

# **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

March 2023

---

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

© 2023 David Smith

---

# Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
<b>2</b>	<b>Background</b>	<b>2</b>
2.1	What is the slide guitar? . . . . .	2
2.2	What is physical modeling and digital wave guides? . . . . .	4
2.2.1	Fundamental Components . . . . .	5
2.2.2	Applied to String Modeling . . . . .	7
2.3	Development of a basic slide guitar digital wave guide model . . . . .	7
2.3.1	Energy Compensation . . . . .	8
2.3.2	Loop Filter . . . . .	8
2.3.3	CSG Development . . . . .	8
2.4	Other Developments and Approaches . . . . .	15
<b>3</b>	<b>Description of Slide Guitar Synthesis Model</b>	<b>16</b>
3.1	Introduction of Architecture . . . . .	16
3.1.1	Diagram Conventions . . . . .	16
3.1.2	Single String Slide Synthesizer . . . . .	17
3.1.3	Control Signal Processor . . . . .	18
3.1.4	String DWG . . . . .	19
3.1.5	Contact Sound Generator . . . . .	21
3.1.6	Generating Control Signals . . . . .	26
3.2	Limitations of Model . . . . .	26
3.2.1	Loop Filter Magnitude Response . . . . .	26
3.2.2	Non-constant Phase Delay of Filters . . . . .	30
3.2.3	Bandwidth Limitations . . . . .	31

---

<b>4 Verification Of Slide Model and Constituent Components</b>	<b>36</b>
4.1 Filters . . . . .	36
4.1.1 Resonator . . . . .	37
4.1.2 Longitudinal Modes . . . . .	37
4.1.3 DC Blocker . . . . .	40
4.1.4 Smoothing Filter . . . . .	40
4.2 CSP and Components . . . . .	41
4.2.1 Components . . . . .	41
4.2.2 Control Signal Processor . . . . .	43
4.2.3 Loop Filter . . . . .	43
4.3 Noise Generation Objects . . . . .	47
4.3.1 Impulse Train . . . . .	47
4.3.2 Exponential Decay . . . . .	47
4.3.3 Noise Pulse Train . . . . .	50
4.3.4 Noise Burst Generator . . . . .	52
4.4 Harmonic Accentuators . . . . .	54
4.4.1 Harmonic Resonator Bank . . . . .	54
4.4.2 Reso + Tanh . . . . .	54
4.5 Contact Sound Generator . . . . .	55
4.5.1 Wound Variant . . . . .	56
4.5.2 Unwound Variant . . . . .	61
4.6 String DWG and Components . . . . .	61
4.6.1 Interpolated Delay Line . . . . .	61
4.6.2 Energy Scaler . . . . .	65
4.6.3 String Digital Waveguide . . . . .	69
4.7 Slide Synthesizer . . . . .	72
<b>5 Physical Measurements of Slide Sounds</b>	<b>74</b>
5.1 Experimental Setup . . . . .	74
5.2 String Winding Density . . . . .	75
5.3 Transverse to Longitudinal Coupling . . . . .	76
5.3.1 Setup and Method . . . . .	77
5.3.2 Results . . . . .	77
5.4 Contact Noise Spectrum . . . . .	78
5.4.1 Method . . . . .	78
5.4.2 Results . . . . .	79

5.5	Decay Rate of Single Winding Impact . . . . .	81
5.5.1	Methods . . . . .	81
5.5.2	Results . . . . .	83
5.6	Playing Examples . . . . .	84
<b>6</b>	<b>Sound Design and Controlling Parameters</b>	<b>85</b>
6.1	Alternate Architecture Considerations . . . . .	85
6.2	Tuning Parameters . . . . .	85
6.2.1	$T_{60}$ . . . . .	85
6.3	Waveform Initialization . . . . .	87
6.3.1	Removing DC Component from String DWG . . . . .	88
6.4	Control Signal Parametrization . . . . .	89
<b>7</b>	<b>Conclusion</b>	<b>92</b>
7.1	Future Work . . . . .	92
<b>A</b>	<b>String Winding Measurement Photos</b>	<b>93</b>
<b>B</b>	<b>Class Hierarchy</b>	<b>94</b>
<b>C</b>	<b>Code</b>	<b>95</b>

# List of Tables

4.1	Parameter changes for Lagrange test #3 . . . . .	63
5.1	Mass of various slides used in grams . . . . .	74
5.2	String Winding Densities . . . . .	76

## List of Acronyms

CSG	Contact Sound Generator
DMI	Digital Music Instrument
DWG	Digital Waveguide
FIR	Finite Impulse Response
IIR	Infinite Impulse Response

## Chapter 1

# Introduction

Fill me in later

## Chapter 2

# Background

Fill me in later. In this chapter we give an overview of what the slide guitar is as well as 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 development of a basic outline for a slide guitar model will be developed. This will be a basic overview as more details will be developed later in the thesis. The end of provides an overview and comparison of more recent slide guitar modeling developments.

### 2.1 What is the slide guitar?

ADD SOMETHING ABOUT GUITAR ANATOMY, INCLUDE THE DEFINITION OF SCALE LENGTH

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 acts as a string termination and influences the vibration of the string by creating a new load termination in between the nut and the bridge (**evangelista’physical’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

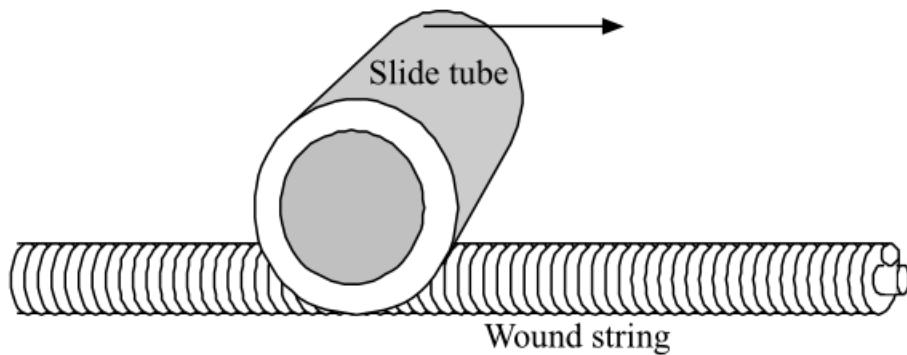
related to the slide's velocity (**pakarinen'virtual'2008**). Figure 2.1 shows a player using a chrome slide on an acoustic guitar.



**Fig. 2.1** 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'analysis'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'analysis'2007**). Figure 2.2 shows a close up of a slide interacting with a wound string.

Slides are traditionally made from ceramic or metal and unwound strings are made with metal (**bhanuprakash'finite'2020**). Figure 2.3. These are smooth/polished materials. 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



**Fig. 2.2** Close up of a slide on wound string

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'virtual'2008**).



**Fig. 2.3** A selection of slides made from different materials

## 2.2 What is physical modeling and digital wave guides?

Physical modeling is a discipline which attempts to recreate physical phenomena using computational algorithms. 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.

### **2.2.1 Fundamental Components**

Three of the fundamental components of digital waveguide models are: digital filters, digital delay lines and fractional delay lines. The inter-connection of these different components can model a variety of linear wave propagation phenomena. Incorporating different noise sources and initialization waveforms facilitates an enumerable number of different sound synthesis algorithms.

#### **Digital Filters**

Conceptually digital filters are extremely similar 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 LTI systems is now made available to the algorithm designer.

They come in two varieties: FIR and IIR filters. FIR filters consist of only delayed and scaled copies of the input signal while IIR filters incorporate the out of the filter via a feedback line. This ultimately is what gives rise to the "infinite" aspect of their name.

They do suffer from the nature of transients. These occur whenever there is a change in the coefficients associated with a filter 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 in this instance as their feedback lines ultimately cause the transient to propagate for a much longer time (depending on their coefficients).

#### **Digital Delay Lines**

One of the fundamental components of DWG based modeling is the digital delay line. It's main purpose is to provide a computational model of the physical traveling wave. It represents spatial samples of the physical medium which is being modeled. 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 and  $X_s$  is the spatial sampling period. An inherent limitation to digital delay lines is the fact that the fundamental unit of discretization in the time-domain is the sample. Signals can only be delayed integer numbers of samples with these. In many physical modeling applications, this is a limitation due to the fact that the physical world and its associated problems often require knowing a physical quantity which doesn't correspond to a sampling location.

### Fractional Delay Components

Due to the limitations of purely integer based delay-lines, as mentioned in the end of the previous section, various approaches have been developed in an attempt to approximate the signal values in between samples. These approaches can be referred to as fractional delay lines. These approaches are implemented using digital filters.

One popular approach to fractional delay line implementations is the Lagrange approach. In this technique an FIR filter is used where the filter coefficients implement Largange interpolation to allow for sub-sample accuracy. They are also generated via the maximally flat criteria. The order of the filter determine the order of the polynomials involved. 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. These benefit from a constant phase delay under certain conditions.

Another popular approach is referred to as Thiran interpolation. The basis of these is the all-pass filter.

### 2.2.2 Applied to String Modeling

In string modeling the main variable of interest is often the transverse displacement of the string at various spatial locations across different points. In combination with the characteristic impedance of the string, various acoustic quantities can be derived. Accordingly, the digital wave guide model of a string consists of a digital delay line. The terminations can be represented by simple reflection coefficients. The effects of the bridge/body connection can be modeled using a loop filter to represent the appropriate losses.

Add some stuff about how controlling the string length changes the fundamental frequency.

Also add some stuff about how the length of the delay line maps on to the length of the signal.

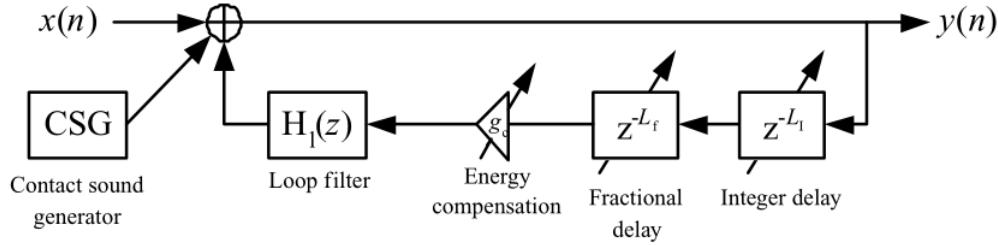
Add some stuff about how the development of the string model has been researched quite a bit, especially computational approaches to it.

Add some stuff about how the length of the string controls the pitch, from the previous paper on the development of a guitar synthesizer.

### 2.3 Development of a basic slide guitar digital wave guide model

The slide guitar model which serves as the basis for this thesis originally comes from a computer music journal described in (**pakarinen virtual 2008**). The basis for this model is the single-delay line string as described earlier with an additional component to model the slide/string surface interactions as well as energy compensation gain block. The model is shown in figure 2.4. A special loop filter was also designed where the coefficients used were generated by recording the output of a professional player, picking various notes across the length of the neck.

All the components in the model are effectively controlled by the relative length of the string. This is depicted as  $L(n)$  in the figures from the original paper. Naturally, this maps onto the notion of how a guitar works in general. Given that frets normally change the relative length of the string, with the fret spacing corresponding to the tuning system the guitar is designed for (12-TET being the most common), the relative length of the string being modeled is a natural



**Fig. 2.4** Slide guitar DWG model from ([pakarinen·slide·2008](#))

control signal. The other main command/method of interacting with this model is via a "pluck" where the waveform in string model is initialized and the simulation begins to run. In the fretted paradigm of playing this  $L(n)$  signal would be restricted to a finite set of values based on the number of frets of the guitar as well as their locations. In the slide guitar model, this is not the case. We now have an entire continuum of pitches to explore so theoretically the  $L(n)$  could take values on the interval  $(0, 1]$ . However, the limitations on the valid values are imposed by configurations of the constituent signal processing blocks (as will be discussed later in the thesis).

