

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

April 2023

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

© 2023 David G. Smith

Contents

1	Introduction	1
1.1	Motivation	1
1.2	Thesis Organization	2
2	Background	5
2.1	What is physical modeling and digital waveguides?	5
2.1.1	Fundamental Components	5
2.1.2	Applied to String Modeling	7
2.2	Development of Slide Guitar Model	10
2.2.1	Energy Compensation	11
2.2.2	Loop Filter	12
2.3	CSG Development	12
2.3.1	Contact Sound Analysis	13
2.3.2	Adapting From Finger to Slide	16
2.3.3	Model Description	16
2.4	Recent Developments and Approaches	18
2.4.1	Gianpaoalo's DWG Approach	18
2.4.2	Belfast Finite-Difference Technique	20
3	Description of Slide Guitar Synthesis Model	21
3.1	Introduction of Architecture	21
3.1.1	Diagram Conventions	21
3.1.2	Single String Slide Synthesizer	21
3.1.3	Control Signal Processor	22
3.1.4	String DWG	23
3.1.5	Contact Sound Generator	24
3.2	Limitations of Model	27

3.2.1	Minimum Relative String Length	27
3.2.2	Non-constant Phase Delay of Filters	31
3.2.3	Impulse Train Implementation	31
4	Verification Of Slide Model and Constituent Components	36
4.1	Filters	36
4.1.1	Resonator	36
4.1.2	Longitudinal Modes	36
4.1.3	DC Blocker	39
4.1.4	Smoothing Filter	39
4.2	CSP and Components	39
4.2.1	Components	39
4.2.2	Control Signal Processor	41
4.2.3	Loop Filter	41
4.3	Noise Generation Objects	46
4.3.1	Impulse Train	46
4.3.2	Exponential Decay	46
4.3.3	Noise Pulse Train	48
4.3.4	Noise Burst Generator	50
4.4	Harmonic Accentuators	52
4.4.1	Harmonic Resonator Bank	52
4.4.2	Reso + Tanh	52
4.5	Contact Sound Generator	53
4.5.1	Wound Variant	53
4.5.2	Unwound Variant	58
4.6	String DWG and Components	58
4.6.1	Interpolated Delay Line	58
4.6.2	Energy Scaler	62
4.6.3	String Digital Waveguide	65
4.7	Slide Synthesizer	68
5	Physical Measurements	70
5.1	Experimental Setup	70
5.2	String Winding Density	71
5.3	Transverse to Longitudinal Coupling	71
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
5.5	Decay Rate of Single Winding Impact	76
5.5.1	Methods	76
5.5.2	Results	77
5.6	Playing Examples	77
6	Sound Design and Control Signals	79
6.1	Alternate Architecture Considerations	79
6.2	Tuning Parameters	79
6.2.1	Decay Rate	79
6.3	CSG Configurations	80
6.3.1	Computational Complexity	80
6.4	Waveform Initialization	81
6.4.1	Removing DC Component from String DWG	81
6.5	Control Signal Parametrization	82
6.5.1	Generating L[n]	82
6.5.2	Analysis of Signals from Playing Examples	82
6.5.3	Purposeful Instability	84
7	Conclusion	88
7.1	Future Work	88
A	String Winding Measurement Photos	91
B	Class Hierarchy	92
C	Code and Sound Examples	93

List of Acronyms

AES	Audio Engineering Society
CMJ	Computer Music Journal
CSG	Contact Sound Generator
DFT	Discrete Fourier Transform
DWG	Digital Waveguide
FIR	Finite Impulse Response
IIR	Infinite Impulse Response
IR	Impulse Response
LPC	Linear Predictive Coding

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. Additionally, the interaction of the slide’s surface with that of the string adds a new timbral component related to the slide’s velocity [Pakarinen, Puputti, and Välimäki 2008]. 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

In the case of wound strings, this adds two new sounds. 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 1.2 shows a close-up illustration of a slide interacting with a wound string.

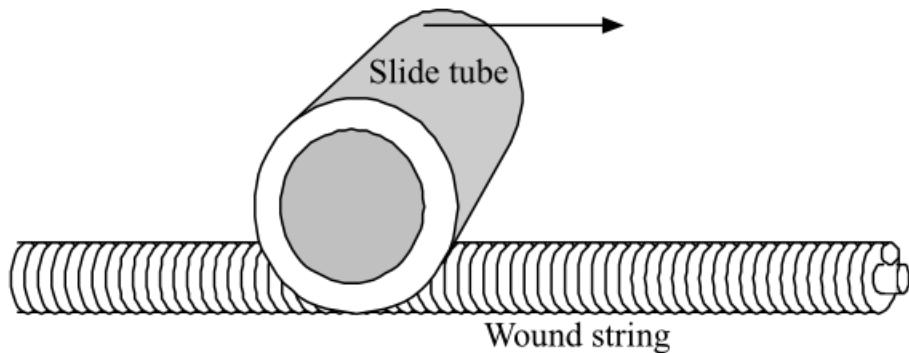


Fig. 1.2 Close up of a slide on wound string [Pakarinen, Puputti, and Välimäki 2008]

Slides are traditionally made from ceramic or metal and unwound strings are made with metal [Bhanuprakash, Stapleton, and Walstijn 2020]. These are smooth/polished materials as can be seen in Fig. 1.3. Correspondingly, the coefficient of friction between the string and slide is comparatively much lower than in the wound-string case. Unwound strings also have a uniform surface, lacking the ridges created by windings which a slide impacts while traveling the length of the string. 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 scaled by the slide velocity and lacks a harmonic component. [Pakarinen, Puputti, and Välimäki 2008].

The goal of this project is to explore the model which was introduced in Pakarinen, Puputti, and Välimäki 2008 more fully than was described from a digital signal processing standpoint. 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.

1.2 Thesis Organization

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



Fig. 1.3 A selection of slides made from different materials, all of which are comparatively smooth.

Chapter 1 - Introduction

This section 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 resources if an interested reader prefers to get 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 provide 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 we will describe the physical experiments which were performed during the development of this model. 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

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 we give an introduction to the theoretical framework of digital waveguides which form the basis for the synthesis model developed in this thesis. After these ideas 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 What is 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. It is an extremely computationally efficient technique which has been incorporated 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 delay lines. The interconnection of these different components models a variety of wave propagation phenomena. Through the incorporation of different noise sources and initialization waveforms, an enumerable number of different sound synthesis algorithms can be created.

Digital Filters

Conceptually digital filters are extremely simple devices. They merely add scaled time-delayed versions of their inputs and outputs. Through various combinations of delay amounts and scaling, a range of different frequency domain effects can be achieved. They are linear systems themselves, which is useful as the entire branch of linear-time invariant (LTI) systems is now made available to the algorithm designer.

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 as theoretically the feedback line makes it so the filter’s output in response to an impulse signal would never end (ignoring the 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 the input changes from steady-state. FIR filters have a transient whose length corresponds to the length of the filter. IIR filters pose more of a problem with regards to transients as their feedback lines 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.

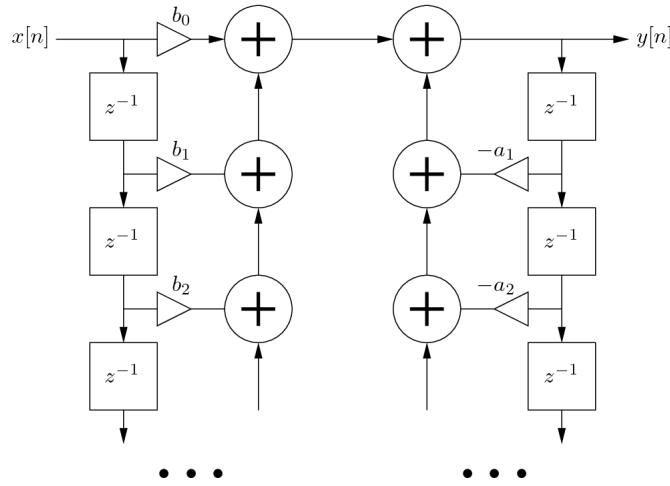


Fig. 2.1 A generic digital filter structure. The b_n coefficients refer to the FIR part while the a_n coefficients refer to the IIR part [Scavone 2023c].

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 medium in which the wave is traveling. The digital 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, c is the wave propagation speed in the particular media being modeled [Scavone 2023d]. Figure 2.2 illustrates a pictorial 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. The fact that the fundamental unit of discretization in the time-domain is the sample. This means that signals can only be delayed-modeled using integer numbers of samples with this structure. In many physical modeling applications, this is a problematic due to the fact that the physical world and its associated laws are not spatially quantized. It is often required to know physical quantity which lies between two spatial samples.

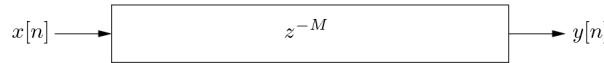


Fig. 2.2 Pictorial representation of a digital delay component which is M samples in length [Scavone 2023a].

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]. These approaches can be referred to as delay line 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 done 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 the Lagrange interpolation. In this technique an FIR filter is used where the filter coefficients implement Largange 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:

$$h[n] = \prod_{k=0}^N \frac{D - k}{n - k}, \text{ for } n = 0, 1, \dots, N \text{ and } k \neq n \quad (2.2)$$

The order of the filter determine 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 scenarios where the phase delay is constant and how the filter has low-pass characteristics.

2.1.2 Applied to String Modeling

A vibrating string which has been put into motion through a plucking excitation has a solution to the wave equation which consists of two traveling waves moving in opposite directions and being reflected at each of the different string terminations. This is d'Alembert's solution to the wave

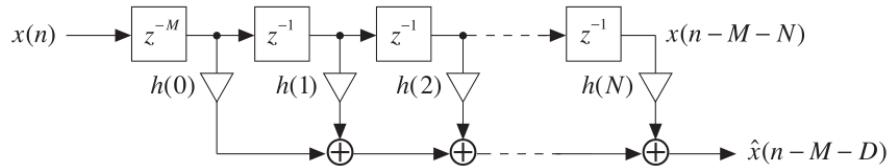


Fig. 2.3 An FIR structure implementing Lagrange interpolation in series with an M sample delay line [Välimäki 1995]

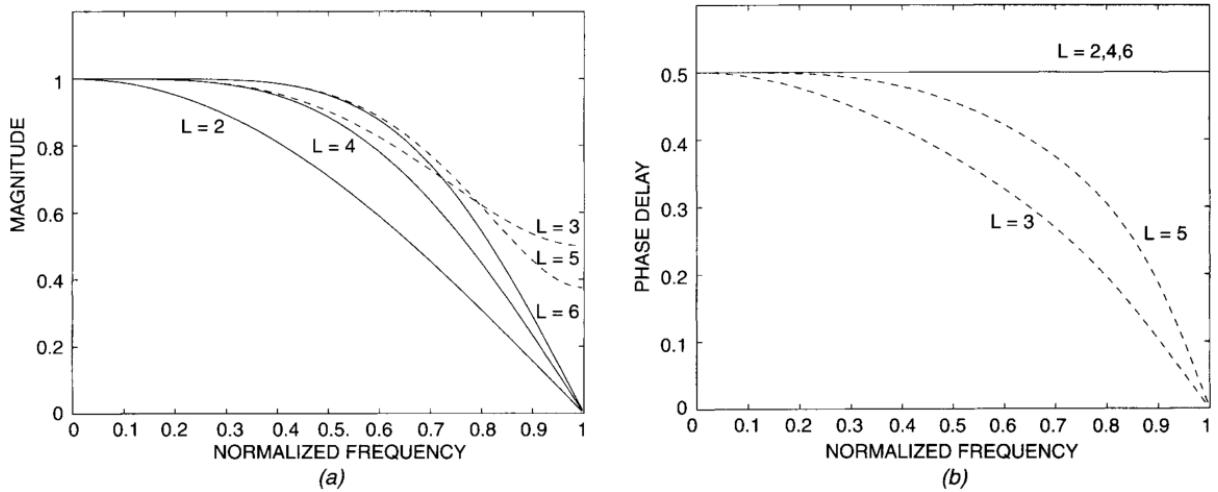


Fig. 2.4 a) Magnitude and b) phase delay responses of Lagrange interpolating filters of length $L = 2, 3, 4, 5$, and 6 with $d = 0.5$ [Laakso et al. 1996]

equation [Smith 2023a]. 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 of the end terminations. The traveling wave variables could be any physical quantity related to the string movement (i.e. displacement, acceleration, velocity...) depending on what is desired for the synthesis model, however transverse displacement is commonly used for string models. In combination with the laws of physics and other physical quantities, like the characteristic impedance, it is easy to calculate the other physical aspects which are needed. Figure 2.5 illustrates a signal flow diagram for a bi-directional string DWG.

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 digital delays 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. The bridge is commonly used as the output location for string models.

By exploiting the linearity of the systems involved, as well as the fact that is generally only desired to know the output at one specific spatial location, it is possible to commute the different

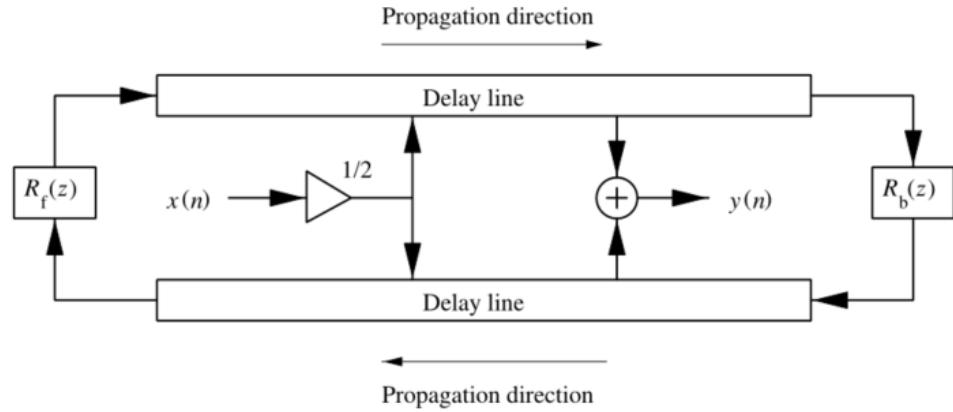


Fig. 2.5 A bi-directional DWG structure for a string terminated at both ends [Karjalainen, Välimäki, and Tolonen 1998]

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 $H_l(z)$. Additionally, it is no longer required to simulate both traveling waves so only delay line is required (as the structure's name would imply).

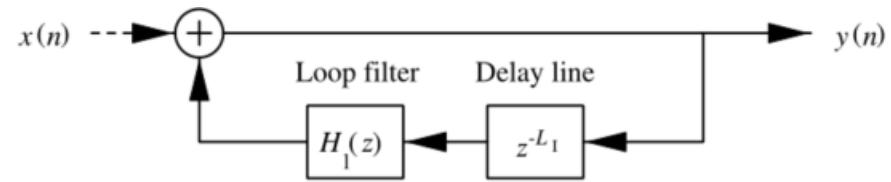


Fig. 2.6 A single delay line DWG structure for a string [Karjalainen, Välimäki, and Tolonen 1998]

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 of the pitch will be scaled inversely proportional to the relative string length [Välimäki and Tolonen 1998]. If L represents a the relative string length, which exists on the interval $(0, 1]$, the fundamental at a specified relative string length can be expressed as:

$$F_L = \frac{F_0}{L} \quad (2.3)$$

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:

$$\text{DWG Length} = \frac{F_s}{F_L} \quad (2.4)$$

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 wave guide 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 breakdown 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:

$$\text{DWG Length} = \tau_{H_L} + L_f + L_I \quad (2.5)$$

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 the loop filter and fractional approximation method used in the model.

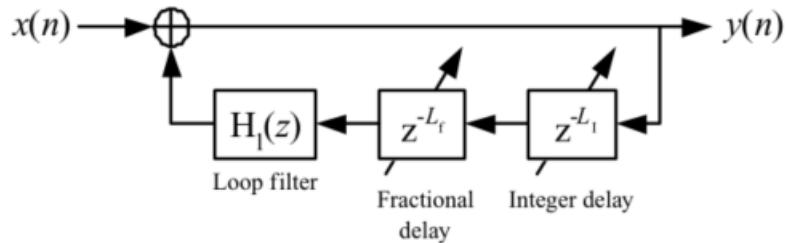


Fig. 2.7 SDL expanded [Karjalainen, Välimäki, and Tolonen 1998]

2.2 Development of Slide Guitar Model

The slide guitar model which serves as the basis for this thesis originally comes from a Computer Music Journal article [Pakarinen, Puputti, and Välimäki 2008]. The basis for this model is the

SDL structure with additional components to model the slide/string surface interactions and compensation energy for changes in the string length. The model is shown in Fig. 2.8. The loop filter was designed by approximating the losses based on string length based on recordings of a professional guitar player playing notes at the different fret locations across the neck. The loop filter's coefficients have been expressed as a polynomial which is a function of relative string length to allow for the synthesis of notes in between the frets.

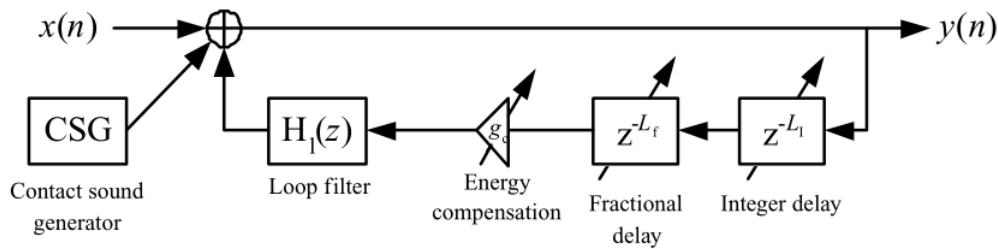


Fig. 2.8 Slide guitar DWG model from [Pakarinen, Välimäki, and Puputti 2008]

The fundamental control signal for the system is the time-varying relative length of the string, notated 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 the Contact Sound Generator section) 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 which does not require another “pluck” to instantiate the changes. 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 volume of the synthesized tone, 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 half the string length, then half the samples of the DWG would removed and an unnatural drop in the volume of the sound occurs. Similarly, sliding downwards results in the addition of samples to the DWG which could result in a sudden increase in volume.

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

$$p_c[n] = \sqrt{1 - \Delta x} p[n] = g_c p[n] \quad (2.6)$$

where Δx 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].

2.2.2 Loop Filter

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

$$H_l(z) = g \frac{1 + a}{1 + az^{-1}} \quad (2.7)$$

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 equations:

$$g = g_{pol}(0) + g_{pol}(1)m_{fret} \quad (2.8)$$

$$a = a_{pol}(0) + a_{pol}(1)m_{fret} \quad (2.9)$$

$$m_{fret} = -12 \log_2(L) \quad (2.10)$$

where m_{fret} represents the fret number and is derived from the relative length based on the rules of the 12-tone equal temperament tuning system. This illustrates how they are interpolated via a first-order polynomial. The polynomial coefficients change for each string and are specified in tables 2.1 and 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 frequency 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 Coefficients for first-order polynomial fit to loop-gain data [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 and 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.

	$a_{pol}(0)$	$a_{pol}(1)$
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 Coefficients for first-order polynomial fit to filter cut-off frequency [Välimäki and Tolonen 1998]

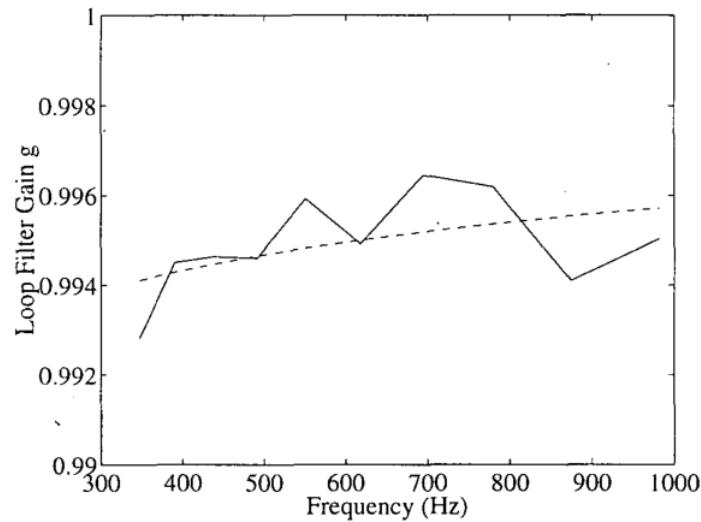


Fig. 2.9 Loop gain g for modeling string 1 (solid line) and first-order polynomial fit (dashed line) as a function of frequency. [Välimäki and Tolonen 1998]

2.3.1 Contact Sound Analysis

Figure 2.10 shows the spectrogram of the noise generated by dragging a finger tip across the surface of a wound guitar string [Pakarinen, Penttinens, 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 [Penttinens et al. 2006] where the guqin, a fretless Chinese stringed instrument using wound strings, was examined.

