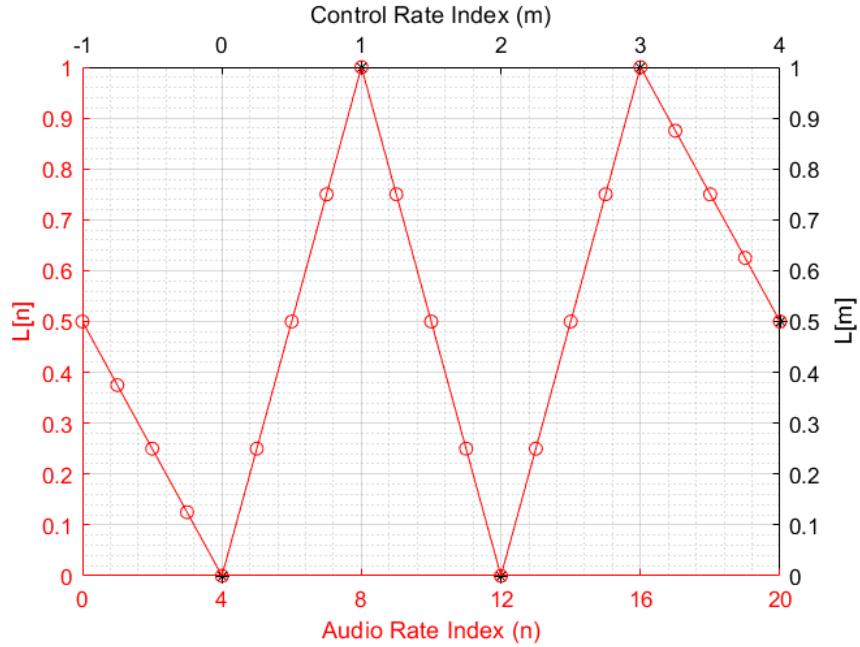
3. **Smoothing Filter** - This block smooths the interpolated signals to mitigate any discontinuities introduced during processing.

The Smoothing Filter was verified earlier in Sec. 4.1.3. What remains to be shown is that the Interpolator and Slide Speed Extractor operate correctly as well as the Control Signal Processor as a whole, given that it is created by the interconnection of these components.

#### 4.2.1 Components of Control Signal Processor

##### Interpolator

The interpolator block operates via linear interpolation. It was tested by specifying a control signal  $L[m]$  and running it through the interpolator. Figure 4.6 illustrates this. The original  $L[m]$  is plotted with black \*. The interpolated  $L[n]$  output is plotted as red circles with linear line segments to help ensure linear trajectories were maintained. Red and black are used to help disambiguate the input and output. For this figure, the audio rate was specified as 48,000 Hz while the control rate was specified as 12,000 Hz. This would give a ratio of  $R = \frac{48000}{12000} = 4$ , meaning 3 interpolated audio samples would need to be calculated for every 1 control sample.  $R$  can only take on integer values. It is also necessary to specify an initial value for the interpolation to start from. In this figure, 0.5 was used for the initial value. As is shown, the linear interpolation is performed correctly.



**Fig. 4.6** The output from the Interpolator test. The black axes and \* represent the original control rate signal. The red axes and circles represent the interpolated output. Line segments have been added to help illustrate the linearity. The initial value used to start the interpolation is 0.5 and R = 4. The different colors are to help disambiguate input from output.

### Slide Speed Extractor

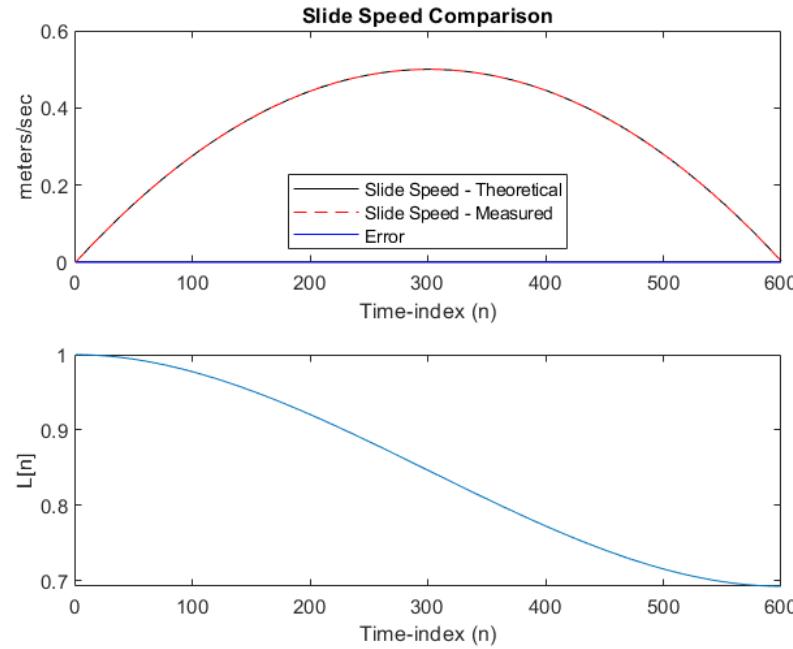
The Slide Speed Extractor was tested by taking a theoretical curve which represents a parabolic *slideSpeed[n]* trajectory. From this, the corresponding relative length signal for a specified string length was generated using the equation:

$$L[n] = L[n - 1] - \frac{slideSpeed[n]}{F_s \times StringLength} \quad (4.5)$$

This curve was then fed into the Slide Speed Extractor object to generate the corresponding *slideSpeed[n]*. The error between the theoretical and measured values was calculated. The output of this is shown in Fig. 4.7. Given that the error signal is a constant 0, this illustrates correct operation of the slide signal extractor.

#### 4.2.2 Control Signal Processor as a Whole

With the constituent components verified as working, the test here confirms that everything is linked together correctly inside the Control Signal Processor. Figure 4.8 illustrates the output of the Control Signal Processor to a specified *L[m]* control signal. This was chosen to be linear



**Fig. 4.7** The top plot illustrates the correctness of the slide speed extraction algorithm while the lower plot displays the  $L[n]$  signal which was used as a stimuli.

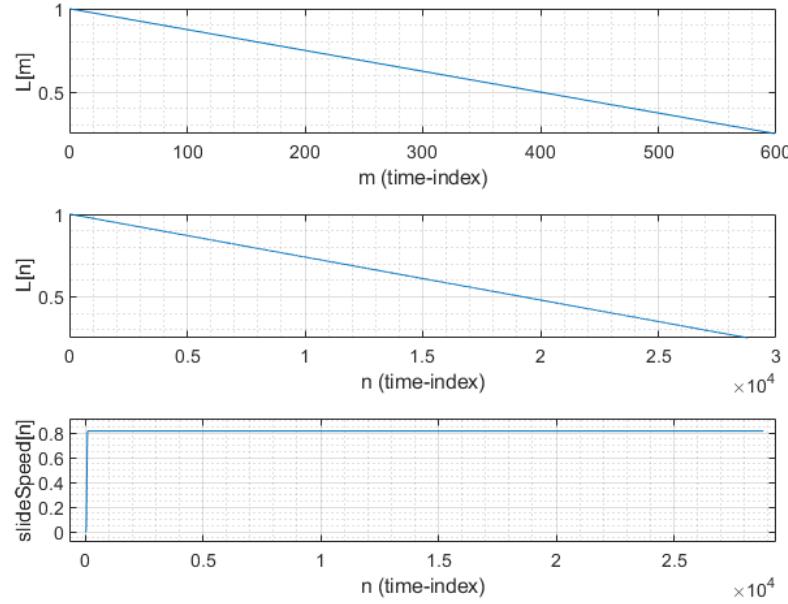
to make the interpolation and smoothing easier to verify. The  $slideSpeed[n]$  signal can be seen as indicating the slide starts from rest and then gradually ramps up to constant speed. This is consistent with what would be expected as we have specified that the starting  $L[m]$  value in the Control Signal Processor corresponds to when  $m = 0$ .

### 4.3 Noise Generation Objects

#### 4.3.1 Impulse Train

The Impulse Train object responds to the  $f_c[n]$  signal which controls its firing rate. Accordingly, various artificial  $f_c[n]$  signals were generated to ensure the different run-time use cases would execute correctly during synthesis. These test are:

1.  $f_c[n]$  is a constant during operation and impulses are generated at regular intervals.
2.  $f_c[n]$  decreases during operation and the number of samples between impulses increases.
3.  $f_c[n]$  increases during operation and the number of samples between impulses decreases.
4.  $f_c[n]$  is set to zero during operation and no more impulses are produced .
5.  $f_c[n]$  changes from 0 to a positive value and impulses begin being produced.

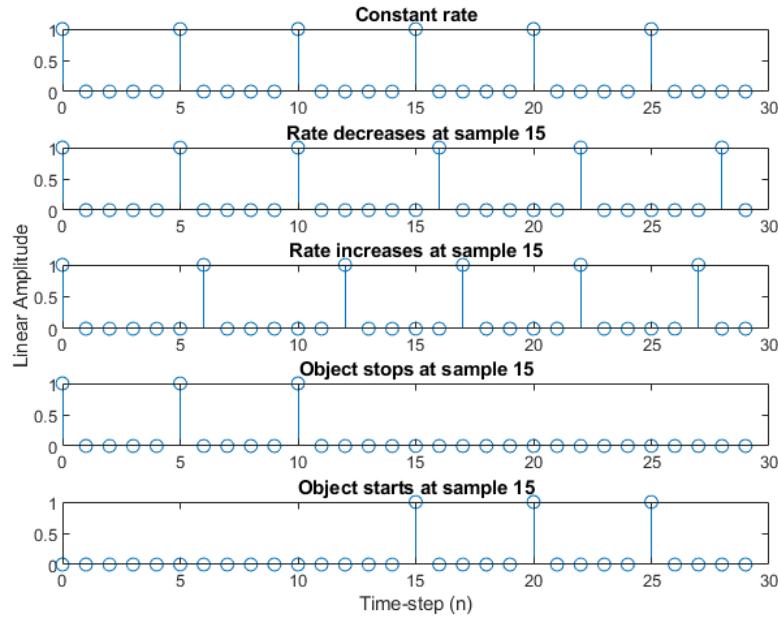


**Fig. 4.8** The  $L[m]$  input signal and corresponding output signals for CSP test.

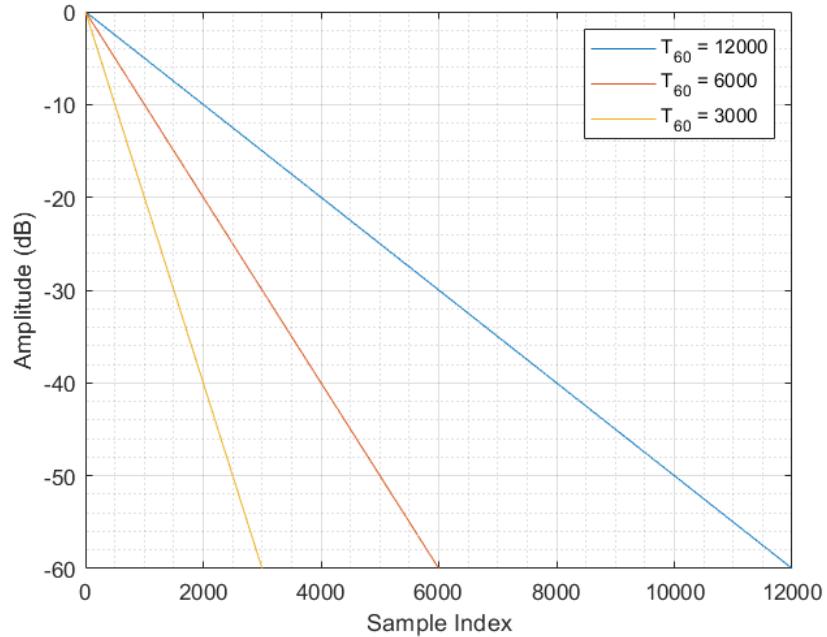
The full details can be found in the link in Appendix B. The different test cases are illustrated in the same order in Fig. 4.9. As can be seen in the figure, the Impulse Train object operates as expected.

#### 4.3.2 Exponential Decay

The Exponential Decay object is composed of an Impulse Train which excites a one-pole filter whose feedback coefficient is tuned to match the specified  $T_{60}$  parameter. As the Impulse Train was tested separately, it was necessary to ensure that the  $T_{60}$  parameter was implemented correctly. Figure 4.10 illustrates correct functioning of the Exponential Decay object for three different  $T_{60}$  values. The plots illustrate the envelopes generated by the Exponential Decay. The  $T_{60}$  were initially specified as  $\frac{1}{4}$  seconds,  $\frac{1}{8}$  seconds and  $\frac{1}{16}$  seconds. The dB scale is used for the y-axis to make it easier to identify the point where the envelope has decayed by 60 dB. The  $T_{60}$  values have been converted into samples by multiplication with  $F_s = 48,000$  kHz, to allow for easier verification.



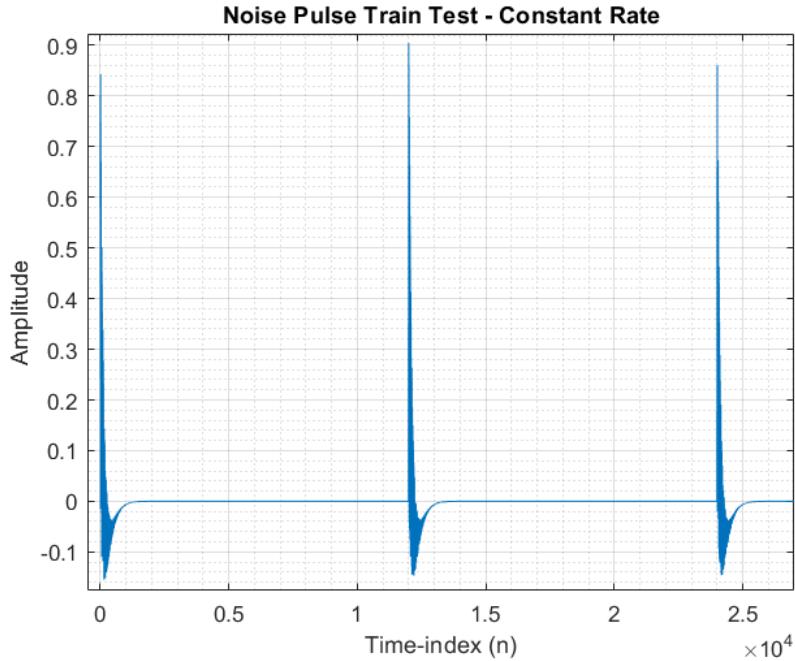
**Fig. 4.9** The Impulse Train test output.



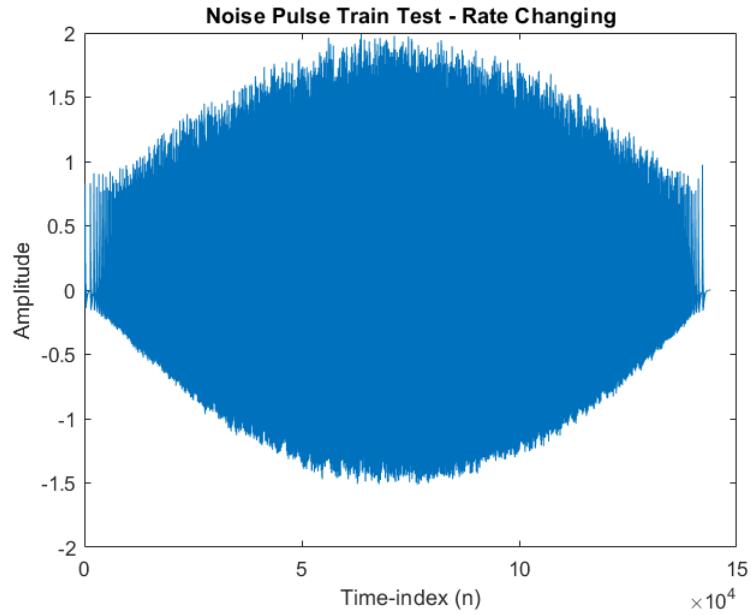
**Fig. 4.10** The test results for three different  $T_{60}$  decay parameters specified in samples.

### 4.3.3 Noise Pulse Train

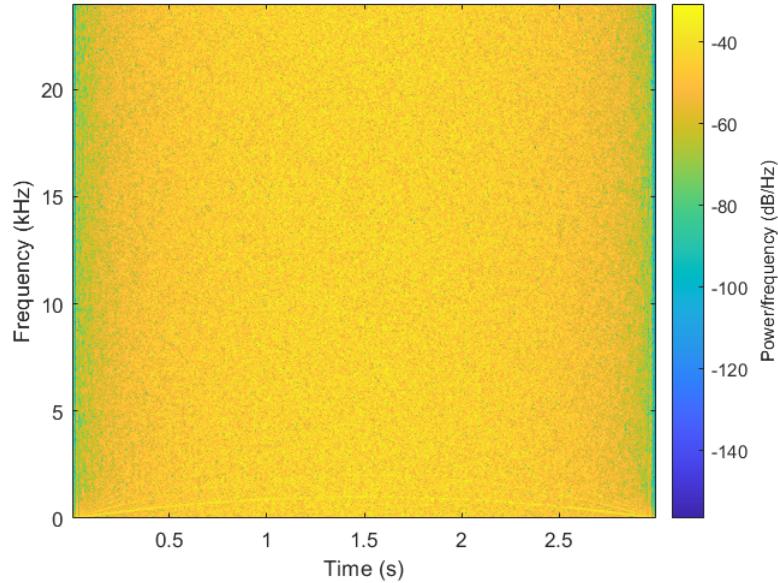
The Noise Pulse Train object was tested with two different scenarios: a constant firing rate producing 12 distinct pulses with no overlap and a swept firing rate corresponding to the same parabolic trajectory used to verify the Slide Speed Extractor in Sec. 4.2.1. The results are illustrated in Figs. 4.11-4.13. Note the overlap of the individual pulses in the second test due to the firing rate being faster than the decay rate. This overlap causes the signal to become more noise-like. The spectrum also illustrates how the signal has a harmonic component in the lower end. The output of each test can be heard in the files *NoisePulseTrain-test1.wav* and *NoisePulseTrain-test2.wav*. Through these examples, as well as their spectrograms, the Noise Pulse Train object has been verified as working correctly both subjectively and objectively. The “noisiness” of a signal at higher values of  $f_c$  can be controlled by the  $T_{60}$  parameter as will be discussed in Sec. 6.1.1.



**Fig. 4.11** The Noise Pulse Train output for a constant rate. Note the negative values introduced by the DC Blocker. These have no perceptual impact, but is a deviation from the original design described in [Pakarinen, Puputti, and Välimäki 2008].



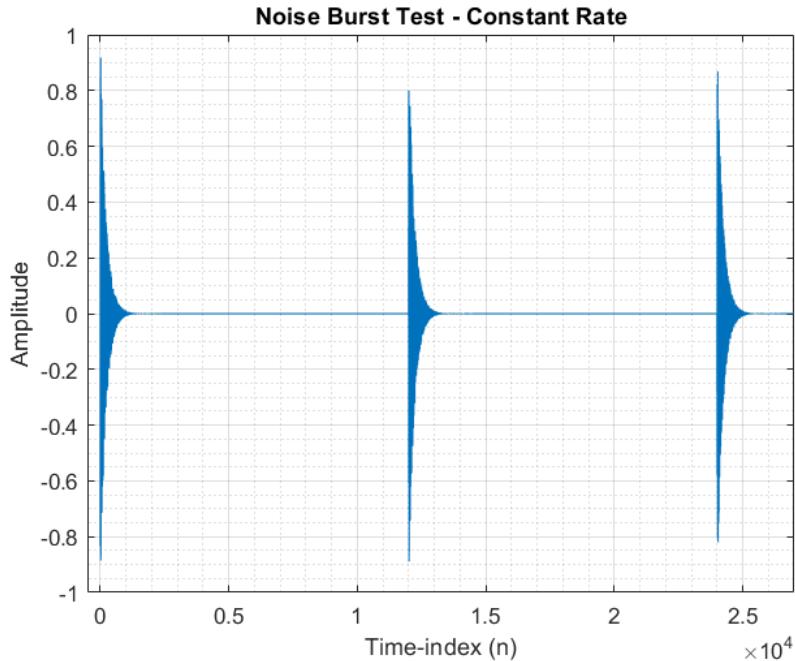
**Fig. 4.12** The Noise Pulse Train output in response to an  $f_c[n]$  sweep. Note the overlapping build-up of the individual impulses. Individual pulses can be seen at the ends.