### 2.3.1 Energy Compensation

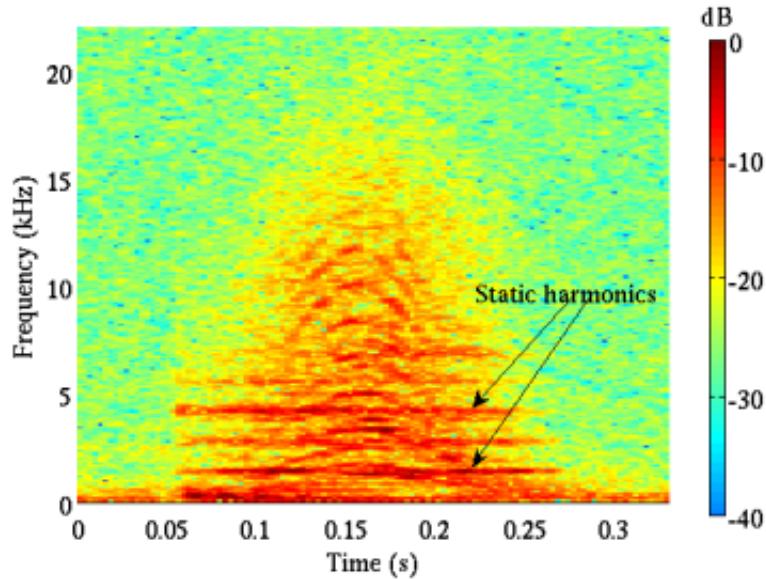
Talk about the energy compensation block here and the limitations and everything on it. Also, what it fundamentally serves to do.

### 2.3.2 Loop Filter

Add stuff in here about the loop filter, its coefficients, how they were extract and interpolated as well as what the loop filter fundamentally serves to emulate. Add the equations for the polynomial approximations.

### 2.3.3 CSG Development

The handling noises based on wound string and finger interactions have previous been investigated in ([pakarinen·analysis·2007](#)) and ([penttinen·model-based·2006](#)). The slide itself is slightly



**Fig. 2.5** Spectrogram of handling noise generated by sliding a finger on a wound guitar string from (**pakarinen·analysis·2007**)

different, however the underlying framework developed by the finger noise analysis is extremely valid and can easily be adapted to the slide scenario. Hence, we will commence with the analysis of the finger noise first.

### Finger Noise Analysis

Figure 2.5 shows the spectrogram of the noise generated by dragging a finger tip across the surface of a wound guitar string (**pakarinen·analysis·2007**). As can clearly be seen, there are two components to the sound. The first is a time-varying harmonic component which corresponds to the interactions of the finger surface with the spatially periodic windings which appear on the string. The second is a static component which is due to the longitudinal modes which are stimulated by the finger impacting the windings. Similar results have been shown in (**penttinens·model-based·2006**) where the guqin, a fretless Chinese stringed instrument, was examined. We can think of the contact noises as consisting of an exciter and resonator part. The finger-string impacts generate the excitation while the longitudinal string vibrations act as the

resonator.

An object moving along a wound string creates a harmonic force excitation to the string. This is velocity dependent and can mathematically be expressed as:

$$F(t) = \left[ \sum_k \delta(t - t_k) \right] * f(t) \quad (2.2)$$

where  $f(t)$  is the impulse response from a single finger-winding collision,  $t$  is time,  $t_k$  is the time instant where the  $k$ th impulse response is generated and  $\delta$  is Dirac's delta function. Equation 2.2 indicates we can interpret the force as a periodic impulse train being filtered by the transfer function of a single finger-to-winding collision.

The next obvious question to ask, is how are the  $t_k$  values established? To simplify, we can start by assuming the finger motion is a constant velocity. In this case  $t_k$  can be expressed as:

$$t_k = \frac{k}{n_w f_{speed}} \quad (2.3)$$

where  $n_w$  represents the linear winding density of the string (winds per meter),  $f_{speed}$  represents the finger tip speed (meters per second). The quantity  $n_w f_{speed}$  has units of windings per second and represents the number of collisions which occur at each second. We can see that by either increasing the linear winding density or finger speed we have control over the frequency at which the impulse responses are generated. From there, it is not difficult to see that as the  $n_w$  parameter is constant per string, the fundamental of the harmonics generated can be 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 wave worm. 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.

This theory can easily be verified by observing the spectrum in figure 2.5. In this figure the finger starts at rest, accelerates to reach its maximum velocity halfway through the slide, at which point it begins to decelerate as the slide comes to an end. The minimum and maximum values of

frequency trajectories corresponding to the different harmonics illustrate the same behavior in this spectrogram. It is also interesting to note that these harmonics follow a differentiable trajectory so we can conclude that the finger velocity is also differentiable and contains no discontinuities (something which is necessary to note when attempting to synthesize sounds which mimic human players).

As shown experimentally in (**pakarinen·analysis·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 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 equation 2.2.

The other component from the sound, which has been labeled in figure 2.5, is a static component due to the longitudinal modes of the string. The partial differential equation describing the longitudinal string motion, as derived in (**bank·physics-based·2006**), is the following:

$$\mu \frac{\partial^2 \xi}{\partial t^2} = ES \frac{\partial^2 \xi}{\partial x^2} - 2R(f)\mu \frac{\partial \xi}{\partial t} + d(x, t) \quad (2.4)$$

where

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

The wave propagation speed is  $c_L = \sqrt{\frac{ES}{\mu}} = \sqrt{E\rho}$ , where  $\rho$  is the density of the material. Contrary to transverse modes, the longitudinal propagation speed for a wave does not depend on tension.

If only one point of excitation exists, then the spatial force distribution can be approximated as:

$$d(x, t) = \delta(x_{exec})F(t) \text{ where } x_{exec} \text{ is the point of excitation.}$$

As shown in (**pakarinen'analysis'2007**) and (**morse'vibration'1981**), if an excitation force  $F(t)$  is applied at  $x_{exc}$ , then the bridge force can be expressed as:

$$F_b(t) = \frac{ES}{\mu L^2} \sum_{k=1}^{\infty} \left[ \frac{k}{f_k} e^{-tR(f_k)} \sin(2\pi f_k t) \right] * \left[ \sin\left(\frac{k\pi x_{exec}}{L}\right) F(t) \right] \quad (2.5)$$

where the longitudinal modal frequencies are  $f_k = k \frac{c_L}{2L}$ . Equation 2.5 clearly shows that the force signal excites a set of parallel resonances where the excitation amplitude depends on  $x_{exec}$ . In general,  $x_{exec}$  has a strong shape over the spectrum and in the cases where a mode as a node at the point, the harmonic will be eliminated. This theory was experimentally verified in (**pakarinen'analysis'2007**).

This analysis was completed based on the interactions between a finger tip and a wound string. Extrapolating this to the more rigid slide object is quite 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 equation 2.2. This would now represent the impulse response of the slide impacting a single winding.

## CSG Model Development/Analysis

The Contact Sound Generator (CSG) of the slide synthesis model can be seen as a discretization of the exciter-resonator model developed in the previous section. Figure 2.6 shows a high-level signal flow diagram of the original CSG presented in (**pakarinen'vertual'2008**).

A noise pulse train is chosen as the excitation signal, which is labeled as block (a) in figure 2.6. 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 6.1. The time-interval between noise pulses is controlled by the slide velocity as

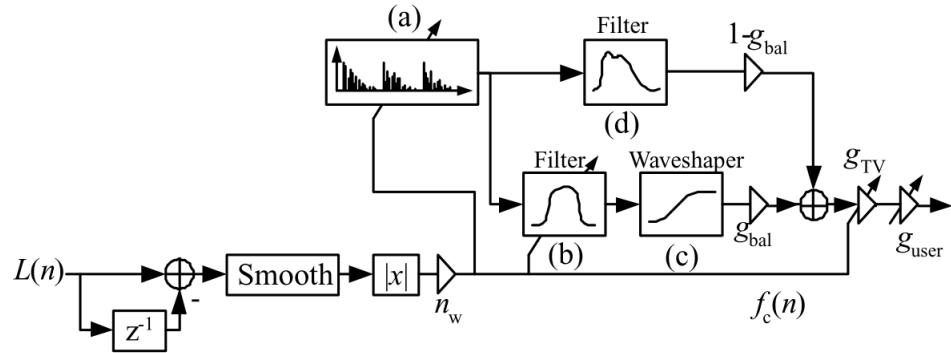


Fig. 2.6 CSG model from (pakarinen'slide'2008)

illustrated in equation 2.3. 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 rates and correspondingly durations of the noise pulses are parameters associated with each different string thickness.

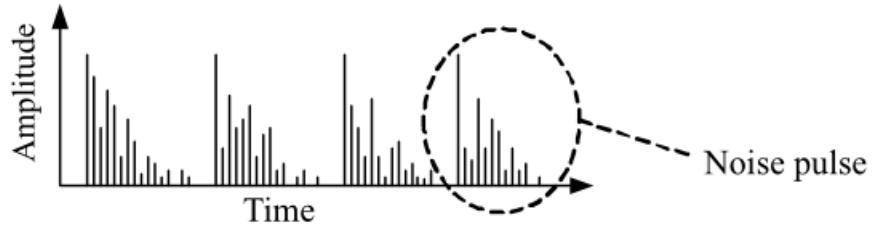


Fig. 2.7 Noise Pulses from (pakarinen'slide'2008)

At the lower-left branch of the CSG we have the control signal  $L(n)$  coming in. This represents the relative length of the string. The first-order derivative of  $L(n)$  is approximated to generate the relative slide-velocity. This is valid given that the position of the slide controls the relative length of the string. From there the signal runs through a block labeled "Smooth". This block in fact performs two operations. The first is a smoothing operation, which helps handle any discontinuities which arise during the differentiation process. This is necessary, as discontinuities rarely

happen, if ever, while play slide guitar due to the human controlled nature of the slide motion. The second operation this 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 in a more natural/gradual way as well and helps avoid transients in digital filters. In this scenario we are filtering noise 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. After this 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 2.3. Accordingly, it controls the firing rate of the noise pulse generator as well as scales the output via the gain block  $g_{TV}$ .

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. The 2nd-order resonator has its center frequency controlled by the aforementioned  $f_c(n)$  in order to extract the lowest time-varying harmonic from the noise source. The series-connection with the waveshaper creates the higher time-varying harmonics in a computationally efficient manner. The number of harmonics can be controlled via a scaling factor to the input of the hyperbolic tangent.

The upper branch serves to emulate the static longitudinal modes. The filter there is a 4th-order IIR which approximates the two most prominent longitudinal modes. 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) (**pakarinen'virtual'2008**). The  $g_{bal}$  controls the balance between the longitudinal modes and the harmonic contact sound components.

#### **2.4 Other Developments and Approaches**

## 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'virtual'2008**) and (**puputti'real-time'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.

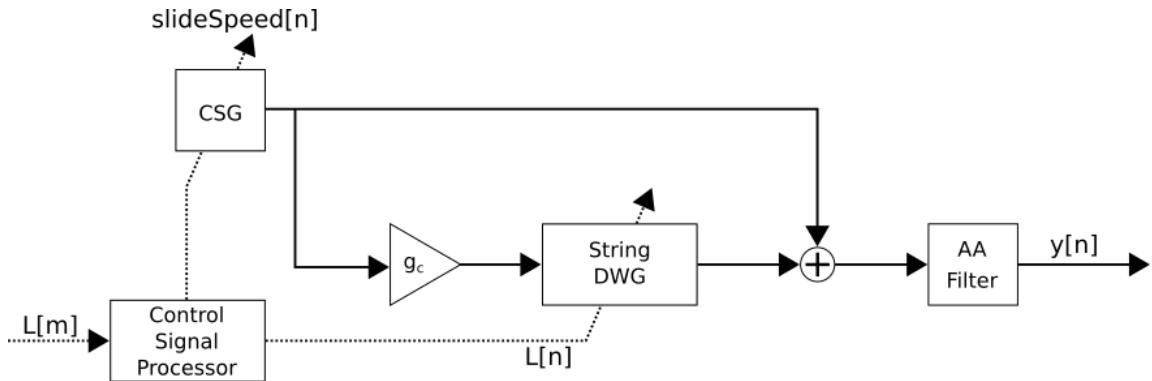
#### 3.1.1 Diagram Conventions

The following conventions will be adhered to in this sections diagrams. This was done to improve clarity and reduce ambiguity as compared to the diagrams from the original papers

(pakarinen'virtual'2008) and (puputti'real-time'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



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

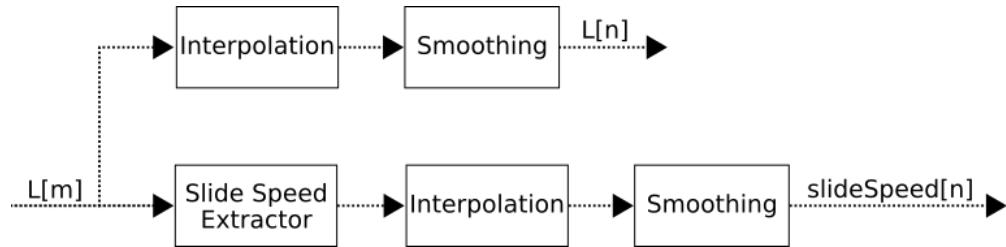
The highest-level component of the synthesis system is depicted in figure 3.1. This is a synthesizer for a single string where the pitch is controlled by a slide. Similar to the model introduced in Chapter 2, this consists of a module which represents a variable length string digital waveguide as well 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 with longitudinal motion in its algorithm. This phenomena was experimentally observed on a

per string basis, as will be shown in Chapter 3. There is also an anti-aliasing filter added to the output, as the changing of the DWG length during synthesis can be viewed as a re-sampling operation and the appropriate measures need to be in place to prevent unwanted artifacts such as aliasing.