Harmonic Component

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

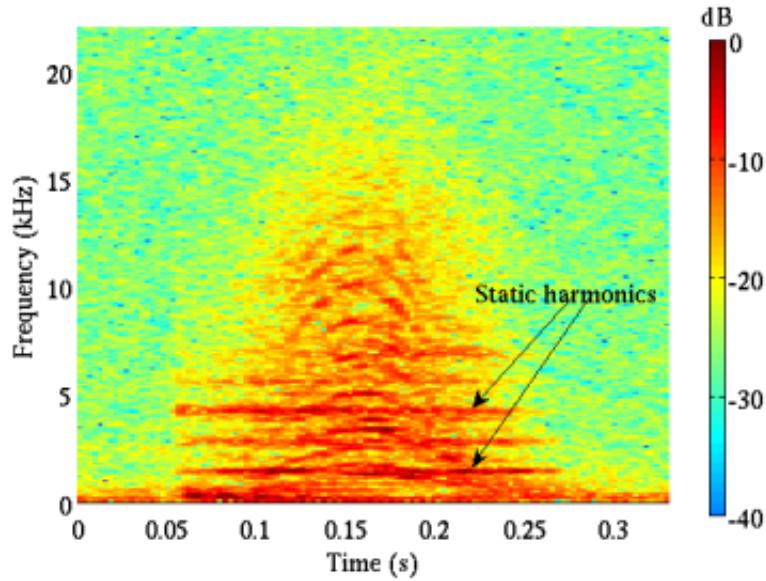


Fig. 2.10 Spectrogram of handling noise generated by sliding a finger on a wound guitar string from [Pakarinen, Penttinen, and Bank 2007]

In terms of a finger moving along a wound string, this can mathematically be expressed as:

$$F(t) = \left[\sum_k \delta(t - t_k) \right] * f(t) \quad (2.11)$$

where $f(t)$ is the impulse response from a single finger-winding collision, t_k is the time instant where the k th impulse response is generated and δ is Dirac's delta function. This equation can also be interpreted as a periodic impulse train which is filtered by the transfer function of a single finger-to-winding collision.

To understand how the t_k values 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:

$$t_k = \frac{k}{n_w f_{speed}} \quad (2.12)$$

where n_w represents the linear winding density of the string (winds per meter), f_{speed} represents the finger tip speed (meters per second). This means the quantity $n_w f_{speed}$ 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 fundamental frequency of the generated harmonic component is controlled by the speed at which the finger moves. The faster the finger moves, the shorter the period between impacts is and the higher the fundamental of

the resulting waveform. A time-varying finger velocity generates a time-varying harmonic signal where the periodic waveform corresponds to the impulse response of a single finger-to-winding impact for that particular string.

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, Penttinen, 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. 2.11.

Static Component

The other component from the 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 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:

$$\mu \frac{\partial^2 \xi}{\partial t^2} = ES \frac{\partial^2 \xi}{\partial x^2} - 2R(f)\mu \frac{\partial \xi}{\partial t} + d(x, t) \quad (2.13)$$

where:

- $\xi(x, t)$ is the longitudinal displacement
- E is Young's modulus
- S is the string's cross sectional area
- μ 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 ρ is the density of the material. Contrary to transverse modes, the longitudinal propagation speed does not depend on the tension in the string. Suppose that the string is only excited at one spatial location, x_{exc} . In this case then the spatial force distribution can be approximated as: $d(x, t) = \delta(x_{exc})F(t)$ where $F(t)$ is the excitation force applied.

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:

$$F_b(t) = \frac{ES}{\mu L^2} \sum_{k=1}^{\infty} \left[\frac{k}{f_k} e^{-tR(f_k)} \sin(2\pi f_k t) \right] * \left[\sin\left(\frac{k\pi x_{exc}}{L}\right) F(t) \right] \quad (2.14)$$

where the longitudinal modal frequencies are $f_k = k \frac{c_L}{2L}$. Equation 2.14 clearly 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 the 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, $F(t)$, whose amplitude depends on x_{exc} .

2.3.2 Adapting From Finger to Slide

Extrapolating this to the more rigid slide object is rather intuitive. The physical properties of the string do not change, only the surface of the object interacting with the string. In terms of impact on the analysis, the only part of the equations which would change is the impulse response function $f(t)$ in Eq. 2.11. This would now represent the impulse response of the slide impacting a single-to-winding. All the other analysis was represented more generally and the changes would follow suit. 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

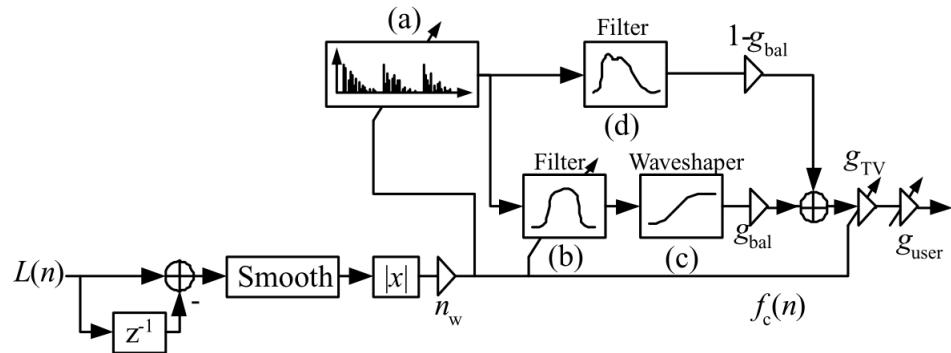


Fig. 2.11 CSG model from [Pakarinen, Välimäki, and Puputti 2008]

chosen as the 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 modeled as an exponentially decaying noise burst. Example noise bursts are shown in Fig. 2.12. The time-interval between noise pulses is controlled by the slide velocity as illustrated in Eq. 2.12. In a

certain regard, the CSG can be viewed more as a periodic impact synthesis model more so than a frictional model, which matches with how the slide makes contact with the windings. 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. 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 the same 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.

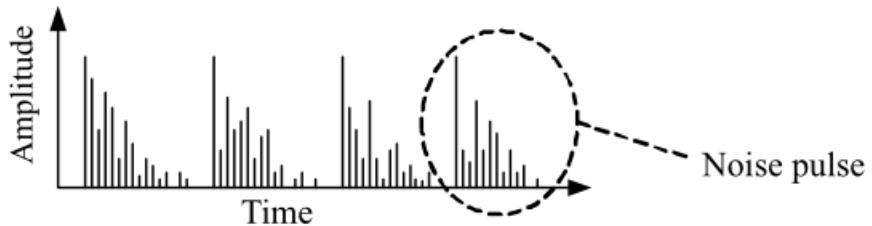


Fig. 2.12 Noise Pulses from [Pakarinen, Välimäki, and Puputti 2008]

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. 2.12. 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 which 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

The Computer Music Journal (CMJ) article mentioned before 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 CMJ. The theoretical refinements add computational complexity at the potential cost of a real-time implementation.

2.4.1 Gianpaoalo's DWG Approach

In 2012, Gianpaolo Evangelista published a paper as part of an AES convention which proposes a physically inspired DWG based model. 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 for 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.

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.

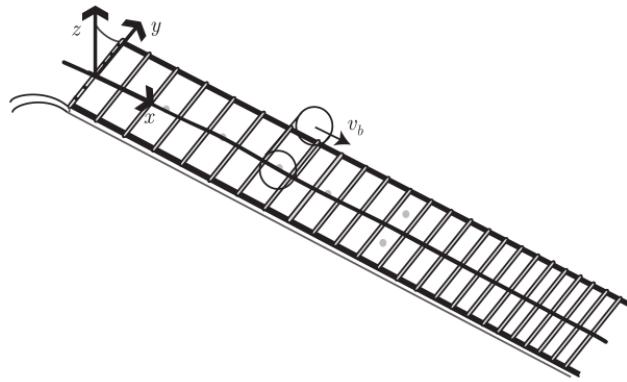


Fig. 2.13 Coordinate system used to describe string and slide motion in Evangelista's approach [Evangelista 2012].

A movable 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.

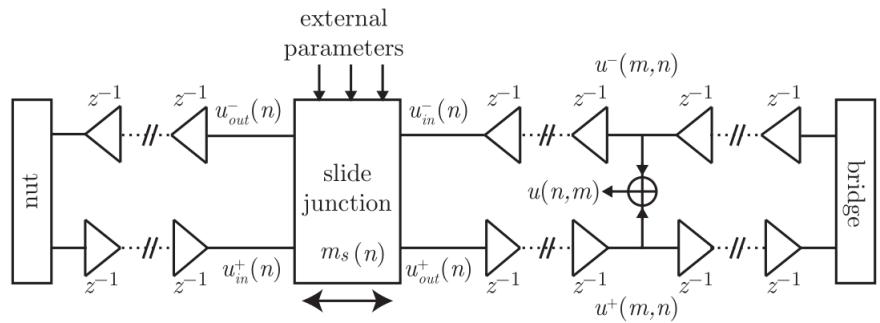


Fig. 2.14 Slide junction model introduced in [Evangelista 2012].

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 to 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>.

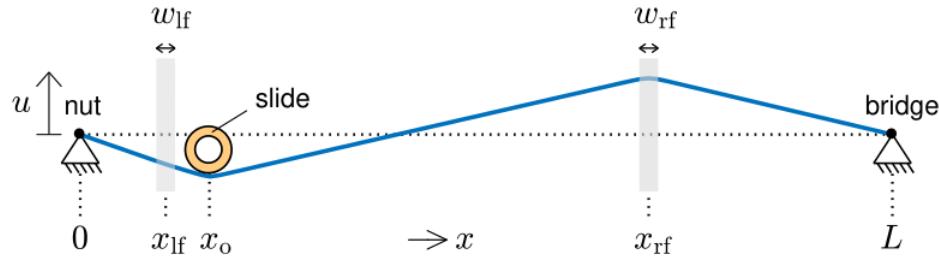


Fig. 2.15 Spatial layout for finite difference approach. The gray areas represent finger force regions [Bhanuprakash, Stapleton, and Walstijn 2020].

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 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 approaches. 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 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. Explicit studying of the player's actions based on observation of online videos as well done. Additionally an embodied understanding of the motions required to generate musical examples as obtained through study of the playing techniques on a physical instrument. 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 the Introduction. 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. 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 figure 3.1.

3.1.2 Single String Slide Synthesizer

The highest-level component of the synthesis system is depicted in figure 3.1. This is a synthesizer for a single string where the pitch is controlled by a slide. Similar to the model introduced in

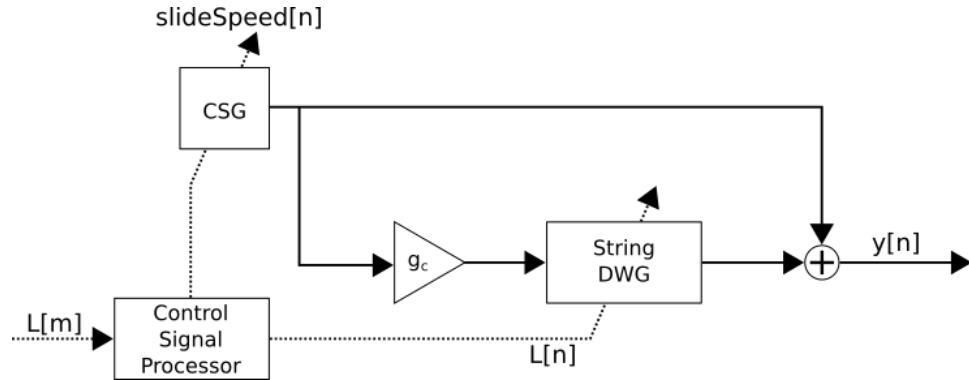


Fig. 3.1 High level architecture for a single string slide synthesizer

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 the chapter covering physical measurements.

The Control Signal Processor is another new block which will be more carefully detailed in the next section. It was placed at high as possible in the architecture for commonization purposes. Placement here removes the need for the individual objects to perform any additional processing on the control signals themselves. This improves computational efficiency by removing redundant computations 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

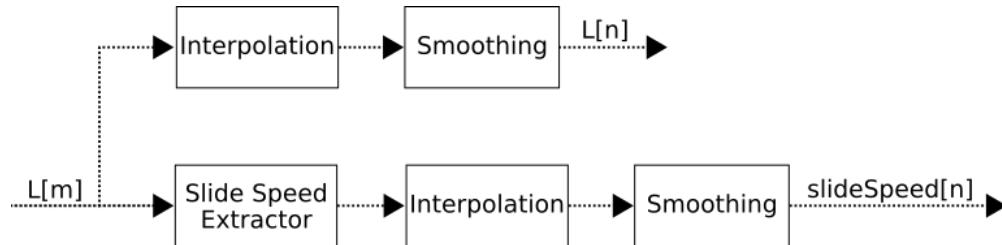


Fig. 3.2 Signal flow diagram for control signal processor block

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 the slide speed as well as relative length signal at audio-rate. The purpose of this block is to extract the speed signal as well as change the control-rate signals to audio-rate.

The interpolation is done via linear-interpolation and is this operation upsamples 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 (like transients). Supposing that R represents the ratio between the control-rate and audio-rate, then for each one control-sample, $R - 1$ audio 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

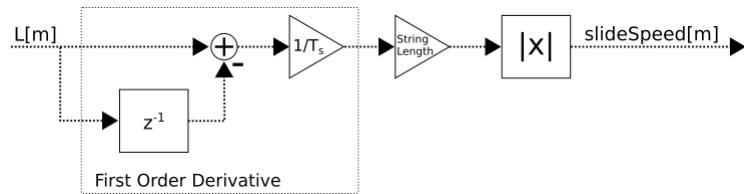


Fig. 3.3 Signal flow diagram for slide speed extractor block

Figure 3.3 shows a signal flow diagram for the slide speed extractor. This is extremely 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 code.

Energy Scaler

Figure 3.5 illustrates the signal flow diagram of the energy scaler. It implements the energy scaling as described in the Chapter 2.

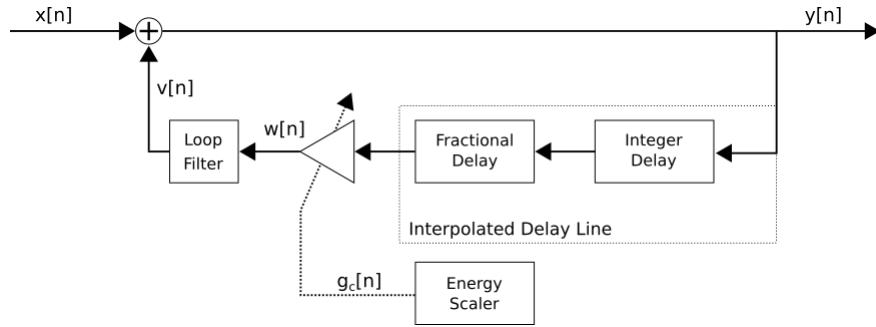


Fig. 3.4 Signal flow diagram for the time-varying string DWG. $L[n]$ is not depicted as every object consumes it in some fashion.

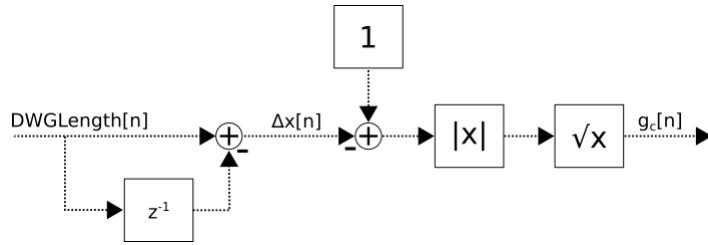


Fig. 3.5 Signal flow diagram for energy scaler block

Loop Filter

The loop filter is implemented via a single-pole design and simulates the losses associated with string vibration. It has been described previously in the Background chapter. 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].

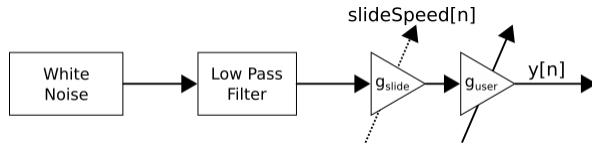


Fig. 3.6 Signal flow diagram for unwound CSG block

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/winding collisions occur. The n_w parameter is stored here to keep all the information specific to the string’s physical properties in a single location.

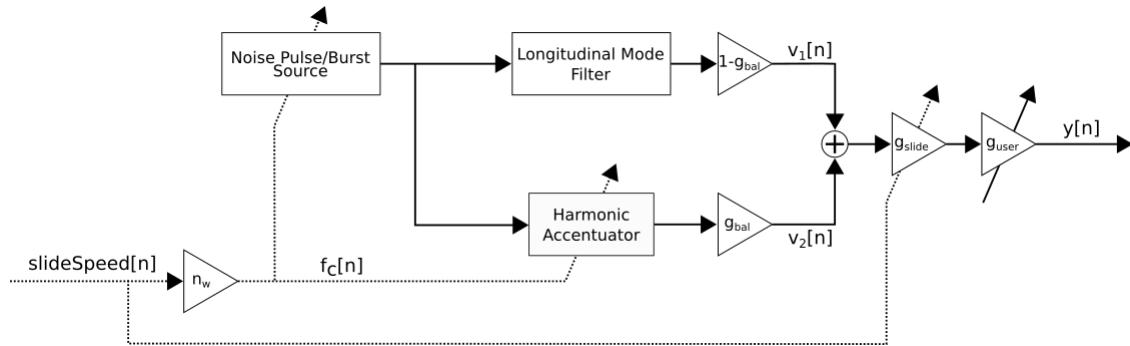


Fig. 3.7 Signal flow diagram for wound CSG block

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 quite a bit 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 the Physical Measurements chapter). 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 help prevent unwanted DC components from tarnishing the sound as well as building up in the string digital wave guide. The difference between this and not using an absolute value block will be explained in the Sound Design chapter.

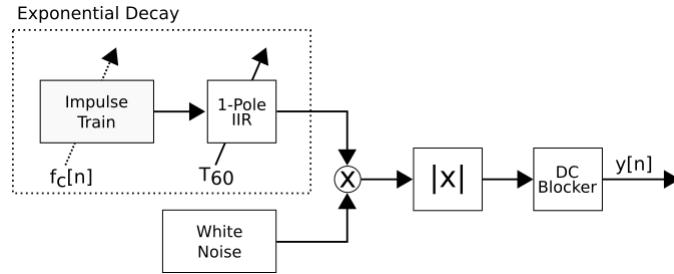


Fig. 3.8 Signal flow diagram for the Noise Pulse Train generator

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.

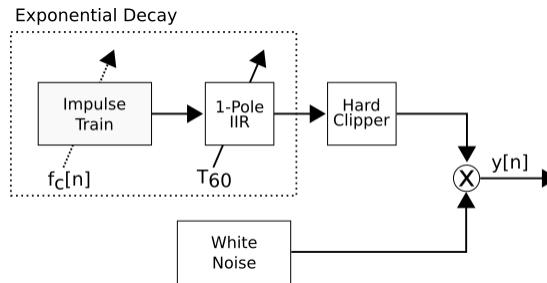


Fig. 3.9 Signal flow diagram for the Noise Burst Generator

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 [Penttinens 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.

Resonator + Tanh The first method is a second-order resonator in series with a hyperbolic tangent function as illustrated in figure 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.

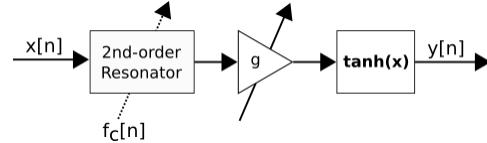


Fig. 3.10 Signal flow diagram for the Resonator + Tanh waveshaper

Harmonic Resonator Bank The second approach is illustrated in figure 3.11. This method is more computationally expensive, but provides much more control over the strength, number and location of the different harmonics. 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 the spectrograms in [Pakarinen, Puputti, and Välimäki 2008], however the object itself supports an arbitrary number.

3.2 Limitations of Model

3.2.1 Minimum Relative String Length

Although not explicitly mentioned, 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].

The paper describes the method by which the polynomials used to generate the filter's a and g values as being 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

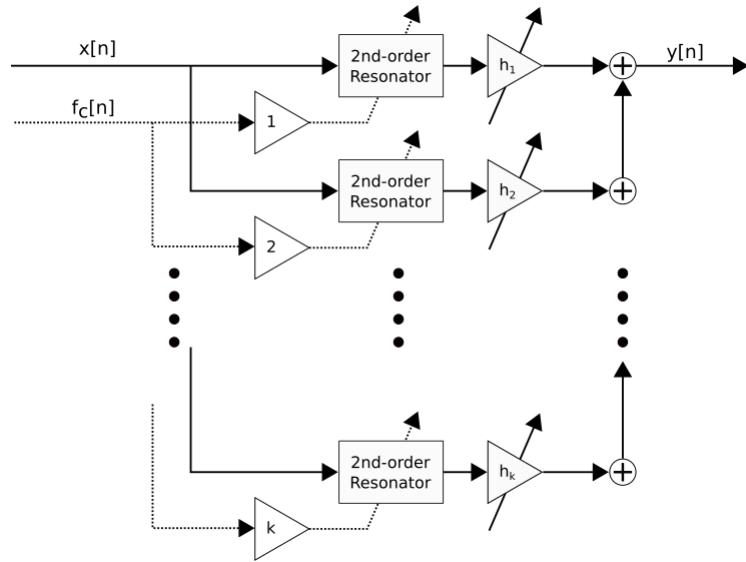


Fig. 3.11 Signal flow diagram for the Harmonic Resonator Bank

paper, one could look up the number of frets on a manufacturer's website. This information is not available. 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 Background chapter illustrates the loop gain g across different frequencies for the high e string. By combining Eq. 2.3 and Eq. 2.10, 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. 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.

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

Table 3.1 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 the high E and D strings across a variety of upper frets to show how the transition occurs. The general pattern to note 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 digital waveguide structure, this creates a positive feedback loop and ultimately 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. However, this would be considered a limitation as often in slide playing, the slide 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.