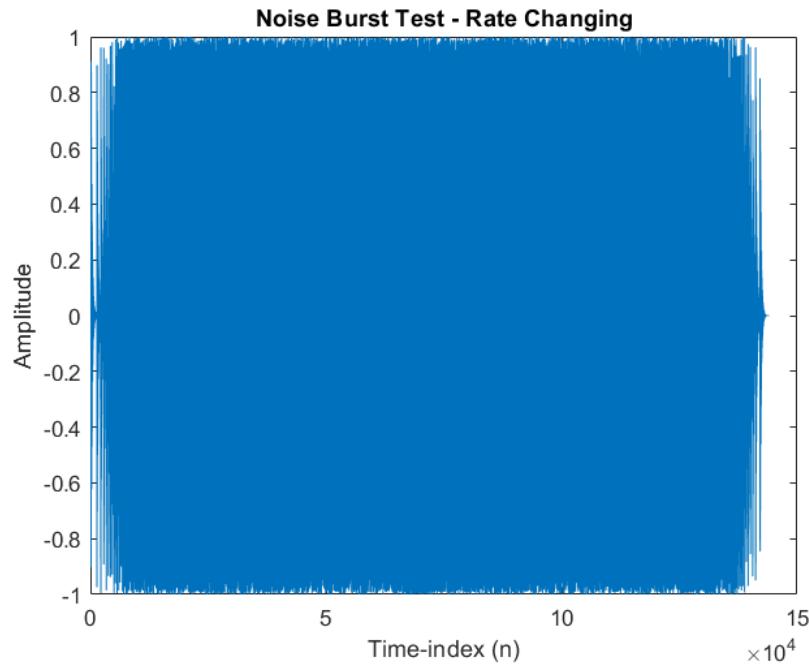
**Fig. 4.13** This spectrum corresponds to Fig. 4.12. Note the emergence of harmonics at the lower end as compared to the results shown in Fig. 4.15.

#### 4.3.4 Noise Burst Generator

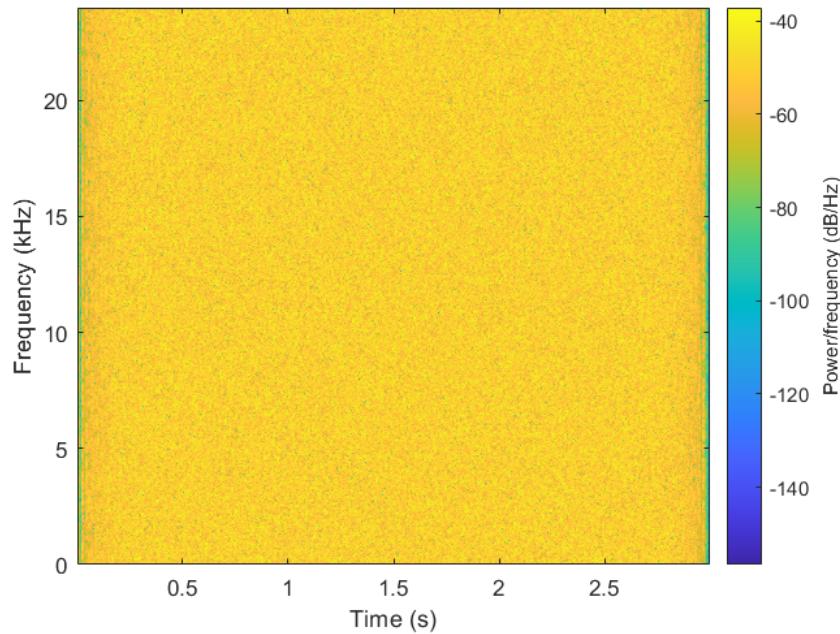
The Noise Burst generator was tested with the same two input signals as the Noise Pulse Train. The outputs are shown in Figs. 4.14-4.16. The sounds can be heard in the files *NoiseBurstGen-test1.wav* and *NoiseBurstGen-test2.wav*. The noisiness in the second example was by design as explained in Sec. 3.1.5. Each of the different noise sources were originally designed to work with only one of the Harmonic Accentuation techniques described in Sec. 3.1.5. The Noise Burst Generator operates best with the Harmonic Resonator Bank as will be elaborated upon in Sec. 6.1.2.



**Fig. 4.14** The Noise Burst Generator output for a constant rate. Note the approximate symmetry about the x-axis. The symmetry eliminates the need for a DC blocker.



**Fig. 4.15** The Noise Burst Generator output in response to an  $f_c[n]$  sweep. Note the effects of the hard-clipping and how the signal transitions into noise as was intended.

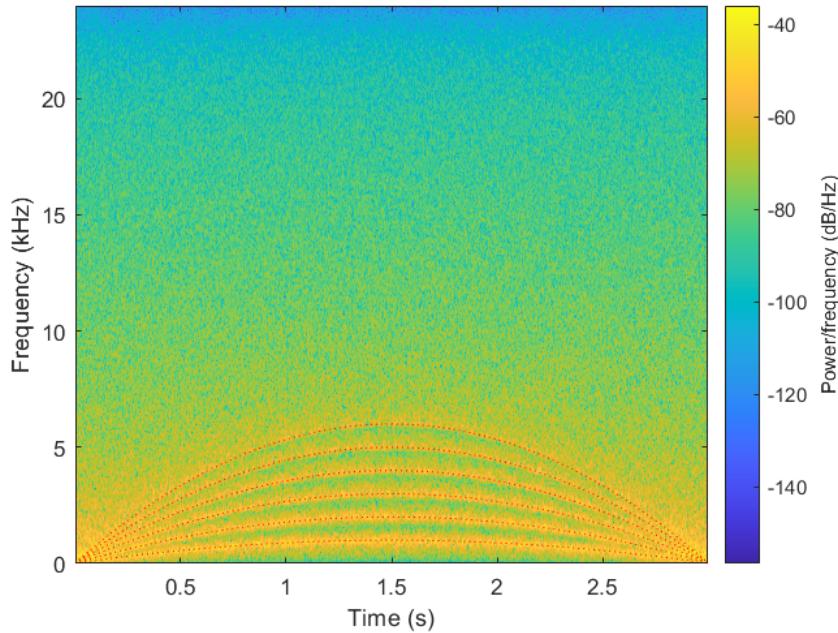


**Fig. 4.16** The spectrum corresponding to Fig. 4.15. Note the lack of harmonics as compared to Fig.4.12.

## 4.4 Harmonic Accentuators

### 4.4.1 Harmonic Resonator Bank

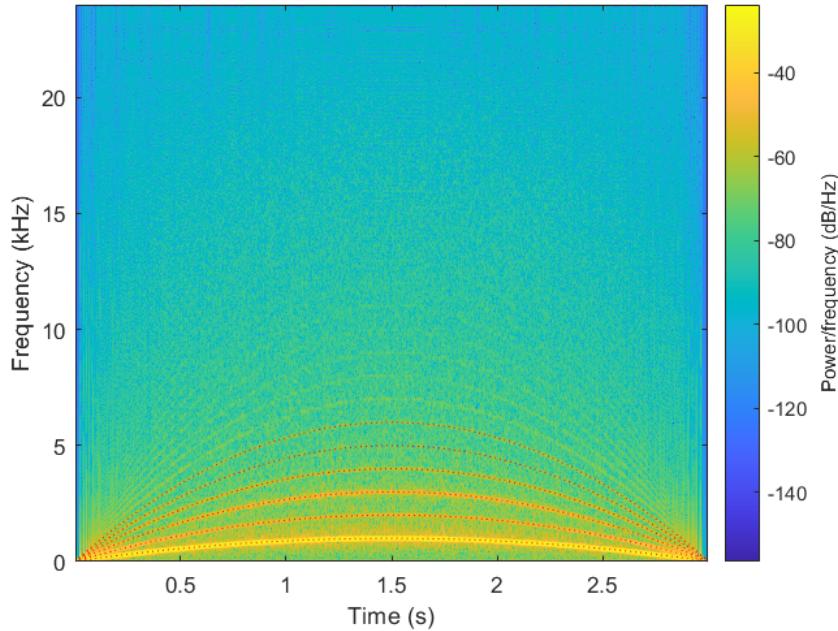
Given that the resonators were verified in Sec. 4.1.1, the Harmonic Resonator Bank was verified by running white noise through the system while performing a sweep on the  $f_c$  control parameter. The sweep follows the same parabolic trajectory from Sec. 4.2.1. The functionality of this block predicts that we would observe six harmonically linked bands in the output spectrum. This is shown in Fig. 4.17 where the different harmonics' trajectories are overlaid in red. The output can be heard in *HRB-test.wav*. Both the recording and spectrogram provide objective and subjective evidence that the Harmonic Resonator Bank works as is expected and is free from unwanted artifacts.



**Fig. 4.17** The output from Harmonic Resonator Bank test. The dotted red lines indicate the theoretical harmonic trajectories.

### 4.4.2 Reso + Tanh

The Reso + Tanh block (Sec. 3.1.5) was tested using the same  $f_c$  sweep. The stimuli was changed to use the Noise Pulse Train object instead as the  $\tanh()$  function relies on a periodic signal being input in order to achieve the desired effect of accentuating/creating harmonics. Figure 4.18 illustrates the output spectrum from the test. Note the much finer concentration of energy in the harmonic bands, the emphasis of the fundamental due to the different noise source and more than 6 harmonics being generated. This helps provide a finer sense of perceptual fusion for the generated sound. The output of this can be heard in *ResoTanh-test.wav*.



**Fig. 4.18** The output from Reso + Tanh test. The dotted red lines indicate the theoretical harmonic trajectories.

## 4.5 Contact Sound Generator (CSG)

Both the wound and unwound CSGs were verified using the following three test cases based on the qualities of the slide motion:

1. No slide motion
2. Slide motion with a constant slide velocity
3. Slide motion with a time-varying slide velocity

These were selected as they covered the basic use cases which would arise during synthesis. The constant slide velocity test uses a slide velocity which generates an  $f_c[n]$  of 250 Hz. The time-varying slide velocity is configured to generate a parabolic sweep of  $f_c[n]$  from 0 Hz to 1kHz and back to 0 Hz (as with the previous similar tests). This mimics the speed experienced by a slide which starts at rest and moves between two positions on the fingerboard. Test scenario 1 (no slide motion) is mentioned for completeness purposes. Results and figures from that test will not be discussed due to its simplicity.

### 4.5.1 Wound Variant

The audio producing tests were further sub-divided into three other tests by controlling the balance between the two sound components. This was done to ensure proper functioning of each

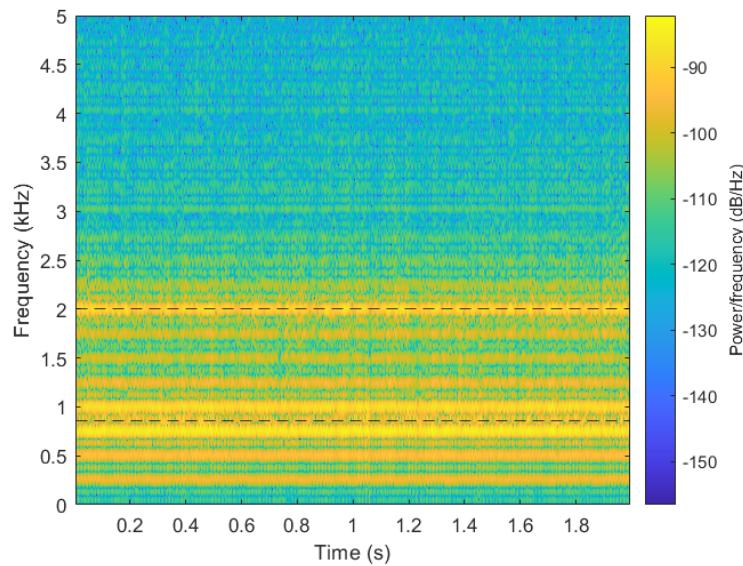
branch as well as help determine each branch's audible contribution in the combined sound. As part of the sound design process, the four different combinations of noise sources and harmonic accentuation techniques were tested (as will be elaborated upon in the Sound Design chapter). The following figures were generated using the Noise Pulse Train and Harmonic Resonator Branch configuration in order to reduce the total number of figures.

1. Longitudinal branch isolated
2. Harmonic branch isolated
3. Both branches combined

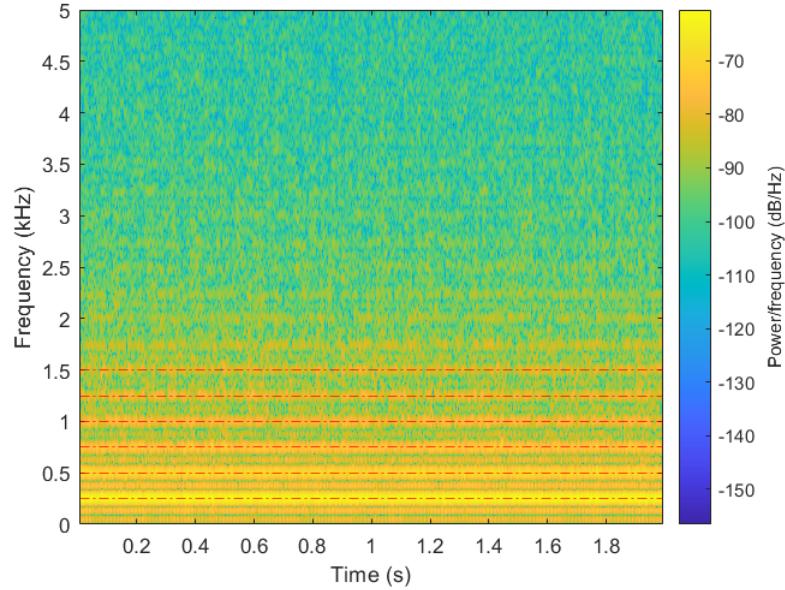
In the spectrograms showing the results, the dashed black lines represent the specified longitudinal mode frequencies and the dotted red lines indicate the theoretical harmonic trajectories.

#### Static $f_c[n]$

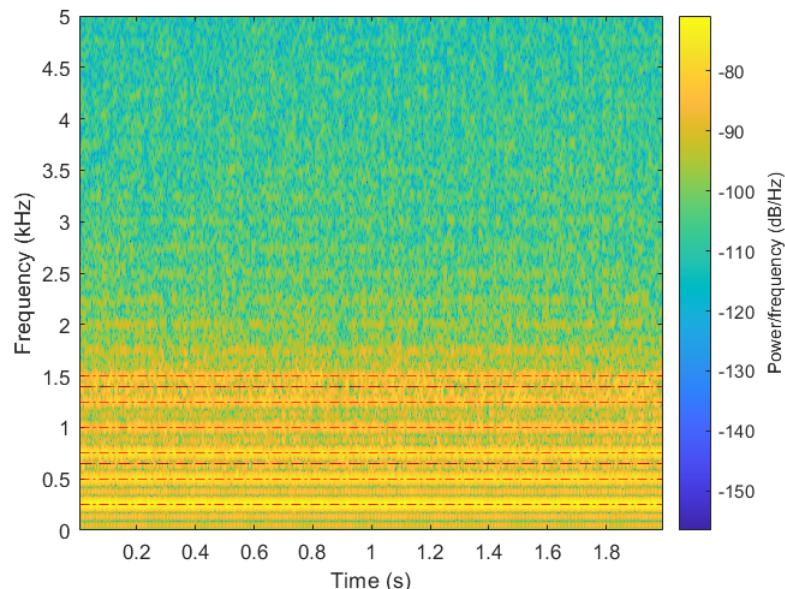
The results for the constant slide velocity scenario are shown in Figures 4.19-4.21. Figure 4.19 illustrates how the original source stimuli produced by the Noise Pulse Generator does not contain strong frequency components at the 1st longitudinal mode frequency. Also illustrated is that the harmonic branch extracts and emphasizes the fundamental while retaining many of the upper harmonics in decreasing strength. The corresponding audio can be heard in *CSG-Wound-Static-Long.wav*, *CSG-Wound-Static-Harm.wav* and *CSG-Wound-Static-Both.wav*.



**Fig. 4.19** The output from the Wound CSG test using a static slide velocity for the longitudinal branch ( $v_1[n]$  in Fig. 3.7). The dashed black lines indicate longitudinal mode frequencies. As the stimuli does not contain frequencies near the first mode, there is less energy there.



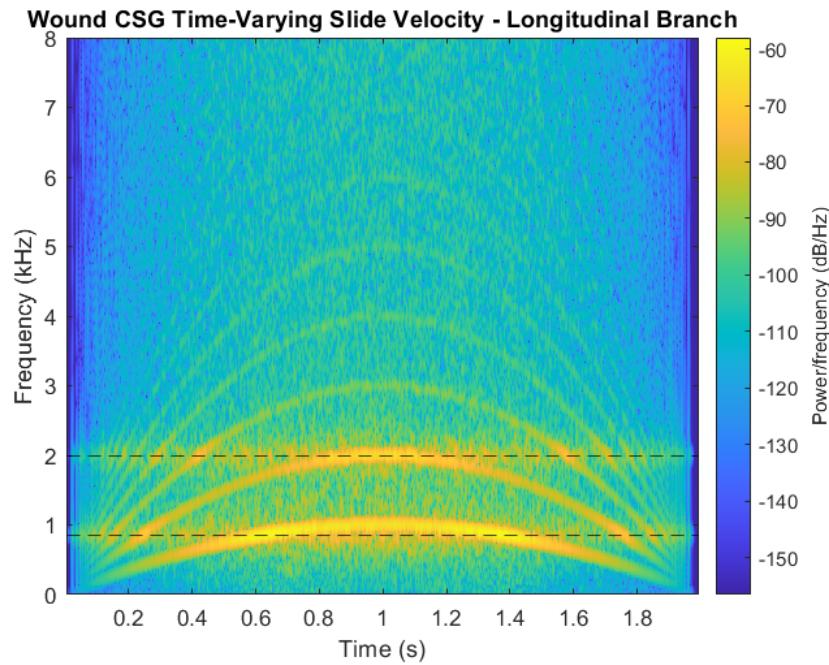
**Fig. 4.20** The output from the Wound CSG test using a static slide velocity for the harmonic branch ( $v_2[n]$  in Fig. 3.7). The dotted-dashed red lines indicate the frequencies for the first six harmonics.



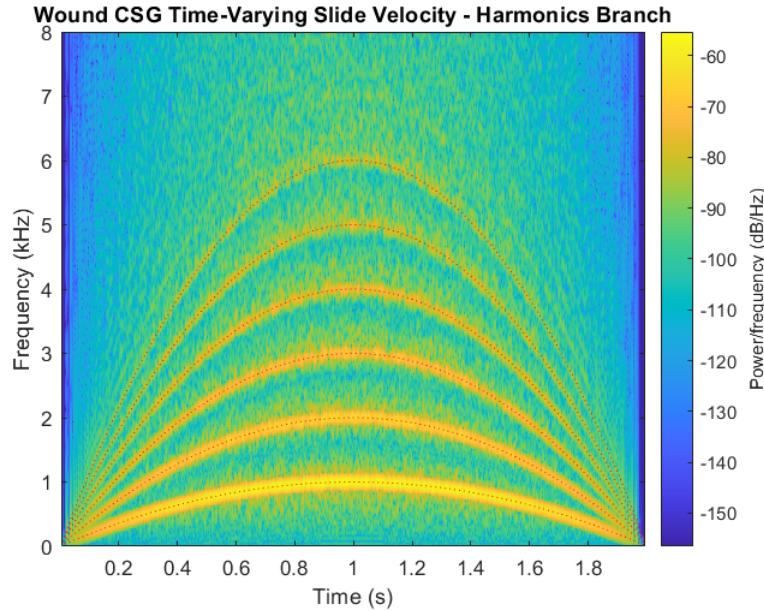
**Fig. 4.21** The output from the Wound CSG test using a static slide velocity for both combined branches. Notice how this is a combination of Fig. 4.19 and Fig. 4.20.

### Time-Varying $f_c[n]$

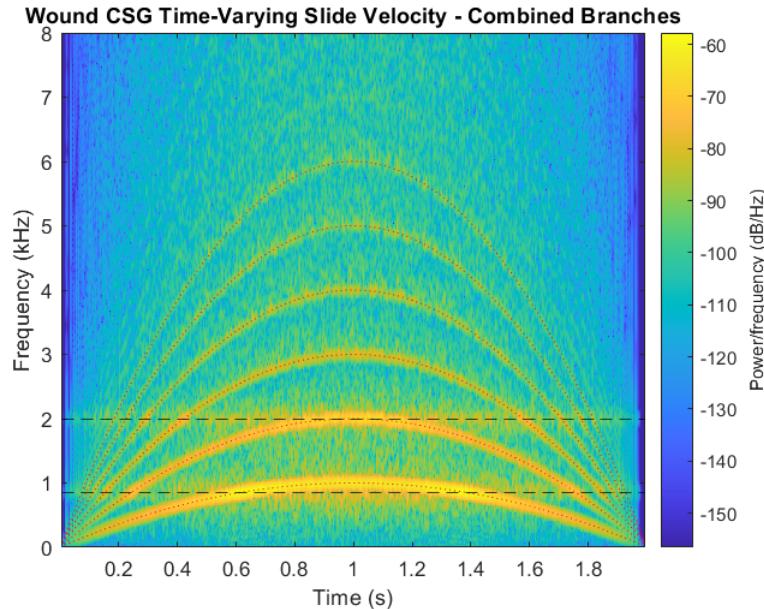
The results for the time-varying slide velocity scenario are shown in Fig. 4.22-4.24. Dashed red lines indicate the theoretical harmonic trajectories. Dashed black lines correspond to the static longitudinal modes. As is clearly shown in Fig. 4.22, the longitudinal modes are only stimulated when the frequencies the filters are tuned to are present in the incoming signal. It also illustrates how some of the harmonic components leak through, which will be reinforced when added to the output from the harmonic branch. Only  $\frac{1}{3}$  of the spectrum is plotted to emphasize these points. The corresponding audio can be heard in *CSG-Wound-TV-Long.wav*, *CSG-Wound-TV-Harm.wav* and *CSG-Wound-TV-Both.wav*.