The Control Signal Processor is another new block which will be more carefully detailed in the next section. It was placed here to remove the need for the individual objects to perform any processing on the control signals themselves. This improves computational efficiency by removing redundant computations as well as ensures all the constituent components are operating on control values derived from the same source.

### 3.1.3 Control Signal Processor



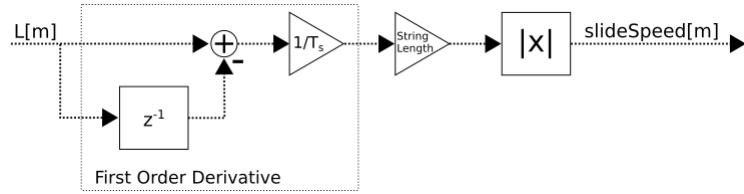
**Fig. 3.2** Signal flow diagram for control signal processor block

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 where the control-rate signals are up-sampled to the audio-rate. This allows more gradual changes to be implemented in the audio-rate objects which can help prevent 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 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

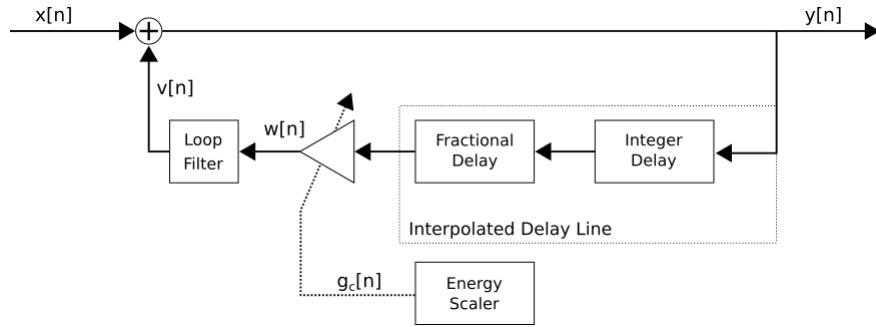


**Fig. 3.3** Signal flow diagram for slide speed extractor block

Figure 3.3 shows a signal flow diagram for the slide speed extractor. This is extremely similar to the model introduced in (**pakarinen'virtual'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 as the Contact Sound Generator is agnostic to the direction the slide moves.

#### 3.1.4 String DWG

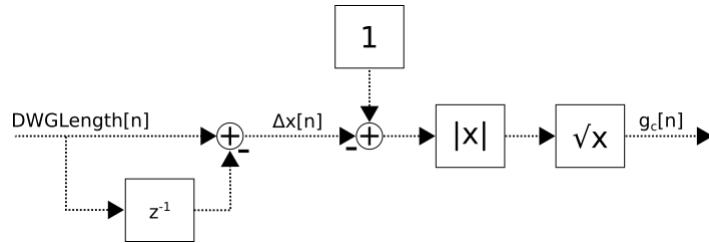
Figure 3.4 illustrates the string digital wave guide model. The model itself does not differ from the original as described in (**pakarinen'virtual'2008**). However, the diagram here differs in an attempt to improve clarity as compared to what was originally introduced.  $L[n]$  is not depicted as all the signal processing blocks 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.



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

### Energy Scaler

Figure 3.5 illustrates the signal flow diagram of the energy scaler. It implements the energy scaling as described in the Introduction chapter. The  $DWGLength[n]$  signal is derived from the  $L[n]$  signal via the equation  $Pitch_{F_0}[n]/F_s$  where  $Pitch_{F_0}[n] = \frac{OpenString_{F_0}}{L[n]}$  and  $OpenString_{F_0}$  is the fundamental frequency of the string when  $L[n] = 1$ .



**Fig. 3.5** Signal flow diagram for energy scaler block

### Loop Filter

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

$$H(z) = g \frac{1 + a}{1 + az^{-1}} \quad (3.1)$$

where  $a$  controls the cut-off frequency and  $g$  controls the gain. The  $a$  and  $g$  parameters are interpolated via a first-order polynomial as described in (valimaki'development'1998) using

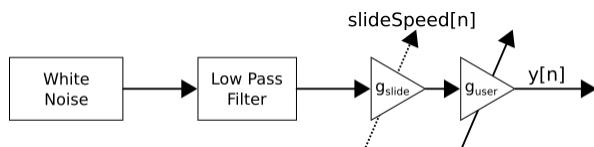
the values specified in tables 1 and 2 of the same paper.

### 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 which models the sound produced when the slide interacts with the smooth surface of an unwound string. After this, 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'virtual'2008**) and implemented in (**puputti'real-time'2010**).



**Fig. 3.6** Signal flow diagram for unwound CSG block

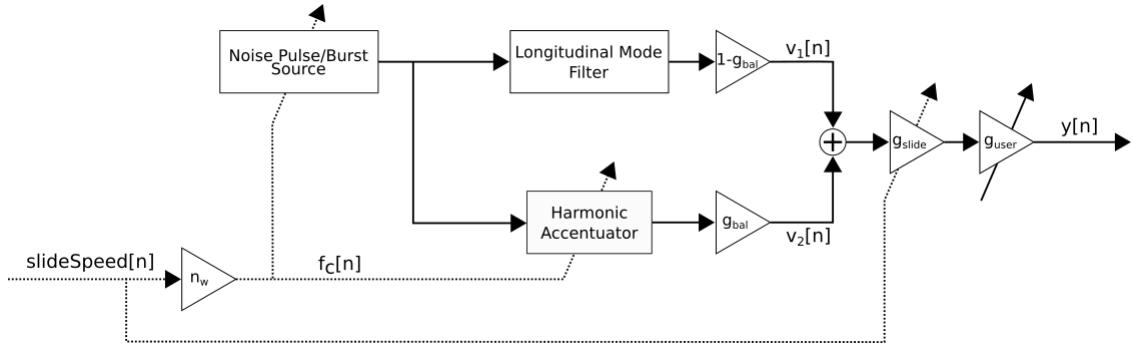
#### Wound Strings

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'virtual'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<sup>real-time</sup> 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<sup>virtual</sup> 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 allows the overall sound level to be specified, same 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.

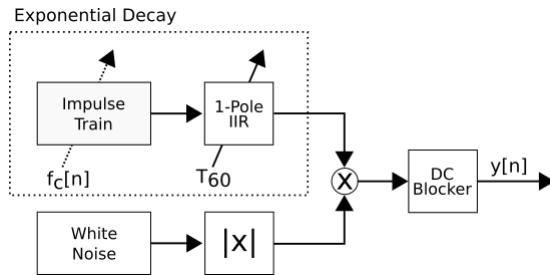


**Fig. 3.7** Signal flow diagram for wound CSG block

**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<sup>virtual</sup> 2008**), however in implementation it differs quite a bit from the version in (**puputti<sup>real-time</sup> 2010**). The second is more akin to what was introduced in the guqin model in (**penttinien<sup>model-based</sup> 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

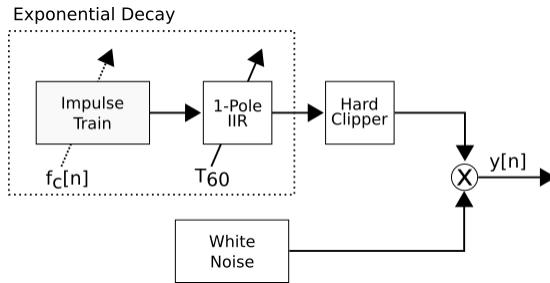
the impulse train is controlled by the  $f_c[n]$  signal which mimics the generation of impulses from the slide hitting windings as it moves. The decay rate of the impulse response of the one-pole filter is controlled by the  $T_{60}$  value measured for each string (which will be elaborated upon in 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 shown in (**penttinens model-based 2006**) with more string specific characteristics as shown in (**pakarinen virtual 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 becomes pure white noise and the harmonic component is lost. 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, it would be a pure step-function and not allow more gradual changes.

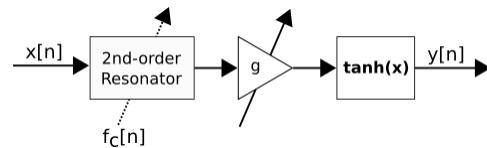
**Harmonic Accentuation** Two variations for accentuating and generating the harmonics of the wound contact sounds were investigated. The first method is the same as what is described in



**Fig. 3.9** Signal flow diagram for the Noise Burst Generator

(pakarinen'virtual'2008), while the second is more akin to the method proposed for the guquin model (penttinen'model-based'2006).

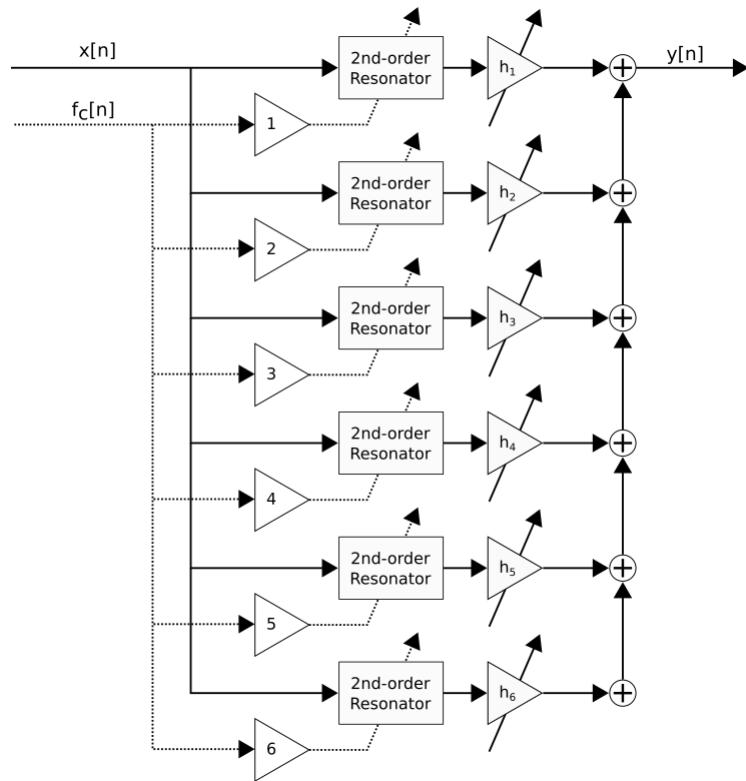
**Resonator + Tanh** The first method is a second-order resonator in series with a hyperbolic tangent function as illustrated in figure 3.10. The second-order resonator has its center frequency controlled by  $f_c[n]$  and its  $r = .99$ . This configuration allows the filter to isolate the fundamental 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 virtual 2008**).