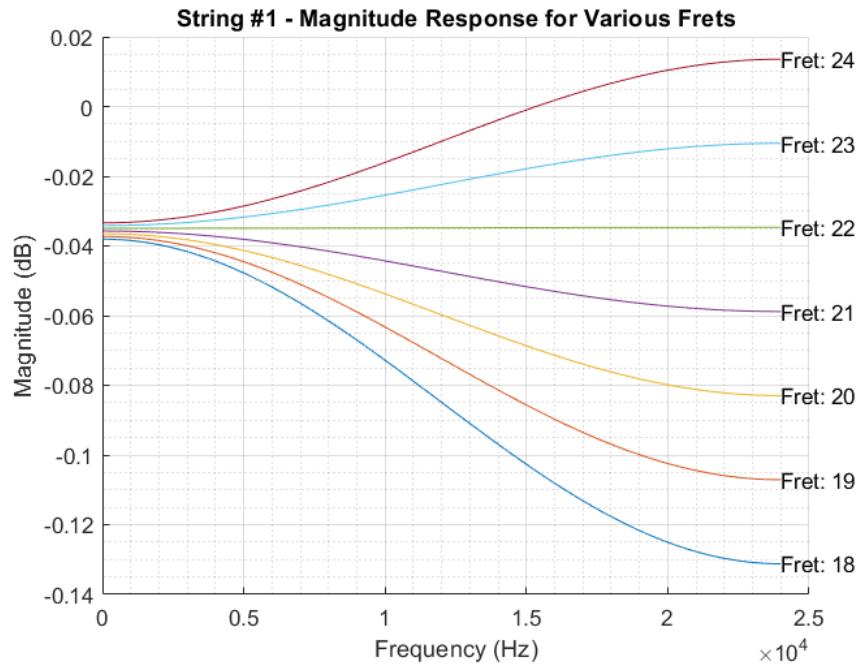


Fig. 3.12 Loop filter magnitude response for the high E string at a variety of frets to indicate its transition to a positive gain in the upper frequencies.

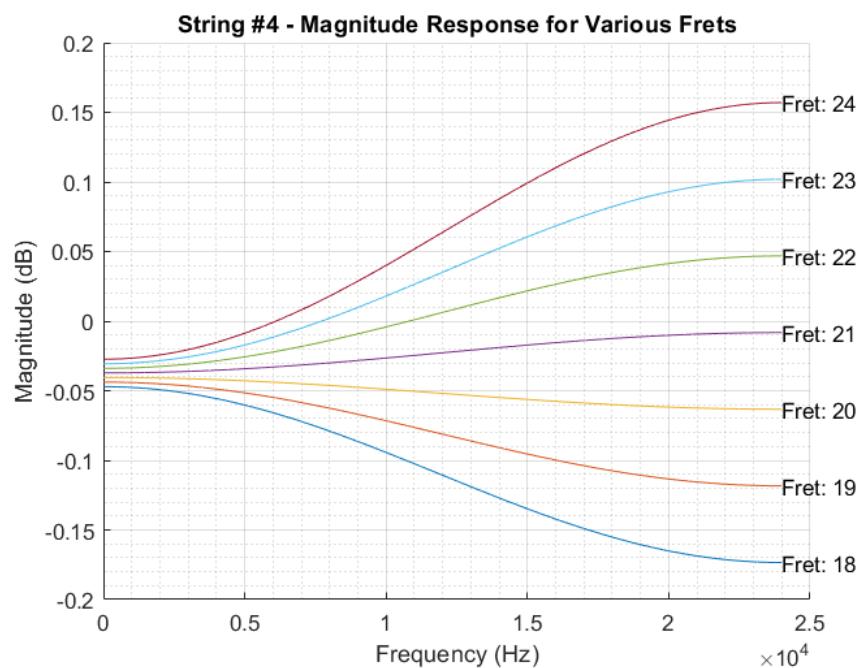


Fig. 3.13 Loop filter magnitude response for the D string at a variety of frets to indicate its transition to a positive gain in the upper frequencies.

3.2.2 Non-constant Phase Delay of Filters

Both the loop filter as well as the interpolation filter illustrate non-constant phase delays. This is shown in figures 3.14 and 3.15. The loop filter illustrates this as it is an IIR filter and 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 .5) [Laakso et al. 1996]. Many strings in reality exhibit some form of stiffness which results in different wavespeeds 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, 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.

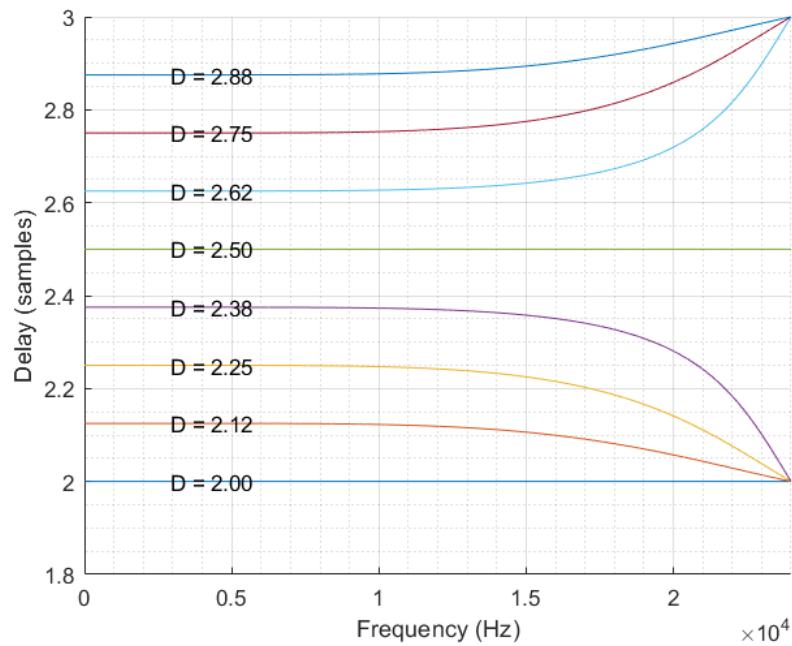


Fig. 3.14 Phase delays for Lagrange interpolating filters with order = 5 at various delay values

3.2.3 Impulse Train Implementation

The Impulse Train has a limitation based on its implementation which results in a quantization of the frequencies it can operate at. Internally, the Impulse Train object counts the number of samples which have passed until it is time to generate an impulse signal. The period in samples

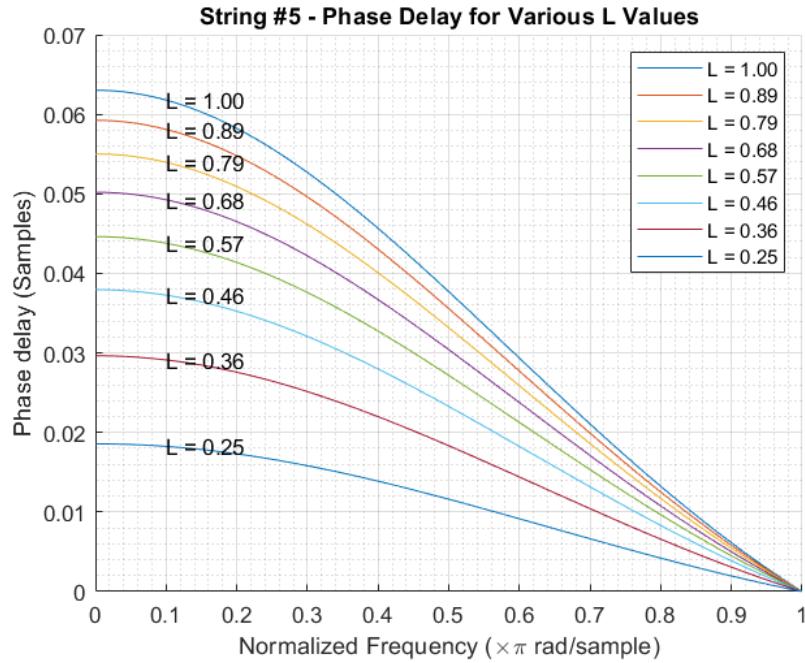


Fig. 3.15 Loop Filter phase delay

(which determines the count value) is computed using the follow equation:

$$\text{period} = \text{round} \left(\frac{F_s}{f_c[n]} \right) \quad (3.1)$$

This results in the period of the impulse train signal being limited to purely integer sample counts. Given the reciprocal nature between frequency and period, this ends up quantizing the frequencies which the Impulse Train can generate signals. 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 which 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.

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 object to the parabolic $f_c[n]$ sweep are shown in Fig. 3.18 and can be heard in *f_cSweepQuantization-NPT.wav*. 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.

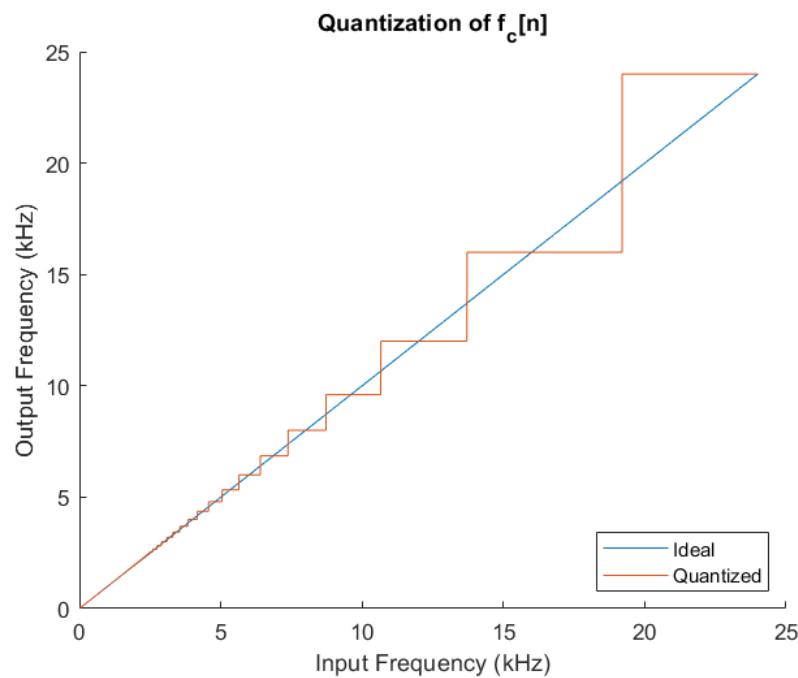


Fig. 3.16 Illustration of the quantization effects from Eq. 3.1 for a sampling rate of 48 kHz

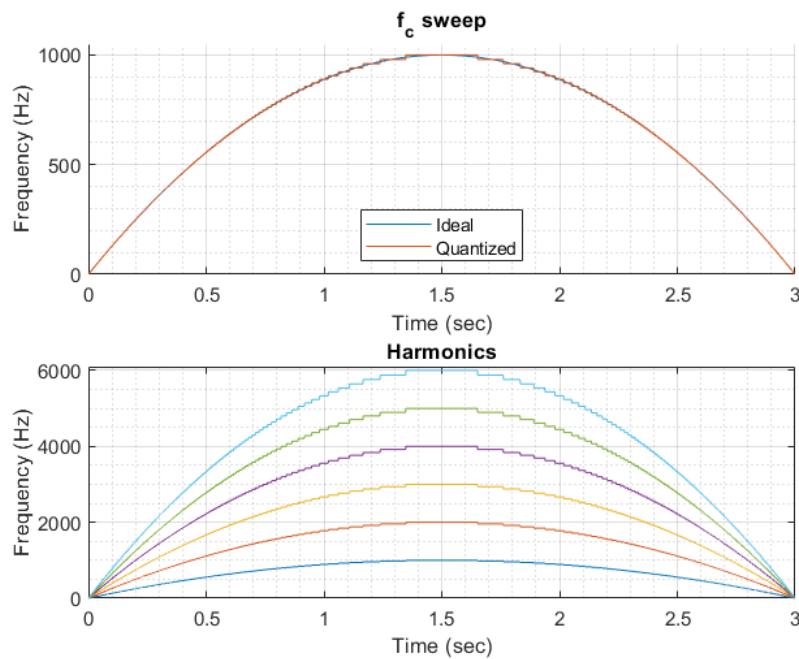


Fig. 3.17 Illustration of the quantization effects for a parabolic sweep and the corresponding impact on the first six harmonics. Note how the effects are amplified in the higher harmonics.

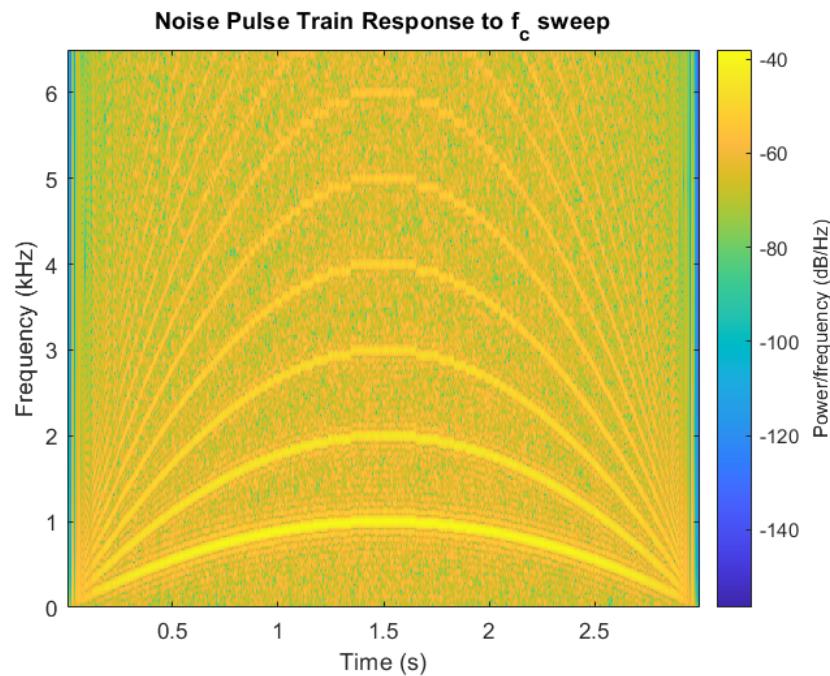


Fig. 3.18 Spectrogram of the output of the Noise Pulse Train in response to the parabolic $f_c[n]$ sweep from Fig. 3.17. The plot has been zoomed in to emphasize the effects on the first six harmonics. The output can be heard in *f_cSweepQuantization-NPT.wav*.

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 (i.e. integer delay lines, filters) were verified first before being integrated into the larger constituent components (i.e. noise generators, 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 attached code of the Appendix.

4.1 Filters

All the filters implemented using a class built around MATLAB's *filter()* function. 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 its corresponding section.

4.1.1 Resonator

As a test, the second-order resonator 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 figure 4.1.

4.1.2 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

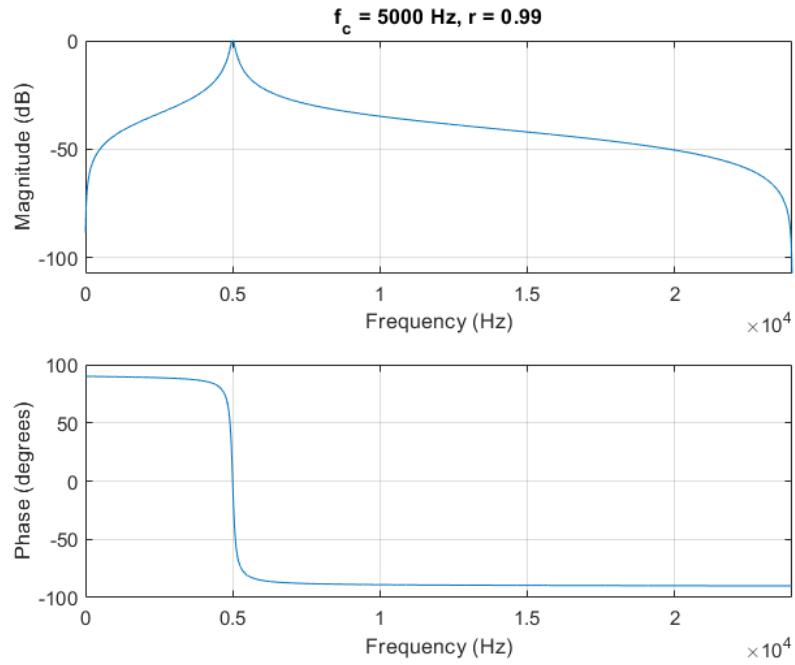


Fig. 4.1 Frequency response for 2nd-order resonator test

filters are what is provided for an implementation. Magnitude responses for the different filters as well as the linear-prediction estimates are provided as shown in figure 4.2. These plots for the 4th-order approximations were recreated for the implemented longitudinal mode filters as shown in figure 4.3. Verification was done through visual comparison of the plots as it was the best option available.

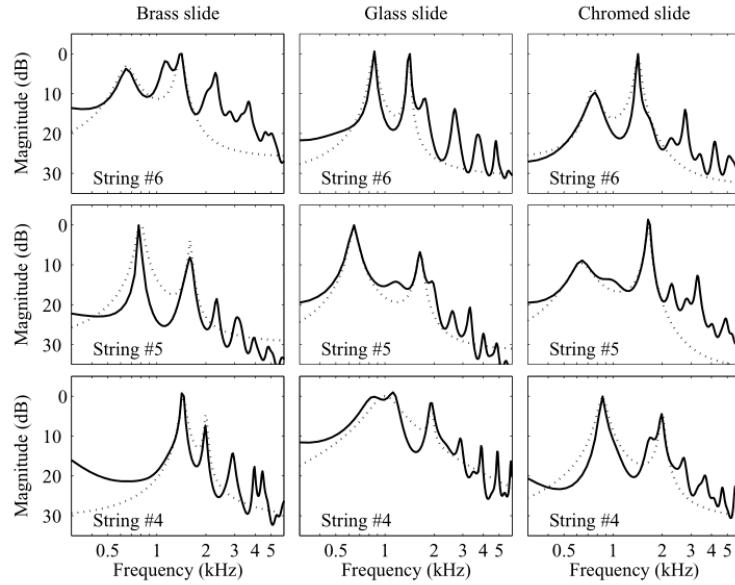


Fig. 4.2 Original figure from [Pakarinen, Puputti, and Välimäki 2008] for comparison purposes. Solid lines represent spectral estimates using linear-prediction filter of order 100. Dashed lines indicate modal filter magnitude responses.

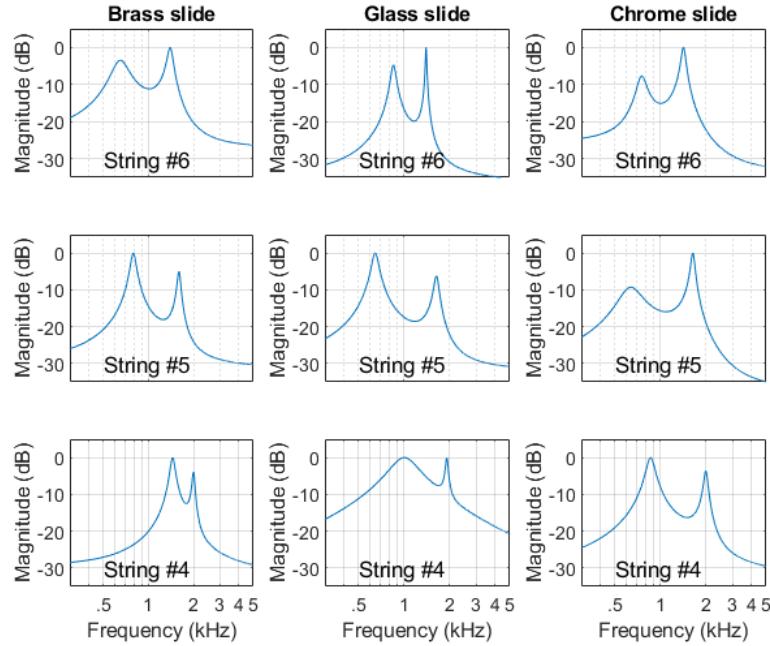


Fig. 4.3 Plot of the different contact sound filter responses for modal filters.

4.1.3 DC Blocker

Figure 4.4 illustrates the DC blocker's frequency response for a value of $R = .995$.

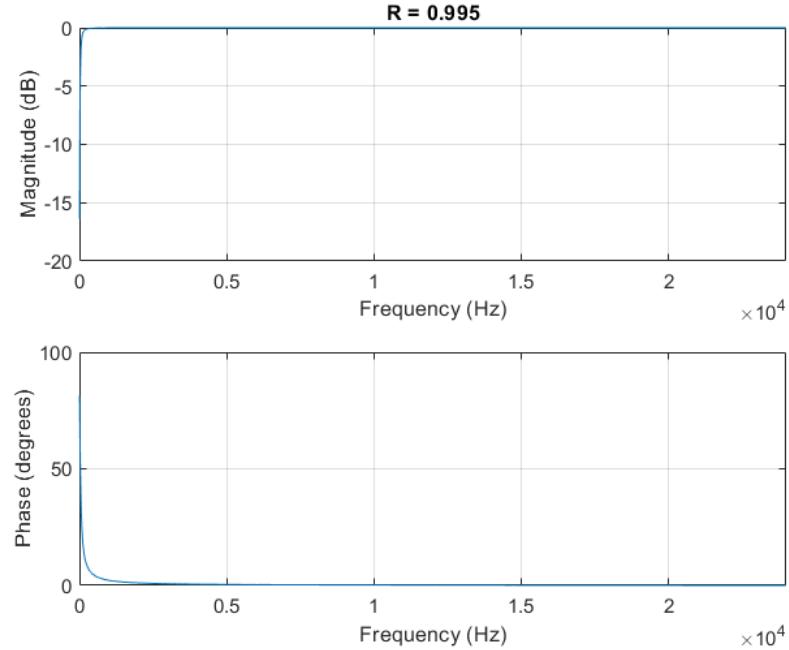


Fig. 4.4 DC Blocker Frequency Response

4.1.4 Smoothing Filter

The smoothing filters is implemented via a 10-point moving average where every sample is weighted evenly. Accordingly, it is expected that the impulse response will be 10 impulses all scaled by $\frac{1}{10}$ as the impulse response of an FIR is the same as its coefficients. This is verified by the output shown in figure 4.5. 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 is paired to the frequency response via the Fourier Transform, verification for this was done in the time-domain.