**Fig. 4.22** The output from the Wound CSG test using a time-varying slide velocity for the longitudinal branch. The dashed black lines indicate longitudinal mode frequencies. When the fundamental of the sweep matches a modal frequency it is reinforced more.



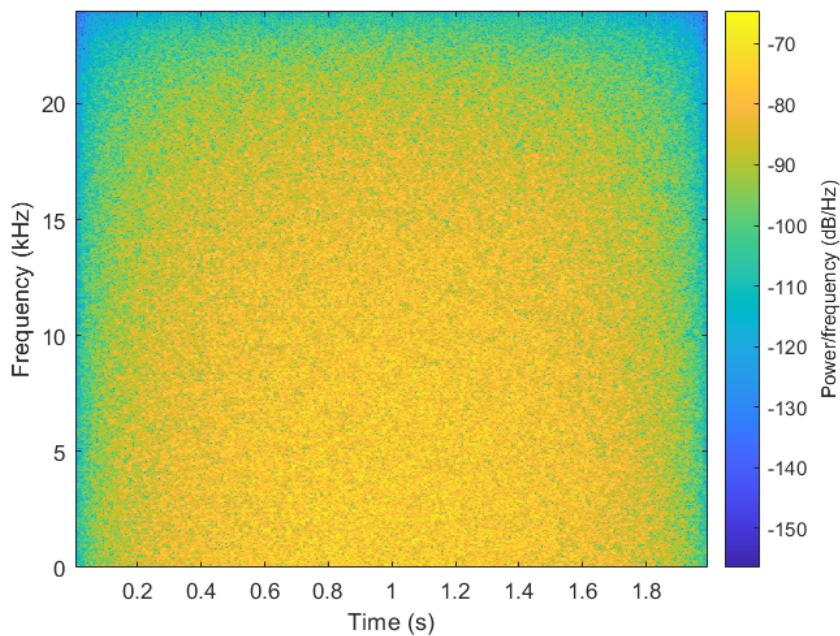
**Fig. 4.23** The output from the Wound CSG test using a time-varying slide velocity for the harmonic branch. The dotted red lines indicate the frequencies for the first six harmonics.



**Fig. 4.24** The output from the Wound CSG test using a static slide velocity for both combined branches. Notice how this is a combination of Fig. 4.22 and Fig. 4.23.

#### 4.5.2 Unwound Variant

Figure 4.25 illustrates the output of the unwound Contact Sound Generator for the same time-varying slide speed signal. It is substantially less interesting but still necessary for the purposes of verification. As can be seen, the low-pass filter applied creates a roll-off at the top of the spectrum while the ramp-in and ramp-down create the variations in spectral energy near the beginning and end of the sound. This can be heard in the file *CSG-Unwound-TV.wav*. The rest of the scenarios were also run on this module but are not shown here.



**Fig. 4.25** The output from unwound branch for time-varying slide velocity.

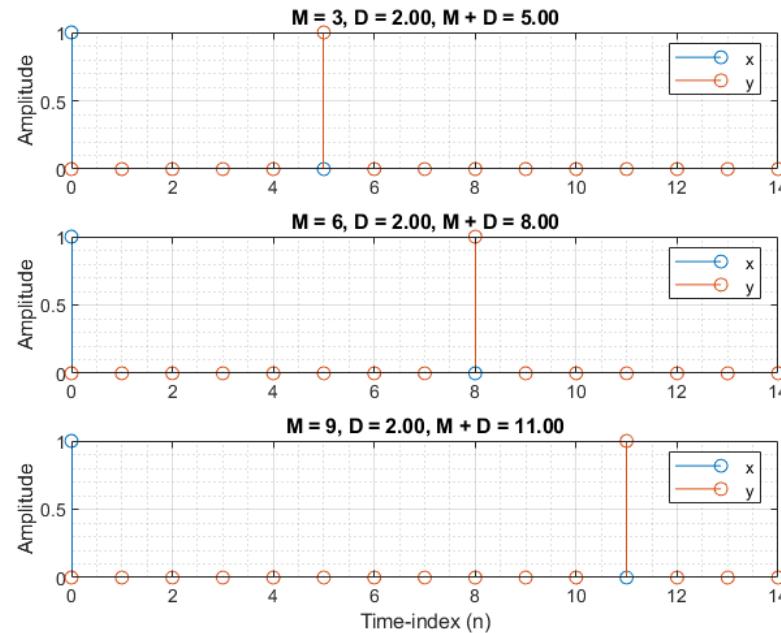
### 4.6 String DWG and its Components

#### 4.6.1 Interpolated Delay Line

The core-class which this object was built around is a circular buffer. The circular buffer class was tested thoroughly itself (see source code). Illustrated here is the verification of the Lagrange interpolation as this is crucial to ensuring the correct functioning of the synthesis algorithm. In this test,  $D$  represents the nominal delay implemented by the Lagrange interpolator. It can also be expressed as  $D = \lfloor D \rfloor + d$ , where  $d$  is the fractional component. The order of the Lagrange interpolator is set to 5. At this order,  $D$  could take values on the interval  $[0, 5)$  samples. However, Lagrange interpolation works best near the mid-point so  $D$  is constrained to  $[2, 3)$ .  $M$  represents delay in samples as implemented by the integer delay line, which precedes the Lagrange interpolator in the overall interpolated delay line structure. See 2.3 for reference. Given that the

order of the Lagrange interpolator creates a lower bound on the length of the interpolated delay line, limits are set based on a specified highest-fret (which results in a minimum  $L[n]$ ) to ensure this lower bound is avoided.

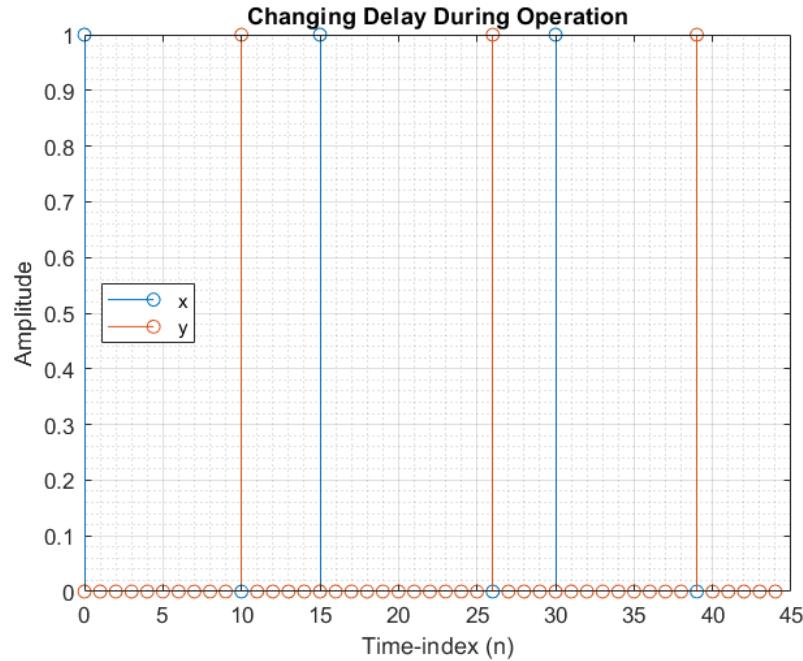
The first test that was performed was a test to ensure that it could indeed operate as an integer delay. Various Interpolated Delay Line objects were constructed, ensuring that no fractional delay would be required. The output of this test is shown in Fig. 4.26 where  $x$  is the input signal and  $y$  is the output. As this figure clearly shows, the interpolated delay line clearly implements this functionality.



**Fig. 4.26** The test results of integer values for the interpolated delay line object.  $x[n]$  is the input impulse stimuli and  $y[n]$  is the resulting delayed output. The delay values are indicated above each plot.

The second test which was performed ensures that the delay line value can be updated during the run-time. This differs from the previous test where a new object was constructed each time. Figure 4.27 illustrates this test. The delay line starts out with an initial delay of 10 samples. After 15 samples have passed this is incremented by 1. After another 15 samples have passed this is decremented by two. Impulses are fed into the delay line every 15 samples to show the changes.

The third test which was performed is similar to the second, however now the fractional component was changed to ensure the coefficients of the Lagrange FIR are calculated correctly. Table 4.1 illustrates the parameter changes which occur every 15 samples. Under these conditions, the length of the interpolation filter is 6 samples, while  $M$  is 8 samples. The output of this test is shown in Fig. 4.28. As is clearly shown, the impulse response of the Lagrange interpolator, corresponding to the different fractional delays, appears in the six samples after the eight zeros (which correspond to the integer delay component of the structure).



**Fig. 4.27**  $x[n]$  is an impulse train with a period of 15 samples and  $y[n]$  represents the delay line output. The delay value starts at 10, increases to 11 at  $n = 15$  and decreases to 9 at  $n = 30$ . This illustrates that the interpolated delay line can change delay values during operation.

<b>n</b>	<b>M</b>	<b>d</b>	<b>D</b>
0	8	.25	10.25
15	9	.50	11.5
30	8	.75	10.75

**Table 4.1** The parameter changes for the test shown in Fig.4.28.

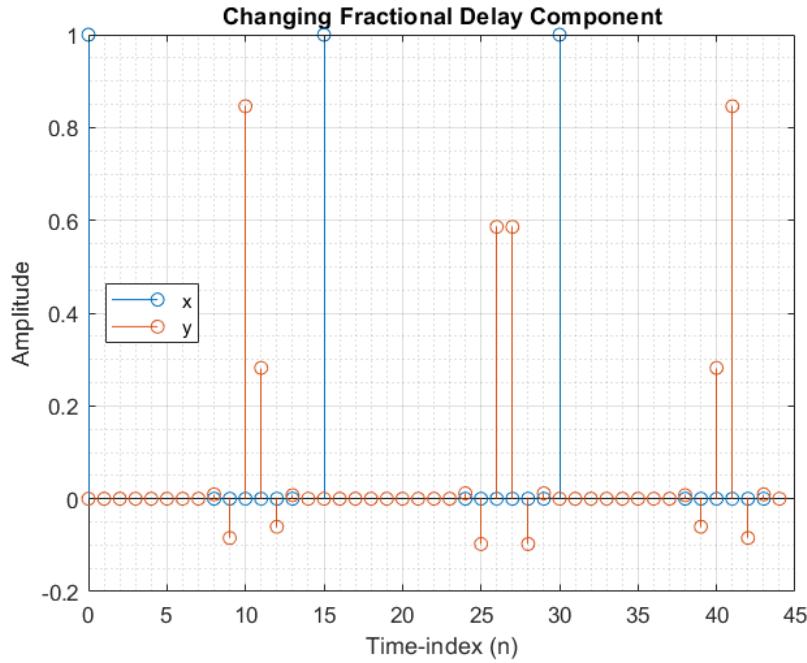
#### 4.6.2 Energy Scaler

The energy scaling coefficient (Sec. 3.1.4) is governed by the following equation:

$$g_c[n] = \sqrt{1 - \Delta x[n]} \quad (4.6)$$

where  $\Delta x[n]$  is the change in the length of the digital waveguide, in samples, at time-step  $n$ . Testing of this block was done by specifying two different curves representing the digital waveguide length in samples over time. As  $x[n]$  could take an infinite variety of different sorts of functions, it was decided to limit the curves to be linear and quadratic. The theoretical output of what the energy scaler should produce was derived and then the output error was calculated by subtracting the measured results from the derived results.

Assuming we start with a continuous signal, a quadratic DWG length signal can be expressed



**Fig. 4.28**  $x[n]$  is an impulse train with a period of 15 samples used as input.  $y[n]$  is the corresponding output. The parameter changes are specified in Table 4.1. The output illustrates the appropriate number of integer delay values followed by the impulse response of a filter approximating the Lagrange interpolation.

as:

$$DWGLength(t) = at^2 + c \quad (4.7)$$

Given  $c = DWGLength(0)$ , the starting point of the sweep, then  $a$  can be expressed as:

$$a = \frac{sweepEnd - sweepStart}{sweepDuration^2} \quad (4.8)$$

The continuous signal can then be sampled to produce the following discrete-time expressions:

$$DWGLength[n] = DWGLength(nT_s) = a(nT_s)^2 + c \quad (4.9)$$

$$DWGLength[n-1] = a(n^2 - 2n + 1)T_s^2 + c \quad (4.10)$$

Given that  $\Delta x = DWGLength[n] - DWGLength[n-1]$ , we can express  $g_c[n]$  as:

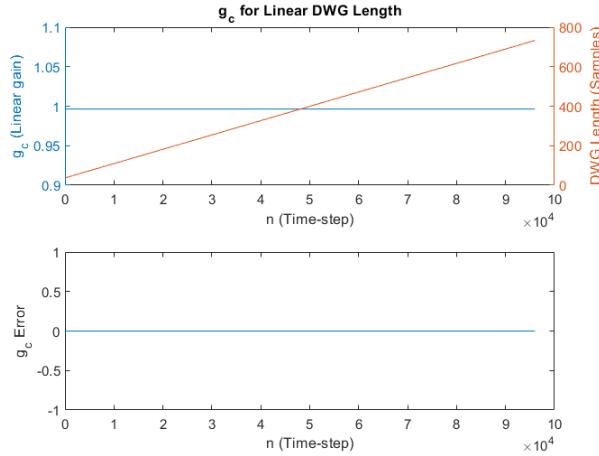
$$g_c[n] = \sqrt{|1 - a(2n - 1)T_s^2|} \quad (4.11)$$

and use the expressions for  $a$  and  $c$  to develop a parameterized theoretical curve for the ideal output to a quadratic input. A similar procedure can be followed for the simpler linear case.

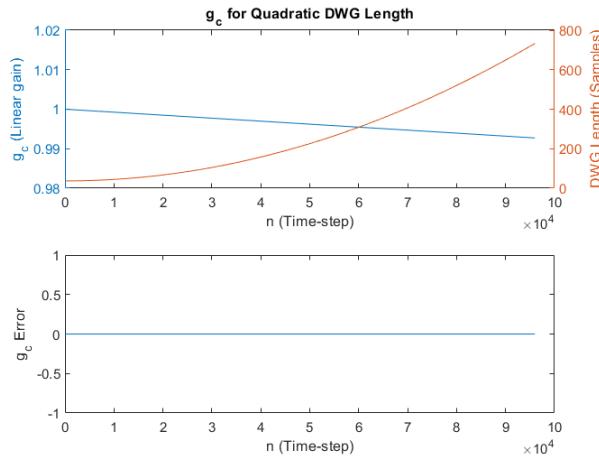
Figures 4.29 and 4.30 show the results for the linear and quadratic DWG Length functions

respectively. The output to a linearly increasing function is a constant gain factor less than one. Based on Eq. 4.6, this is expected.  $\Delta x$  for a linearly increasing DWG Length is a positive constant, making  $g_c[n]$  on the interval  $[0, 1)$ .

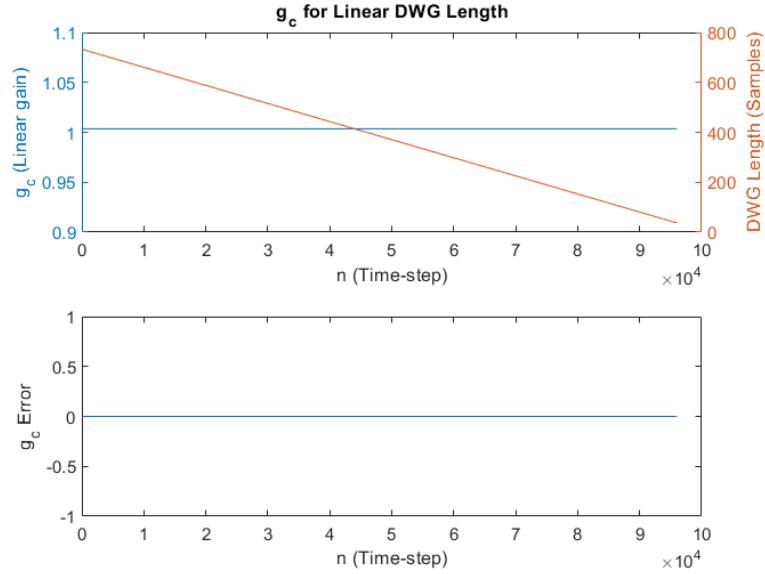
Figures 4.31 and 4.32 illustrate the output to the same functions which are now decreasing. As is illustrated, the absolute error remains zero and the gain operates in the same pattern but applying amplification.



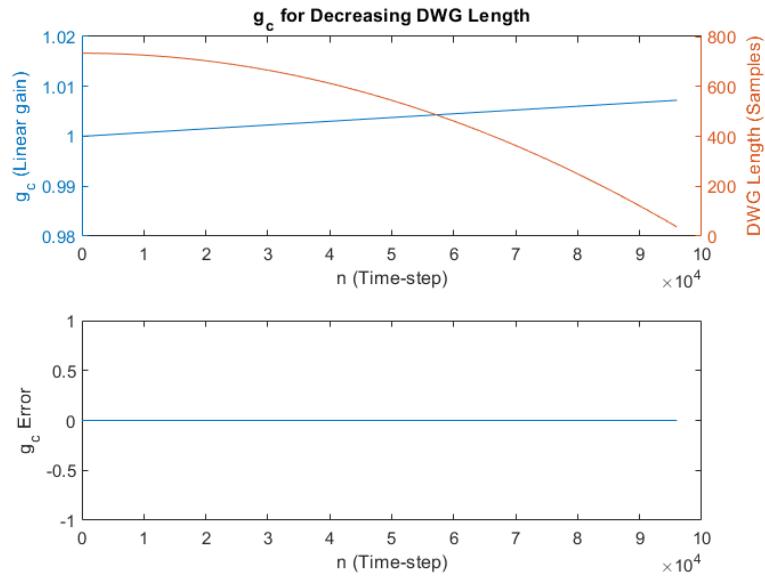
**Fig. 4.29** The Energy Scaler output in response to a linearly increasing input. The top plot illustrates the input/output signals as they evolve over time. The bottom figure illustrates correctness as the absolute error is 0.



**Fig. 4.30** The Energy Scaler output in response to a quadratically increasing input. The top plot illustrates the input/output signals as they evolve over time. The bottom figure illustrates correctness as the absolute error is 0.



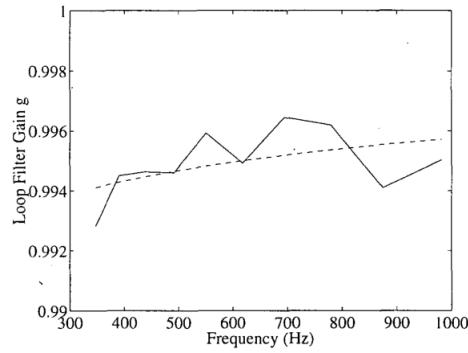
**Fig. 4.31** The Energy Scaler output in response to a linearly decreasing input. The top plot illustrates the input/output signals as they evolve over time. The bottom figure illustrates correctness as the calculated error is 0.



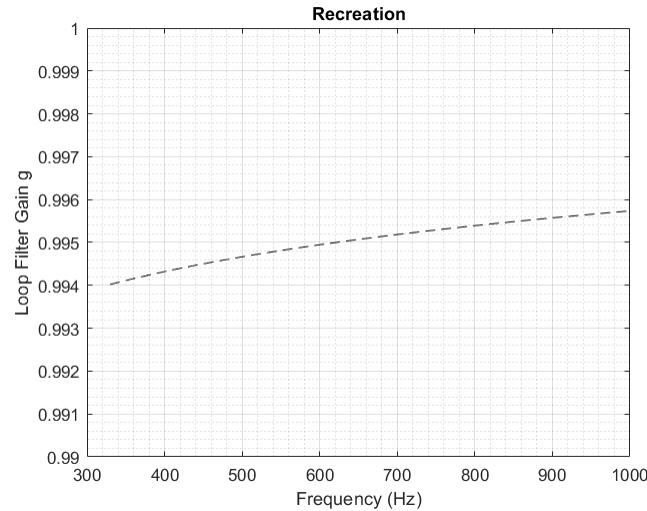
**Fig. 4.32** The Energy Scaler output in response to a quadratically decreasing output. The top plot illustrates the input/output signals as they evolve over time. The bottom figure illustrates correctness as the calculated error is 0.