**Fig. 3.11** Signal flow diagram for the Harmonic Resonator Bank

### 3.1.6 Generating Control Signals

TODO: FILL IN HOW CONTROL SIGNALS ARE GENREATED

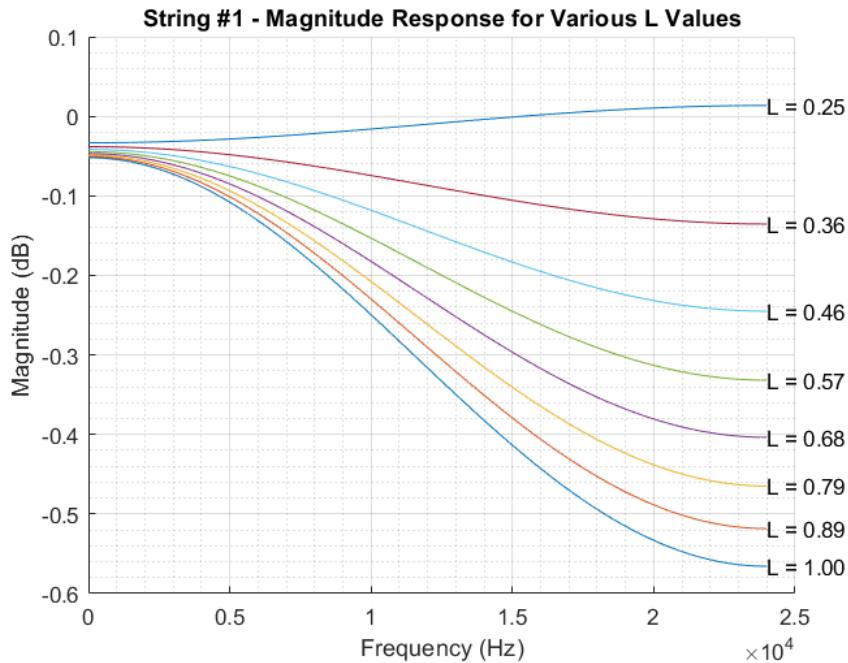
## 3.2 Limitations of Model

### 3.2.1 Loop Filter Magnitude Response

As detailed in (**valimaki development'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 3.12 and 3.13 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.

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 3.14 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 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

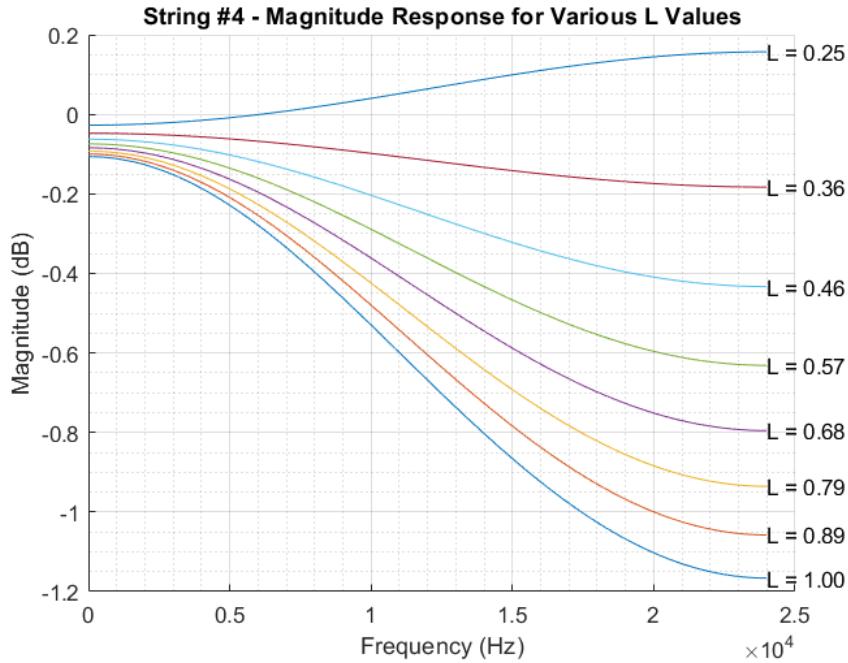


**Fig. 3.12** Unstable Loop Filter magnitude response for high E string

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

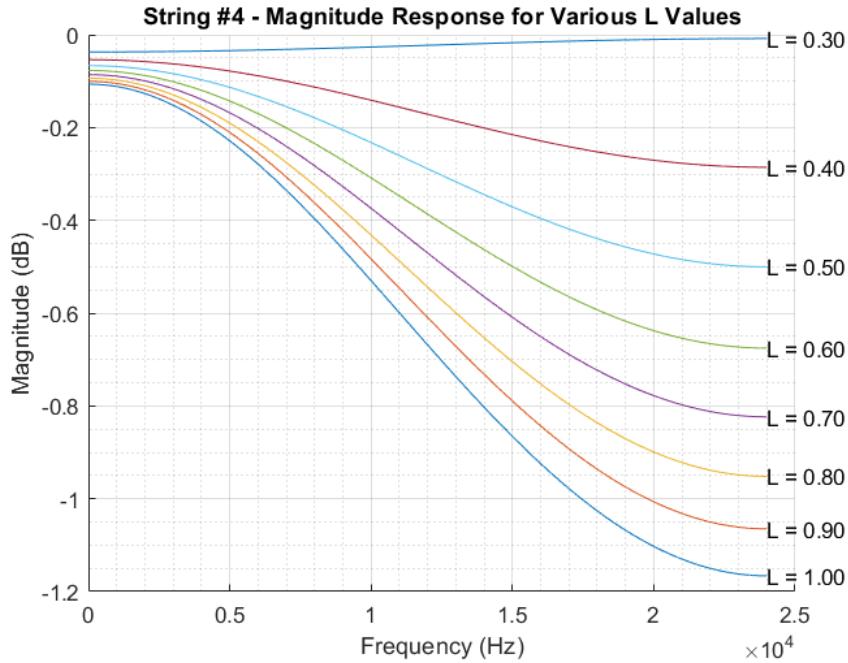


**Fig. 3.13** Unstable Loop Filter magnitude response for D string

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 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:

$$L = \frac{\text{OpenString}F_0}{F_s} \times \text{DWGLength} \quad (3.2)$$

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

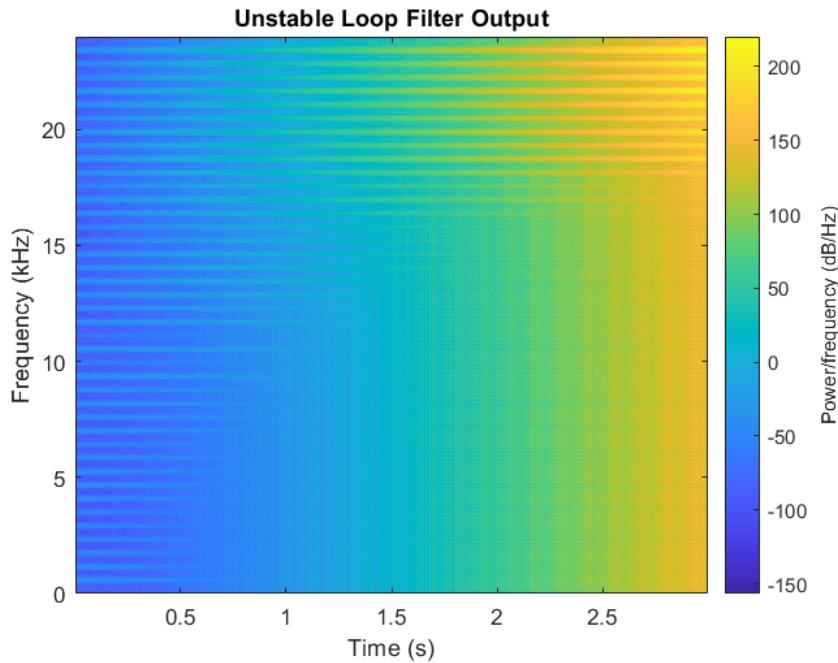


**Fig. 3.14** Stable Loop Filter magnitude response for D string

system without going beyond the 24th fret:

$$L = \frac{146.83}{48,000} \times 82.25 \approx .2516 \quad (3.3)$$

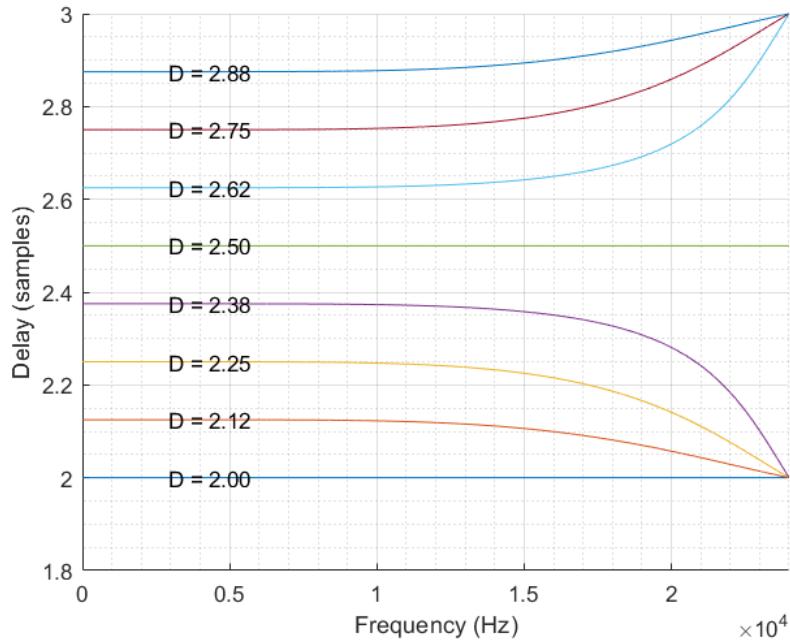
The results of this system can be heard in the example *UnstableLoopFilter-scaled.wav* and are illustrated in figure 3.15. This clearly shows how the upper frequencies are amplified over time as the system maintains an unstable state. The system grows quite rapidly near the end of the signal so in order 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. 3.15** Corresponds to *UnstableLoopFilter-scaled.wav*

### 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 3.16 and 3.17. The loop filter illustrates this as it is an IIR filter and this is inherent in their design ([Oppenheim et al. 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 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. 3.16** Phase delays for Lagrange interpolating filters with order = 5 at various delay values

### 3.2.3 Bandwidth Limitations

One limitation, which was discovered while testing the extremes values for slide conditions, is related to bandwidth limitations inherent in a digital model. As we are simulating the system digitally here, the Nyquist rate forms the upper limit on the frequencies which can be represented. For a system run at 48 kHz, this corresponds to 24 kHz. In this particular model, there is a permanent loss in harmonics when you slide upwards as well as a limitation in the number of harmonics when sliding downwards. These are illustrated in figures 3.18 and 3.19 as well as the files *ExtremeDownwardSlide-NoLoopFilter.wav* and *ExtremeUpwardSlide-NoLoopFilter.wav*. The loop filter has been disabled in order to allow the harmonics to exist for longer and help illustrate the issue. The attenuation which occurs in these examples is due to the effects of the interpolation filter as well as the energy scaler.