4.2 CSP and Components

4.2.1 Components

Interpolator

The interpolator block operates via linear interpolation. It was tested by specifying a control signal $L[m]$ and running it through the interpolator. The original $L[m]$ was plotted with linear line segments connecting between the points. The interpolated $L[n]$ output was plotted as individual

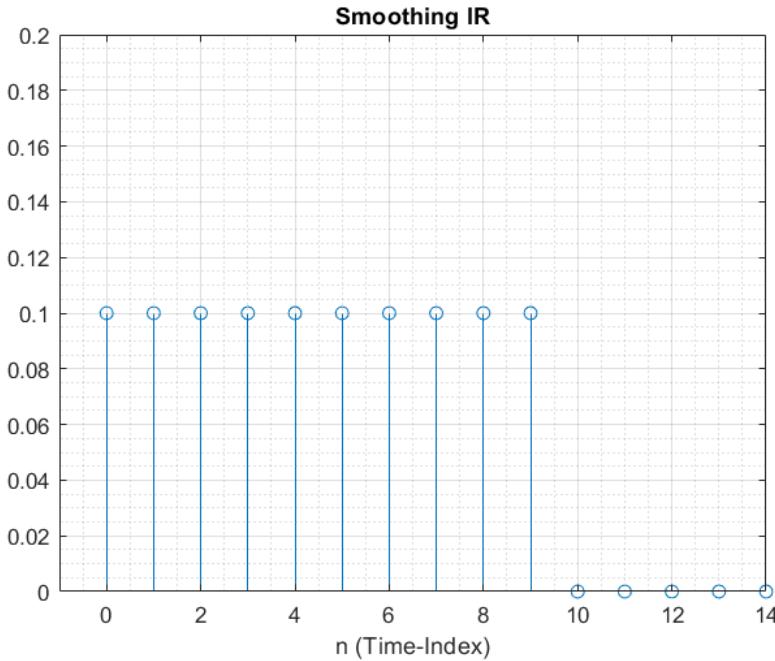


Fig. 4.5 Verification for smoothing filter

samples overlaid onto the original to ensure that the linear trajectories were maintained. Figure 4.6 illustrates this. 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. The black dashed lines in the figure correspond to the boundaries between the interpolation frames. It is also necessary to specify an initial value for the interpolation to start from. In this figure .5 was used.

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:

$$L[n] = L[n - 1] - \frac{slideSpeed[n]}{F_s \times StringLength} \quad (4.1)$$

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.

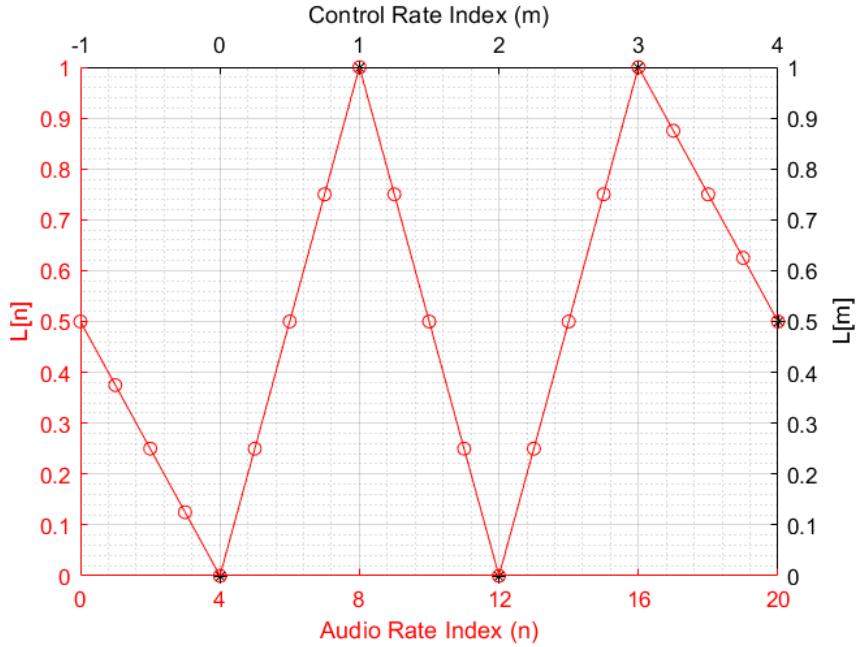


Fig. 4.6 Output from Interpolator test. Red line segments have been added to help illustrate the linearity. The initial value used to start the interpolation from is .5 and $R = 4$.

4.2.2 Control Signal Processor

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 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 CSP corresponds to when $m = 0$.

4.2.3 Loop Filter

As described before, 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 (as described in the introduction chapter). Limitations of this approach have been shown which illustrated that the filter itself is operational through examining its various frequency response characteristics. 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

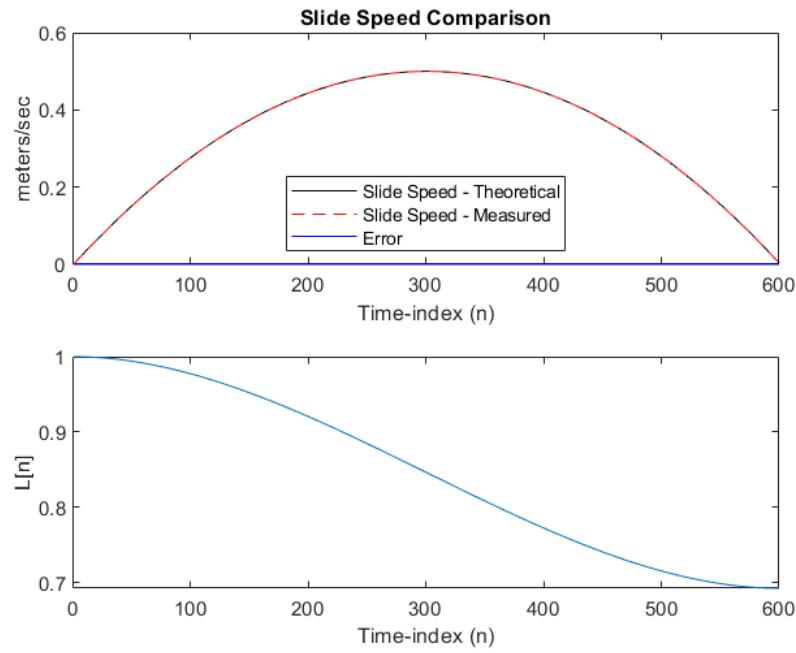


Fig. 4.7 The top plot illustrates the correctness of the slide speed extraction algorithm while the lower plot displays the $L[n]$ signal which was used as a stimuli.

correctly. Figure 4.9 and Fig. 4.11 show the original plots while Fig. 4.10 and Fig. 4.12 show the recreations respectively.

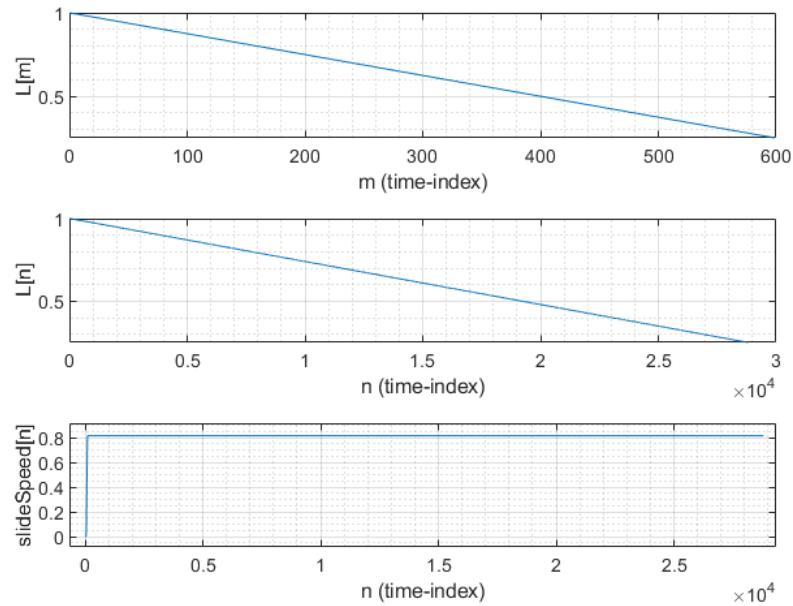


Fig. 4.8 $L[m]$ input signal and corresponding output signals for CSP test

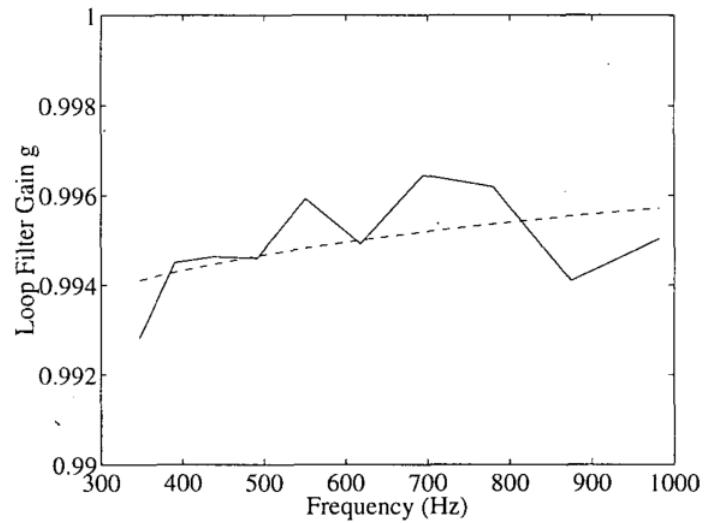


Fig. 4.9 Loop gain g for modeling string 1 (solid line) and first-order polynomial fit (dashed line) from [Välimäki and Tolonen 1998]

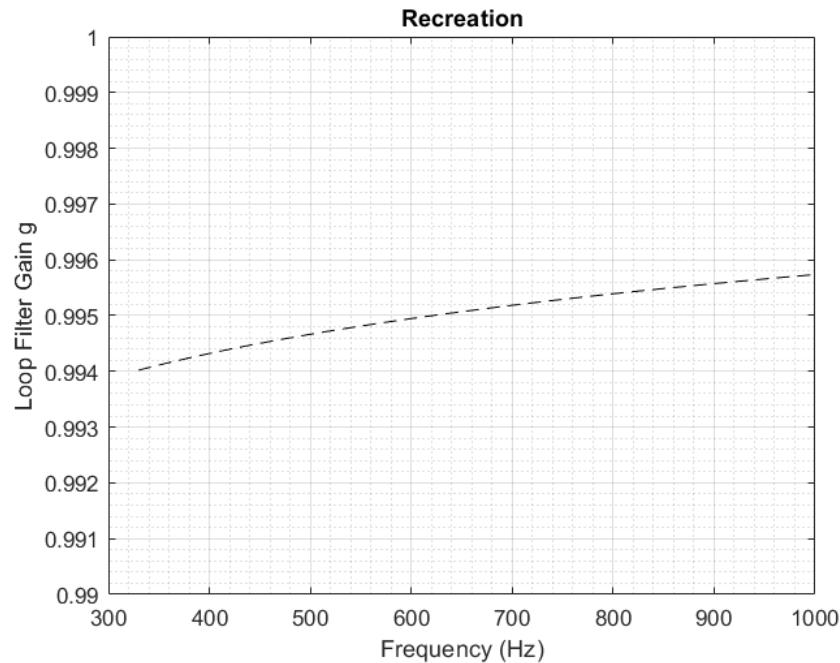


Fig. 4.10 Loop gain g polynomial for string 1

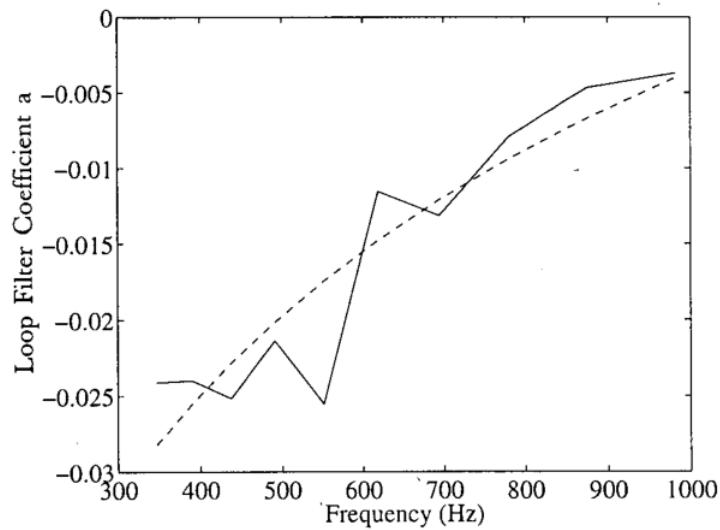


Fig. 4.11 Loop-filter a for string 1 (solid line) and first-order polynomial fit (dashed line) from [Välimäki and Tolonen 1998]

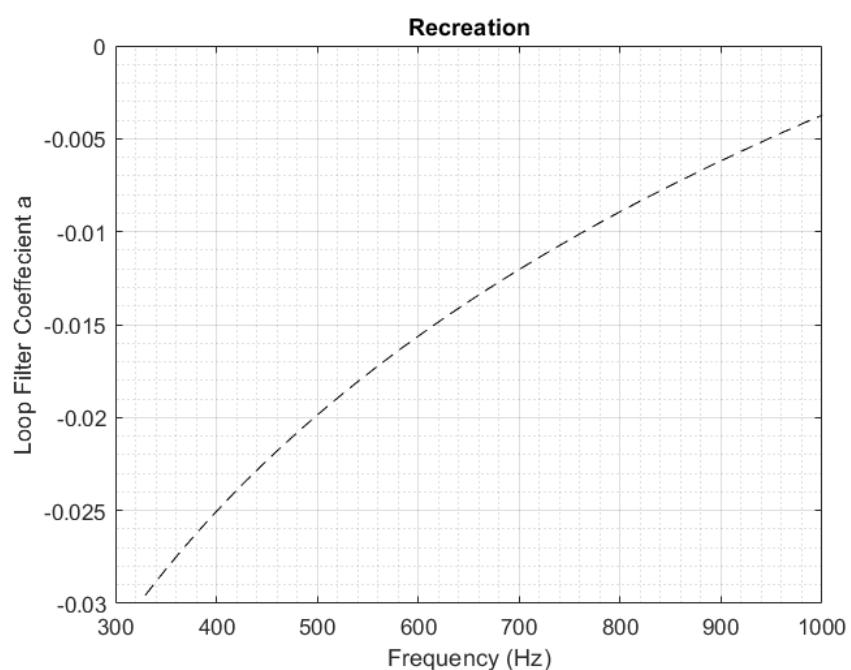


Fig. 4.12 Loop-filter a polynomial for string 1

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. Figure 4.13 provides a summary of the tests and output. The full details can be found in the Appendix.

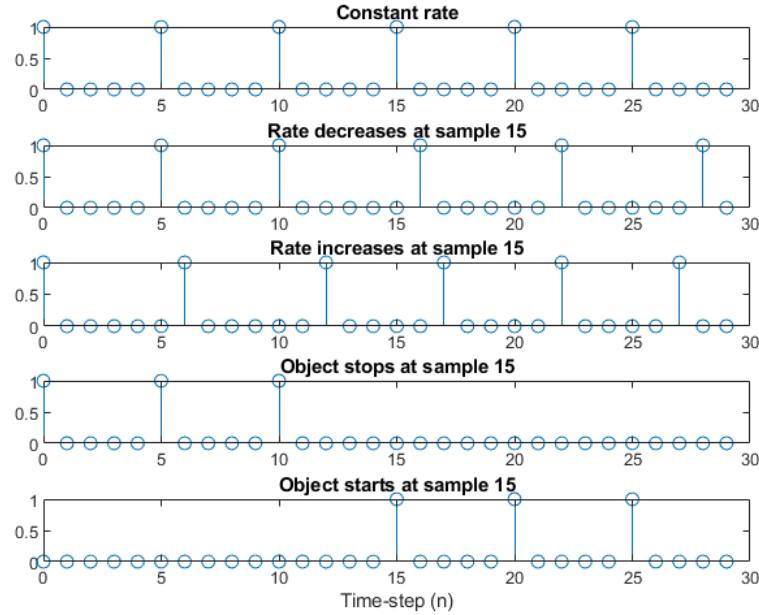


Fig. 4.13 Impulse Train test output

4.3.2 Exponential Decay

The Exponential Decay object is composed of a Impulse Train fed into a one-pole IIR 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 4.14 illustrates correct functioning of the object for three different T_{60} values where they have been converted from seconds to samples.

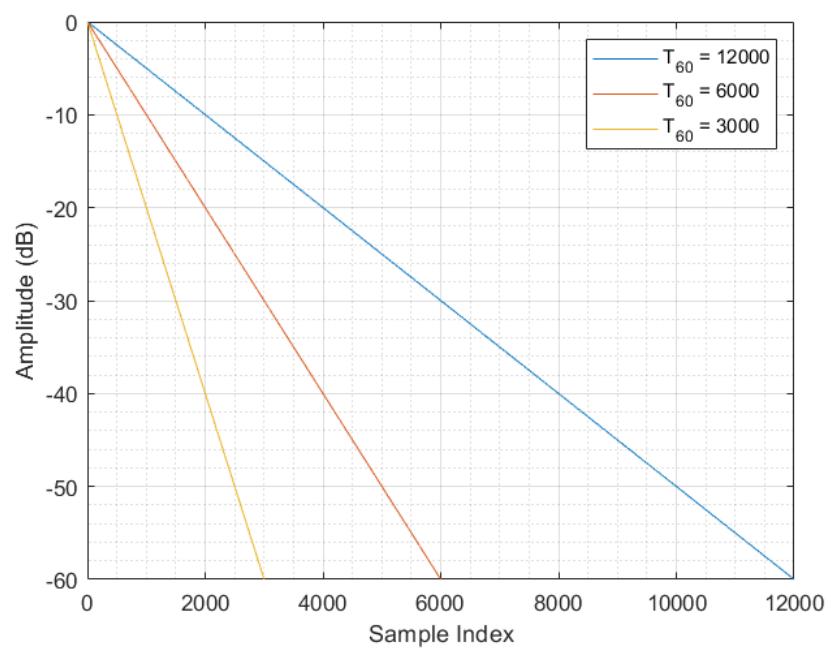


Fig. 4.14 Test results for different T_{60} decay parameters specified in samples

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 an swept firing rate corresponding to the same parabolic trajectory used in previous tests. The results are illustrated in figures 4.15, 4.16 and 4.17. Note the overlap of the individual pulses in the second test due to the firing rate being faster than the decay rate. 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*. The “noisiness” of a signal at higher values of f_c can be controlled by the T_{60} parameter as will be discussed in the Sound Design chapter.

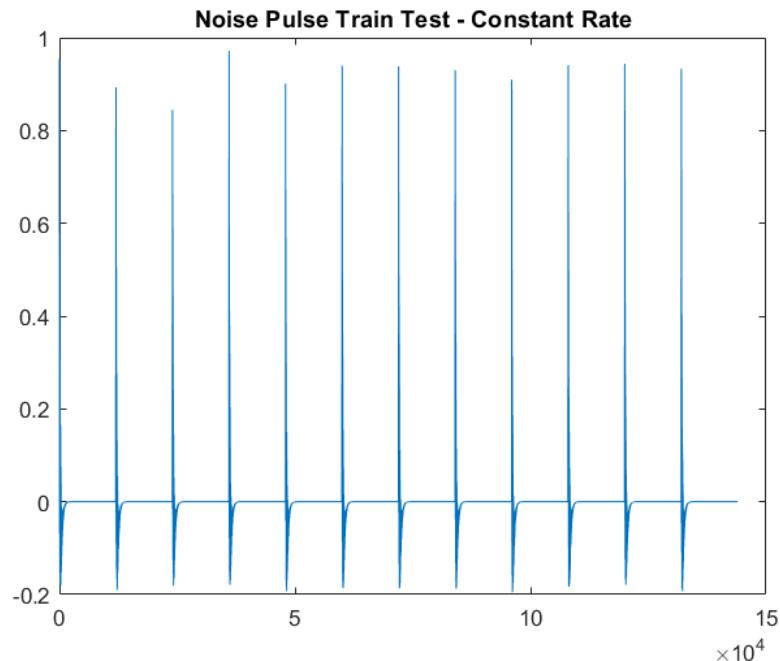


Fig. 4.15 Noise Pulse Train output for a constant rate. Note the negative values introduced by the DC Blocker.

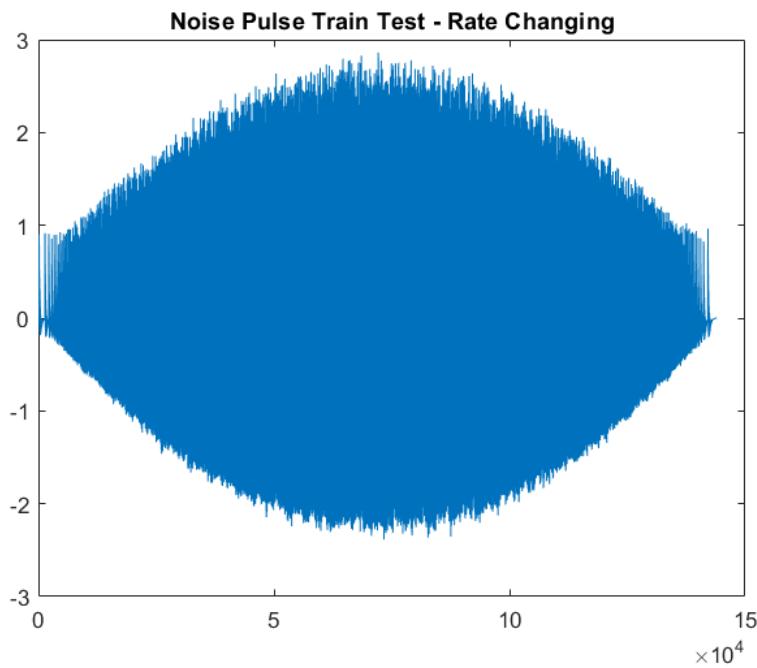


Fig. 4.16 Noise Pulse Train output in response to an $f_c[n]$ sweep. Note the overlapping build-up of the individual impulses. Individual pulses can be seen at the ends.