### 4.6.3 Loop Filter

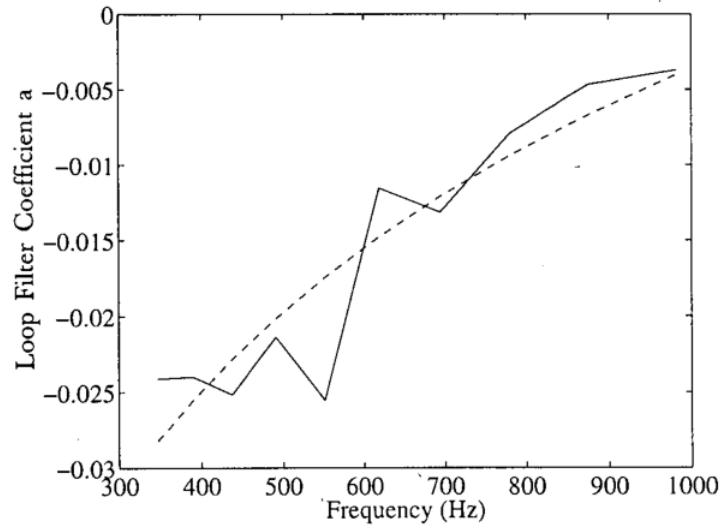
As described in Sec. 2.2.2, the loop filter was designed to approximate the various losses associated with vibrating string motion. It is a simple one-pole filter which uses a first-order polynomial approximation to generate  $a$  and  $g$  coefficients at various relative string length values (see Sec. 2.2.2). Aspects of this approach have been examined in Sec. 3.2.1 which illustrated that the filter itself is operational. The original paper provides the equations for the polynomial approximation as well as the polynomial coefficients. No frequency response plots are provided. In terms of the polynomial approximation, results are only provided for the first string. Accordingly the verification approach here involves recreating the original figures and relying on the fact the other coefficients have been copied correctly. Figures 4.33 and 4.35 show the original plots while Fig. 4.34 and 4.36 show the recreations respectively.



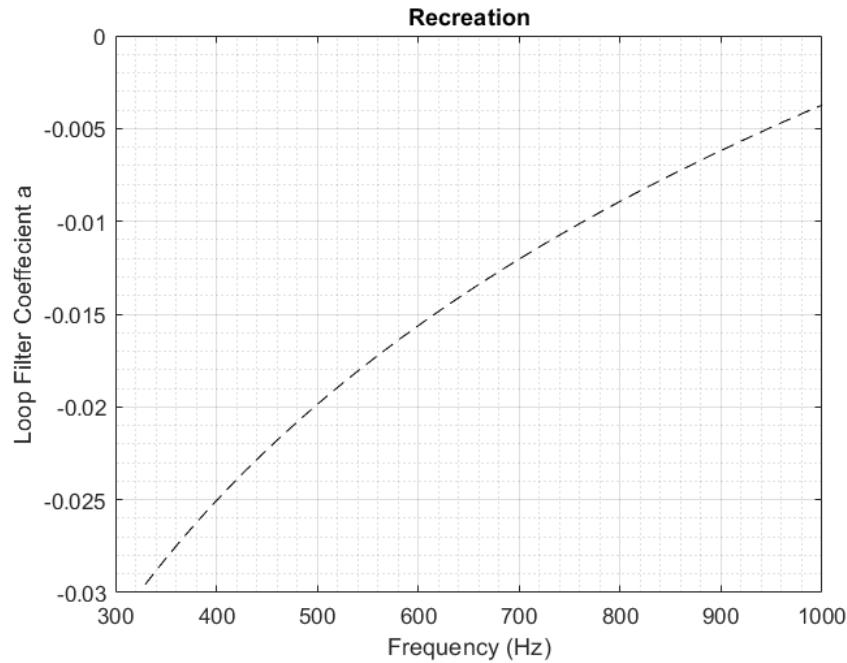
**Fig. 4.33** The loop gain  $g$  for modeling string 1 (solid line) and first-order polynomial fit (dashed line) from [Välimäki and Tolonen 1998].



**Fig. 4.34** The loop gain  $g$  polynomial for string 1 used in the slide synthesis model from this thesis.



**Fig. 4.35** The loop-filter  $a$  for string 1 (solid line) and first-order polynomial fit (dashed line) from [Välimäki and Tolonen 1998].



**Fig. 4.36** The loop-filter  $a$  polynomial for string 1 used in the slide synthesis model from this thesis.

#### 4.6.4 String Digital Waveguide

The string digital waveguide model was first verified to be free from transients and artifacts. This was achieved by running a series of sweeps on the relative length signal across its valid value range. After this was ensured, the tuning accuracy was verified via parabolic spectral interpolation as outlined in [Smith 2023b]. This method relies on computing the Discrete Fourier Transform (DFT) of the signal to sample its spectrum at N evenly spaced analysis frequencies. As a result, these analysis frequencies are harmonically related.

Ideally it would be best to have the harmonics generated by the string match the analysis frequencies. This would mitigate spectral leakage and reduce the need for interpolation in general. The fundamental frequency of the synthesized pitch was selected to align with a DFT bin but also be high enough so that several DFT bins exist in between the different harmonics. This is another method for mitigating spectral leakage. A lower fundamental frequency would produce a much more spectrally dense sound, which would be harder to get a clean estimate for.

The initial assumption in the algorithm development was that the strongest resonance present in the signal would be the fundamental frequency. In practice this was not the case, likely due to the initial waveform being initialized by noise as well as the non-idealities in the Loop Filter's magnitude response. The assumption had to be modified and a search range of the algorithm introduced. This search range is limited to  $1.5 * F_{0,bin}$ , where  $F_{0,bin}$  is the DFT bin associated with the synthesized sound. Other reasons for violations of the assumption could include the non-constant phase delay of both the loop as well as interpolation filter causing the frequencies to not all experience the same travel time and create a slight shift away from a perfectly harmonic signal.

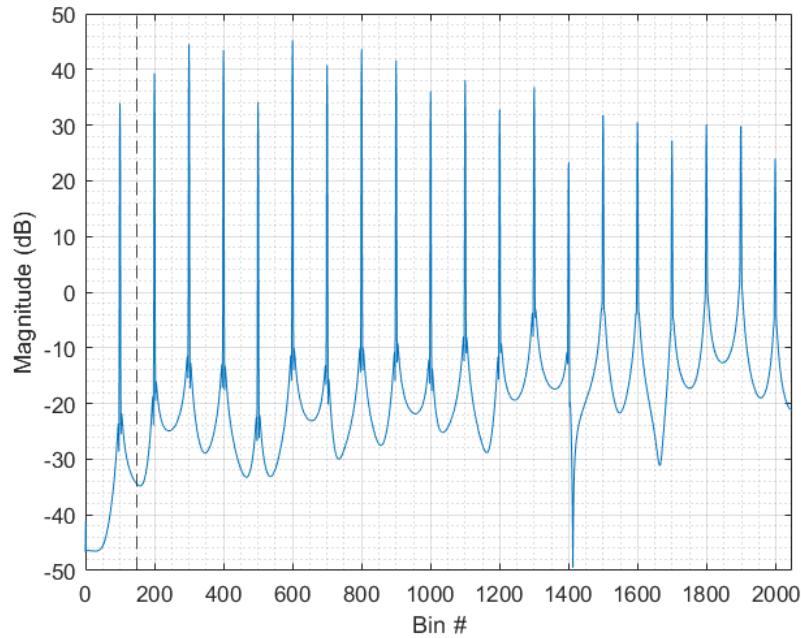
To analyze the test signal, the following Short-Time Fourier Transform (STFT) analysis parameters were used:

- Window Type = Hamming
- $F_s = 48,000$  Hz
- $N = 4096$
- Overlap = 75
- Window length = 12 ms = 576 samples

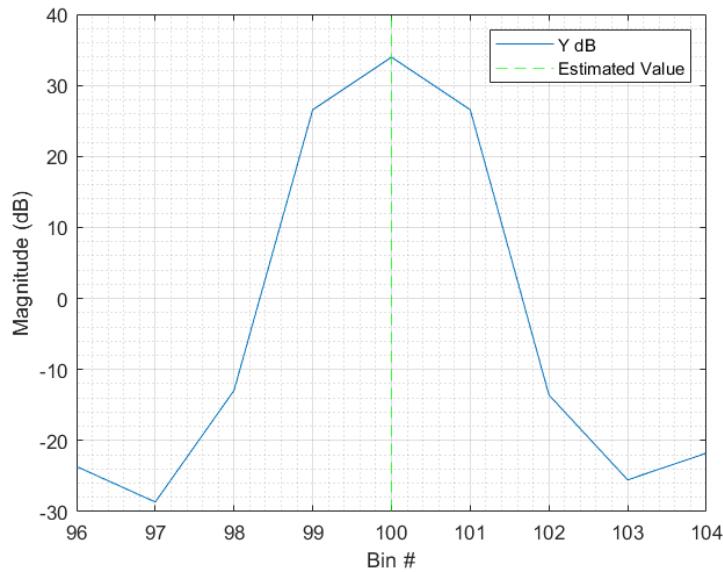
The DWG was configured to generate a signal with a fundamental frequency corresponding with  $F_{0,bin} = 100$ , or 1,171.9 Hz. Following the method described in [Smith 2023b], the calculated bin error was  $8.3499 \times 10^{-5}$ , which in Hertz is  $9.7850 \times 10^{-4}$  Hz. As this is extremely small, the tuning was considered to be accurate and verified. Figures 4.37-4.39 illustrate this process.

### 4.7 Slide Synthesizer

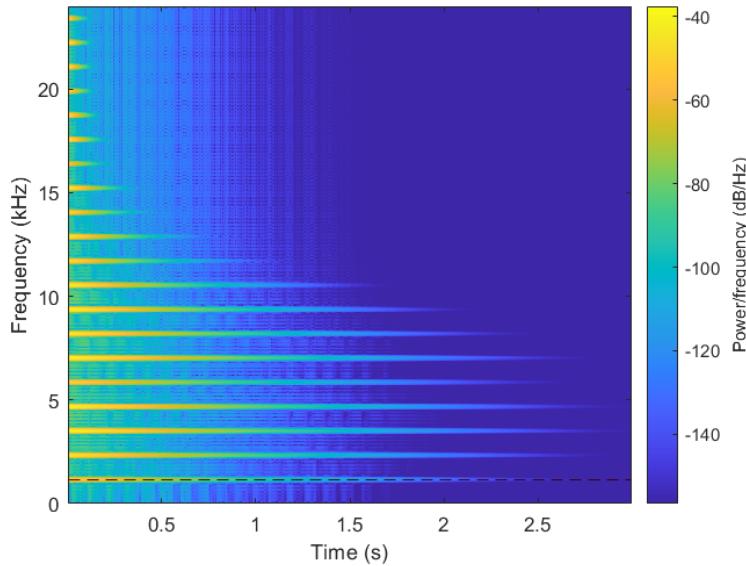
After all the individual components had been verified, a series of tests were run to check the overall functioning of the slide synthesizer. These tests were not meant to necessarily be physically accurate or musically useful, but purely as an approach to benchmark its basic behavior and



**Fig. 4.37** The DFT of the tone produced for tuning verification. Note the upper harmonics which surpass the fundamental in strength. The dashed-black line indicates the upper search limit.



**Fig. 4.38** A close-up of Fig. 4.37 near the estimated value.



**Fig. 4.39** The spectrogram for the synthesized verification tone. The black dashed line is the estimation of fundamental. Note how the fundamental is not the strongest harmonic in the signal.

determine the relationships between the CSG and DWG sound components. These were run with the different combinations of noise sources and harmonic accentuators as well as on the different string types.

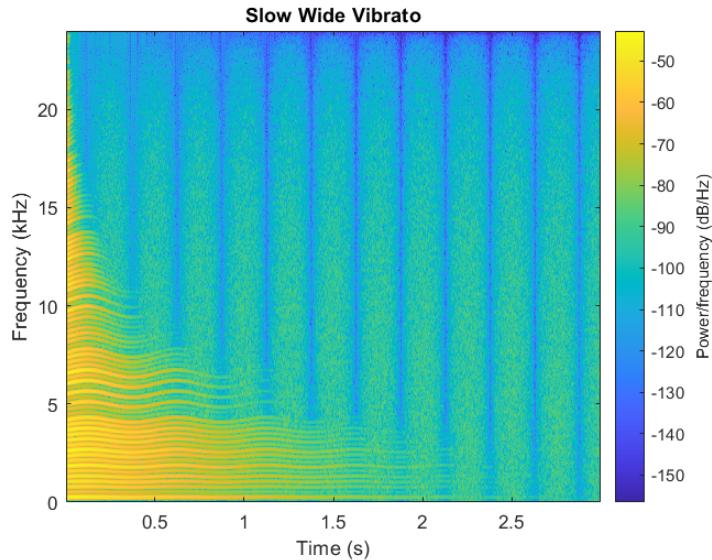
The testing scenarios are:

1. Basic pluck with no slide motion
2. Sliding up/down one fret over three seconds
3. Sliding up/down three frets over one second
4. Sliding up/down five frets over .5 seconds
5. Sliding up/down extremes of relative string length
6. Narrow/wide vibrato

The output files and spectrograms were generated using the Noise Pulse Train and Harmonic Resonator Bank configuration using a mixture of the different strings. The filenames are *SlideSynth-Test-#-direction.wav*, where # is replaced with the corresponding test number and *direction* is either up or down.

Figures 4.40 and 4.41 show the spectra from the different vibrato tests (case #6). These are shown as they illustrate correctness of the more basic tests due to their comparative complexity. As can clearly be seen, the harmonics follow a sinusoidal trajectory corresponding to the parameters

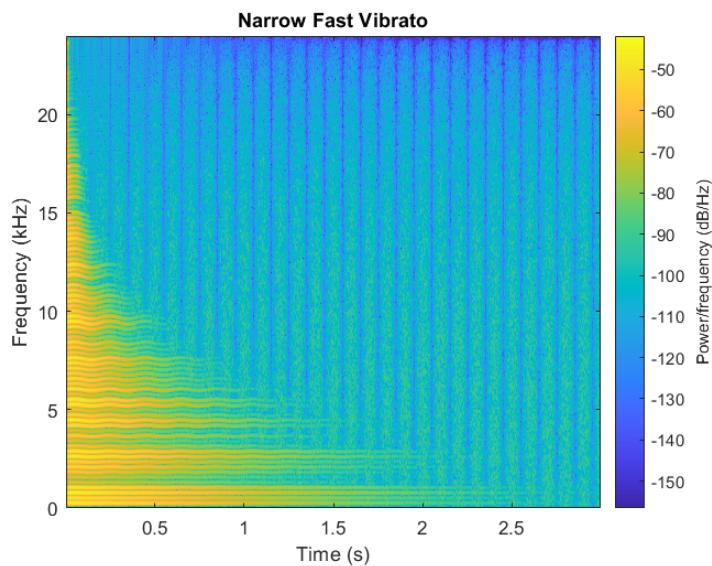
of the specified vibrato. The variations in the contact sound intensity can be seen as well. As is also shown, the contact sound dominates the spectrum as the string dies out, which is similar to what happens in the physical world.



**Fig. 4.40** The spectrogram corresponding to the wide vibrato test. Note how both the harmonics and contact sounds respond corresponding to the sinusoidal relative string length.

## 4.8 Real-Time Feasibility

The model used in this chapter is implemented in MATLAB due to its rich feature set related to digital signal processing which allows for easier verification of the implementation. The MATLAB implementation is not configured to run in real-time. However, it entirely seems feasible that the model could easily be implemented in real-time based on previous results. The model which serves as the basis for this thesis, described in [Pakarinen, Puputti, and Välimäki 2008] and [Puputti 2010], is shown operating in real-time in the following video: <https://www.youtube.com/watch?v=aIJ-8kd8rFs>. The implementation is in Pure Data and includes 6-strings simultaneously to recreate a full-guitar. Additionally, the computer is running image recognition software to facilitate gestural control as well as other software to simulate reverb and distortion. Given that this was achieved in 2010 using a 2.66 GHz Intel Pentium 4 CPU with 1 GB of RAM [Puputti 2010], it entirely seems feasible that the model would be easily implemented using today's hardware, 13 years later, based on the rate at which computing power regularly increases.



**Fig. 4.41** The spectrogram corresponding to the narrow vibrato test. Note how both the harmonics and contact sounds respond corresponding to the sinusoidal relative string length.

---

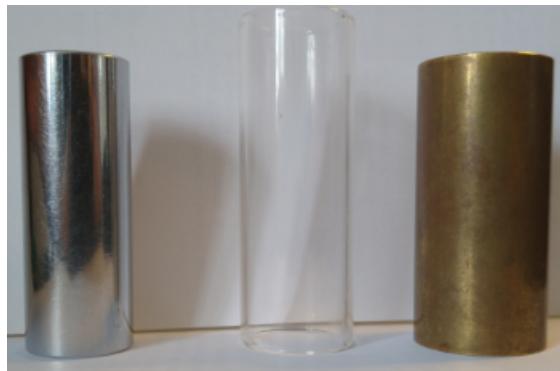
# Chapter 5

# Physical Measurements

In this section, the physical measurements performed as part of the model development will be described. These were performed for two purposes. Some experiments are meant to recreate the results from the papers which the model was based on, which help confirm the theoretical aspects which inform the synthesis model (e.g., contact sound spectrum verification). Others were performed in an attempt to refine the model and make it more physically accurate as not all the synthesis parameters had a strong physical basis in the original papers describing the slide guitar synthesis model (e.g.,  $T_{60}$  values).

## 5.1 Experimental Setup

All tests were done on a Yamaha-502 steel string acoustic guitar which was fitted with medium-gauge D'Addario phosphor bronze strings. The strings were aged so their surface was rougher and less uniform as compared to a fresh set of strings. This was done because the sounds of aged strings are often considered more preferable on an acoustic guitar by many players. The slides used were Dunlop brass/chrome/glass slides. Figure 5.1 contains a picture of the slides. The mass of the slides in descending order is: 125 grams (brass), 50.0 grams (chrome) and 17.5 grams (glass).



**Fig. 5.1** The slides used in the measurements.

The audio interface used was an RME Fireface UC. The microphones used were an AKG C480B and a DPA 4011-TL. The AKG was selected to be able to compare against the original measurements if need be as [Pakarinen, Penttinen, and Bank 2007] and [Pakarinen, Puputti, and Välimäki 2008] list it as part of the experimental setup. The DPA 4011-TL was selected due to its flatter response which would allow more accurate measurements to be made. Contact microphones were considered as a potential option, however the extra mass they add to the guitar based on their placement could influence the measurements. For this reason, they were not used. The audio was captured at a sampling rate of 48,000 kHz using a MATLAB script to control the audio interface via ASIO.

For basic measurements (e.g., testing for the existence of coupling), the Centre for Interdisciplinary Research in Music Media and Technology's (CIRMMT) *A818 - Performance and Recording Lab* was used for measurements. When more precise measurements were required CIRMMT's *A816 - Spatial Audio Lab* was used as it is a hemi-anechoic chamber.

## 5.2 String Winding Density

The first measurement made was to determine the linear string winding density for the different strings present on the guitar. This was done using a caliper set to a distance of 1 cm and taking a photo of it aligned with the beginning of a string winding through a magnifying glass. This allowed for the number of windings per centimeter to be counted which could then be multiplied by 100 to determine  $n_w$ , the density per meter, for each of the wound strings. Table 5.1 provides a summary of these measurements. Figure 5.2 illustrates the method for the E-string, while Appendix A contains the photos for all the strings measured.

String	Windings cm	Windings m
E	20	2000
A	26	2600
D	38	3800

**Table 5.1** Measured string winding densities.



**Fig. 5.2** The E-string winding density measurement. The distance between the inner edges of the caliper is 1 cm. Approximately 20 windings lie on this distance.

### 5.3 Longitudinal to Transverse Coupling

The original model assumes coupling between the longitudinal motion of the slide and transverse vibrations of the strings. This is indicated by the injection of the CSG output into the string DWG structure as shown in Fig. 2.8. Two potential sources for this coupling are due to reflections from an imperfect bridge impedance as well as imperfections in the slide/winding collisions, or friction between the string/slide surface (especially in the case of the unwound strings). The goal of these measurements is to confirm that the coupling exists and qualitatively determine the strength to facilitate rough comparisons across different string and slide combinations. These measurements were performed in *A818* as the the precision required for a successful experiment did not require the less reflective attributes of *A816*.