In order to eliminate the effects of the interpolation filter and energy scaler, the same sweep was run on a string digital waveguide model which consists only of an integer delay line. Figures

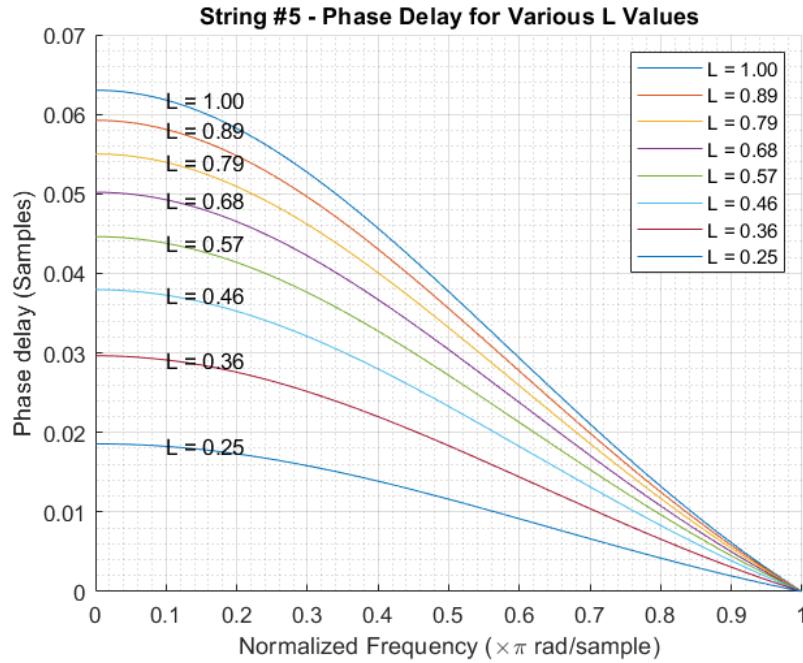
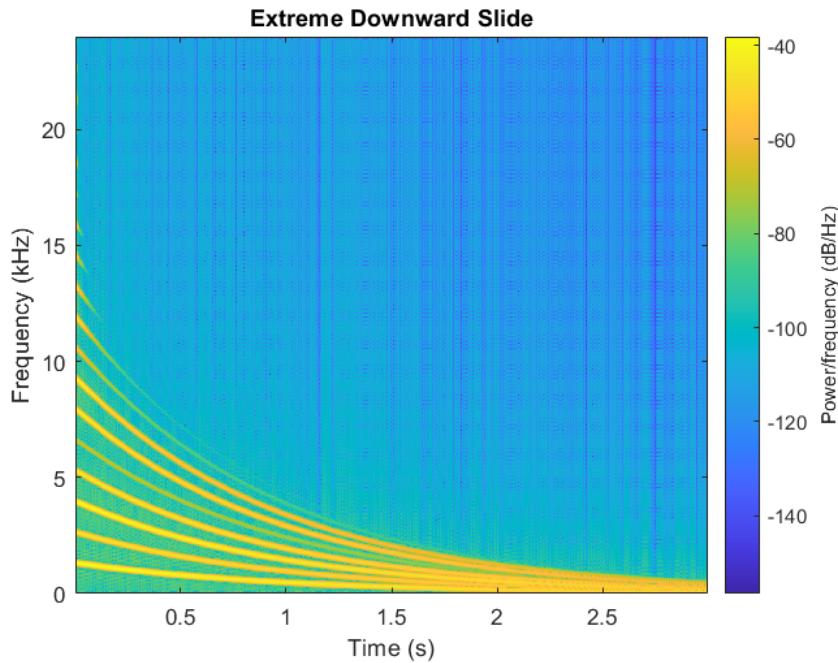


Fig. 3.17 Loop Filter phase delay

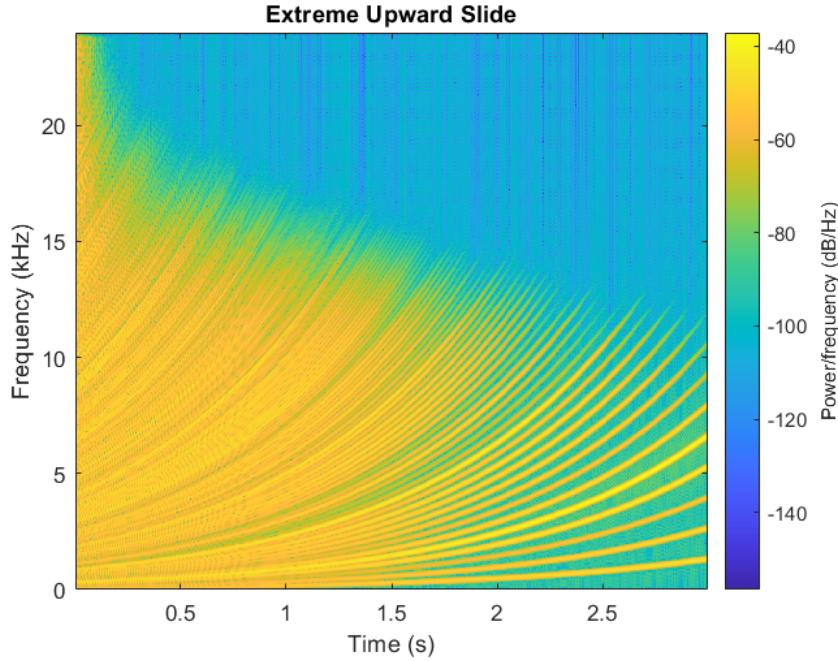
3.20 and 3.21 illustrate the spectra of this scenario. These can be heard in *StringDWGInt-Ascending.wav* and *StringDWGInt-Descending.wav*. As is clearly shown in the ascending figure, the harmonics get spaced out further apart as the fundamental frequency increases. The signal gets less spectrally dense over time as the number of harmonics is limited by the cap imposed by the sampling rate. The descending figure illustrates that the upper harmonics are gradually re-introduced as the delay line becomes longer and longer. From this, we can conclude that the loss of harmonics in the descending slide case is due to the attenuation effects of the loop and interpolation filters.

It is difficult to test the physical accuracy of the complete loss of harmonics given that there is coupling between the longitudinal motion of the slide and transverse vibrations on a real-physical system (as will be shown in the chapter on physical measurements). Completely eliminating that would require a measurement setup where the moving string termination had an extremely low-coefficient of friction in relation to the string surface. The true nature of a string with a

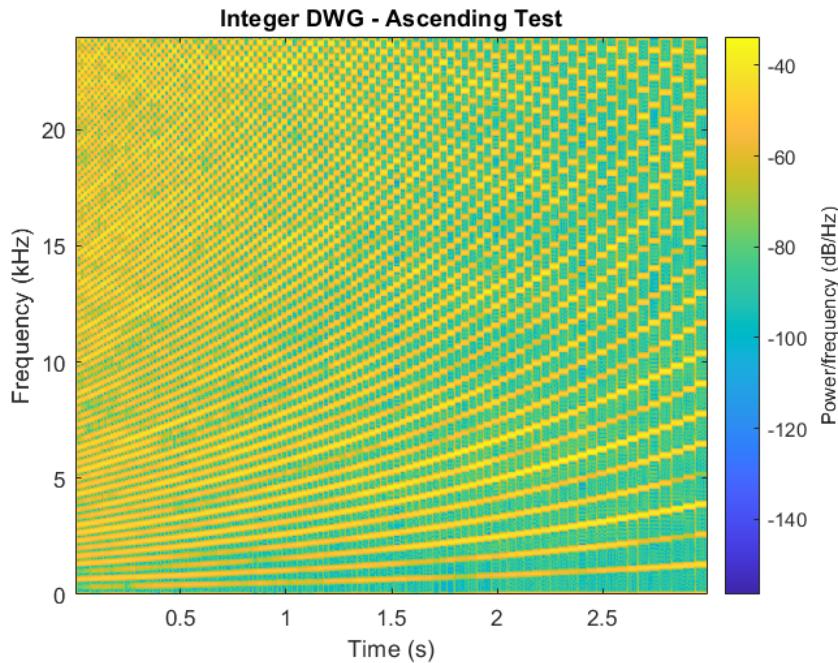


**Fig. 3.18** Extreme downward slide with the Loop Filter disabled to help illustrate reduced harmonics

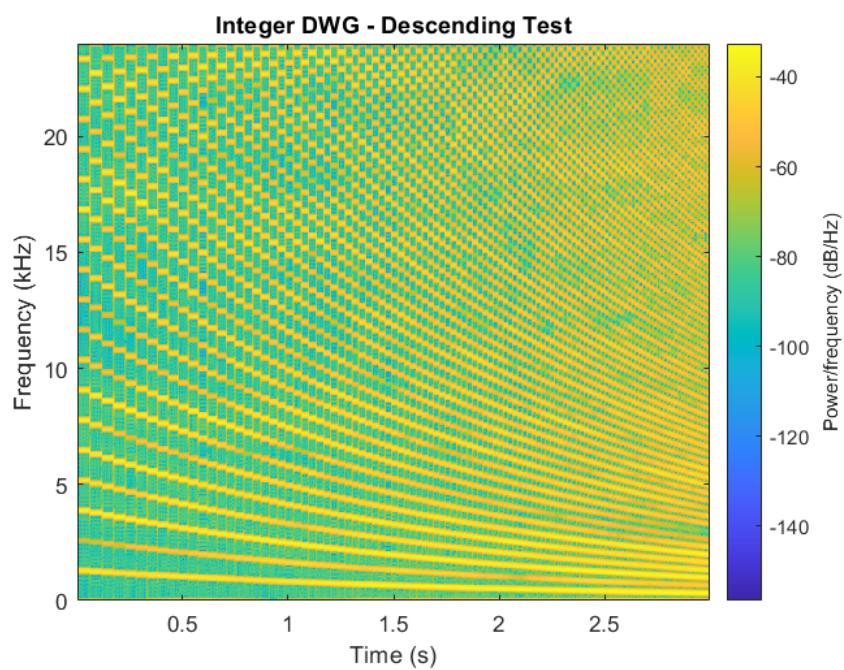
time-varying length is controversial ([pakarinen virtual 2008](#)), however there is no reason why the upper harmonics of a string would be limited from a physical standpoint. The loss of the harmonics is due to the inability of the system to represent them at that sampling rate. In summary, the inherent bandwidth limitations imposed by the sampling rate create limitations on the physical accuracy of the model.



**Fig. 3.19** Extreme upward slide with Loop Filter disabled to help illustrate reduced harmonics



**Fig. 3.20** Note the quantization in frequency as well as the loss of harmonics



**Fig. 3.21** Note the quantization in frequency as well as the reintroduction of harmonics

## 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 on a unit-level 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 as will be explained in their section. Full details of the testing scenarios described can be found in the attached code of the Appendix.

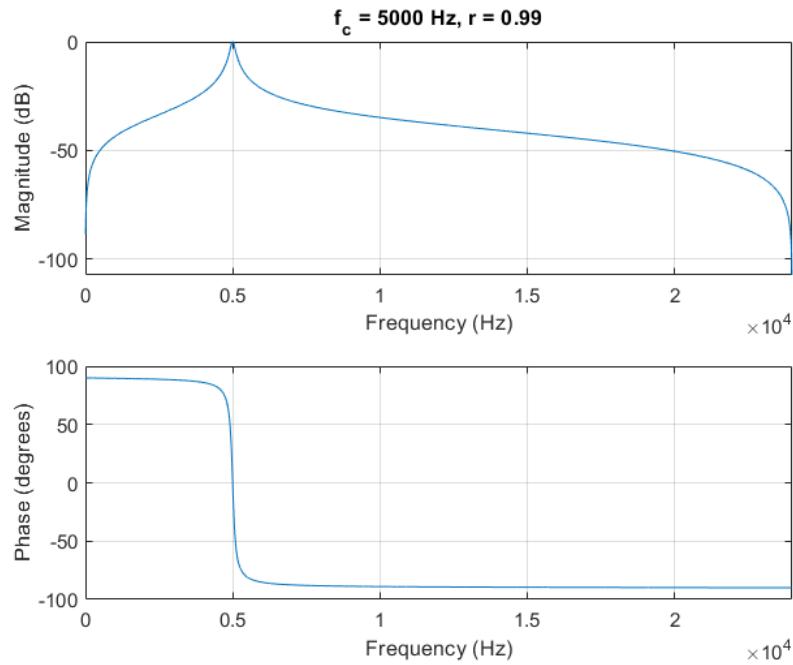
### 4.1 Filters

All the filters are a class built around MATLAB's *filter()* function. Accordingly, it is not necessary to recreate testing of this function given that MATLAB has done that for us. 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:  $F_s = 48,000\text{Hz}$ ,  $f_c = 5,000\text{Hz}$  and  $r = .99$ . This is illustrated below in figure 4.1.