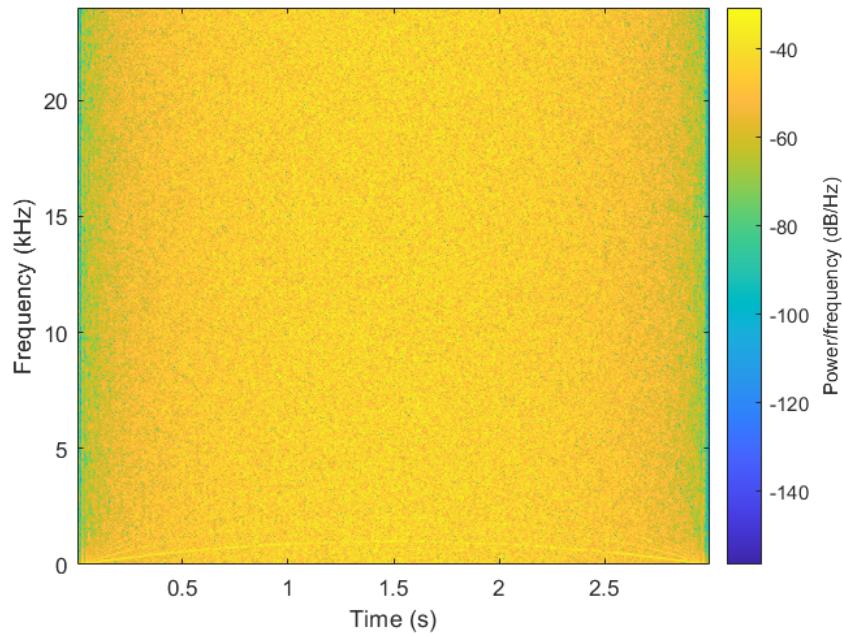


Fig. 4.17 Spectrum corresponding to figure 4.16. Note the emergence of harmonics at the lower end.

4.3.4 Noise Burst Generator

The Noise Burst generator was tested with the same two scenarios as the Noise Pulse Train. The outputs are shown in figures 4.18, 4.19 and 4.20. 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 Chapter 3. Each of the different noise sources were originally designed to work with a different Harmonic Accentuation technique. The Noise Burst Generator pairs best with the Harmonic Resonator Bank.

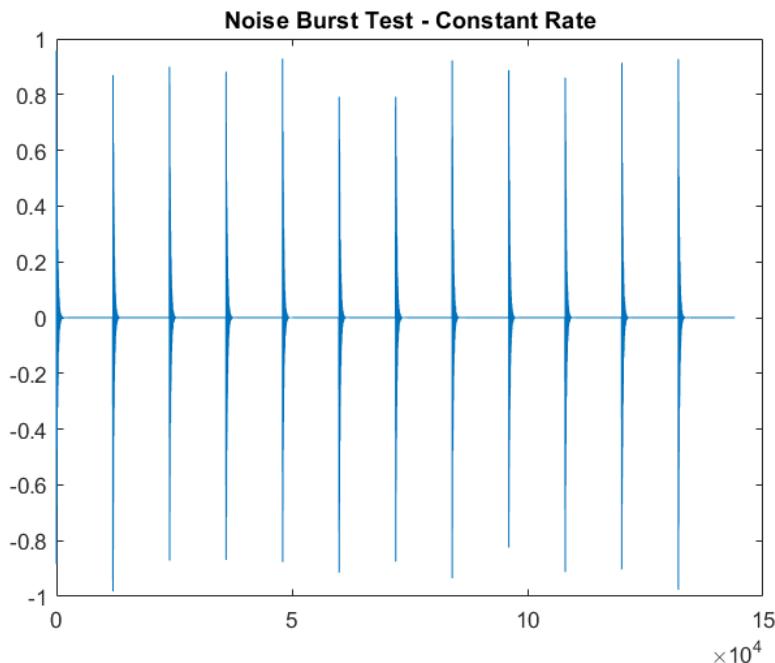


Fig. 4.18 Noise Burst Generator output for a constant rate. Note the approximate symmetry about the x-axis.

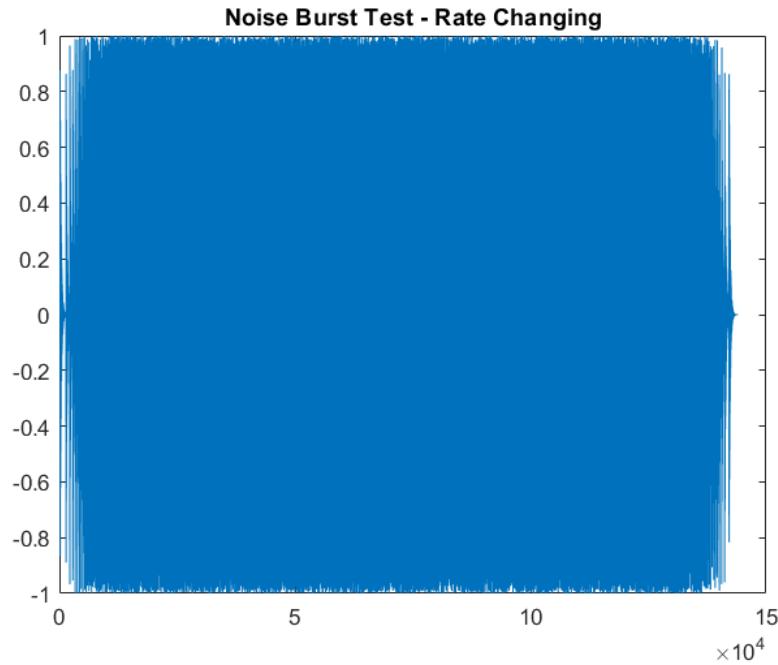


Fig. 4.19 Noise Burst Generator output in response to an $f_c[n]$ sweep. Note the effects of the hard-clipping.

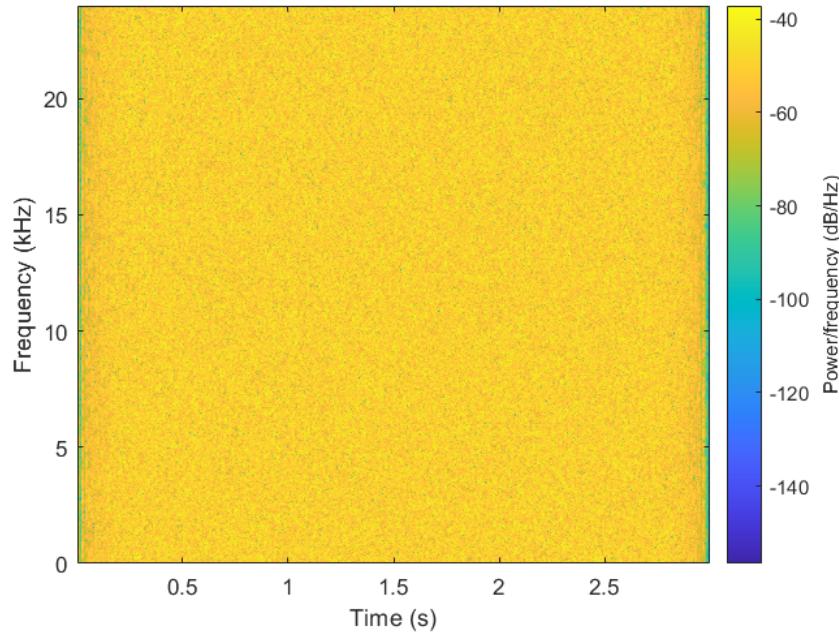


Fig. 4.20 Spectrum corresponding to figure 4.19. Note the lack of harmonics.

4.4 Harmonic Accentuators

4.4.1 Harmonic Resonator Bank

Given that the resonators were tested already, 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 as before. The functionality of this block predicts that we would observe six harmonically linked bands in the output spectrum. This is shown in Fig. 4.21 where the different harmonics' trajectories are overlaid in red. The output can be heard in *HRB-test.wav*.

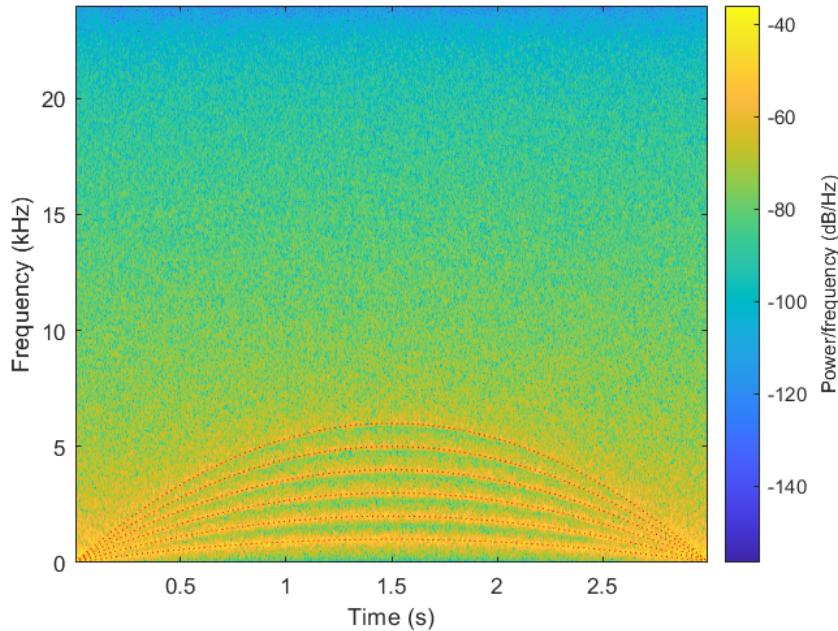


Fig. 4.21 Output from Harmonic Resonator Bank test

4.4.2 Reso + Tanh

The Reso + Tanh block 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 harmonic signal being input in order to achieve the desired effect of accentuating/creating harmonics. Figure 4.22 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. The output of this can be heard in *ResoTanh-test.wav*.

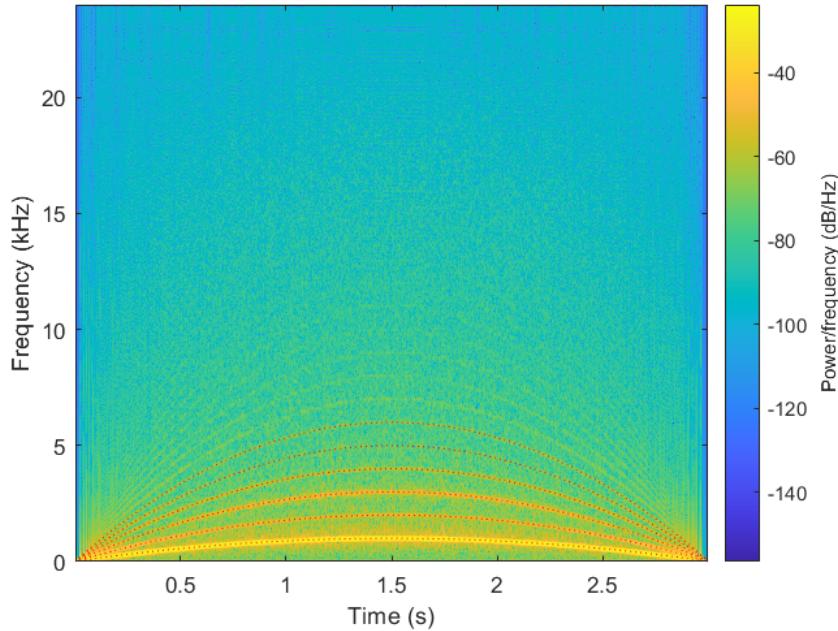


Fig. 4.22 Output from Reso + Tanh test

4.5 Contact Sound Generator

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 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 get a sense for the audible contribution of each one 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 4.23, 4.24 and 4.25. Figure 4.26 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*.

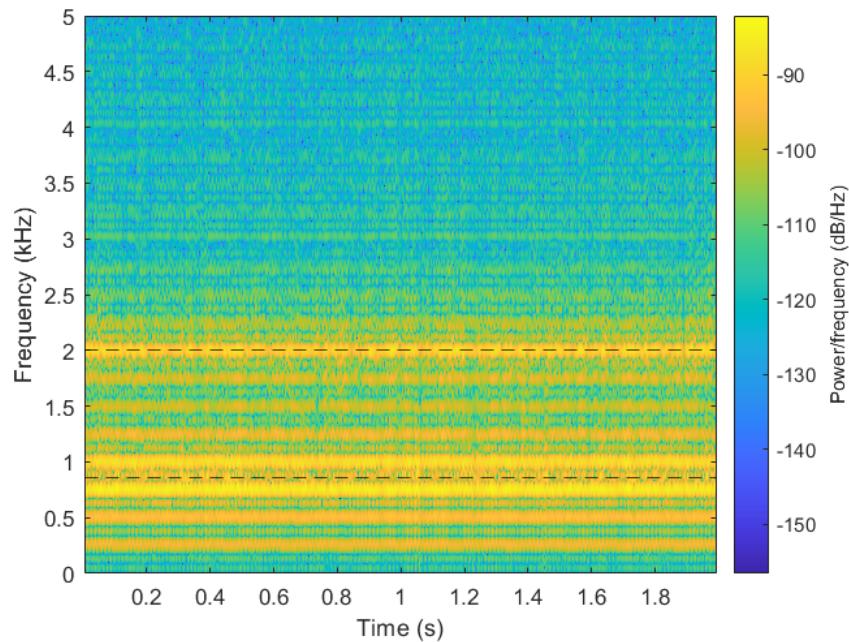


Fig. 4.23 Wound CSG and Static Slide Velocity - Longitudinal Branch

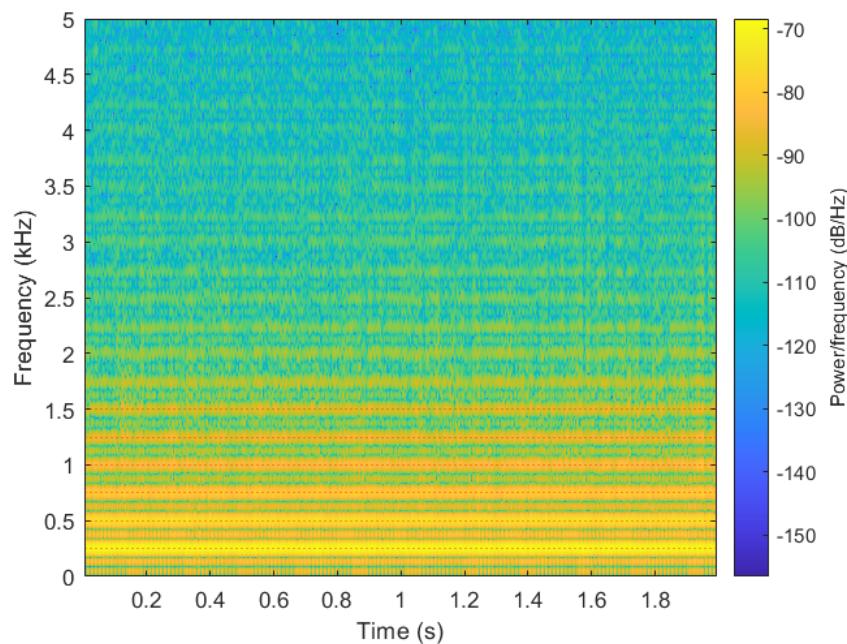


Fig. 4.24 Wound CSG and Static Slide Velocity - Harmonic Branch

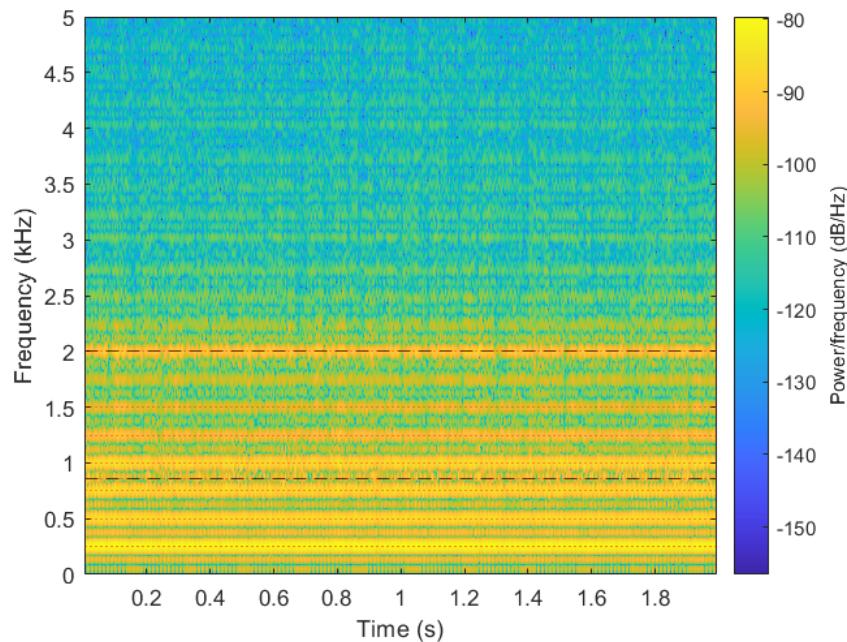


Fig. 4.25 Wound CSG and Static Slide Velocity - Both Branches

Time-Varying $f_c[n]$

The results for the time-varying scenario are shown in figures 4.26, 4.27 and 4.28. Dashed red lines indicate the theoretical harmonic trajectories. Dashed black lines correspond to the static longitudinal modes. As is clearly shown in Figure 4.26, 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 due to imperfections in the filter. 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*.

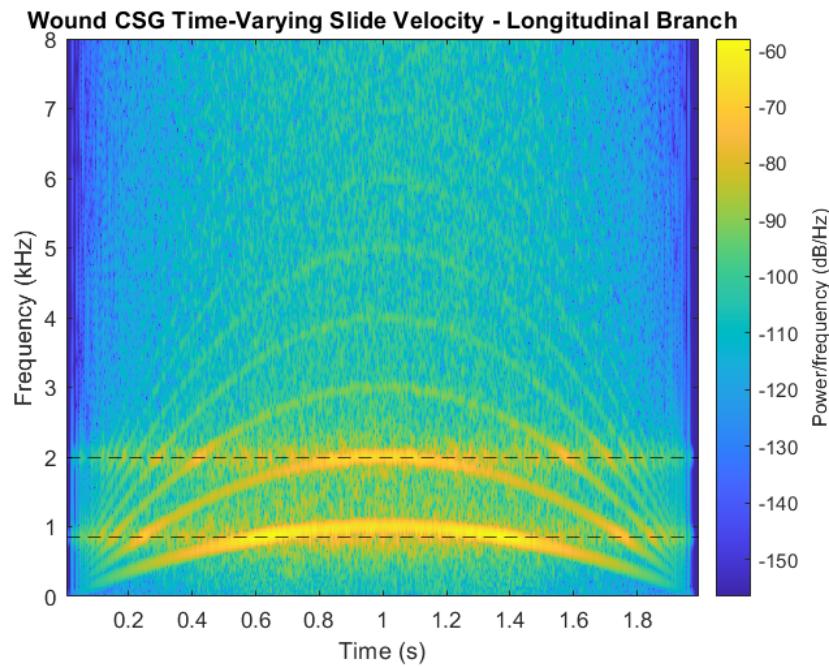


Fig. 4.26 Dashed black lines = Longitudinal modes

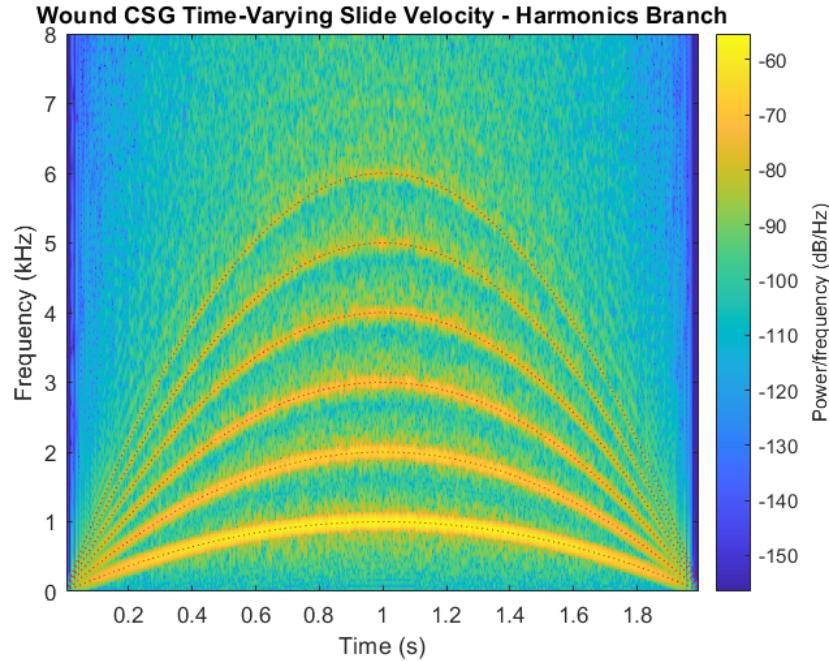


Fig. 4.27 Dotted red lines = harmonic trajectories

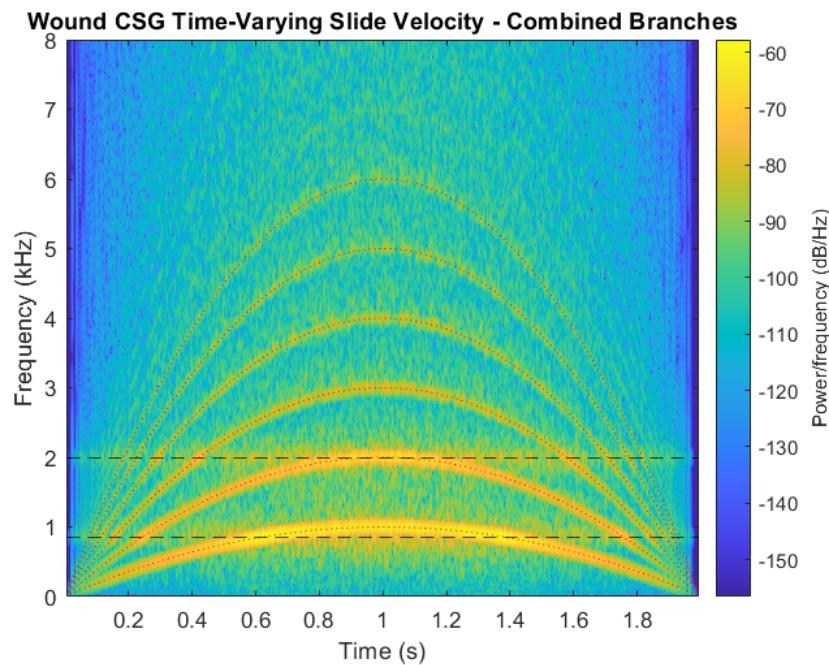


Fig. 4.28 Dashed black lines = Longitudinal modes, Dotted red lines = harmonic trajectories

4.5.2 Unwound Variant

Figure 4.29 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 (as can be seen in the Appendix) but aren't shown here.