#### 5.3.1 Setup and Method

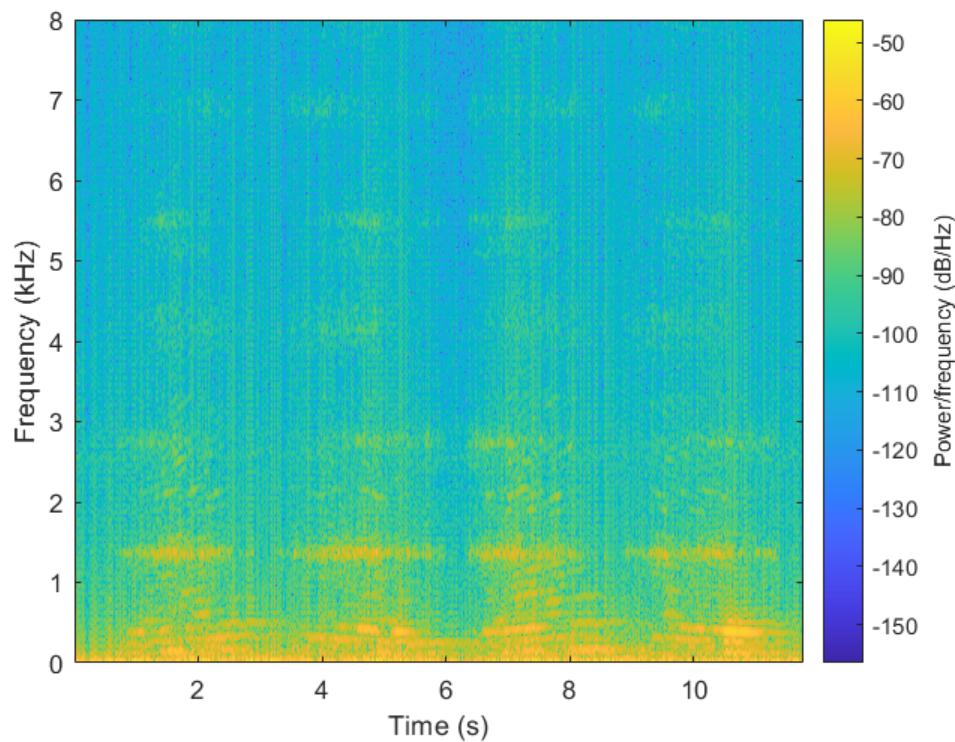
The general method for testing the coupling on a string is described as follows. First, the other strings were muted with ear plugs to prevent any sort of sympathetic vibrations or other unwanted coupling. The slide was held in the left hand and the right hand was used to dampen the string while the slide was placed on it. This was done as the initial contact between the slide and string often creates a click/impulse, which is unwanted in this case. The muting helps ensure that there are no vibrations in the string before the slide moves. After ensuring the system would be starting from rest, the slide was moved along the string from the 7th fret up to the 12th and back. This motion was repeated twice in a recording. The mic was placed over the 12th fret as this produced the best results. The neck rest used was *Music Nomad Cradle Cube* which consists of hard plastic covered in a softer blue gel polymer. The guitar itself rests on top of a *Music Nomad Premium Instrument Work Mat*, which is made of a flexible foam, placed on top of a movable table. The neck rest and mat aim to both protect the instrument as well as provide decoupling between in and the supporting table. Figure 5.3 shows a photograph of the setup.



**Fig. 5.3** The setup used to confirm longitudinal-to-transverse coupling. Ear plugs are used to mute strings not being measured.

### 5.3.2 Results

For all the strings it was confirmed that coupling between the longitudinal slide motion and transverse vibrations of the strings exists. Figure 5.4 illustrates this for the brass slide being moved along the low E-string. The recording used to generate this plot can be heard in *CouplingMeasurement-E-string-Brass.wav*. As can clearly be observed in the plot and heard on the recording, the appropriate harmonics of the transverse vibrations corresponding to the slide's placement along the string are being stimulated by the slide's longitudinal motion.



**Fig. 5.4** This spectrogram corresponds to the coupling measurements for the brass slide moved along the E string.

Two patterns were observed relating the strength of the coupling to the slide used as well as the string. For a specific string, the slides produced coupling from greatest to least in the following order: brass, chrome and glass. This order corresponds to listing the slides in order from heaviest to lightest as well as roughest to smoothest. Intuitively this makes sense as the heavier slides have more mass and would proportionally transfer more energy to the string on each slide/winding collision as compared to the lighter slides. The same would apply from a friction standpoint, where the rougher surfaces would provide more opportunities to create kinetic friction and transfer energy into the string.

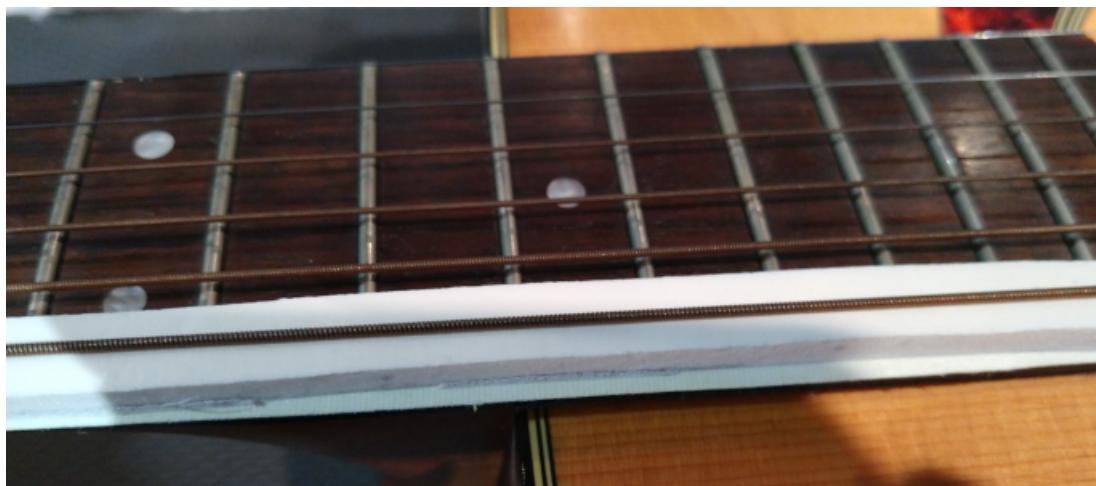
When comparing the coupling across different strings for the same slide type, the pattern observed was that the strings produced coupling from greatest to least in the following order:

E, A, D, G, B and e. The first four are wound strings, and in order of decreasing winding size. The pattern could be explained by the fact that the larger windings provide more surface area to collide with allowing more coupling to be generated. In terms of the unwound strings, a similar argument could be made regarding the thickness of the string providing more surface area for the slide to interact with. However, given the strings are rather old and exhibit corrosion, it is hard to definitively conclude the coupling source regarding the unwound strings.

## 5.4 Contact Noise Spectrum

### 5.4.1 Method

The contact noise spectrum measurements were done in the *Spatial Audio Lab*. Ear plugs and window sealing foam were used to mute the transverse vibrations of the string being measured. As one side of the foam was adhesive, a strip of tape was applied to ensure the fretboard wouldn't be damaged by the adhesive. The other strings were muted with the right hand near the sound hole while the measurement was being performed. Figure 5.5 illustrates the foam-strip muting technique.



**Fig. 5.5** A foam strip placed underneath the E string to dampen its transverse vibrations. Note how the string protrudes nicely without sinking too far into the foam. This allows it to be more easily simulated, without the foam influencing the motion of the stimulation tool.

In order to mitigate the non-deterministic aspect of a human-controlled slide, the following approach was used. A metronome was set to 100 bpm to provide a consistent pulse. The slide was then moved up one length on a beat and then down the same length the following beat, with the up-down pattern performed twice in a row for each set of frets. The first distance was defined by frets 5 & 7, the second by frets 5 & 9 and the third by frets 5 & 12. These different distances were selected to have the top speed reached by the slide increase on each measure. As the contact sound is agnostic to the direction of the slide motion, one up-down pattern would produce a pair of extremely similar contact sounds.

### 5.4.2 Results

#### 5.4.3 Wound Strings

Figure 5.6 illustrates a spectrogram of measurements from the DPA microphone for the experiment being performed on the low E string with the brass slide. This can be heard in the file *NoiseCharacterization-E-brass.wav*. The results are extremely similar to the analysis performed in [Pakarinen, Puputti, and Välimäki 2008] for the slide guitar and [Pakarinen, Penttinen, and Bank 2007] for finger noises. There are clearly both the time-varying harmonic as well as static modal components to the sound. It is also quite clear that the fundamental frequency of the harmonic varies with the slide velocity.

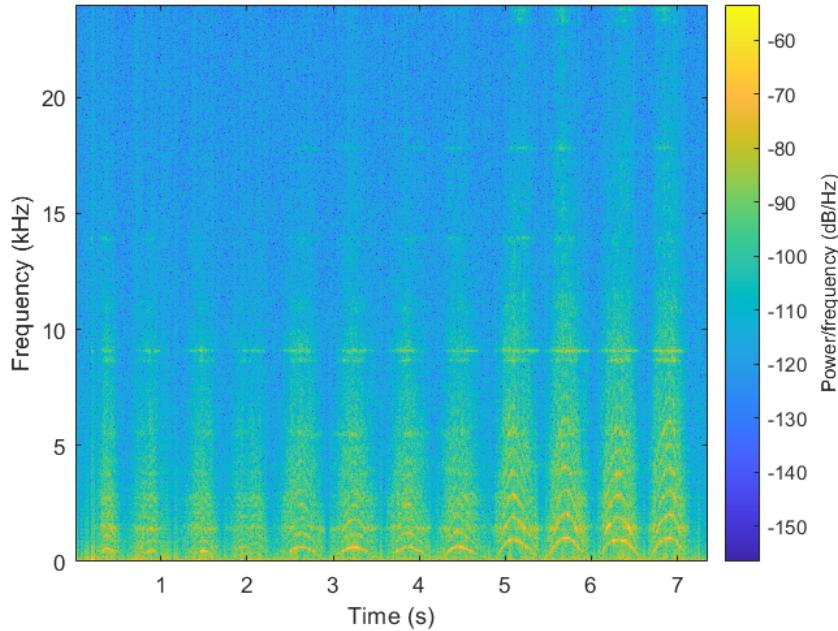
There are also resonances which appear in the measurements that could be due from either higher longitudinal modes or coupling between the guitar and its supporting structure. They appear all throughout the spectrum but are most strongly stimulated around 9 and 8.5 kHz and reach all the way to the upper end of the spectrum near 23.4 kHz and 23.8 kHz when the slide is moving fastest. [Pakarinen, Puputti, and Välimäki 2008] did not attempt to control for the slide speed, nor is there a detailed explanation of the physical measurement setup. As mentioned in Sec. 5.3.1, foam and rubber were placed between the guitar body and neck in an attempt to decouple them from the table. However, the decoupling might be imperfect resulting in energy being transferred to the table and its resonances stimulated. Equation 2.17 indicates that the strength of the modes is dependent on the excitation force which supports the idea that the higher slide velocity is stimulating modes which were weakly stimulated before. It is hard to make a definitive statement regarding the source without further measurements.

#### 5.4.4 Unwound Strings

The unwound strings (the B and e) were more difficult to characterize as they had a tendency to sink into the foam more so than the wound strings. As a result the slide would make contact with the protruding foam. This simultaneously made it difficult to move the slide in a consistent and repeatable manner as well as introduced its contact sound due to the noise between the foam and the slide. An attempt was made to cut the foam into a thinner strip to reduce the protrusions, but this was difficult to do uniformly and the same problem was exhibited. Figure 5.7 shows the results of this attempt. Accordingly, the parameters of the synthesis model for unwound strings could not be as precisely confirmed.

### 5.5 Decay Rate of Single Winding Impact

Based on the information in [Pakarinen, Puputti, and Välimäki 2008] and [Puputti 2010], it is unclear as to how the values for the decay rate associated with the noise pulses were obtained. Two methods were investigated in an attempt to derive a more physically informed value for these parameters. In either case it is not necessary to know the input force precisely, as the relative changes in signal strength over time are what is of interest. This drastically reduces the complexity of the tools required in the physical setup. The foam used for muting would likely impede the longitudinal motion of the string slightly and as it is also used in the noise characterization setup then the same effect would be seen in both experiments. As this experiment is used to inform the



**Fig. 5.6** The measured spectrum for contact noises of the brass slide on E string. A 12 ms rectangular window with 75% overlap was used. The slide was moved 3 different distances, 4 times each.



**Fig. 5.7** A failed attempt to reduce the thickness of the foam.

tuning process by ear, it is not necessary for it to be the most precise. The small amount of error the foam introduces is acceptable as it narrows down the search range of tuning parameters later on and provides a solid starting point.

### 5.5.1 Methods

Both methods here rely on an assumption of linearity. In both setups, the string being measured was muted using the foam strips as in the contact sound experiments. The object used to stimulate the strings was a utility knife blade removed from its casing. Attempts were made to use it inside the casing, however this setup did not provide the same rigidity as when removed from the case. The interaction between the knife edge and the string windings is different than the slide and string, however the assumption of linearity allows the results to be generalized. Additionally the goal here was not to develop a precise characterization, but have a stronger physical basis for the

$T_{60}$  parameter used to control the synthesizer.

### Strike Multiple Windings

Figure 5.8 illustrates the experimental setup for the *Strike Multiple Windings* method. In this setup, the knife edge was held at an angle relative to the longitudinal axis of the string. An angle allows it to more easily slide across the windings of the surface. This is desirable in this case as the goal is to strike a small number of windings sequentially and extract the  $T_{60}$  parameter from the resulting signal which either contains overlapped impulse responses or several individual ones depending on the relationship between the knife-edge velocity and duration of the impulse responses. The approach was developed based on difficulties in controlling the physical setup in the *Hold and Release* method.

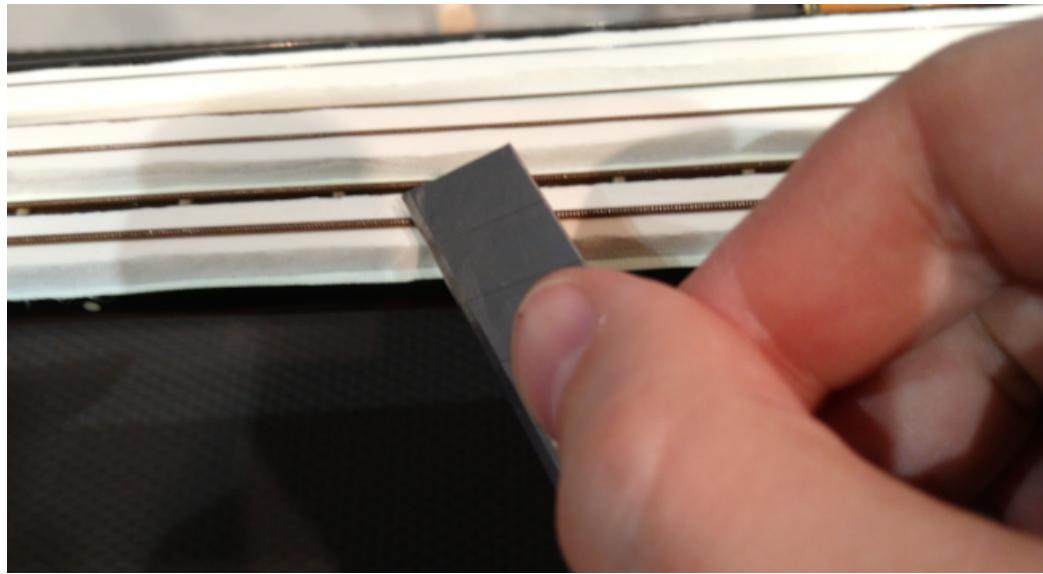
### Hold and Release

The experimental setup in the *Hold and Release* case is nearly identical to the previous method, differing in the angle and position of the knife edge. The knife is now placed orthogonally to the string and its edge is wedged between two string windings. From there, force was applied in the longitudinal direction until slippage occurred allowing the knife edge to dislodge from between the string windings. Once dislodged, the utility knife does not make contact with the string again and the system is left to vibrate undisturbed. At the point of dislodging, all the potential energy stored in the string longitudinally would start to release and be exchanged into kinetic energy, starting the oscillation process. Capturing this process would ideally allow the system's overall decay rate to be measured and used to tune the model.

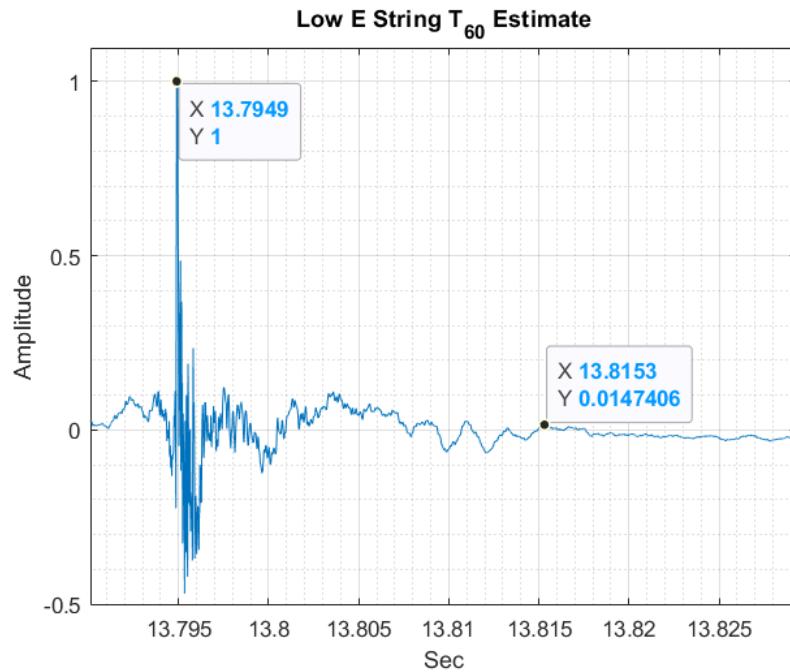
#### 5.5.2 Results

Unfortunately, neither of the two methods worked as exactly as hoped due to the noise floor of the measurement system. However, the results can be used to help determine a starting point from which to begin tuning the  $T_{60}$  parameter for each string as well as providing values to compare against the ones listed in Fig. B.6 of [Puputti 2010] where the implementation is described.

Figure 5.9 illustrates the strongest peak for the low E string which was obtained from the single winding measurements. The data have been normalized to make it easier to find the point which corresponds to a decay of 60 dB. Unfortunately, due to the noise in the system, finding this value was not easy. However, an extremely rough measurement of  $T_{60}$  can be obtained by estimating the point at which the noise becomes strongest after the peak. Based on the values shown in the graph, this results in a measured  $T_{60}$  value of approximately 20.4 ms. Clearly this measurement is extremely rough and not the most precise or accurate, however for the same string [Puputti 2010] lists a value of 200 ms. The measured value is roughly an order of magnitude smaller and as the method by which the parameter values were obtained is not elaborated upon in the source material, it provides a stronger physical basis from which to begin tuning the parameter.



**Fig. 5.8** The experimental setup for the striking test. Alternating strings have muting foam as it was easier to prepare the setup for the next string measurements. Note how the width of the knife edge is at an angle to the longitudinal direction. In the hold-and-release test, it is closer to orthogonal.



**Fig. 5.9** The strongest peak for the single winding experiment on the low E string which is used to roughly estimate  $T_{60}$ . The estimate here puts  $T_{60}$  at approximately .0204 seconds.

### 5.6 Playing Examples

The last measurements made were recordings of a particular slide example. This was done to be able to capture real-world examples for comparison to the synthesized examples. The spectral aspects of the sound are useful from the standpoint of determining the parameters of the synthesis model (e.g., balance between the longitudinal modes and harmonic component of wound string contact sounds). The fundamental frequency as a function of time is also of interest as this can be used to inform the  $L[m]$  control signal to make the articulations more realistic. This will be discussed more in the subsequent chapter on sound design.

## Chapter 6

# Sound Design and Control Signals

This chapter provides a description on different aspects of the model . First, decisions regarding architectural aspects of the Wound Contact Sound Generator model are described as well as how the corresponding model was tuned. Second, aspects relating to the waveform used to initialize the digital waveguide are discussed. The chapter concludes with a discussion on how  $L[n]$  can be configured to generate different sounds with the complete model.

### 6.1 Wound Contact Sound Generator

Comparatively speaking, the wound contact sound generator model is more complex than its unwound counterpart. There are architectural decisions to make in terms of which noise source as well as harmonic accentuator to use. Once these have been decided, the parameters associated with the different Contact Sound Generator (CSG) components need to be tuned.

#### 6.1.1 Decay Rate

The first element to consider for the CSG model is the value of  $T_{60}$  associated with each string. This is what determines the length of the noise pulse/burst which is meant to mimic the impulse response of a single string-to-winding impact. As mentioned before in Sec. 5.5, the original source material did not contain a detailed explanation regarding how the values were determined for each string. Section 5.5 also explained the difficulties in measuring the value precisely and provided a physical measurement by which the tuning can begin.

In terms of the perceptual aspects, as the CSG is not meant to be an exact computational model of the physical system, the  $T_{60}$  parameter controls the noisiness of the excitation signal fed into the resonators. It could conversely be thought of as controlling the excitation's impulsiveness as well. The effects of the parameter are intimately tied to the value of  $f_c[n]$ , as this control signal determines the rate at which the pulses/bursts are generated. If the period of the firing rate is shorter than the specified  $T_{60}$ , then the generated signals will overlap. The longer the generated pulses overlap in time, the more the excitation signal transitions from impulse-like to noise-like. The number of overlapping impulses also affects this transition. In the case of the Noise Burst Generator, the hard clipper facilitates the signal becoming completely white noise, while the Noise

Pulse Train will become more noise-like while retaining a harmonic aspect.