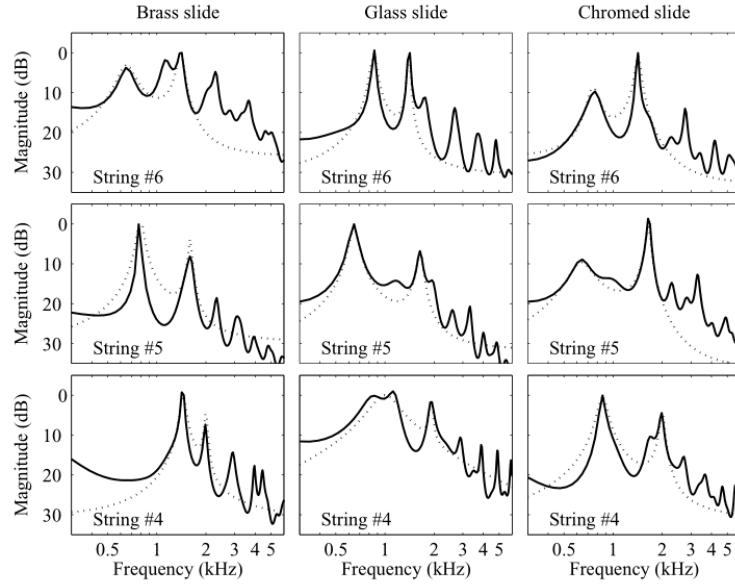


**Fig. 4.1** Frequency response for 2nd-order resonator test

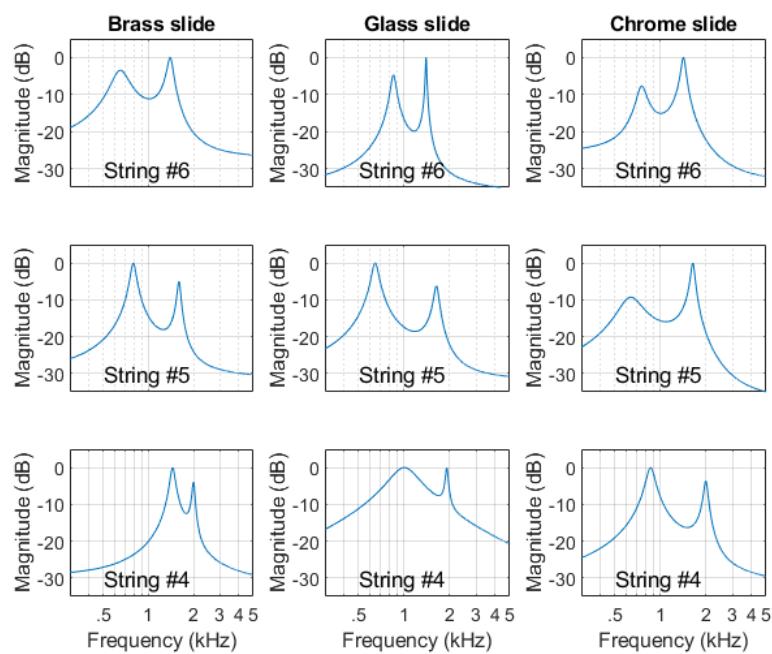
#### 4.1.2 Longitudinal Modes

The precise method by which the longitudinal mode filters were derived and designed is not completely specified in either (**puputti'real-time'2010**) or (**pakarinen'vertual'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 there a 4th-order IIR filter was derived based on the most prominent resonances for which they give the pole/zero locations.

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.



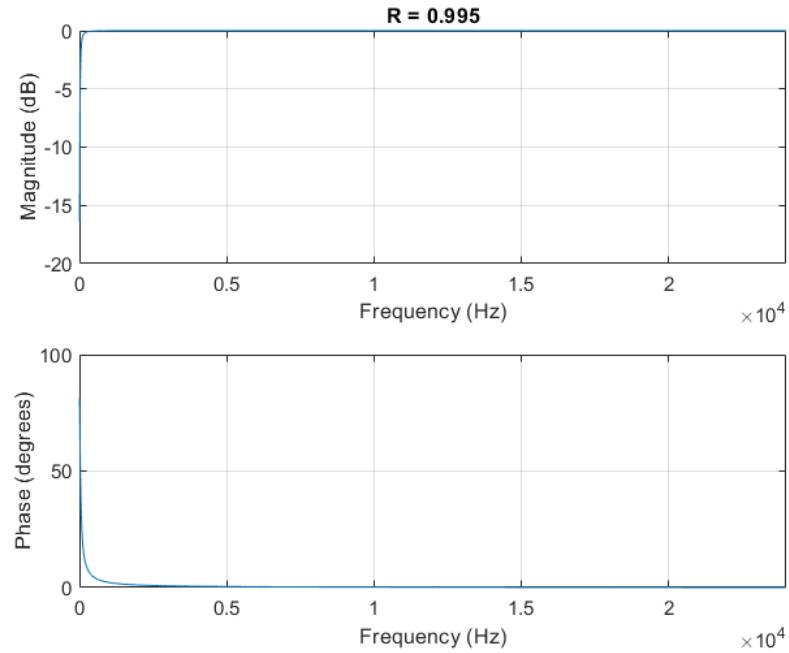
**Fig. 4.2** Original figure from (**pakarinen'virtual'2008**) for comparison purposes. Solid lines represent spectral estimates using linear-prediction filter of order 100. Dashed lines indicate modal filter magnitude responses.



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

#### 4.1.3 DC Blocker

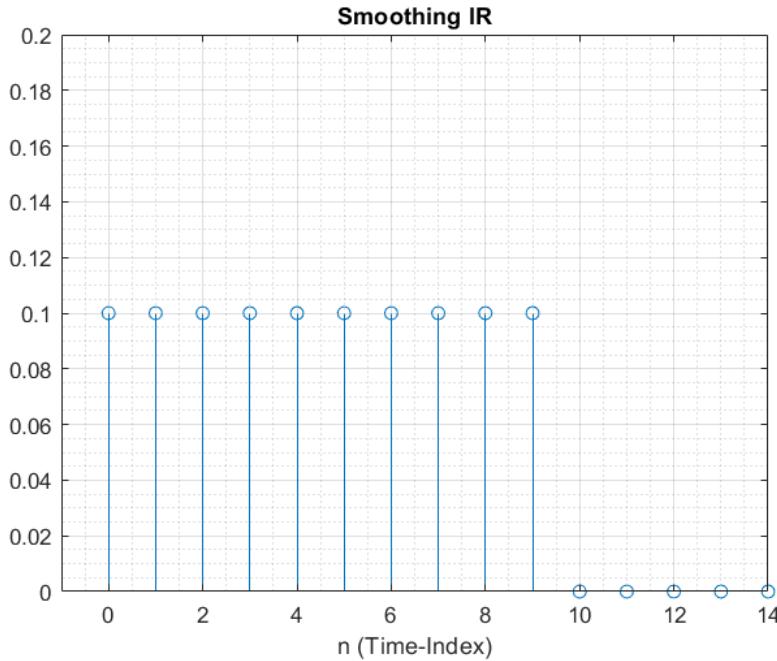
Figure 4.4 illustrates the DC blocker's frequency response for a value of  $R = .995$ .



**Fig. 4.4** DC Blocker Frequency Response

#### 4.1.4 Smoothing Filter

This filter was verified by computing the impulse response. As the filter is a 10th order smoothing filter, 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  $n = 9$ ).



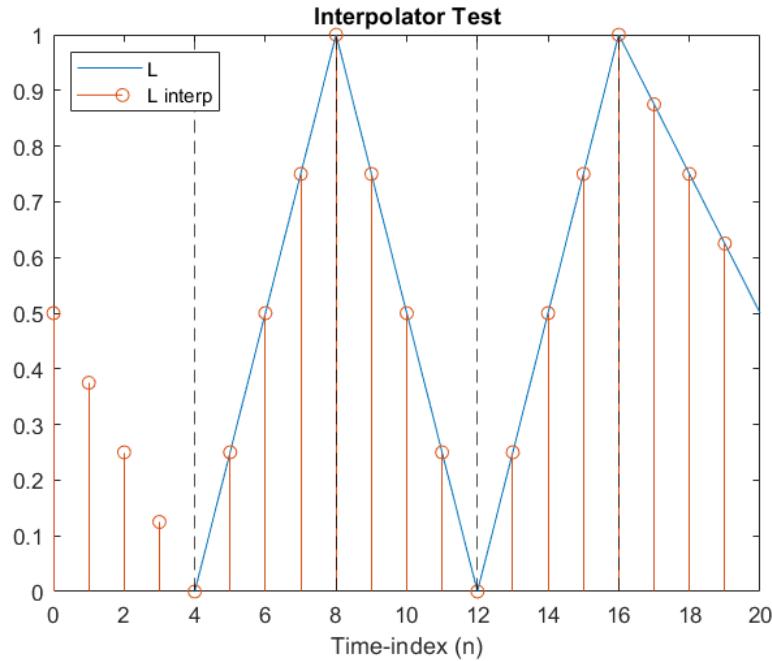
**Fig. 4.5** Verification for smoothing filter

## 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 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. .5 was used in this case.



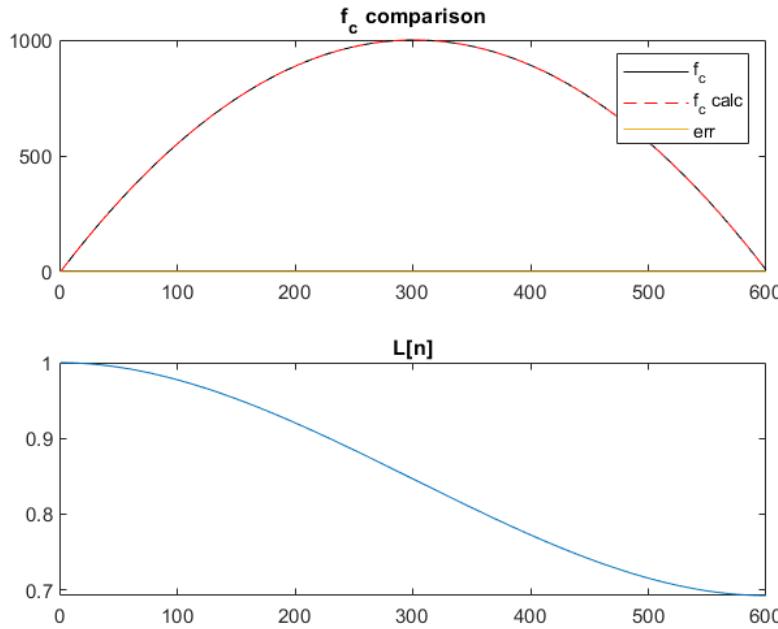
**Fig. 4.6** The black dashed lines represent the separation between frames.

### Slide Speed Extractor

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

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

This curve was then fed into the Slide Speed Extractor object to generate the corresponding  $slideSpeed[n]$ . This output was then multiplied by the  $n_w$  parameter for the corresponding string to ensure that the calculated output matched the specified  $f_c[n]$  form which the relative length control signal was derived. The output of this is shown in figure 4.7 below.



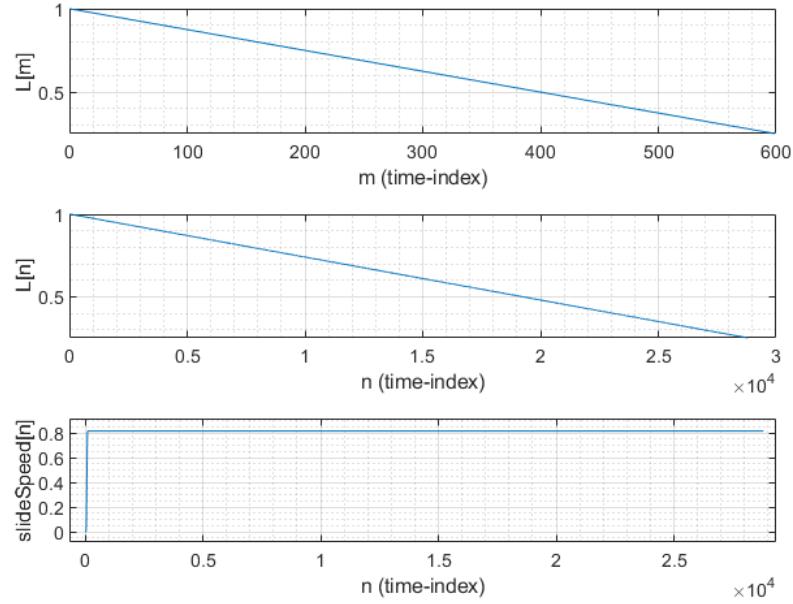
**Fig. 4.7** Slide speed verification output as well as corresponding  $L[n]$  stimuli

#### 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 object. 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 is can be seen as indicating the slide starts from rest and then gradually ramps up to constant speed. This is consistent as what would be expected as we have specified that the starting  $L[m]$  value in the CSP corresponds to when  $m = 0$ .