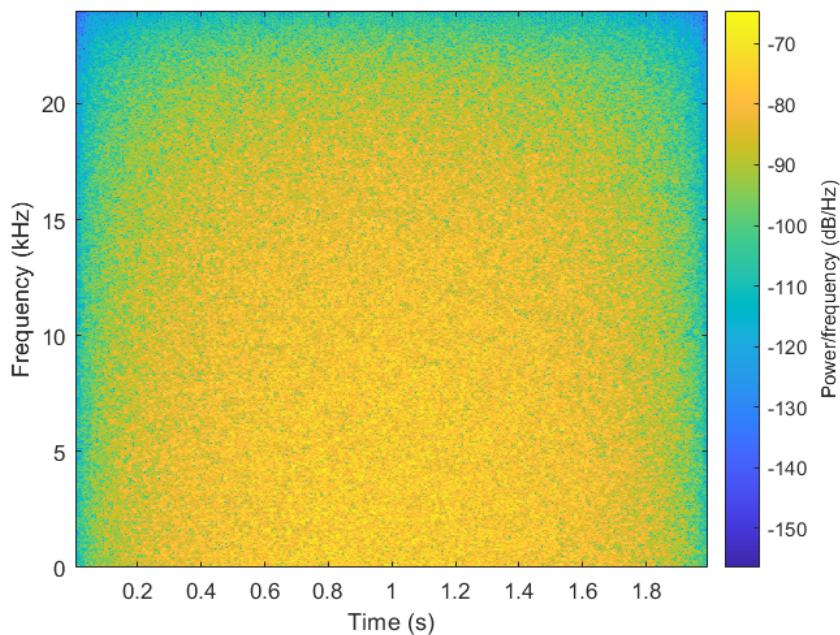


Fig. 4.29 Output from unwound branch for time-varying slide velocity

4.6 String DWG and 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 (as will be shown in the Appendix). Illustrated here will be the verification aspects of the Lagrange interpolation as these are crucial to ensuring the correct functioning of the synthesis algorithm. In this test, M represents delay in samples as implemented by the integer delay line while D represents the total delay implemented by the Lagrange interpolator. d is the fractional delay. $D = \lfloor D \rfloor + d$. The order of the Lagrange interpolator is set to 5.

The first test which was performed was a test to ensure that it could indeed operate as an integer delay. Various Lagrange objects were constructed, ensuring that no fractional delay would be required. The output of this test is shown in Fig. 4.30 where x is the input signal and y

is the output. As this figure clearly shows, the interpolated delay line clearly implements this functionality.

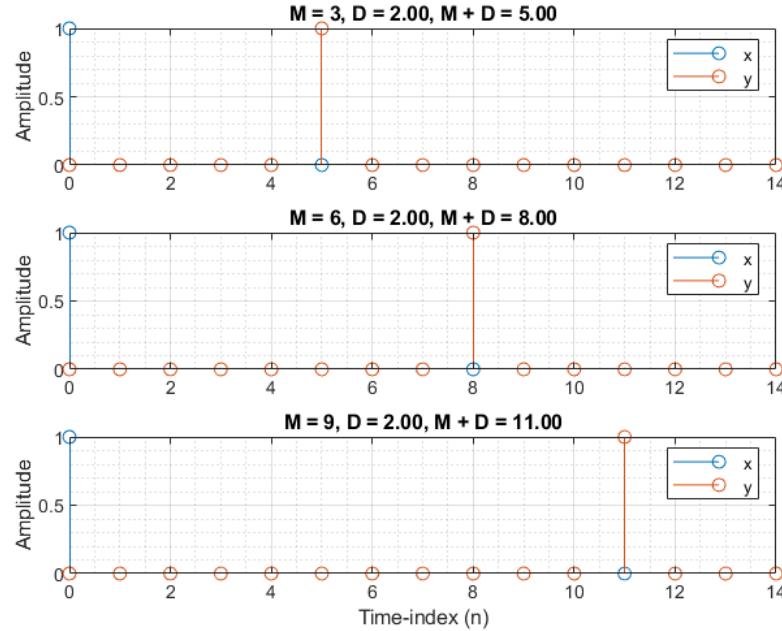


Fig. 4.30 Test results of integer values for fractional delay object. $x[n]$ is the input impulse stimuli and $y[n]$ is the resulting delayed output. The delay values are broken down above each plot.

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 4.31 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 done 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 figure 4.32. 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).

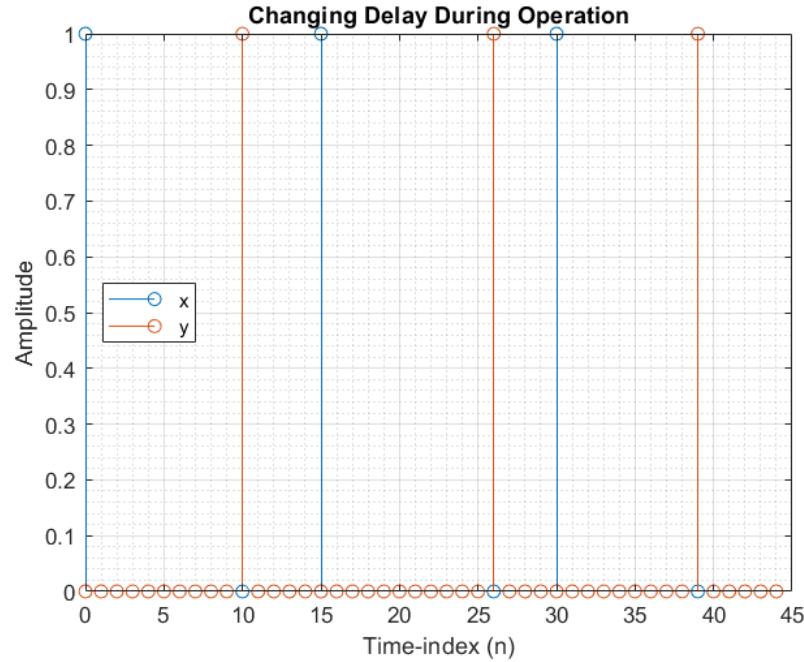


Fig. 4.31 $x[n]$ is an impulse train with a period of 15 samples and $y[n]$ represents the delay line output. The delay value starts at 10, increases to 11 at $n = 15$ and decreases to 9 at $n = 30$.

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

Table 4.1 Parameter changes for the test shown in Fig.4.32

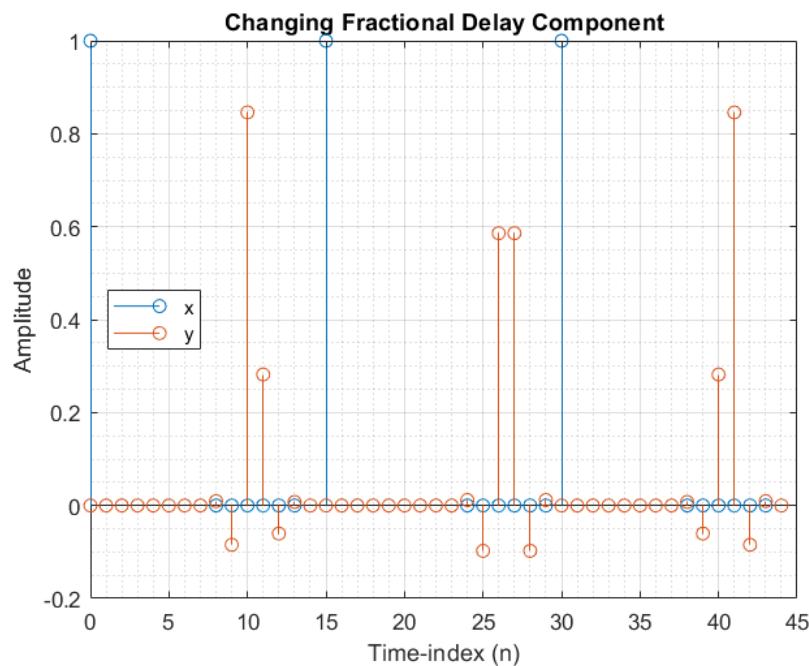


Fig. 4.32 $x[n]$ is an impulse train with a period of 15 samples used as input. $y[n]$ is the corresponding output. The parameter changes are specified in Table 4.1. The output clearly illustrates the appropriate number of integer delay values followed by the impulse response of a filter approximating the Lagrange interpolation.

4.6.2 Energy Scaler

The energy scaling coefficient is governed by the following equation:

$$g_c[n] = \sqrt{1 - \Delta x[n]} \quad (4.2)$$

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 using the derived results.

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

$$DWGLength(t) = at^2 + c \quad (4.3)$$

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

$$a = \frac{sweepEnd - sweepStart}{sweepDuration^2} \quad (4.4)$$

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

$$DWGLength[n] = DWGLength(nT_s) = a(nT_s)^2 + c \quad (4.5)$$

$$DWGLength[n - 1] = a(n^2 - 2n + 1)T_s^2 + c \quad (4.6)$$

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

$$g_c[n] = \sqrt{|1 - a(2n - 1)T_s^2|} \quad (4.7)$$

and use the expressions for a and b 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 4.33 and 4.34 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. Intuitively this is what we would expect as the change in the DWG Length on a per sample basis is a constant value and the length of the DWG is increasing. It is necessary to attenuate the energy slightly as it spreads out across the new DWG Length. The output to a quadratically increasing function is a linearly decreasing function. Intuitively this makes sense as the change in length at each time step is gradually getting larger. It is necessary to attenuate the signal more and more to maintain the same perceptual loudness as the energy spreads out more across the length of the digital waveguide.

Figures 4.35 and 4.36 illustrate the output to the same functions which are now decreasing. As is illustrated, the error remains zero and the gain operates in the same pattern but applying amplification. Amplification is required in order to maintain the same perceptual loudness as samples are being discarded by the shortening length.

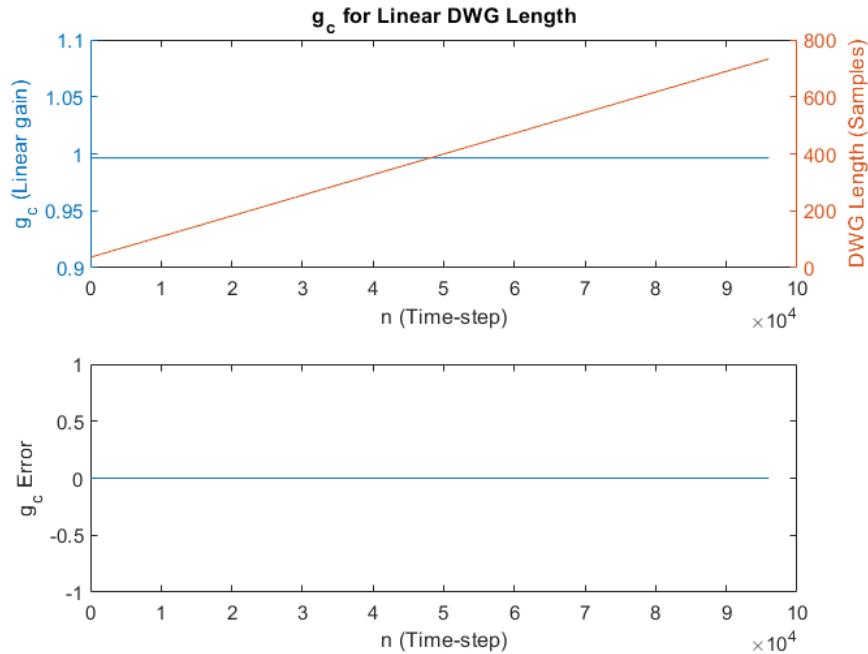


Fig. 4.33 Energy scaler output for linearly increasing input

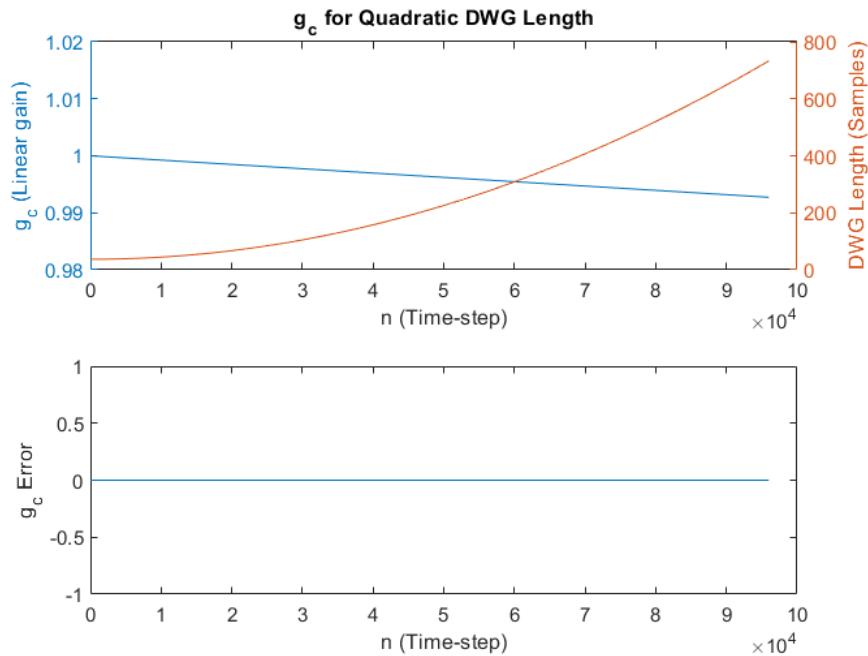


Fig. 4.34 Energy scaler output for quadratically increasing output

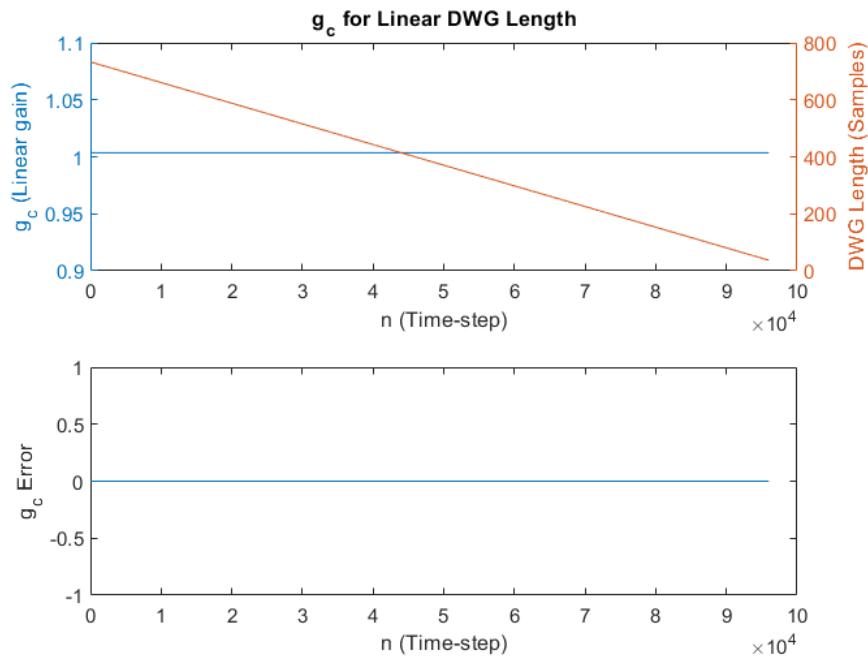


Fig. 4.35 Energy scaler output for linearly decreasing input

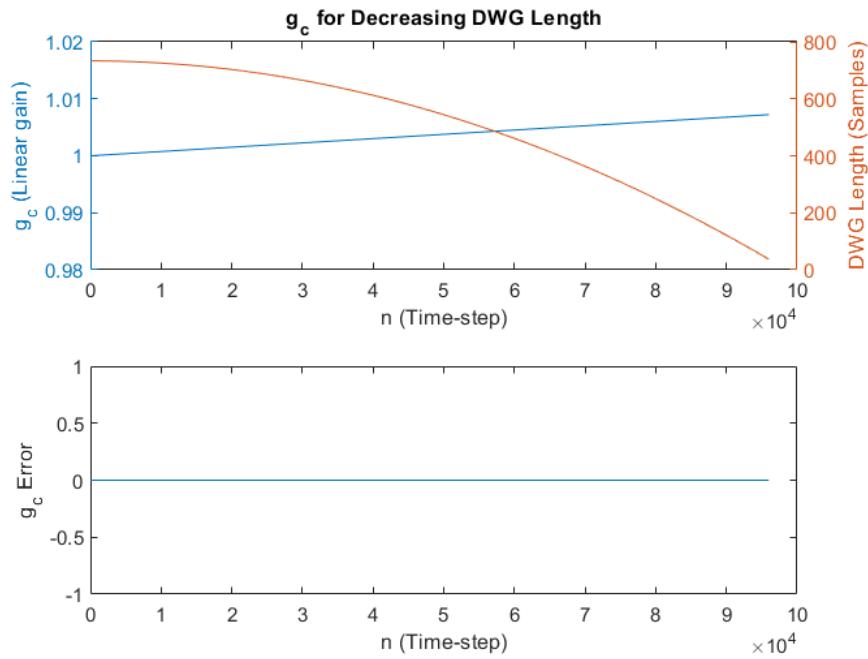


Fig. 4.36 Energy scaler output for quadratically decreasing output

4.6.3 String Digital Waveguide

The string digital waveguide model was first verified to be transient and artifact free. This was achieved by running a series of sweeps across the range of valid relative length parameters. After this was ensured, the tuning accuracy was verified via parabolic spectral interpolation as outlined in [Smith 2023b]. This method relies taking the DFT of the signal to sample its spectrum at N evenly spaced, making them harmonically related, analysis frequencies.

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 the search range of the algorithm is limited $1.5 * F_{0_{bin}}$, 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 the signal, the following STFT analysis parameters were used:

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

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 8.3499×10^{-5} . In hertz this is 9.7850×10^{-4} . As this is extremely small, the tuning was considered to be accurate and verified. Figures 4.37, 4.38 and 4.39 illustrate this process.

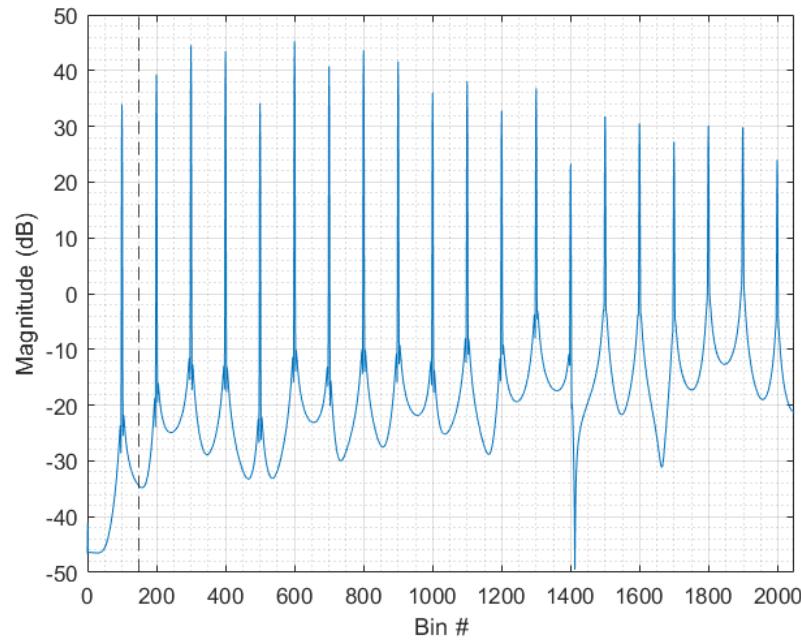


Fig. 4.37 FFT Spectrum of tone for tuning verification. note the upper harmonics which surpass the fundamental. The dashed-black line indicates the upper search limit.