The differences in the effect of  $T_{60}$  can be heard in the following files: *T60-Short-NPT.wav* and *T60-Long-NPT.wav*. A value of  $T_{60} = 20$  ms was used for the long example while a value of  $T_{60} = 2$  ms was used for the short one. Longer  $T_{60}$  values are useful for emphasizing the longitudinal modes as the more noise-like signals stimulate the 4th-order filter more fully. The Noise Burst Generator is not shown, as the effects are not as drastic as in the Noise Pulse Train case and have largely been shown in Sec. 4.3.4.

### 6.1.2 Noise Sources and Harmonic Accentuation Combinations

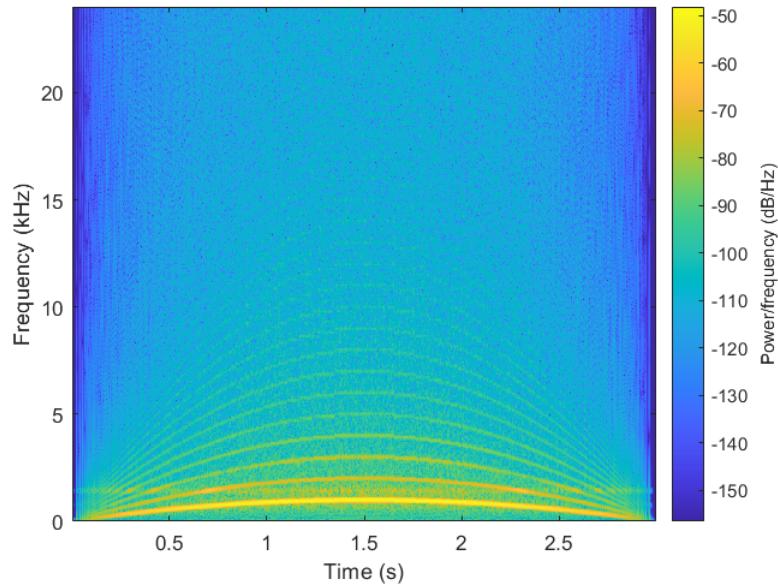
The different noise sources were originally designed with a different harmonic accentuation technique in mind. As mentioned in Sec. 3.1.5, the Reso + Tanh structure works better with harmonic sources while the Harmonic Resonator Bank is more agnostic to its input and will still work with either a periodic or a noisy source. However, experimenting with different combinations outside of their original intent can be an interesting method to gain insight into their operation, how to best utilize them sonically, as well as the trade-offs associated with each different method. The following section elaborates on the different noise source and harmonic accentuation pairings in that manner.

#### Noise Pulse Train

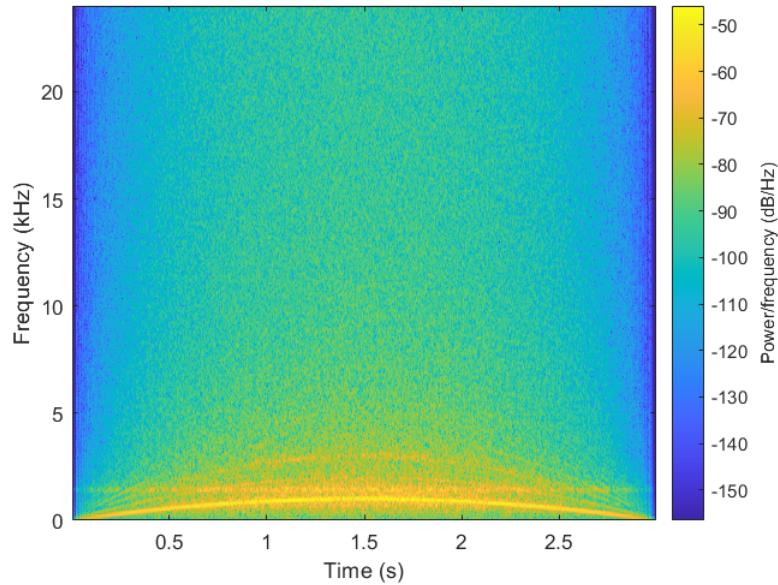
The Noise Pulse Train structure was introduced in [Pakarinen, Puputti, and Välimäki 2008] where it is paired with the Reso + Tanh object to form an efficient algorithm to generate harmonics. This noise source would theoretically work with either harmonic accentuation technique as it generates an inherently harmonic signal. This is useful in the Reso + Tanh case as the `tanh()` function requires a periodic source in order to properly function.

**Reso + Tanh Results:** Figures 6.1 and 6.2 illustrate the effects of the  $T_{60}$  parameter on the architectural pairing of the Noise Pulse Train noise source with the Reso + Tanh harmonic accentuator. The same short and long  $T_{60}$  values of 2 ms and 20 ms from before were used. The control signal is the same parabolic sweep of  $f_c[n]$  used throughout the thesis. The results can be heard in *NPT-ResoTanh-T60-Short.wav* and *NPT-ResoTanh-T60-Long.wav*.

As can clearly be seen/heard when comparing the two results, the shorter  $T_{60}$  value generates a signal with a much more well-defined harmonic presence. The number of harmonics is comparatively much greater than in the longer  $T_{60}$  case. The longitudinal modes are also less present. Both of these can be explained by how a longer  $T_{60}$  value allows for more overlap between adjacent noise pulses. The larger the amount of overlap, the more the individual pulses are obfuscated. This diminishes the fundamental temporal structure which gives rise to the harmonic spectral pattern as the signal becomes more noise-like overall.



**Fig. 6.1** Spectrogram of the output from the Noise Pulse Train paired with the Reso + Tanh structure for  $T_{60} = 2$  ms. Note the more clearly defined harmonic structure and comparatively weaker longitudinal mode presence.

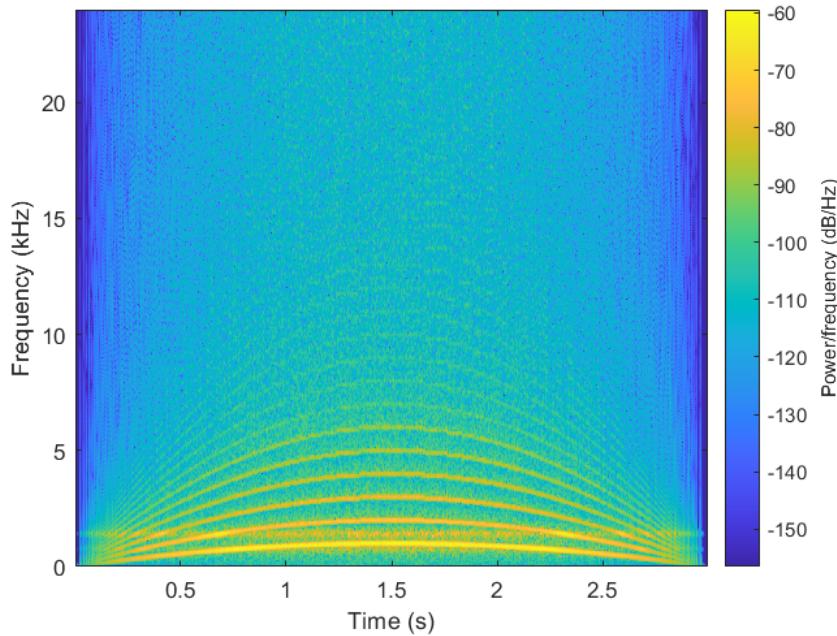


**Fig. 6.2** Spectrogram of the output from the Noise Pulse Train paired with the Reso + Tanh structure for  $T_{60} = 20$  ms. Note the strong longitudinal mode presence and loss of upper harmonics due to the more noise like structure of the excitation signal.

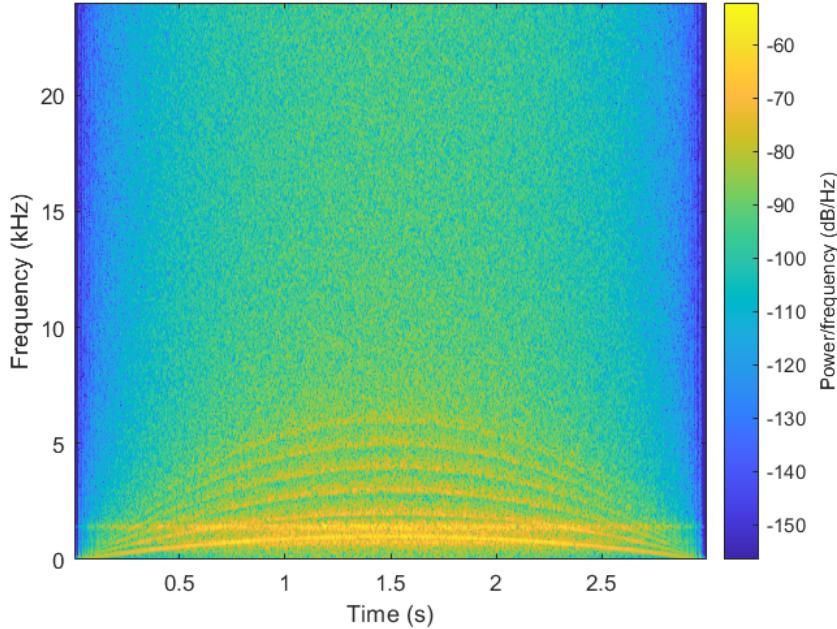
**Harmonic Resonator Bank Results:** Figures 6.3 and 6.4 show the spectrograms of the output of the Noise Pulse Train paired with the Harmonic Resonator Bank (HRB) for the long and short  $T_{60}$  values. The HRB was set to only 6 harmonics. The results can be heard in *NPT-HRB-T60-Short.wav* and *NPT-HRB-T60-Long.wav*.

When comparing Fig. 6.3 to Fig. 6.1, the same strong harmonic structure is present. The use of the HRB puts a strong emphasis on the first six harmonics which can be seen in how Fig. 6.3 indicates they are slightly stronger than in Reso + Tanh case. As one would expect, the perceptual result of this is less emphasis on the fundamental frequency which can be heard in the corresponding .wav files.

When looking at the results of Fig. 6.4, where the HRB is used with the longer  $T_{60}$  value, the most notable feature is the limitation in the number of harmonics corresponding to the configuration of the HRB. The second most prominent aspect is that the energy of the harmonics is spread out over a wider band as compared to the signals generated using the shorter  $T_{60}$ . This is due to the fact that these harmonics are created by the resonator filters in the HRB itself as the excitation signal is more noise-like and lacks a strong harmonic structure. The spread of the energy can be controlled by the width of the resonator. In Fig. 6.3, the resonators act more to accentuate the already existing harmonics in the signal. From a perceptual standpoint, the longer  $T_{60}$  here causes a loss in the impulsive aspects of the sound. This in turn causes the output to sound less like it was generated from a periodic set of impacts/collisions and more like the interaction between two smoother surfaces.



**Fig. 6.3** Spectrogram of the output from the Noise Pulse Train paired with the Harmonic Resonator Bank for  $T_{60} = 2$  ms. Compared to the Reso + Tanh approach the harmonics the first 6 harmonics are more emphasized.



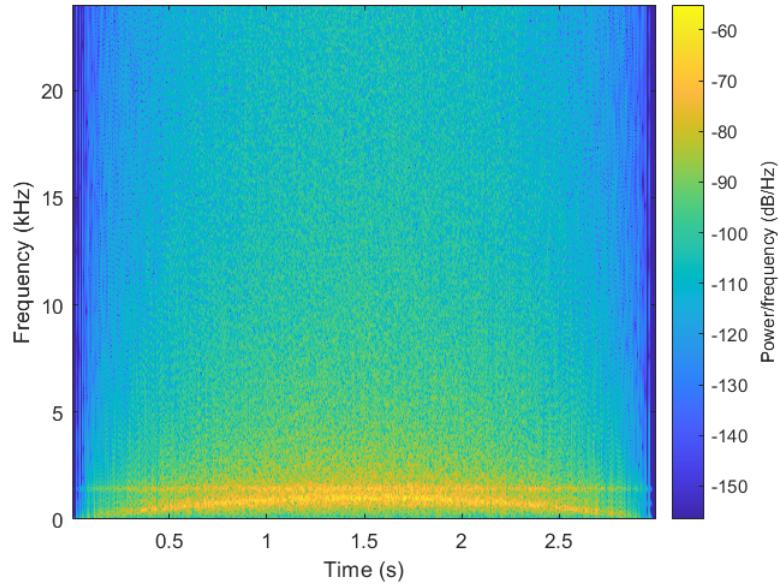
**Fig. 6.4** Spectrogram of the output from the Noise Pulse Train paired with the Harmonic Resonator Bank structure for  $T_{60} = 20$  ms. Note that the number of harmonics is limited by the number specified in the HRB and they aren't as energetically concentrated.

### Noise Burst Generator

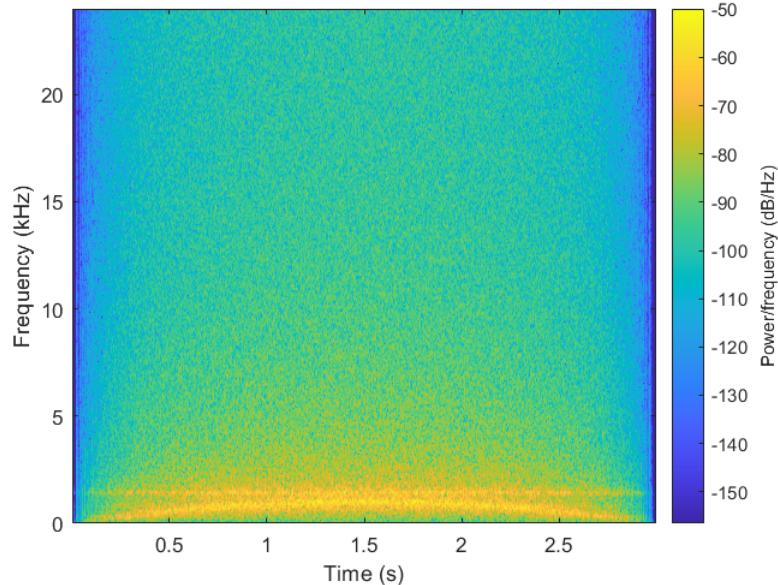
The basis for this structure comes from the guqin model introduced in [Penttinen et al. 2006]. As this signal is not inherently harmonic, it does not operate well with the Reso + Tanh approach. Much better results are achieved using the Harmonic Resonator Bank method as the signal itself is much more noise like. However, the Noise Pulse Train provides much better results in the context of the synthesizer given that it creates a harmonic structure and retains an impulsive aspect when  $T_{60}$  is tuned properly.

**Reso + Tanh Results:** As mentioned before, it was not expected that this combination of components would perform well in the goal of perceptually simulating the slide-to-winding impact sound. Figures 6.5 and 6.6 show spectrograms for the results. The sounds can be heard in *NBG-RT-T60-Short.wav* and *NBG-RT-T60-Long.wav*.

These results clearly indicate a lack of harmonic structure. There are two main spectral components present in the signal: a band of energy corresponding to the fundamental frequency (which is extracted by the resonator in the Reso + Tanh) and a prominent longitudinal mode. Between the two  $T_{60}$  parameters, the difference is that the longer  $T_{60}$  produces more concentrated energy in the resulting bands due to its inherently more noise-like structure.

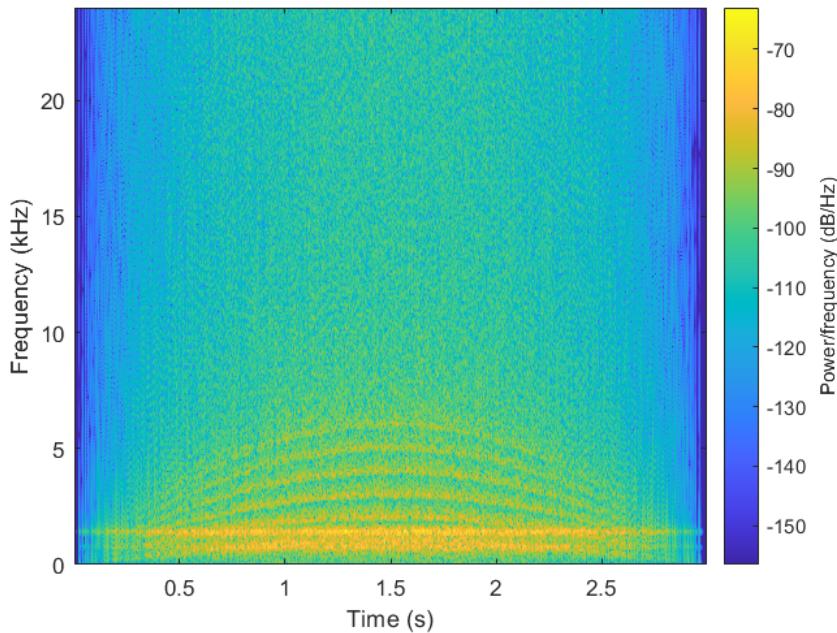


**Fig. 6.5** Spectrogram of the output from the Noise Burst Generator paired with the Reso + Tanh structure for  $T_{60} = 2$  ms. Only the fundamental and some longitudinal modal effects are present.



**Fig. 6.6** Spectrogram of the output from the Noise Burst Generator paired with the Reso + Tanh structure for  $T_{60} = 20$  ms.

**Harmonic Resonator Bank Results:** The results of this test are an improvement over the previous ones, however they do not yield results as good as when the Noise Pulse Train generates the excitation. Figures 6.7 and 6.8 illustrate the spectrograms for the short and long  $T_{60}$  values respectively. The results can be heard in *NBG-HRB-T60-Short.wav* and *NBG-HRB-T60-Long.wav*. There is a much clearer harmonic structure as compared to using Reso + Tanh with the Noise Burst Generator. However, the impulsive aspect is not present. This component is necessary to sound like two objects are colliding as opposed to just rubbing and creating friction sounds.

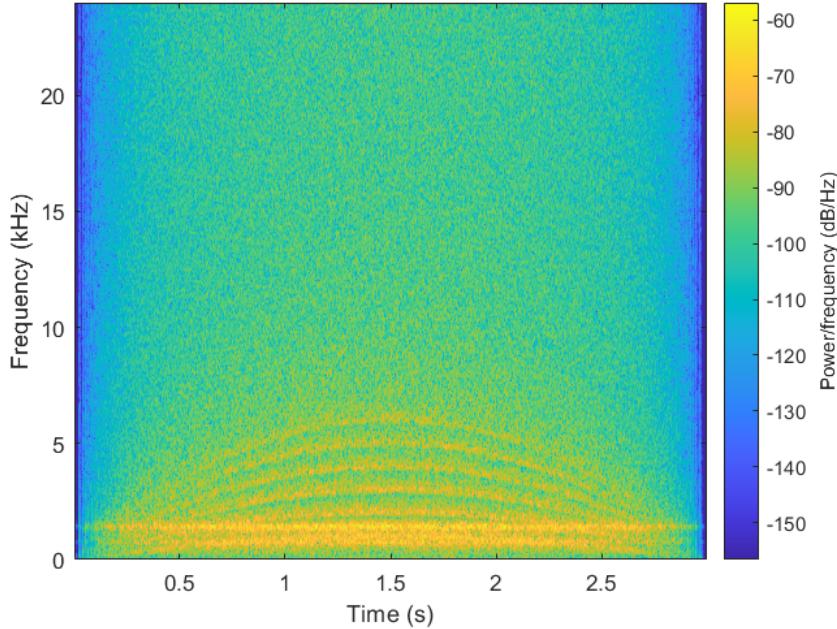


**Fig. 6.7** Spectrogram of the output from the Noise Burst Generator paired with the Harmonic Resonator Bank for  $T_{60} = 2$  ms.

### 6.1.3 Matching the Recordings

Determining which pair of noise source/harmonic accentuation objects to use, as well as how to tune the parameters of the Wound CSG, was done by comparison using the spectrum from Fig. 5.6 and the corresponding measurements as the baseline. A control curve for *slideSpeed[n]* was generated to mimic the speed which the slide experienced during the measurements. Subsequently, this was applied to a Wound CSG object tuned to various sets of parameters. The best results are shown in Fig. 6.9 and can be heard in *Tuned-Wound-CSG.wav*.