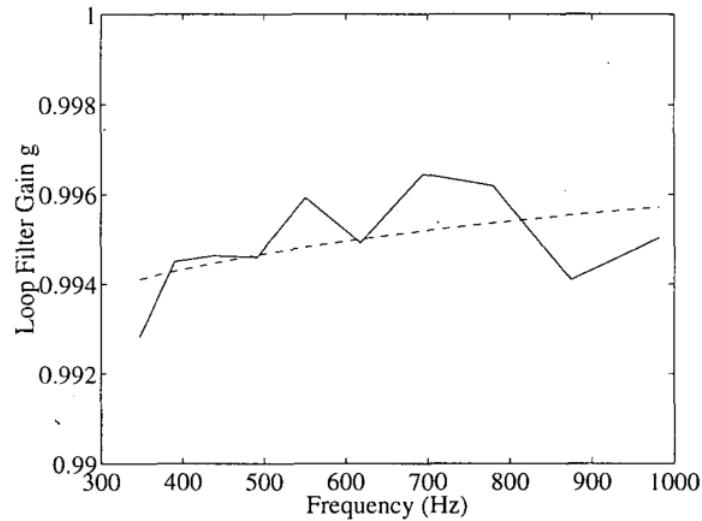
#### 4.2.3 Loop Filter

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

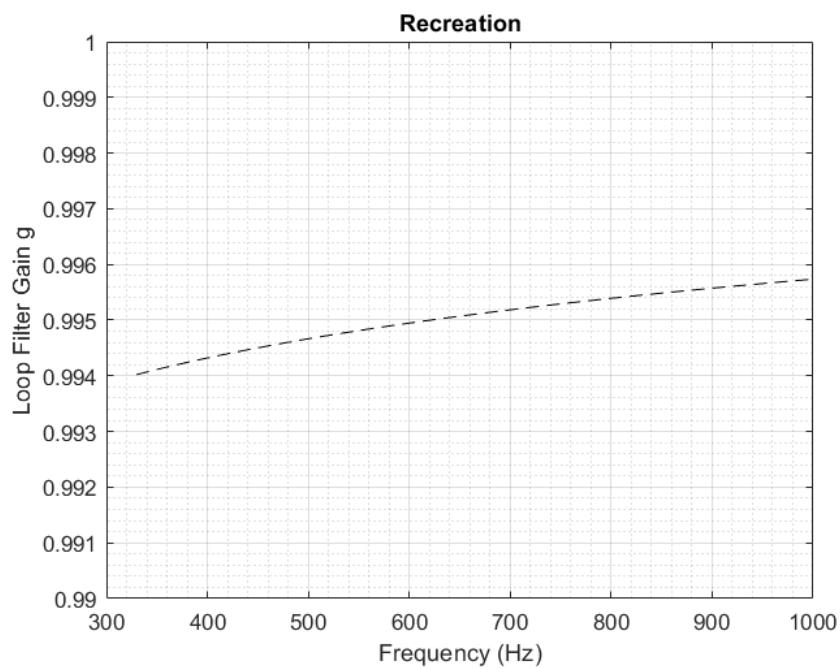


**Fig. 4.8**  $L[m]$  input signal and corresponding output signals for CSP test

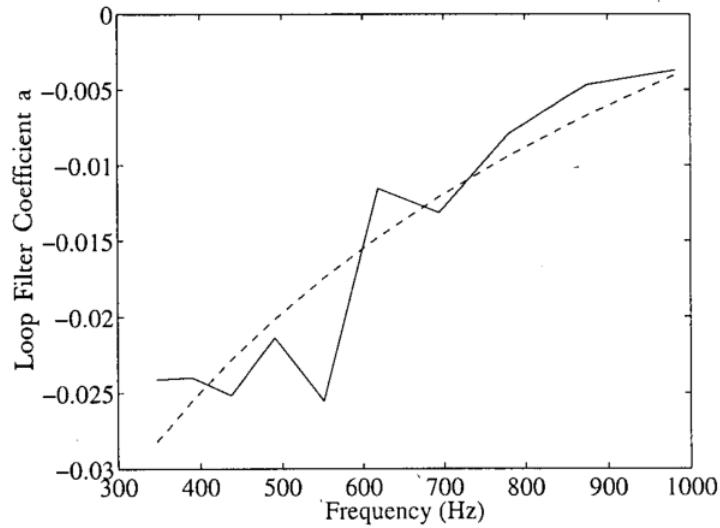
in the introduction chapter). Limitations of this approach have been shown which illustrated that the filter itself is operational through examining its various frequency response characteristics. The original paper provides the equations for the polynomial approximation as well as the polynomial coefficients. 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 correctly. Figures 4.9 and 4.11 show the original plots while figures 4.10 and 4.12 show the recreations respectively.



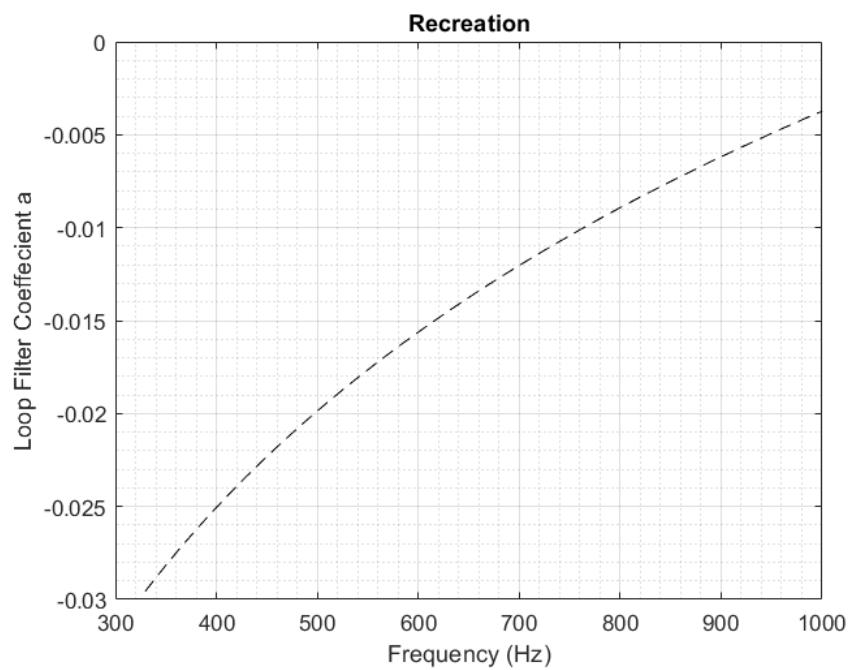
**Fig. 4.9** Loop gain  $g$  for modeling string 1 (solid line) and first-order polynomial fit (dashed line) from (valimaki'development'1998)



**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 (valimaki development 1998)

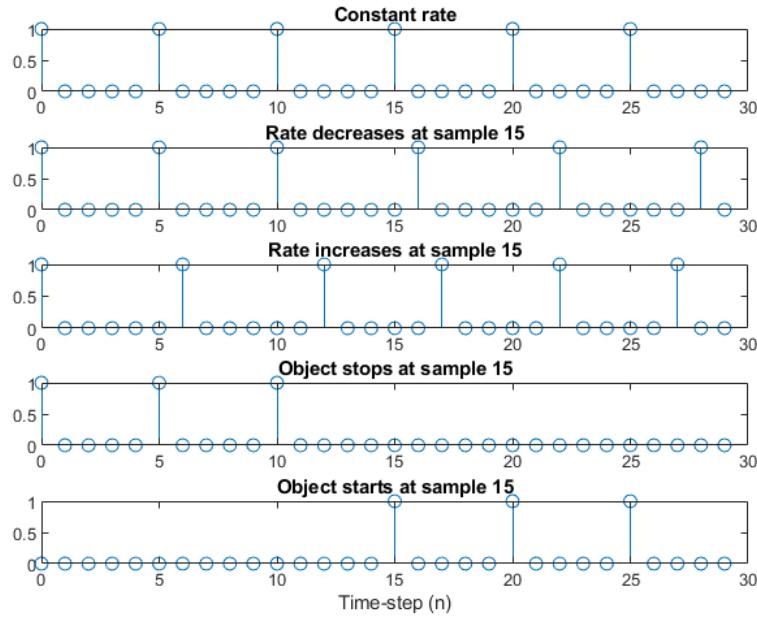


**Fig. 4.12** Loop-filter  $a$  polynomial for string 1

### 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.

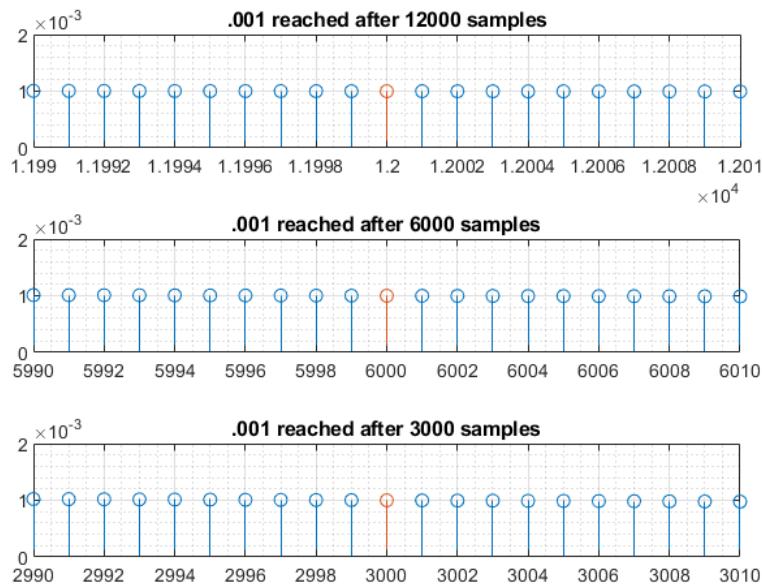


**Fig. 4.13** Impulse Train test output

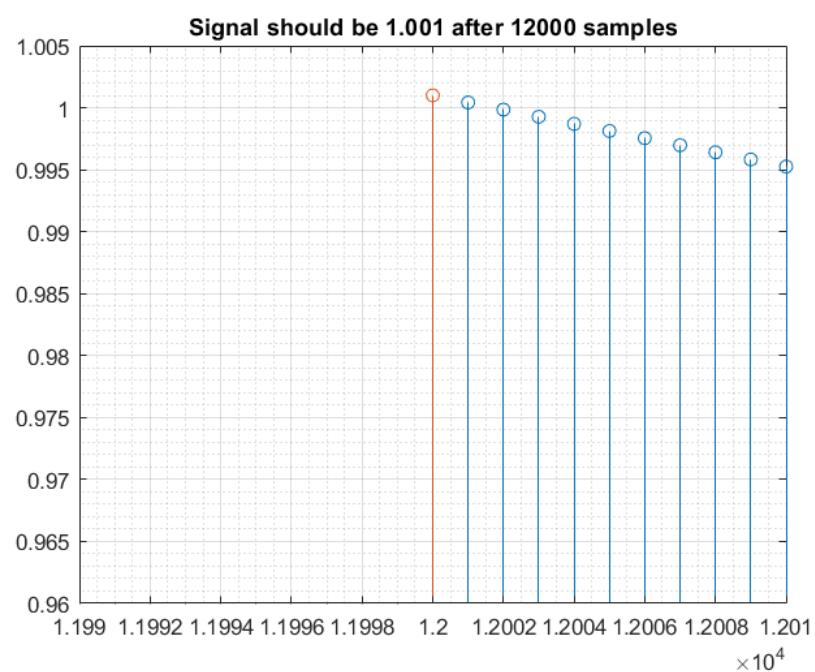
#### 4.3.2 Exponential Decay

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. Figure 4.15

illustrates correct functioning of overlapping impulse responses where the  $T_{60}$  value is set to be the same as the period of the Impulse Train.