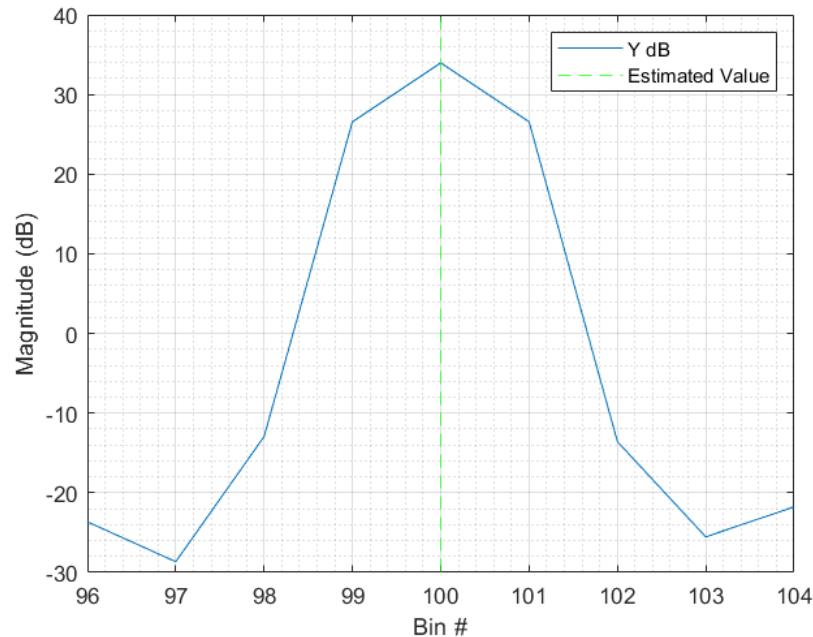


Fig. 4.38 Zoom of FFT for tuning verification.

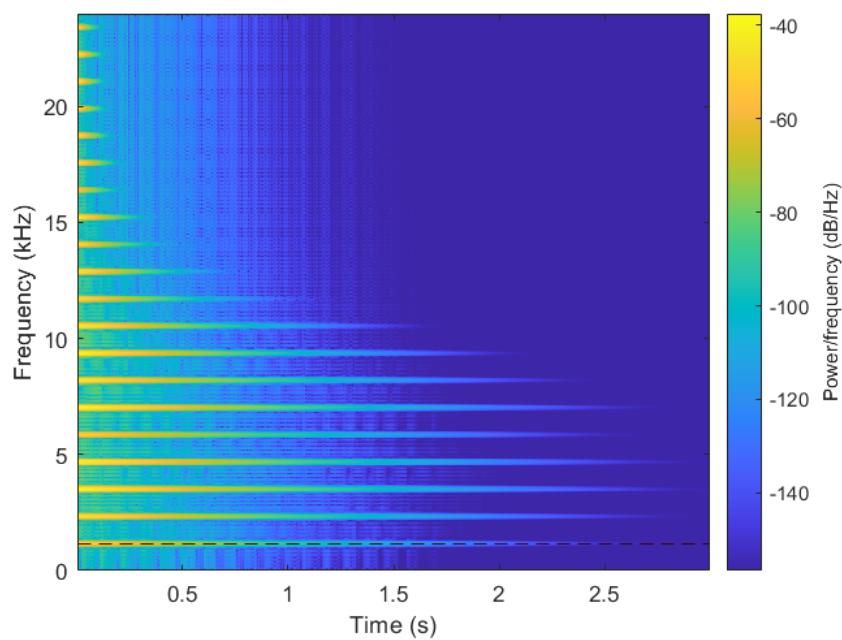


Fig. 4.39 Tuning verification spectrum. Black-dashed line is estimation of fundamental.

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 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-#-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. 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.

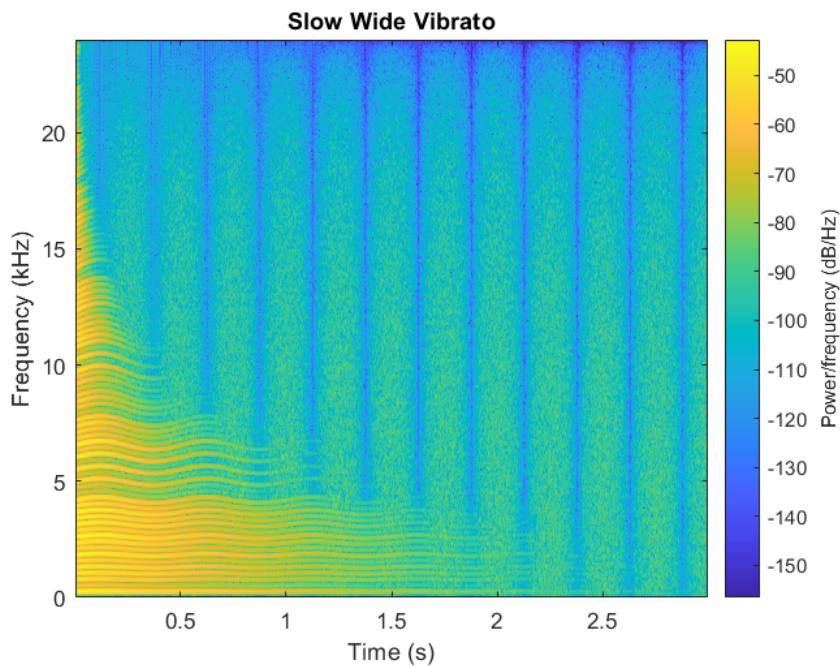


Fig. 4.40

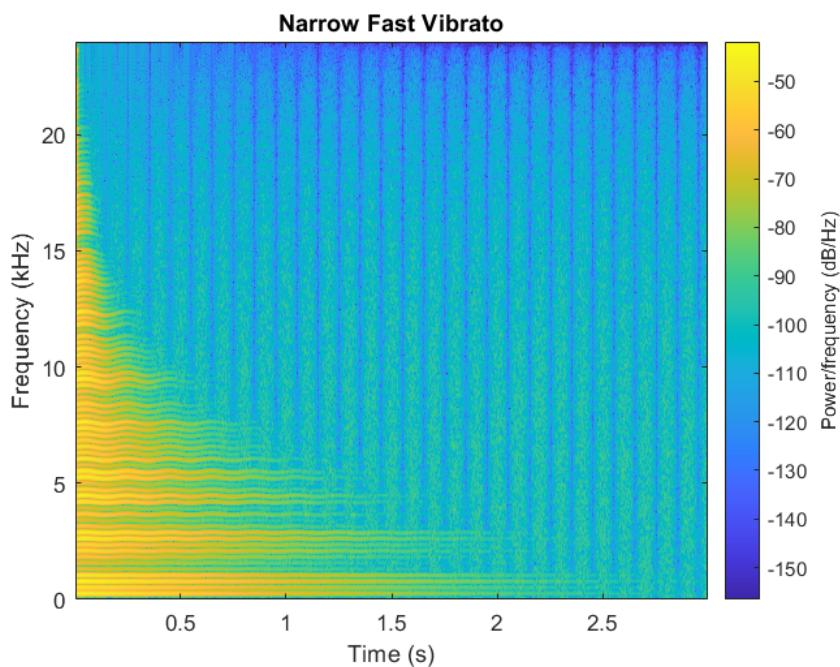


Fig. 4.41

Chapter 5

Physical Measurements

In this section, the physical measurements performed as part of the model development, will be described. These were done for two purposes. Some experiments are meant to recreate the results of the source papers which help confirm the theoretical aspects which inform the synthesis model (i.e. 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 were physically informed in the source papers (i.e. 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).

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 is 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 this was decided against as the extra mass they add to the guitar based on their placement could influence the measurements. 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 (i.e. testing for the existence of coupling), CIRMMT's *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.



Fig. 5.1 Slides used in the measurements

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 the Appendix contains the photos for all the strings measured.

String	Windings cm	Windings m
E	20	2000
A	26	2600
D	38	3800
G	64	6400

Table 5.1 Measured string winding densities

5.3 Transverse to Longitudinal 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 out 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

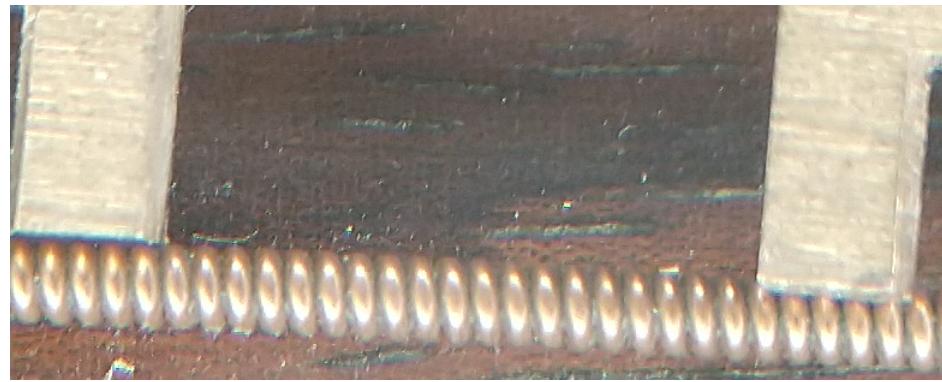


Fig. 5.2 E-string winding density measurement. Distance between inner edges of caliper is 1 cm

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. Figure 5.3 shows a photograph of the setup.



Fig. 5.3 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.

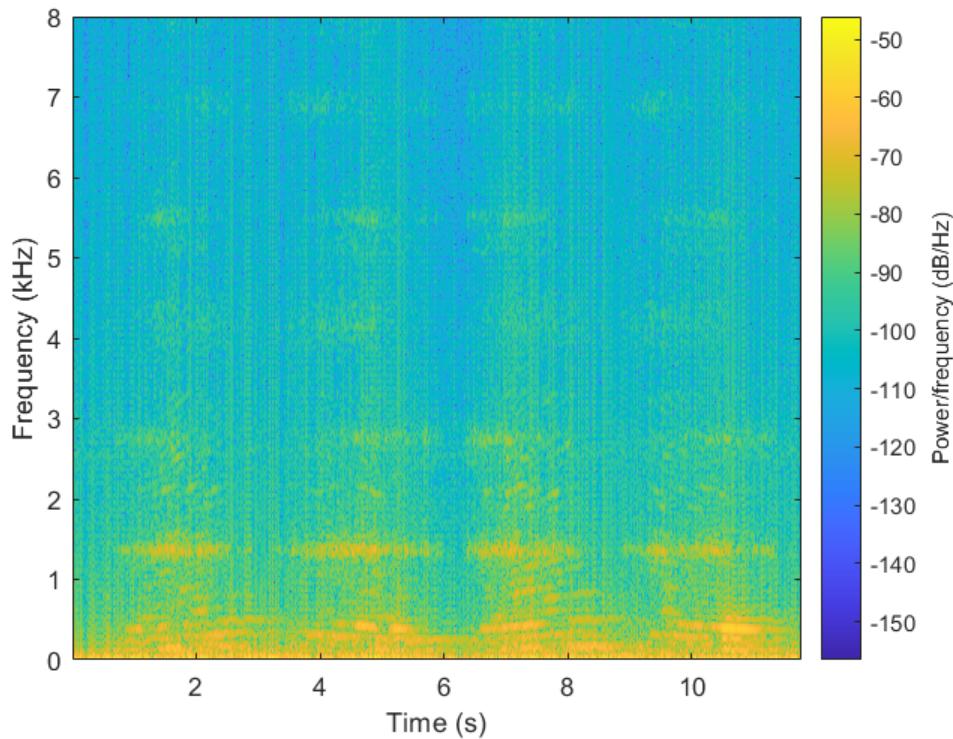


Fig. 5.4 Spectrogram of coupling measurements for the brass slide moved along the E-string

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 foam muting technique.

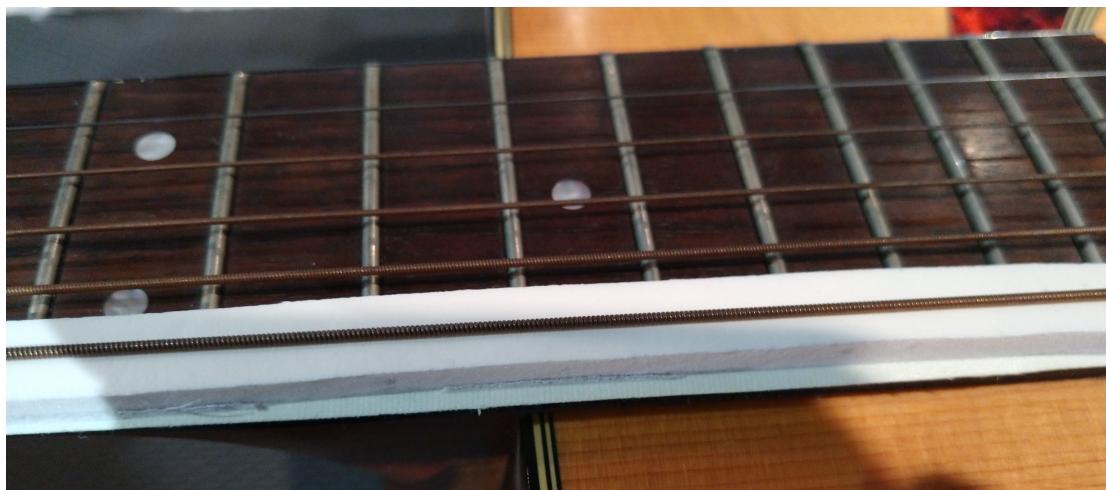


Fig. 5.5 A foam strip placed underneath the E string to dampen its transverse vibrations. Note how the string protrudes nicely without sinking too far into the foam.

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.

5.4.2 Results

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. What wasn't made clear in the previous work though, is that it appears as if there are higher longitudinal harmonics whose strength is correlated with the magnitude of the slide velocity. 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. The previous work did not attempt to control for the slide speed as much, so I suspect this is why it was not mentioned.

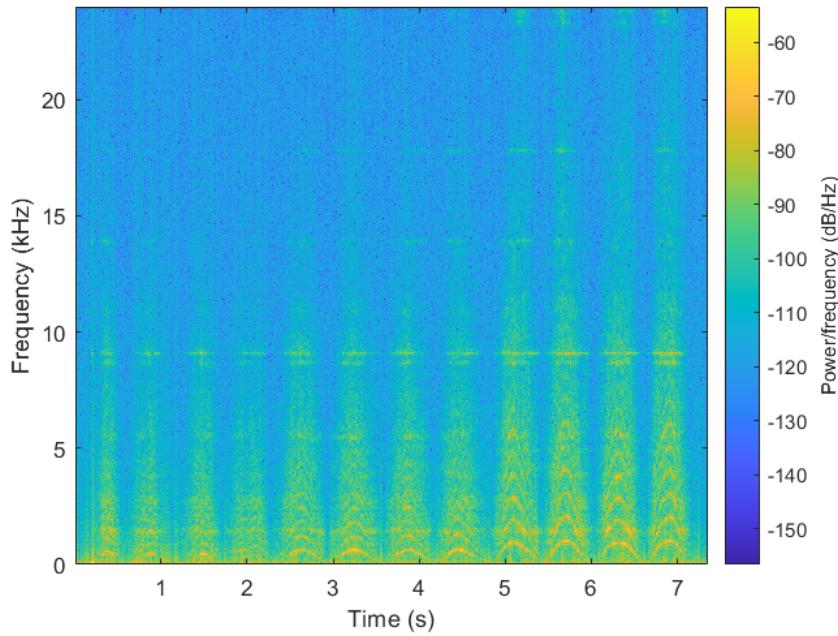


Fig. 5.6 Measured spectrum for brass slide on E string. A 12 ms rectangular window with 75% overlap was used.

It was difficult to perform this measurement on the unwound strings 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. Accordingly, the parameters of the synthesis model for unwound strings could not be as precisely confirmed.



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

5.5 Decay Rate of Single Winding Impact

Based on the material 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. Given this experiment will ultimately be used to inform the tuning process by ear, it isn't necessary to have the most precise method. 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 muting using the foam strips as in the contact sound experiments. The object used to stimulate the strings was an utility knife blade removed from its casing. Attempts were made to use it inside the casing, however this setup didn't 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 isn't to develop a precise characterization, but have a stronger physical basis for the T_{60} parameter used to control the synthesizer.

Hold and Release

Figure 5.8 shows a photograph of a preliminary experimental setup before the knife was removed from the casing and the muting foam was added. In this setup the knife's edge was wedged between two string windings. From there, force was applied in the longitudinal direction until slippage occurred and the knife edge dislodged from the string winding. At this point 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.



Fig. 5.8 Knife edge wedged in between two string windings. This method was revised to be done with the blade out of the casing and on a string muted with foam.

Strike Multiple Windings

Figure 5.9 illustrates the experimental setup for this case. The setup is extremely similar to the previous one, except for in this case the knife edge was held at an angle. Changing the position in this manner would allow it to more easily slide across the winding 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 previously designed experiments.

5.5.2 Results

Unfortunately neither of the two methods were able to be used due to issues in the measurement setup. The noise floor of the audio capturing chain made it difficult to get an accurate estimate of the T_{60} parameter. Ultimately this will be tuned by ear as will be described in the chapter on Sound Design.

5.6 Playing Examples

The last measurements made were recordings of different slide playing examples. The purpose of this was to be able to capture the different sounds for comparison purposes between the synthesized sounds and the real-world examples. The spectral aspects of the sound are useful from the standpoint of determining the parameters of the synthesis model (i.e. 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

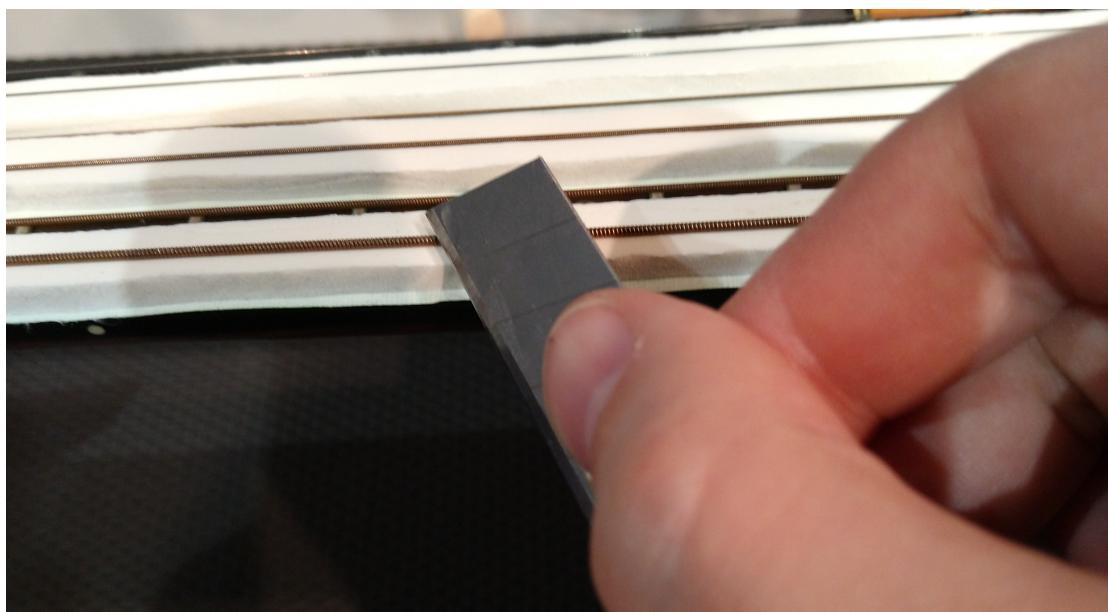


Fig. 5.9 Knife striking setup. Alternating strings have muting foam as it was easier to prepare the setup for the next string measurements.

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

Fill me in later

6.1 Alternate Architecture Considerations

6.2 Tuning Parameters

Explain how DC blocker is different than not using absolute value function in noise source

6.2.1 Decay Rate

THE MORE NOISIE LIKE THIS LETS THE NPT OUTPUT BECOME THEN THE MORE THE f_C quantization is mitigated as the filter's don't have that issue.

T_{60} is not full described in the original [Pakarinen, Puputti, and Välimäki 2008] paper. This paper describes the type of string as determining both the decay time as well as duration and an individual noise pulse. Given that the noise pulses have an exponential amplitude envelope, ultimately only one of these are needed as the decay time and duration can be derived from each other. There is no specification as to how the decay rate is specified either, so the T_{60} value was chosen as this is a widely known and standardized parameter for reverberation time. It has been adapted here to represent the amplitude envelope's time until it decays 60 dB from its original value (and does not correspond to any sort of reverberation).

The implementation of the algorithm as described in [Puputti 2010] does not clearly define how the decay rate and pulse length are specified as well. Their implementation is more akin to the guqin model ([Penttinen et al. 2006]) or noise burst approach where the stacking of the impulses is not allowed to occur in an unrestrained fashion. This makes the signal less harmonic and more noise like as the fundamental repetition of the slide/winding collision impulse response is not allowed to occur in a more sonically meaningful way.

There is also a question as to the physical interpretation of the noise pulse as it has been described originally in [Pakarinen, Puputti, and Välimäki 2008] and implemented in [Puputti 2010]. Figure 6.1 illustrates the noise pulses as originally described. It is clear here that the noise pulses do not take on negative values. This is also reflected in Figure B.21 in [Puputti 2010] where an *abs* block has been applied after the *noise* in the PD implementation.

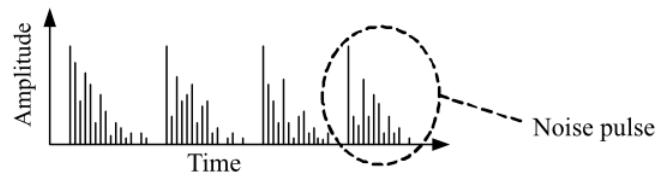


Fig. 6.1 Image of noise pulses take from [Pakarinen, Puputti, and Välimäki 2008]

The physical interpretation of a noise burst is meant to represent the impulse response of a single slide/winding impulse. However, what does an impulse response which does not take on negative values here indicate? This would seem to indicate that there is no oscillatory motion and the signal only decays. Without oscillatory motion, how can there be any sort of wave motion? Equation 1 from [Pakarinen, Penttinen, and Bank 2007] indicates that the noise pulse is meant to represent $f(t)$, which given the existence of the modes would imply some sort of a negative values. $f(t)$ represents the force that an infinitesimally small cross-section of the string would experience over time after having a collision. Additionally, this also adds a DC component to the noise source which can build up in the DWG due to the coupling between the longitudinal and transverse motion.