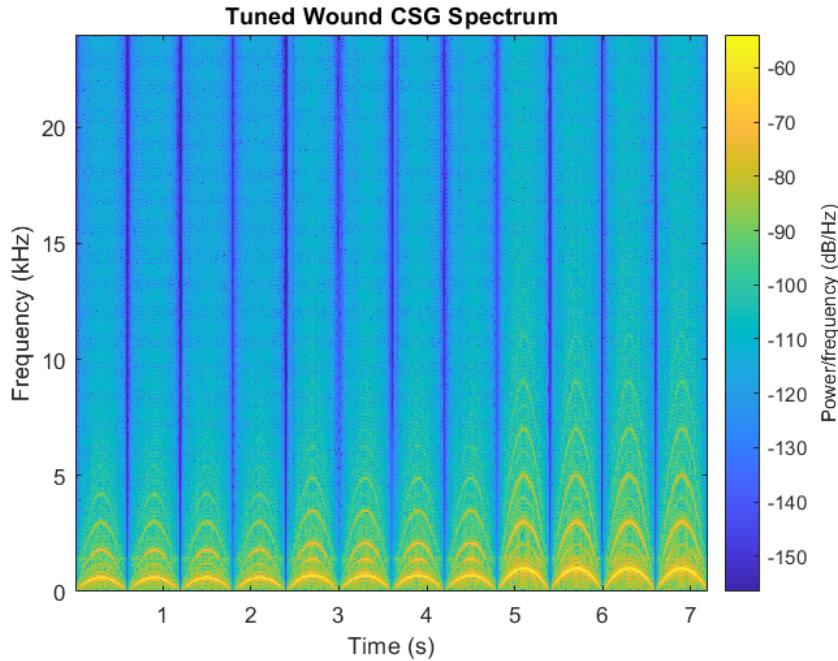
Based on the results of the previous section, it was determined that the best noise source and harmonic accentuator pair is the Noise Pulse Train (NPT) and Reso + Tanh structure. The harmonics produced using this pair sound substantially more realistic as compared to the harmonics produced via filtered noise using the HRB. For the tuned  $T_{60}$  value, this pair produced a similar number of harmonics to what appears in the last slide event of Fig. 5.6 using a pre-tanh() gain



**Fig. 6.8** Spectrogram of the output from the Noise Burst Generator paired with the Harmonic Resonator Bank for  $T_{60} = 20$  ms.

of 30. The relative strengths of the odd-order harmonics are increased due to the imperfections in the filtering mechanism by which the fundamental is extracted. This is unavoidable as the resonator which extracts the fundamental has  $r = .99$  and cannot be made more frequency selective without adversely affecting the harmonic structure of the NPT output. These harmonic imbalances could be corrected by adding an HRB structure after the Reso + Tanh to help control the harmonic proportions more easily. However, this would increase the computational complexity of the implementation potentially making it not feasible for a real-time scenario. It is unknown why the slide measurements which occur earlier in the recording, from Fig. 5.6, do not exhibit the same number of harmonics and overall strength as the corresponding sounds in the synthesized version. There is likely an interaction between the slide and string surfaces which is not described purely by the slide speed or a component which is missing in the model.

In terms of  $T_{60}$ , experimentation showed that a value of 2 ms in combination with  $g_{bal} = .15$  provided the best sounding results while producing a synthesized spectrum which was closest to the measured contact sound characteristics. This  $T_{60}$  value provides a balance between being able to generate a harmonic waveform while at the same time providing enough noise-like characteristics to stimulate the longitudinal mode filters. Shorter values were unable to properly excite the 4th-order filter, while longer values lacked a sufficiently harmonic structure to produce convincing sounds. The 2 ms value is an order of magnitude lower than the rough estimate from Sec. 5.5.2 and two orders lower than the 200 ms listed in Fig. B6 of [Puputti 2010]. This suggests that more research needs to be done to be able to properly measure this quantity and understand its ramifications in the computational model of the sound, whether it be a perceptual approximation



**Fig. 6.9** Spectrogram of the output from the tuned Wound CSG using the Noise Pulse Train, Reso + Tanh structure,  $T_{60} = 2$  ms and  $g_{bal} = .15$ .

or refined to a more accurate physical simulation.

The only differences between strings for this model's implementation are the coefficients for the 4th-order filter and the  $n_w$  parameter. This allows the same tuning values to be used across the other strings. The approach may not be the most physically accurate but is justifiable as the CSG sound is an extremely small component of the total synthesized sound. Other changes to it are extremely difficult to perceive in the total output. Verification was done through the same spectral and perceptual comparison method for the other strings.

## 6.2 Waveform Initialization

A variety of different waveforms and pre-processing techniques can be used to initialize the DWG structure. Several of these are described in [Välimäki and Tolonen 1998] and [Karjalainen, Välimäki, and Tolonen 1998] and the topic could span an entire thesis itself. For the purposes of this implementation, white noise and pink noise are provided as options to initialize the DWG and provide different timbral characteristics to the synthesized tone. Pink noise was used in the [Puputti 2010] implementation while white noise has been used historically [Karplus and Strong 1983]. In either case, it is necessary to pre-process the waveform to remove any DC bias as this can add unwanted artifacts and build up in the DWG model over time.

### 6.2.1 White vs. Pink Noise

Each noise type has different properties regarding its frequency content and spectral makeup. White noise has equal power per frequency interval whereas pink noise has equal power per octave. Accordingly, pink noise is skewed more towards the lower end of the spectrum while white noise contains more energy at the upper end for a specified sampling rate.

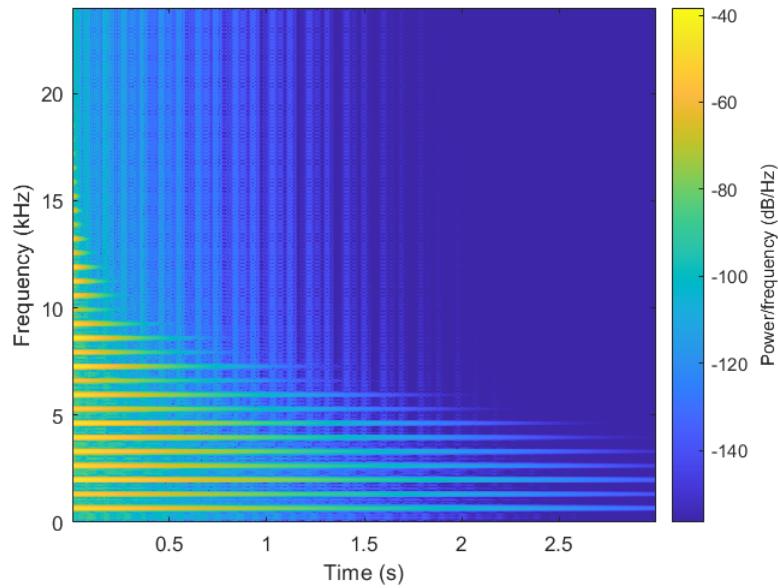
Perceptually, this different frequency content translates into different timbres of the plucked string. White noise generated signal contain more definition/clarity in the attack. This is because it contains the higher frequencies which are necessary to create the sharper transitions associated with the faster transients that create the clarity in the attack. The pink noise generated sounds have more of a “warmer” sound due to their stronger low-frequency content and could be considered more natural sounding. Results from the different noise types can be heard in the files *Pluck-pink-noise.wav* and *Pluck-white-noise.wav*.

### 6.2.2 Removing DC Component

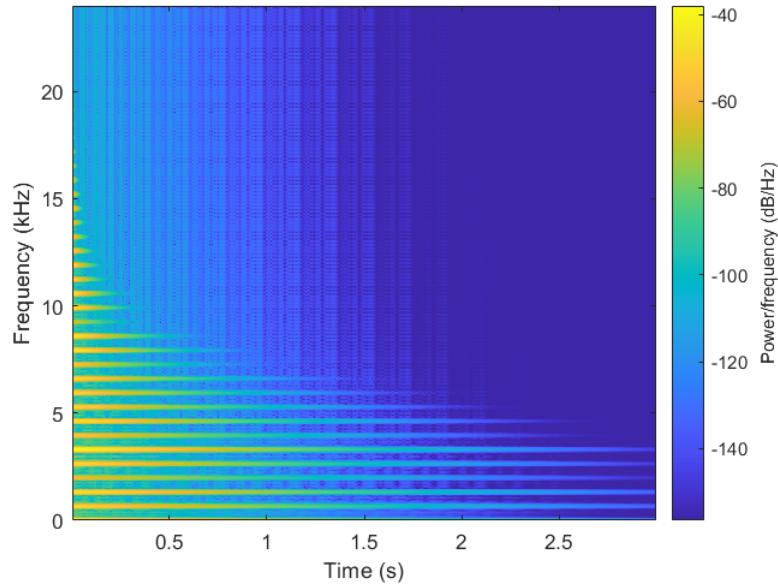
The standard method of removing the DC component from a stored signal is by computing its mean and subtracting this from the waveform. Given that the digital waveguide’s length is distributed across three different components (the integer delay line, the interpolation filter and the loop filter) there is a question of where to store the initial waveform inside the computational structures. Through experimentation it was determined that the integer delay line was the best place to achieve this. Additionally, the input to the loop filter is subject to the effects of the interpolation filter and as the frequency response of both filters are dependent on  $L[n]$ , the process of taking into account their effects when removing the DC bias complicates the matter.

In terms of the actual implementation, a subtle but important distinction to note is the difference between the size of the buffer used to implement the DWG and the length of the DWG itself. Suppose that the data structure for the buffer has been set to have a maximum of 1000 samples but only 250 samples are required for the integer delay line. Only those 250 samples should be considered as those are the only ones which will be used to synthesize the tone. The others will be overwritten as the algorithm computes its output samples due to the feedback connection. Including them in the computations ends up adding an unwanted DC bias as they will influence the computed mean. Contrary to white noise, pink noise is correlated with itself. This property is an additional factor to consider as it results in an initialization signal which is not true pink noise if more samples than necessary are generated.

The files *DC-Bias-Removed.wav* and *DC-Bias-Kept.wav* contain synthesized sounds with and without the DC component removed from the initial waveform. Figures 6.10 and 6.11 illustrate the spectrogram for each of these two samples respectively. The DC component is most easily seen along the bottom axis during the beginning of the sound in Fig. 6.11.



**Fig. 6.10** Spectrogram of a sound synthesized with the DC bias removed from the initial DWG waveform.



**Fig. 6.11** Spectrogram of a sound synthesized with the DC bias kept in the initial DWG waveform. It is most easily identifiable starting in the bottom right corner.

### 6.3 Control Signal Parametrization

A crucial component in any sound synthesis algorithm is the parametrization and generation of its control signals. This section first introduces the basic algorithms to generate  $L[n]$  for different articulations. These results are used to synthesize a musical example of which recordings were also made. The synthesized results are compared to the recorded results in order to understand the differences and limitations of the algorithms described. Musicians have a long history of using equipment outside its intended design. This serves as the creative impetus for the ending discussion of this section where  $L[n]$  is parameterized to purposefully induce an unstable state in the synthesizer.

#### 6.3.1 $L[n]$ Generation

##### Algorithm for Arbitrary Slides

A rudimentary algorithm for generating  $L[n]$  was created that results in the synthesized tone following a linear pitch trajectory by taking into account the fact that the ear hears changes in the fundamental frequency logarithmically [Plack 2018]. Algorithm 1 describes this in more detail. Based on specified starting and stopping points as well as a duration (in seconds), the algorithm calculates the relative change in string length on a per sample basis based on the logarithmic perception aspects of the ear.

---

**Algorithm 1** Generate logarithmic  $L[n]$  from specified end points and duration

---

**Require:**  $L_{init}$ ,  $L_{end}$ ,  $F_s$ ,  $duration$

```

frequencyRatio =  $\frac{L_{init}}{L_{end}}$ 
numSamples =  $F_s \times duration$ 
sampleRatio =  $frequencyRatio^{\frac{1}{numSamples-1}}$ 
n ← 0
for  $n < numSamples - 1$  do
     $L[n] \leftarrow L_{init} \times sampleRatio^{-n}$ 
end for

```

---

## Vibrato

The vibrato signals are generated based on a sinusoidal model which calculates the fret trajectory with the following expression:

$$m_{fret}[n] = A \sin(2\pi f_v T_s n) + C \quad (6.1)$$

$$n = 0, 1, 2, \dots, F_s \times duration - 1 \quad (6.2)$$

$$(6.3)$$

where  $A$  is the vibrato width in semi-tones (or frets),  $f_v$  is the vibrato frequency in Hz,  $C$  is the center fret,  $T_s$  is the sampling period in seconds per sample and  $duration$  is the length of the vibrato in seconds.  $\sin()$  was chosen over  $\cos()$  here as this results in  $L[n]$  starting at the specified

center fret. After computing  $m_{\text{fret}}[n]$ , it is converted to  $L[n]$  using a slightly rearranged version of Eq. 2.13. The result is used to control the synthesis model.

### Notational Slide-In

As will be elaborated upon in the next section, this synthesis model was used to generate sounds corresponding to a musical example which is specified in standard musical notation. This example is shown in Fig. 6.12. To generate the *slide-in* articulations, a short segment is generated following Alg. 1. For the algorithm’s parameters, the *duration* corresponds to a 16th note for the specified tempo,  $L_{\text{end}}$  is the fret corresponding to the target note, and  $L_{\text{init}}$  is one fret below  $L_{\text{end}}$ .

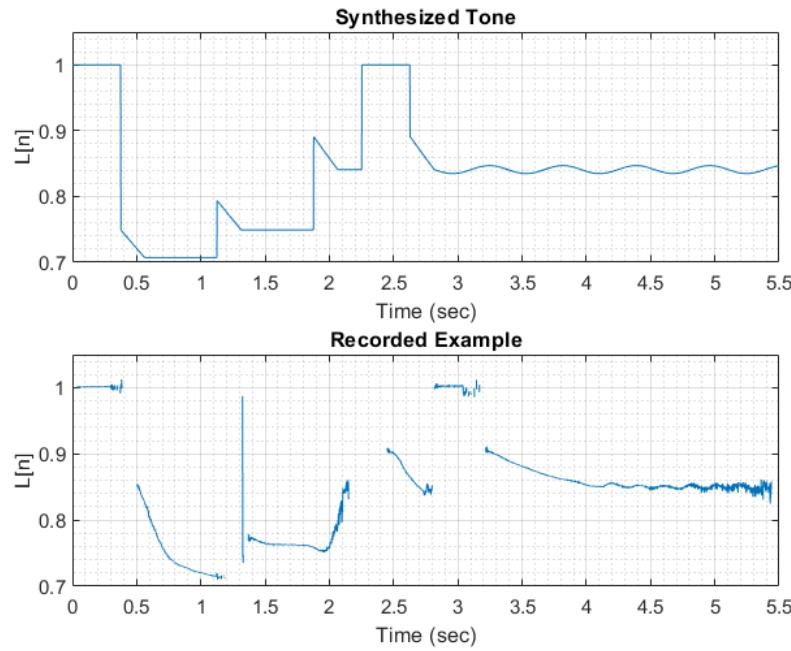
#### 6.3.2 Comparison to Observed $L[n]$

The musical example in Fig. 6.12 was both synthesized and recorded to facilitate a comparison between the synthesized and observed  $L[n]$  signals. For the recorded example,  $F_0[n]$  was first estimated using the YIN algorithm [Chevigné and Kawahara 2002] from which  $L[n]$  was derived based on Eq. 2.6. The results are shown in Fig. 6.13. Only the segments in which the YIN algorithm was extremely confident are plotted, hence the discontinuities in the graph. It tended to fail during note-onset events. Although a bit of noise exists in the computed signal, more general qualitative comparisons between the two plots can still be made. The sounds being compared can be heard in *SlideLick-High-e-Rec.wav* and *SlideLick-High-e-Synth.wav*.



**Fig. 6.12** The slide playing example for the high E string in music notation. This was transposed accordingly for all the other strings when recorded and synthesized.

As can be both seen and heard, the  $L[n]$  trajectories which Alg. 1 produce do not sound quite as nuanced as what is captured in the recorded example. The slopes of the transitions from the recorded example are not constant as they appear in the synthesized tone’s transitions. Visually and audibly the recording has more inflection points and the duration of each “approach” slide is not fixed as in the synthesized example. The second note of the recorded example also includes a slide downwards as it decays. With regards to the vibrato, there is much more of a delay after the initial attack before it is initiated in the recording. Nor is its frequency constant. The synthesized tone transitions much more directly into its vibrato, and maintains the same frequency and amplitude while the note decays (as it was programmed to do). In general, the synthesized example is missing quite a bit of the nuance of the recording. It would be a substantial task in itself to generate mathematical expressions for the recording’s subtle details and translate these into an easily parameterized  $L[n]$  generation algorithm. In summary, the synthesized results are a bit more robotic and comparatively less expressive than the recorded examples.



**Fig. 6.13** The  $L[n]$  used to synthesize musical example and the  $L[n]$  computed from a recording of the same example using the YIN algorithm. Some noise exists due to limitations of the YIN algorithm, however general characteristics can still be compared.

#### 6.4 Comparing Synthesized Results vs. Recordings

An important test of a physical model synthesizer is to compare synthesized sounds to recordings of the same musical example. The following section aims to do this both subjectively and objectively. There are many aspects involved in the recording of a physical slide guitar which are missing from the synthesized model (e.g. the body effects, room reverberation). Attempts were made to mitigate some of these aspects in the measurement setup (e.g. using a hemi-anechoic chamber to reduce reverberation), however, they were not perfect. Given these limitations, it would be unreasonable to expect the results synthesized here to be perfect. Even so, the comparison still provides useful insight into the operation of the model and slide guitar in general.

The measurement setup used to make the comparison recordings is very similar to the one used to measure the Contact Noise Spectrum (see Sec. 5.4.1). The difference is a reversal of the transverse vibration muting scheme. In this scenario, the string being played was unmuted. All the other strings were muted using the right-hand, closer to the bridge. This was done as it is the standard method to mute sympathetic vibrations while playing.

One of the most prominent components missing from the synthesized sounds are the non-musical interaction sounds between the player and instrument generated by touch and collision. These include sounds like the initial contact the slide makes with the string as well as sound introduced by the displacement of the string when a new note is plucked. Restoring this displacement

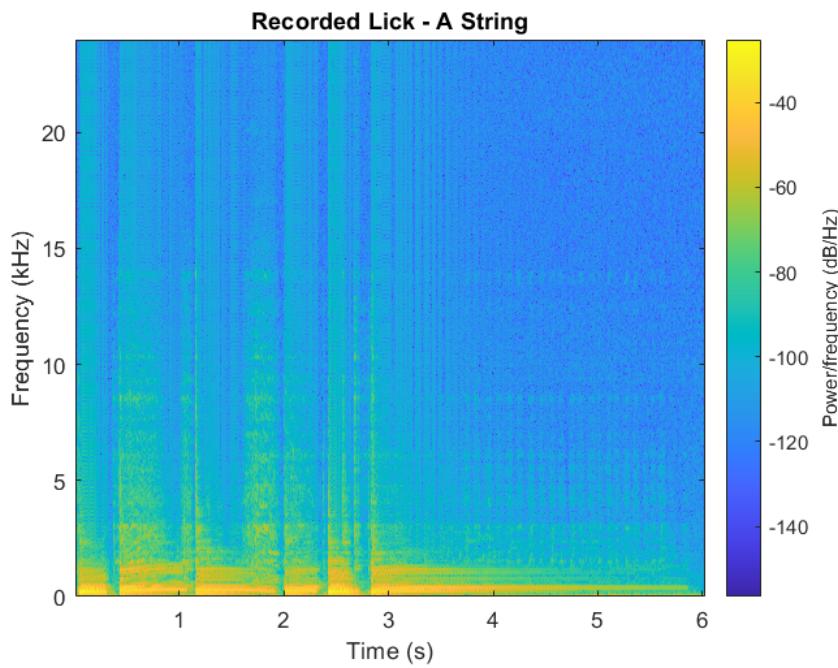
can cause a percussive contact sound to be generated as heard in *SlideLick-E-Rec.wav* in the note attacks. Compare this to the synthesized sound in *SlideLick-E-Synth.wav*. It is entirely expected that these sounds are not included in the synthesized output as the model does not address them. Gianpaolo Evangelista addresses this in the slide model he proposes [Evangelista 2012]. It uses ideas adapted from his previous work in [Evangelista and Eckerholm 2010] which focuses purely on the touch and collision sounds generated while playing guitar.

Another point to note is that the feedback introduced by the longitudinal-to-transverse coupling (see Sec. 5.3) does not sound the same in the synthesized results as compared to the recordings. This is most prominent once the transverse vibrations have decayed substantially and the contact sounds dominate. Compare the sounds in *SlideLick-B-Rec.wav* and *SlideLick-B-Synth.wav* near the end of the files. There is a more “digital” quality to the feedback which appears towards the end of the synthesized sound. This suggests that there is more work to be done in developing a computational model for the longitudinal-to-transverse coupling.