Tuning

Attempts were made to empirically measure the T_{60} value, however the measurement equipment was not sufficiently low noise to allow for a meaningful result to be calculated. Instead, an attempt will be made to determine the correct value based on trying to match the spectrograms where were measured. The crucial component to understanding this observing the relationship between the decay rate and the firing rate. The longer the decay rate is, then the lower the firing rate needs to be in-order to have the impulse responses overlap. The more the noise pulses overlap, the more "noise like" the signal becomes due to the build up of energy. If this build up is too much then the harmonic component of the signal is obscured. Signal-to-noise ratio could be an apt term to describe this, however the standard definition of this implies that the "noise" component is unwanted. Given that the "noise" component is useful as a stimuli for the longitudinal-mode filters which follow it in the CSG', the desirability of "noise" is ultimately subjective as it is a creative decision by the person playing the synthesizer. The T_{60} parameter ultimately controls the harmonicity to noise transition frequency for $f_c[n]$.

6.3 CSG Configurations

6.3.1 Computaional Complexity

The more harmonics which are added to the HRB. The longer it takes to compute the results.

6.4 Waveform Initialization

Two different types of noise were experimented with for specifying different initial conditions for the DWG's buffer: white and pink noise. Each noise type has a different properties regarding its frequency content. White noise contains equal frequency content across the spectrum, whereas pink noise has a spectrum where there is equal energy 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. Experiments were done with both types of noise. There is an audible difference between the two, where the white noise generated signal contains more definition/clarity in the attack. Intuitively this makes sense as the higher frequencies are necessary to create sharper transitions associated with a faster transient. The pink noise generated sounds have more of a “warmer” sound due to their stronger low-frequency content and are more natural sounding. Pink noise was used in the [Puputti 2010] implementation while historically white noise has been used as illustrated by [Karplus and Strong 1983].

TODO: Add sound examples with both types to illustrate the differences. Perhaps add spectrograms and snapshots of the waveform attack.

6.4.1 Removing DC Component from String DWG

In either case it is necessary to remove the DC component from the signal as it doesn't add anything to the sound's timbre and can cause issues with computations if it builds up too much in the digital waveguide structure. It is easiest to achieve this when the waveform is initialized in the memory where the digital waveguide is stored. The standard method of achieving this is by computing the mean of the buffer and subtracting this from the waveform. Given that the digital waveguide's length is distributed across three different components (an integer delay line, an interpolation filter and a loop filter) there is a question of where to store the initial waveform. Through experimentation it was determined that the integer delay line was the best place to achieve this. Fundamentally it is impossible to generate a waveform which has a non-integer number of samples, so this is the only buffer which is guaranteed to be able to be filled at any digital waveguide length. Additionally, the processing effects of the loop and interpolation filter are dynamic depending on the synthesis context and in cascade with each other so the effects of the interpolation filter have an impact on the sample which is stored in the loop filter.

In terms of the actual initialization, it is necessary to only initialize as many samples which correspond to the integer delay line's length for a particular digital waveguide length. 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 get played out. The others will be overwritten as the algorithm computes output samples, so they would introduce an unwanted bias in the waveform. Contrary to white noise, pink noise is correlated with itself. So generating more samples than is necessary would ultimately not result in pink noise as the other samples would never be played. Beyond that, it would be necessary to remove the bias from the entire generated waveform, but given that part of it would never be accessed, an unwanted DC bias would be introduced into the component which is played out. This would be introducing the exact thing which we were trying to prevent.

TODO: ADD SOME PICTURES FOR THIS

6.5 Control Signal Parametrization

6.5.1 Generating L[n]

A rudimentary algorithm for generating control signals was created based on the idea of specifying linear perceived pitch trajectories as the ear perceives. This was done using Algorithm 1, shown below, which calculates $L[n]$ from a specified sampling frequency, duration in seconds as well as starting and ending string lengths.

Algorithm 1 Generate $L[n]$ from specified end points

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

$$frequencyRatio = \frac{L_{init}}{L_{end}}$$

$$numSamples = F_s \times duration$$

▷ $duration$ is in seconds

$$sampleRatio = frequencyRatio^{\frac{1}{numSamples-1}}$$

▷ Ratio change in $L[n]$ per each sample

$$n \leftarrow 0$$

for $n < numSamples - 1$ **do**

$$L[n] \leftarrow L_{init} \times sampleRatio^{-n}$$

6.5.2 Analysis of Signals from Playing Examples

An attempt was made to synthesize the following musical segment.



Fig. 6.2 Slide playing example for the high E string.

This example was also played and recorded to be able to compare the difference between the synthesized figures and the real playing examples. This occurred across multiple different parameters, but the primary one of interest was the accuracy in terms of the specified control signal. The goal of a musical synthesizer is to be able make musical sounds (something which is fundamentally subjective), however one way of determining the “musicality” of a synthesized sound would be to compare it to a recorded example.

Comparing an artificially constructed $L[n]$ to an extracted one is one way of doing this. This is true as there is only one control signal and ultimately one of the defining characteristics of slide guitar is that this can take on values which normally aren't achievable by frets. The recorded example was run through the YIN algorithm and its output is shown in figure 6.3. The blue lines represent areas where the YIN estimation is most reliable, where the green is less reliable and yellow is the least reliable. The note attacks as well as decay at the end are less reliable as they contain less periodic content which is necessary to estimate a fundamental frequency.

The files being compared can be heard in *GETFILENAME.wav* and *GETFILENAME.wav*.

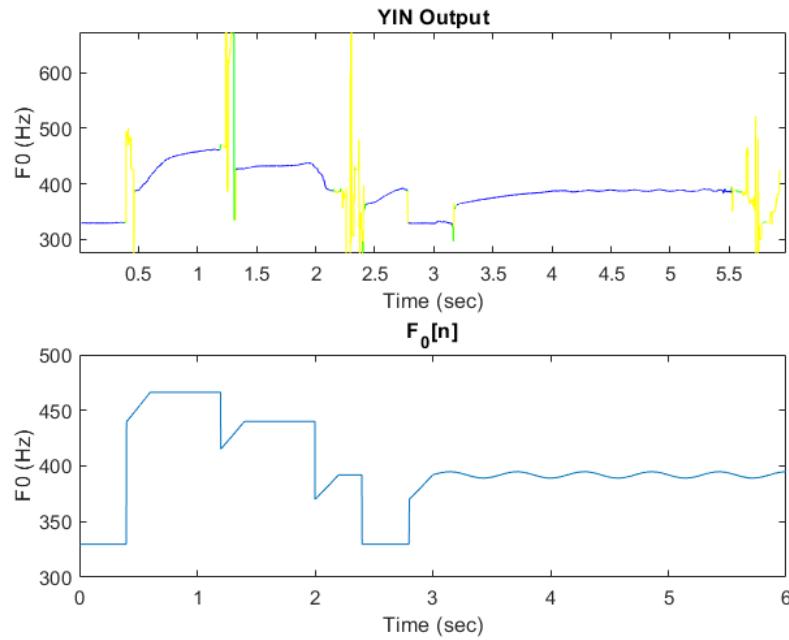


Fig. 6.3 Comparison of Recorded Example vs. Synthesized Signal F0

One of the main things to note is the transitions between notes are substantially more nuanced in the YIN analysis of the recorded playing example. This is due to the human element factor and the fact that no one person ever plays the performance the same way twice. Determining an algorithm which would match these curves would be quite a difficult task. It would be easier to just extract this signal and attempt to use it as a parametrization curve for the component.

The second plot in the figure illustrates the fundamental frequency which should be generated based on synthesis parameters specified in the example. The slides were approximated by using the *generateLCurve()* (ADD BACKGROUND INFO ON THIS FUNCTION) but by specifying that the slide is a one fret slide from below which has a duration of a 16th note. The results are something which sound very much like it was controlled and generated by a computer. The YIN output clearly indicates far more nuance in the different types of slide approaches to notes as well as their articulations. An area of potential future research would be attempting to classify different sorts of slide articulations and algorithmically generating them. Another approach would also be developing an algorithm to cleanly extract the $L[m]$ signal from an recorded example to be able use it to control the algorithm. At points they seem logarithmic (i.e. first big slide from .5 to 1.25 sec), however given the myriad of ways in which articulations can be achieved via a slide, codifying one approach seems to be a poor decision.

The slide transitions are linear in the computed approach for some reason, even though the underlying algorithm has been specified to operate logarithmically. Although for small enough segments it is approximately linear and we are considering only a 1-fret slide in from below over the course of a 16th note at 75 BPM. This note duration is approximately 47 milliseconds.

6.5.3 Purposeful Instability

As detailed in [Välimäki and Tolonen 1998], the a and g coefficients for this filter were derived from recordings of a professional guitar player playing several notes on all the frets of a guitar for each string. Unfortunately there is no mention of numerically how many frets were on the the guitar used in the recordings. Furthermore, there is no standardization of fret-numbers agreed upon by guitar manufacturers. Common values range are 19, 21, 22 and 24 depending on the make and manufacturer. Without a clear number of frets from which the measurements were made, it is hard to establish what range of values the relative length signal could take. This creates there situation, where physically valid fretting options create an unstable system in terms of the loop filter in certain situations. This is further complicated by the fact that slides are often used to play notes above the range of the end of the fingerboard.

Figures 6.4 and 6.5 illustrate this scenario more clearly. In both these figures, the magnitude response for the relative length setting of .25 (which corresponds to the 24th fret) goes above 0 dB at certain frequencies. This creates a positive feedback loop where the total amount of energy in the system increases with each iteration making the system unstable. If the relative length is maintained here for too long the output will grow unbounded and explode.

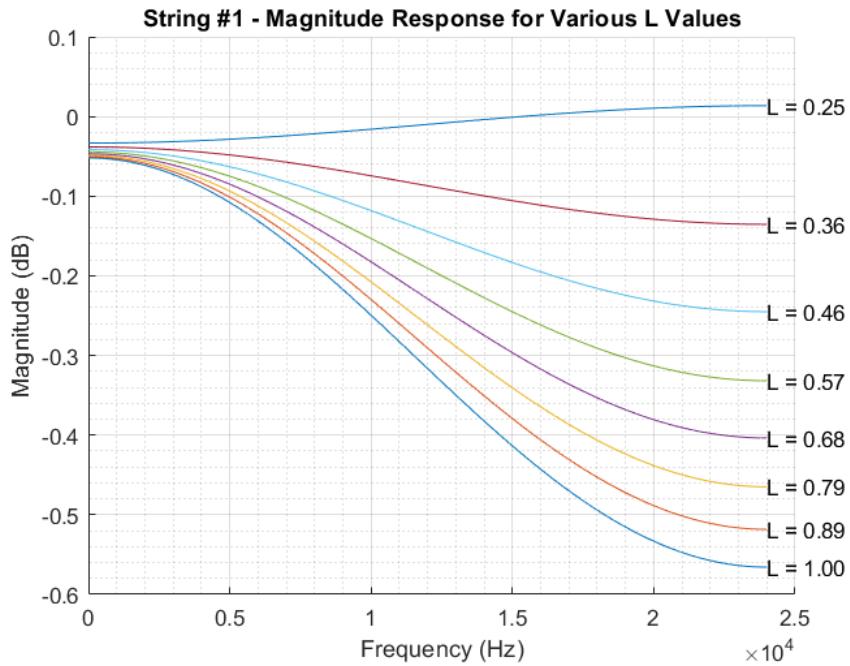


Fig. 6.4 Unstable Loop Filter magnitude response for high E string

Through experimentation it was determined that in order to guarantee that $|H_{loop}(\omega)| < 1$, the maximum fret value needs to be set to 21. This corresponds to $L \approx .30$. However, as shown in figure 6.6 this results in in a frequency response where the lower frequencies are attenuated more rapidly than the higher frequencies. While stable, this is in contradiction to how the modes

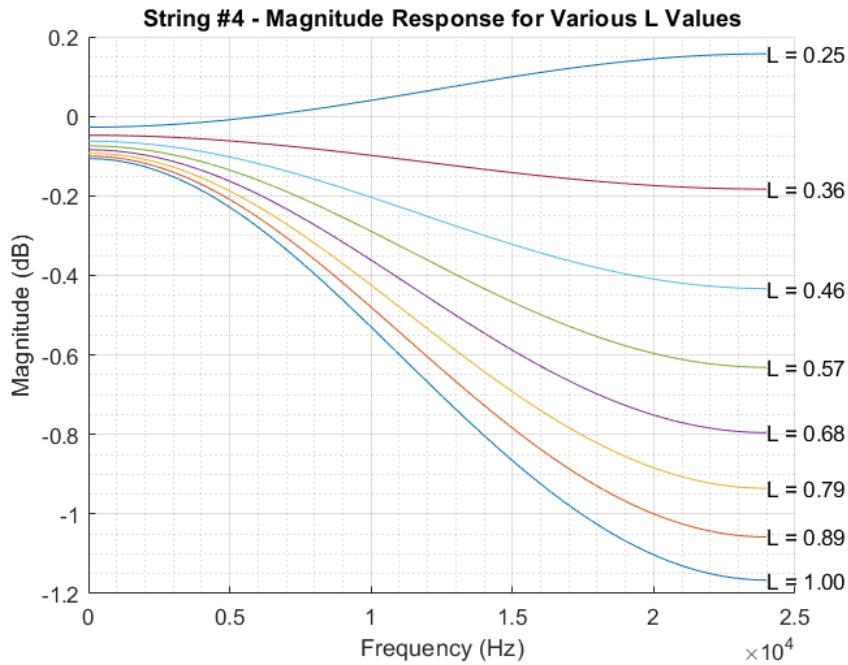


Fig. 6.5 Unstable Loop Filter magnitude response for D string

of a vibrating string are expected to decay. Even taking into account the various other losses (air damping, internal frictional forces, etc.) which the loop filter is approximating this is still not physically consistent and in opposition to the behavior illustrated at the various other relative string lengths.

Given that the filter coefficients are generated through a polynomial approximation derived from approximating the frequency response characteristics at various relative string lengths, I believe that this is an error in the process in general. It could be that the anomalous relative string lengths are not something which was captured in the original measurements. However given that the other strings do not illustrate this behavior I believe that it is more likely a limitation of the polynomial approximation used to generate the different filter coefficients. Given that the purpose of the slide is to expand the pitch palette beyond the fretboard this is a limitation when compared to the physical realities of the slide-guitar. It is often common for players to go far beyond the 24-fret and experiment with extended ranges.

In practice, to achieve this unstable state another condition needs to be imposed on the Lagrange interpolator. The total magnitude response of the loop is determined by the effects of the Lagrange interpolation filter as well as the loop filter as they are in series and the integer delay line has a unity gain. When approximating a fractional delay, the Lagrange interpolator tends to act as a low-pass filter whose attenuation at the higher-frequencies is greater than the amplification of the loop filter in an unstable state. However, if not required, then the Lagrange filter will act as a pure integer delay and have a flat magnitude response and the overall system would be unstable due to the aforementioned positive feedback. This condition occurs when the

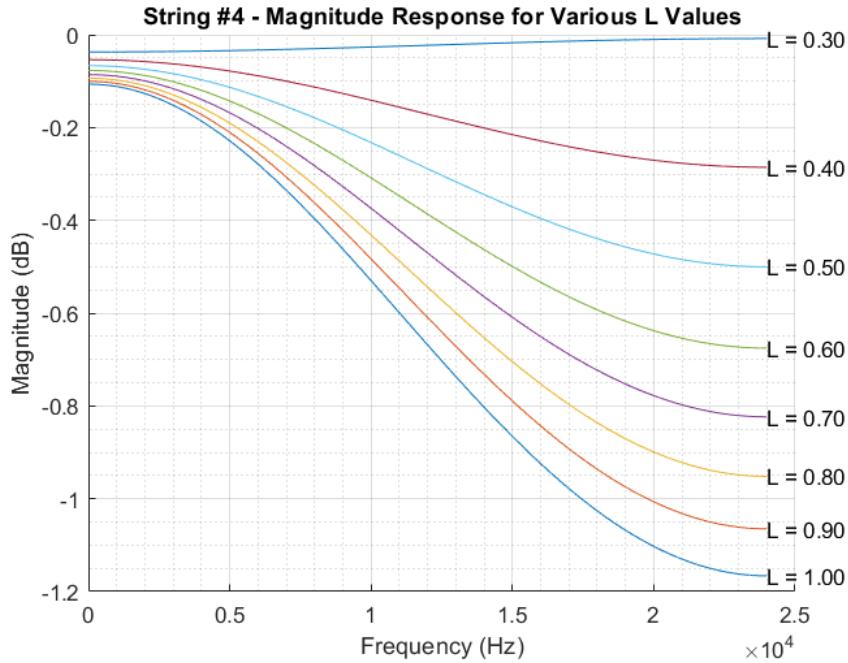


Fig. 6.6 Stable Loop Filter magnitude response for D string

fractional component of the total digital waveguide length can be achieved via the via phase delay of the loop filter. The relative string length can be expressed as:

$$L = \frac{\text{OpenString}F_0}{F_s} \times \text{DWGLength} \quad (6.1)$$

For a given open-string fundamental frequency and sampling rate, the appropriate digital waveguide length needs to be selected. For the D-string running at 48kHz and using an approximation of .25 for the loop filter's phase delay, the following calculation produces an unstable system without going beyond the 24th fret:

$$L = \frac{146.83}{48,000} \times 82.25 \approx .2516 \quad (6.2)$$

The results of this system can be heard in the example *UnstableLoopFilter-scaled.wav* and are illustrated in figure 6.7. This clearly shows how the upper frequencies are amplified over time as the system maintains an unstable state. The system grows quite rapidly near the end of the signal so in order to save it as a wave, it had to be scaled to prevent clipping. It is rather difficult to hear given the high contrast in the different signal levels. In a real-time system this would likely manifest as clipping so correspondingly a clipped signal was output and can be heard in the file *UnstableLoopFilter-clipped.wav*.

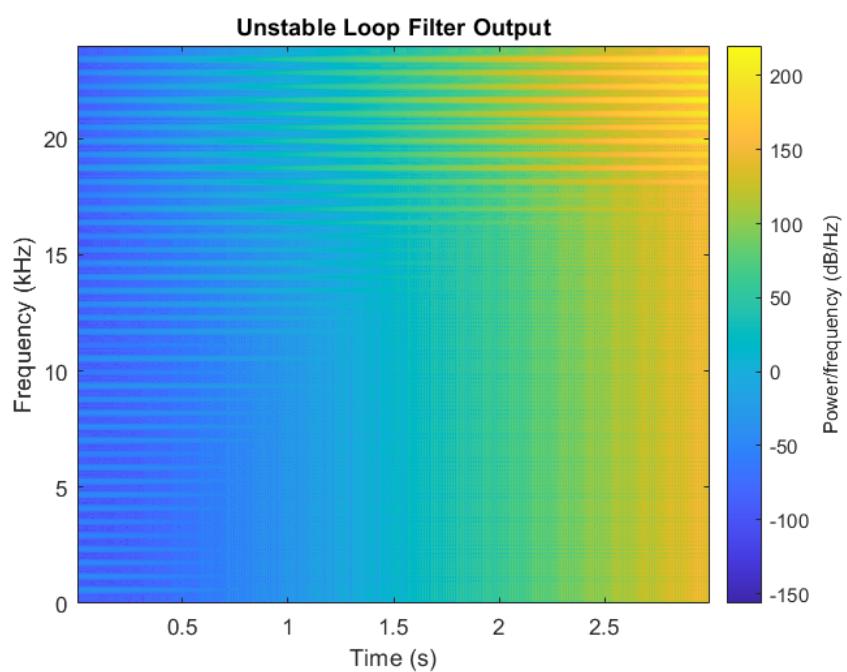


Fig. 6.7 Corresponds to *UnstableLoopFilter-scaled.wav*

Chapter 7

Conclusion

Fill me in later

7.1 Future Work

Interpolated look-up table for tuning phase delay.

Refine loop-filter to not go above 1 for frequency response.

Measure the difference between fresh strings and older corroded strings and the impact it has on the sound and coupling (rusted unwound strings might have longitudinal mode stimulation).

Devise measurement method to capture the contact sound generated by the unwound strings.

Use a harder foam to mute the transverse vibrations?

Add more control over the dynamics of the plucked sound

Add more measurements to more precisely determine the spectral characteristics between different slide and string surfaces of various conditions (brand new, old, a bit of finger grease/sweat).

Determine the precise coupling method and refine the computational model.

Use a BLIT to be able to deal with quantization of f_c

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.
- 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 : illustrations ; 28 cm. 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>.
- 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.
- 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. 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.
- Laakso, Timo I. et al. (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.
- 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, et al. (2005). "Energy Behavior In Time-Varying Fractional Delay Filters For Physical Modeling Synthesis Of Musical Instruments". In: IEEE; 2005, pp. 4–4. ISBN: 0-7803-8874-7.
- Pakarinen, Jyri, Henri Penttinien, 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, Tapani 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 Tapani Puputti (June 2008). "Slide Guitar Synthesizer with Gestural Control". In: *NIME*.
- Penttinien, Henri et al. (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.
- Puputti, Tapani (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 (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

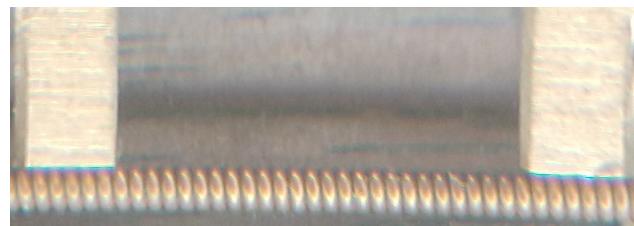


Fig. A.2 Photo of the A string winding count for 1 cm



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

Appendix B

Class Hierarchy

Fill me in later

Appendix C

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>