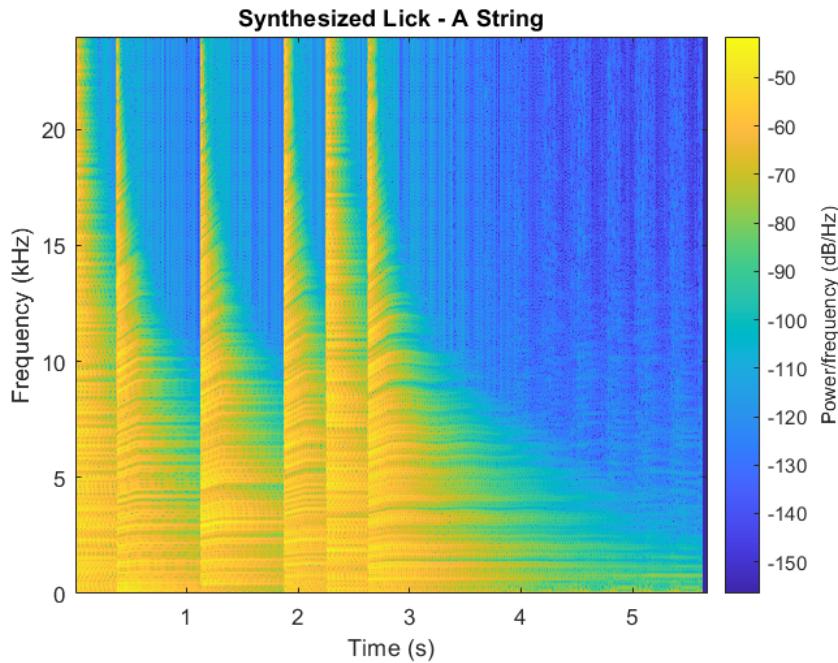
In general the synthesized sounds from the three upper strings (G, B and high E) sound closer to their physical analogs than the three lower strings (low E, A and D) when purely considering the string model. Part of this has to do with the rate at which the harmonics decay. In the recordings, the upper harmonics die out much more rapidly or are not present. Figures 6.14 and 6.15 illustrate the differences. This is consistent with the fact that the recorded guitar was equipped with older strings (see Sec. 5.1). As the strings have decayed they do not support energy in the higher overtones as compared to fresh strings. The loop filter could be modified to take this into account and provide more realistic modeling as it is where the physical losses are modeled in the system.

The friction sounds from the wound CSG do not always perceptually fuse to the transverse vibrations in the same manner as in the recordings. There are instances where they are perceived like a separate acoustic event superimposed on top of the transverse vibrations. The sounds in *SlideLick-A-Rec.wav* and *SlideLick-A-Synth.wav* illustrate this. There almost seems to be a slight delay between when the  $f_0$  changes (corresponding to the start of the slide movement) and when the contact sounds are heard. As the smoothing filters are the same length for both  $L[n]$  and  $slideSpeed[n]$  (see Fig. 3.2) both control signals would incur the same processing delay. This suggests that refinement needs to be made in the mechanism which scales the wound contact sound generator’s output. The more advanced friction model introduced by Evangelista in [Evangelista 2012] supports this.

Similar to the observations made earlier in Sec. 6.3.2, the unnaturalness of the algorithmically generated  $L[n]$  signal has ramifications on the contact sounds as well as the transverse vibration sounds. The idealizations in the vibrato sections at the end of the musical examples cause the synthesized contact sounds to be rather extreme as compared to their recorded counterparts. The contact sounds are periodic to a level of precision a human could not replicate and differ substantially from what is captured during the vibrato sections in the recordings. See *SlideLick-D-Synth.wav* and *SlideLick-D-Rec.wav* for reference. Based on Eq. 6.1 due to the constant  $f_v$ , this is entirely expected.  $f_v$  could be made a time-dependent quantity to allow for more expressive control signals.



**Fig. 6.14** The spectrogram of the musical example played on the A-string of a physical guitar. Note the lower harmonic content as compared to Fig. 6.15.



**Fig. 6.15** The spectrogram of the synthesized using the A-string configuration of the model. Note the higher harmonic content as compared to Fig. 6.14.

## 6.5 Purposeful Instability

Guitarists, and musicians in general, have a long history of using equipment in a fashion which goes beyond its original design. Distortion is an example of this. From an engineering standpoint, it is generally taught that distortion is not desirable and is often defined as “an unwanted component in the output signal”. Since its initial “discovery”, distortion has become an integral part of the tone of an electric guitar to the point where amplifiers are designed to purposely create it in a musical way. The following experiment was done in a similar vein of where the synthesis model is pushed into an unstable (and initially undesirable) state to see the produced output. This has no connection to slide guitar specifically, it is simply meant to be a creative parametrization of the synthesis model which “breaks the rules” of its design. The ultimately usability of this would be determined by a musician using the synthesizer as they are the final arbiters of the musicality of a sound.

As noted in the discussion in Sec. 3.2.1, it is possible to adjust  $L[n]$  in such a fashion that the loop filter has positive gain at certain frequencies. Given its placement in a feedback loop, this could put the synthesis model in an unstable state. In order to ensure the system is unstable, another condition needs to be met. It would be necessary for the attenuation provided by the interpolation filter to be less than the amplification of the loop filter so the net loop gain would be positive. The integer delay component has unity gain so it does not need to be considered here. A Lagrange interpolation filter will have unity gain when it does not need to provide any interpolation on the waveform in the DWG. This occurs when the fractional component of the total digital waveguide length can be achieved via the phase delay of the loop filter.

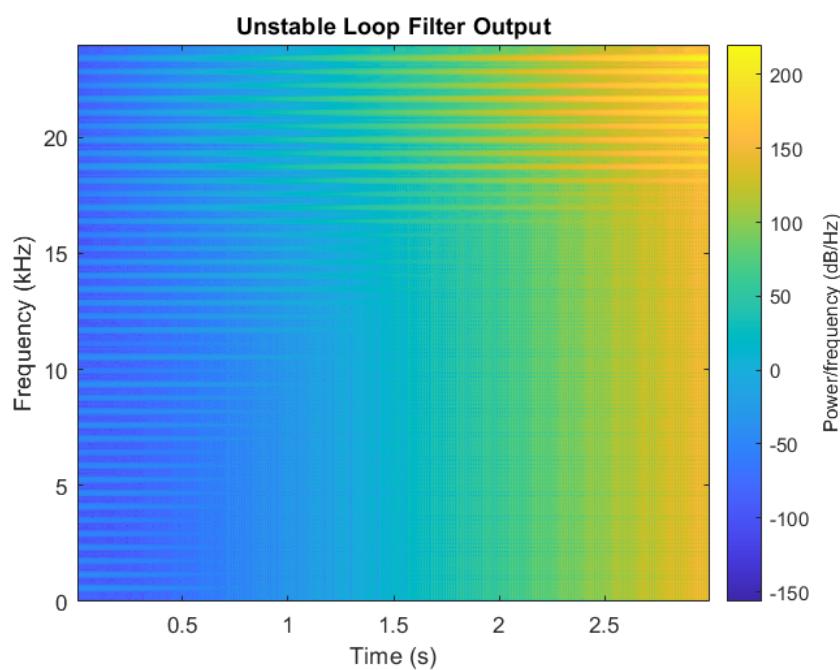
The relative string length can be expressed as:

$$L = \frac{\text{OpenString}_{F_0}}{F_s} \times \text{DWGLength} \quad (6.4)$$

Given an open-string fundamental frequency and sampling rate, only the appropriate digital waveguide length needs to be selected. For the D-string running at 48kHz and using an approximation of 0.25 for the loop filter’s phase delay, the following calculation produces an unstable system without going beyond the 24th fret (keeping it in the range of many electric guitars):

$$L = \frac{146.83}{48,000} \times 82.25 \approx 0.2516 \quad (6.5)$$

The results of this system are illustrated in Fig. 6.16. This clearly shows how the upper frequencies are amplified over time as the system maintains an unstable state. Based on the magnitude responses shown in Fig. 3.13, this is expected behavior. Two output files exist for this example: *UnstableOutput-scaled.wav* and *UnstableOutput-clipped.wav*. In order to be able to save the output without clipping, it was necessary to scale/normalize it first. However, the scaling was so drastic due to the extremes in signal values that the output does not contain much useful information. In a real-time system, the instability would likely manifest as clipping so a file was produced where the clipping was allowed to occur. Ultimately the musicality of this sound is subject to the creative desires of the composer utilizing it, however the harsher and more chaotic sounds of the clipped example could find use in electronic or industrial music genres.



**Fig. 6.16** The spectrogram of the sound synthesized after the model has been put in an unstable state using the  $L$  value from Eq. 6.5.

## Chapter 7

# Conclusion and Future Work

### 7.1 Conclusion

This thesis served the purpose of developing a physically informed synthesis model for the slide guitar. The model is based on the design described in [Pakarinen, Puputti, and Välimäki 2008] and implemented in [Puputti 2010]. These papers also contain an emphasis on the gestural control aspects and comparatively some of the digital signal processing aspects are neglected. The thesis has a secondary goal of extending and elucidating the digital signal processing techniques related to the implementation as the previously mentioned papers were focused on other topics.

First, the slide guitar was described, emphasizing how it differs from the traditional fretted playing style in that the slide acts as a moving string termination. This allows for a continuum of pitches and nuanced articulations to be produced (including vibrato). The basics of digital waveguide modeling and acoustics were introduced to be able to explain how they can be used to create a computational model of a transversely vibrating string. This model was then adapted to support emulation of the slide guitar. An overview to the handful of more recent approaches related to slide guitar sound synthesis was then described, including comparisons between the different approaches.

After the basics were introduced in Chapter 2, a more detailed explanation of the slide guitar model's architecture was explained in Chapter 3. This included a digital waveguide model which allows for a time-varying pitch and more precise tuning via delay-line interpolation. Also included is a Contact Sound Generator to emulate the interactions between the slide and string surfaces. The explanation of the model included a discussion of the limitations inherent in its design. Modifications to the inspiration model include a gain coefficient to control the longitudinal-to-transverse coupling as well as the addition of the Noise Burst Generator and Harmonic Resonator Bank as potential components to increase the creative potential of the synthesizer. The full details of implementation for each component as well as audio examples are provided to develop a more intuitive understanding of each component. This is an extension to the initial model descriptions.

Subsequently in Chapter 4, the verification techniques used to ensure the correct operation of the model and its constituent components were discussed and tested. The results of Chapter 4 illustrated that the various constituent components (e.g. filters, delay lines, string digital waveguides, etc.) worked correctly as well as the single string slide synthesizer as a whole. This is a

substantial improvement over the results presented in [Puputti 2010] as there is comparatively little objective evidence indicating the implementation works correctly.

A discussion then followed in Chapter 5 regarding the physical measurements performed to help verify the model's accuracy as well as refine its implementation. Experiments which prove the existence of the longitudinal-to-transverse coupling as well as attempt to measure  $T_{60}$  for the wound strings are included. The sound design and tuning aspects which these measurements informed were then discussed in addition to some subtle implementation details. This information is not covered in [Pakarinen, Puputti, and Välimäki 2008] or [Puputti 2010]. There is a strong physical foundation which informs the design of the synthesizer.

In Chapter 6, aspects related to the parametrization of the control signal and synthesis parameters are discussed. The best method for configuring the Contact Sound Generator to generate realistic sounds is described. The control signal is examined in terms of the qualities which generate realistic playing examples as well as how to purposefully induce an unstable state in the synthesis model for creative purposes. There is also a comparison between the synthesized and recorded results of the same musical example. Minimal discussion of these topics occurs in [Pakarinen, Puputti, and Välimäki 2008] and [Puputti 2010]. In summary, the model itself produces usable results with similar spectral characteristics to recordings but lacks the nuances associated with playing a physical instrument.

In summary, this thesis implements and expands on the model introduced and implemented in the aforementioned papers. More precise implementation details are described and limitations of the model are examined. Physical measurements have been completed to refine and verify the model's accuracy. Guiding principles behind configuration of the parameters and control signals for sound design are discussed. Auditory examples have been provided throughout to help illustrate the different concepts.

## 7.2 Future Work

There remain many opportunities for improvement of the model. In terms of improving its constituent components the impulse train generator could be modified to a Band-Limited Impulse Train. This would support the generation of impulse signals which are not quantized based on an integer number of sample periods and overcome a current limitation. The Wound CSG could be modified to change the harmonic strengths based on the slide speed to more accurately model the measurements made on the physical system.

In terms of the loop filter, two obvious refinements could be made. Given that the phase delay values at different string lengths are not constant, this could be made more accurate by using the mean of the phase delay across the spectrum instead of a constant approximation. To help reduce the computational complexity and facilitate a real-time implementation, a look-up table based on  $L[n]$  could be developed for each string based on these average values. The loop filter could also be redesigned so that the magnitude response always stays below unity beyond the 19th fret. This would extend the range of usable  $L[n]$  values to incorporate situations which are likely occur during slide playing. Another approach could be investigating the bridge and body of the instrument to account for the coupling between strings. This could be incorporated into a modeling filter which could help provide a more realistic sound.

Many different aspects of the physical measurements could also be refined. The coupling and spectral differences between fresh and older corroded strings could be measured and incorporated into the model to provide more timbral options. An experimental design to capture the contact sound generated by the unwound strings could be developed given the muting issues which occurred. The setup used to measured the  $T_{60}$  for each string has a lot of opportunity for improvement and a more accurate  $T_{60}$  would help illustrate deficiencies in the Wound CSG model. Also, the longitudinal-to-transverse measurements could be made more quantitative and these could be used to help refine the coupling model. A filter, as opposed to a scaling coefficient, would likely be a better candidate given the coupling is likely influenced by the impedance of the string termination at the bridge. All the experiments could benefit from a physical setup where the guitar was guaranteed to be decoupled from its supporting structure (or the coupling was quantified).

# Bibliography

- Bank, Balazs (Jan. 2006). "Physics-Based Sound Synthesis of String Instruments Including Geometric Non-Linearities". PhD Thesis. Budapest University of Technology and Economics. URL: <http://home.mit.bme.hu/~bank/phd/>.
- Bhanuprakash, Abhiram, Paul Stapleton, and Maarten van Walstijn (Sept. 2020). "A Finite Difference Model for Articulated Slide-String Simulation". In: Vienna: DAFx. URL: [http://dafx.de/paper-archive/2020/proceedings/papers/DAFx2020\\_paper\\_68.pdf](http://dafx.de/paper-archive/2020/proceedings/papers/DAFx2020_paper_68.pdf).
- Cheveigne, Alain de and Hideki Kawahara (2002). "YIN: A Fundamental Frequency Estimator for Speech and Music". In: *The Journal of the Acoustical Society of America* 111.4, pp. 1917–1930. ISSN: 0001-4966.
- Erlewine, Dan. (2001). *How to Make Your Electric Guitar Play Great! : The Electric Guitar Owner's Manual*. English. A Guitar Player Book. Section: 133 pages. San Francisco, CA: Backbeat Books ; ISBN: 978-0-87930-601-4.
- Evangelista, Gianpaolo (Apr. 2012). "Physical Model of the Slide Guitar: An Approach Based on Contact Forces". In: *Audio Engineering Society Convention 132*. URL: <http://www.aes.org/e-lib/browse.cfm?elib=16228>.
- Evangelista, Gianpaolo and Fredrik Eckerholm (2010). "Player-Instrument Interaction Models for Digital Waveguide Synthesis of Guitar: Touch and Collisions". In: *IEEE Transactions on Audio, Speech, and Language Processing* 18.4, pp. 822–832. DOI: [10.1109/taslp.2009.2038822](https://doi.org/10.1109/taslp.2009.2038822).
- Jaffe, David A. and Julius O. Smith (1983). "Extensions of the Karplus-Strong Plucked-String Algorithm". In: *Computer Music Journal* 7.2, pp. 56–69. DOI: [10.2307/3680063](https://doi.org/10.2307/3680063).
- Karjalainen, Matti, Vesa Välimäki, and Tero Tolonen (1998). "Plucked-String Models: From the Karplus-Strong Algorithm to Digital Waveguides and Beyond". In: *Computer Music Journal* 22.3. Publisher: The MIT Press, pp. 17–32. ISSN: 01489267, 15315169. DOI: [10.2307/3681155](https://doi.org/10.2307/3681155). URL: <http://www.jstor.org/stable/3681155> (visited on 11/12/2022).
- Karplus, Kevin and Alex Strong (1983). "Digital Synthesis of Plucked-String and Drum Timbres". In: *Computer Music Journal* 7.2, pp. 43–55. DOI: [10.2307/3680062](https://doi.org/10.2307/3680062).
- Laakso, Timo I., Vesa Valimaki, Matti Karjalainen, and Unto K. Laine (1996). "Splitting the Unit Delay: Tools for Fractional Delay Filter Design". In: *IEEE Signal Processing Magazine* 13.1. ISSN: 1053-5888. DOI: [10.1109/79.482137](https://doi.org/10.1109/79.482137).
- Morse 1903-1985., Philip M. (Philip McCord) (1981). *Vibration and Sound*. English. Section: xix, 468 pages : illustrations ; 23 cm. [New York]: American Institute of Physics for the Acoustical Society of America. ISBN: 978-0-88318-287-1.

- Oppenheim, Alan V. 1937- and Ronald W. 1938- Schafer (2010). *Discrete-Time Signal Processing*. English. Third edition. Prentice Hall signal processing series. Section: xxviii, 1108 pages : illustrations ; 25 cm. Upper Saddle River, NJ: Pearson. ISBN: 978-0-13-198842-2.
- Oppenheim, Alan V. 1937-, Alan S. Willsky, and Syed Hamid Nawab (1997). *Signals & Systems*. English. 2nd ed. Prentice-Hall Signal Processing Series. Section: xxx, 957 pages : illustrations ; 25 cm. Upper Saddle River, N.J.: Prentice Hall. ISBN: 978-0-13-814757-0.
- Pakarinen, Jyri, Matti Karjalainen, Vesa Valimaki, Stefan Bilbao, and PA 2005; Mar International conference Philadelphia (2005). “Energy Behavior In Time-Varying Fractional Delay Filters For Physical Modeling Synthesis Of Musical Instruments”. In: *ICASSP*. IEEE; 2005, pp. 4–4. ISBN: 0-7803-8874-7.
- Pakarinen, Jyri, Henri Penttinens, and Balazs Bank (2007). “Analysis of Handling Noises on Wound Strings”. In: *The Journal of the Acoustical Society of America* 122.6. DOI: 10.1121/1.2786607.
- Pakarinen, Jyri, Tapio Puputti, and Vesa Välimäki (2008). “Virtual Slide Guitar”. In: *Computer Music Journal* 32.3, pp. 42–54. DOI: 10.1162/comj.2008.32.3.42.
- Pakarinen, Jyri, Vesa Välimäki, and Tapio Puputti (June 2008). “Slide Guitar Synthesizer with Gestural Control”. In: *NIME*.
- Penttinens, Henri, Jyri Pakarinen, Vesa Välimäki, Mikael Laurson, Henbing Li, and Marc Leman (2006). “Model-Based Sound Synthesis of the Guqin”. In: *The Journal of the Acoustical Society of America* 120.6, pp. 4052–4063. DOI: 10.1121/1.2360422.
- Plack, Christopher J. (2018). *The Sense of Hearing*. 3rd. New York, NY: Routledge. ISBN: 978-1-315-20814-5.
- Puputti, Tapio (May 2010). “Real-Time Implementation of the Virtual Air Slide Guitar”. MA thesis. Aalto University. URL: <https://aaltodoc.aalto.fi/handle/123456789/3258>.
- Scavone, Gary P. (2023a). *A Digital Delay Line*. URL: <https://www.music.mcgill.ca/~gary/618/week1/node15.html> (visited on 03/31/2023).
- (2023b). *Digital Waveguide Models*. URL: <https://www.music.mcgill.ca/~gary/618/week4/node13.html> (visited on 03/31/2023).
- (2023c). *Infinite Impulse Response (IIR) Filters*. English. URL: <https://www.music.mcgill.ca/~gary/618/week1/node5.html>.
- (2023d). *Sampled Traveling Waves*. URL: <https://www.music.mcgill.ca/~gary/618/week4/node12.html> (visited on 03/31/2023).
- Smith, Julius O. (2023a). *Physical Audio Signal Processing*. Online 2010. CCRMA. (Visited on 01/03/2023).
- (2023b). *Spectral Audio Signal Processing*. Online 2011. CCRMA. (Visited on 01/03/2023).
- Välimäki, Vesa (Dec. 1995). “Discrete-Time Modeling of Acoustic Tubes Using Fractional Delay Filters”. English. PhD Thesis. Espoo, Finland: Helsinki University of Technology. URL: [http://users.spa.aalto.fi/vpv/publications/vesa\\_phd.html](http://users.spa.aalto.fi/vpv/publications/vesa_phd.html) (visited on 01/05/2023).
- Välimäki, Vesa and Tero Tolonen (1998). “Development and Calibration of a Guitar Synthesizer”. In: *J. Audio Eng. Soc* 46.9, pp. 766–778. URL: <http://www.aes.org/e-lib/browse.cfm?elib=12131>.

## Appendix A

# String Winding Measurement Photos



**Fig. A.1** Photo of the low E string winding count for 1 cm



**Fig. A.2** Photo of the A string winding count for 1 cm



**Fig. A.3** Photo of the D string winding count for 1 cm

## Appendix B

# Code and Sound Examples

All the source code and sound examples can be found at the following GitHub repo:  
<https://github.com/dgsmith1988/Masters-Thesis>