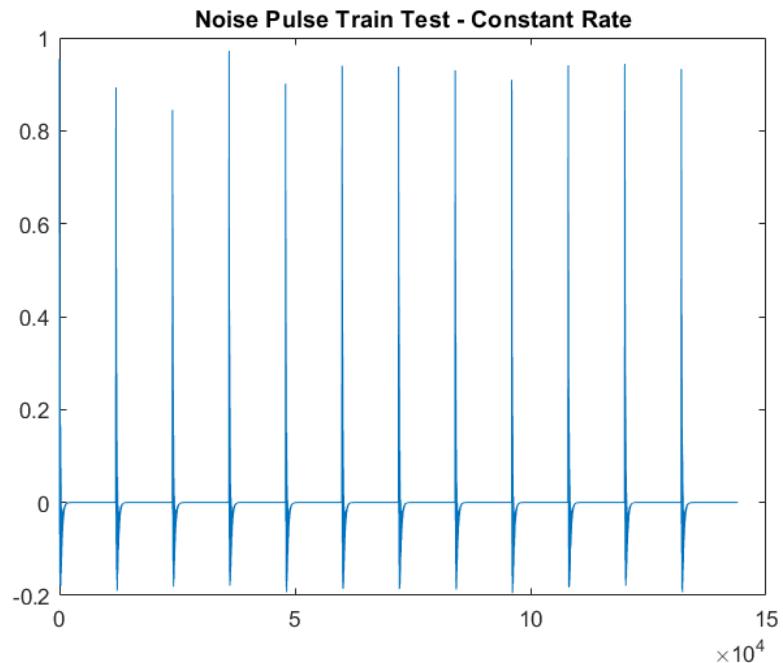
**Fig. 4.14** Test results for different  $T_{60}$  decay parameters



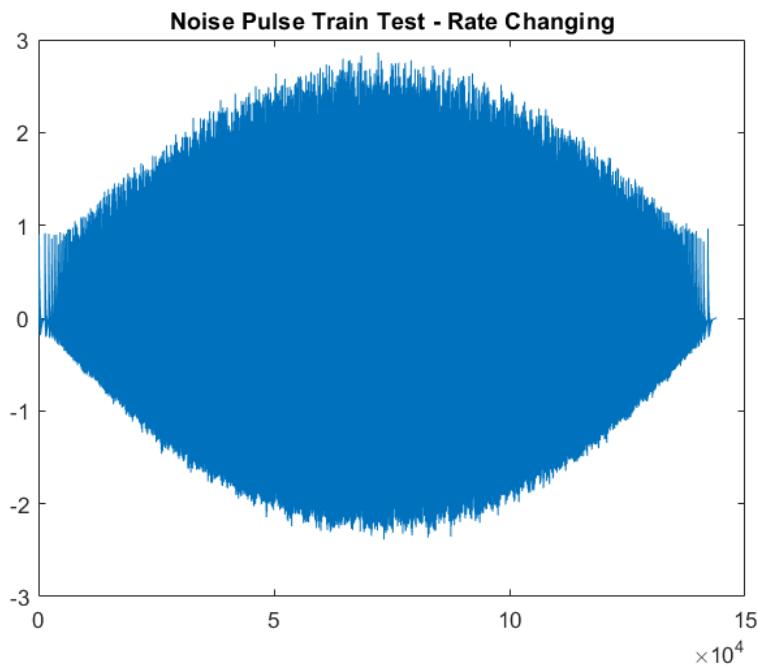
**Fig. 4.15** Test results to ensure overlapping IRs occur

#### 4.3.3 Noise Pulse Train

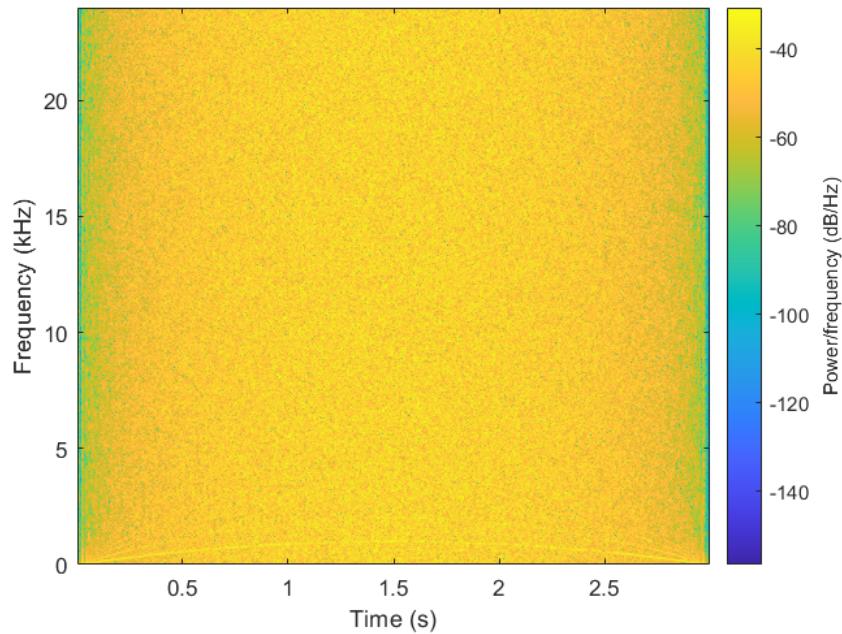
The Noise Pulse Train object was tested with two different scenarios: a constant firing rate producing 12 distinct pulses with no overlap and an swept firing rate corresponding to the same parabolic trajectory used in previous tests. The results are illustrated in figures 4.16, 4.17 and 4.18. Note the 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*.



**Fig. 4.16** Noise Pulse Train output for a constant rate. Note the negative values introduced by the DC Blocker.



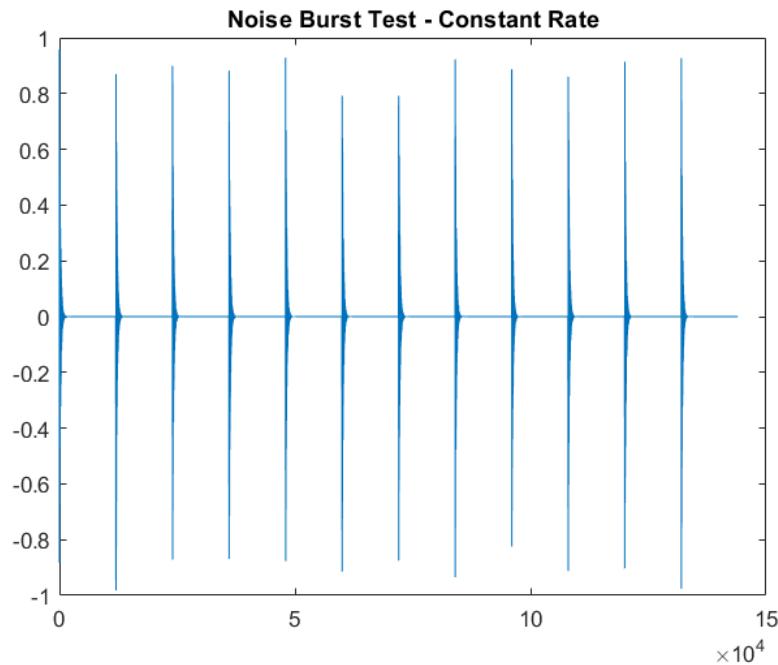
**Fig. 4.17** Noise Pulse Train output in response to an  $f_c[n]$  sweep. Note the overlapping build-up of the individual impulses. Individual pulses can be seen at the ends.



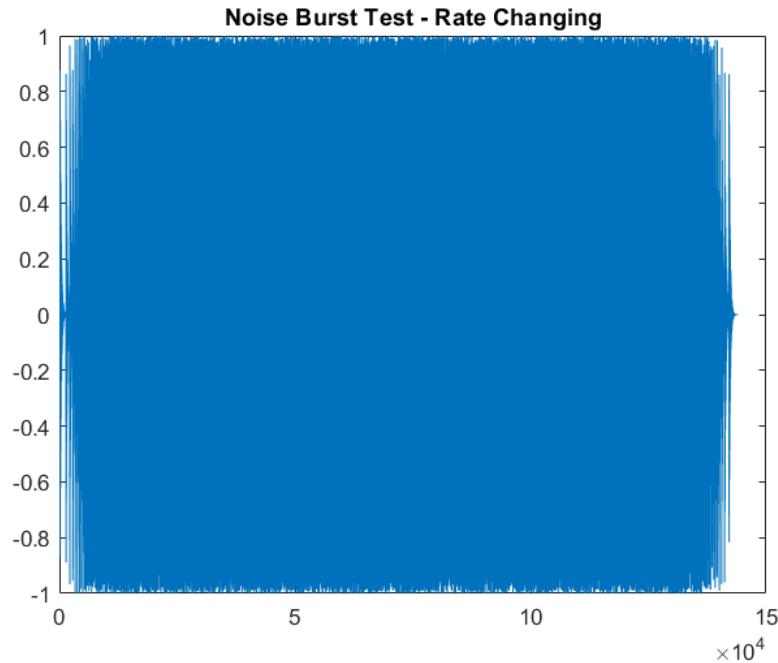
**Fig. 4.18** Spectrum corresponding to figure 4.17. Note the emergence of harmonics at the lower end.

#### 4.3.4 Noise Burst Generator

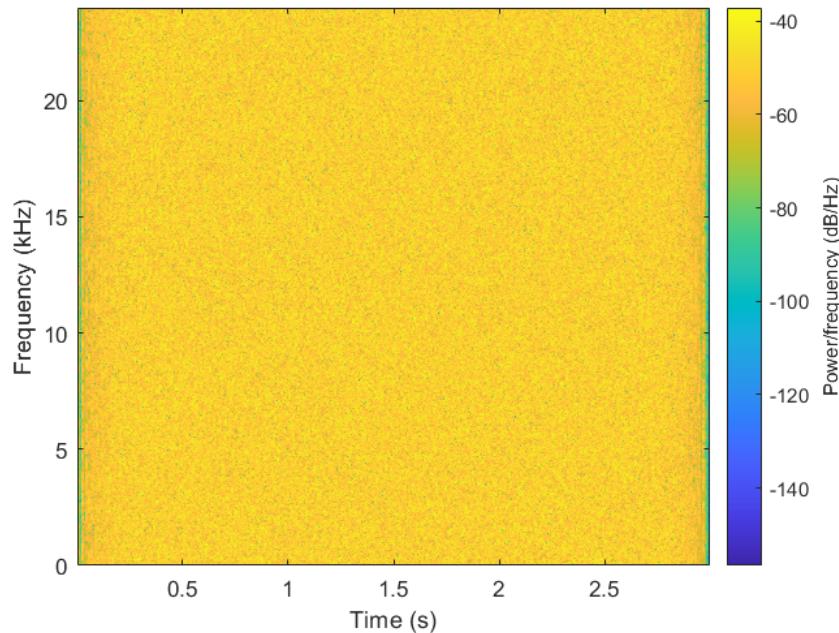
The Noise Burst generator was tested with the same two scenarios as the Noise Pulse Train. The outputs are shown in figures 4.19, 4.20 and 4.21. The sounds can be heard in the files *NoiseBurstGen-test1.wav* and *NoiseBurstGen-test2.wav*.



**Fig. 4.19** Noise Burst Generator output for a constant rate. Note the approximate symmetry about the x-axis.



**Fig. 4.20** Noise Burst Generator output in response to an  $f_c[n]$  sweep. Note the effects of the hard-clipping.

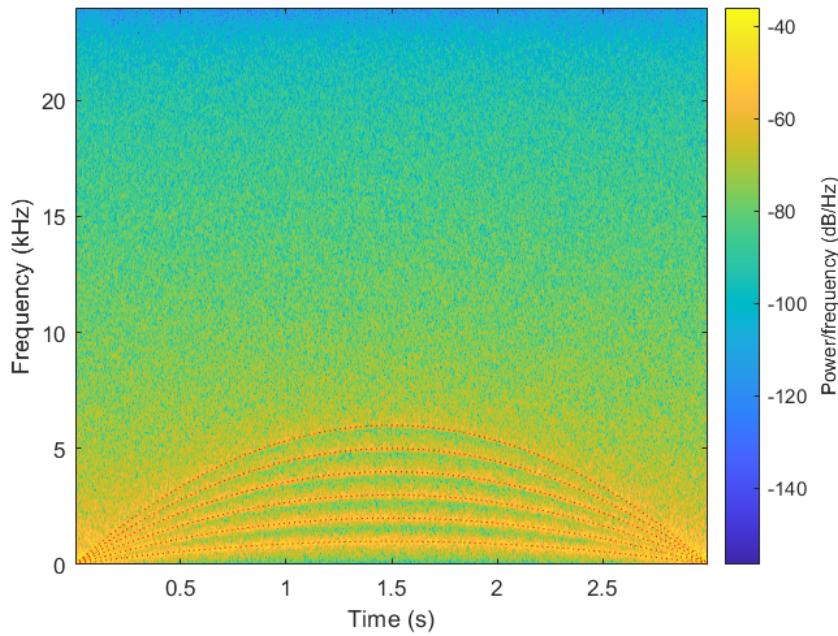


**Fig. 4.21** Spectrum corresponding to figure 4.20. Note the lack of harmonics.

## 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 figure 4.22 where the different harmonics' trajectories are overlaid in red. The output can be heard in *HRB-test.wav*.

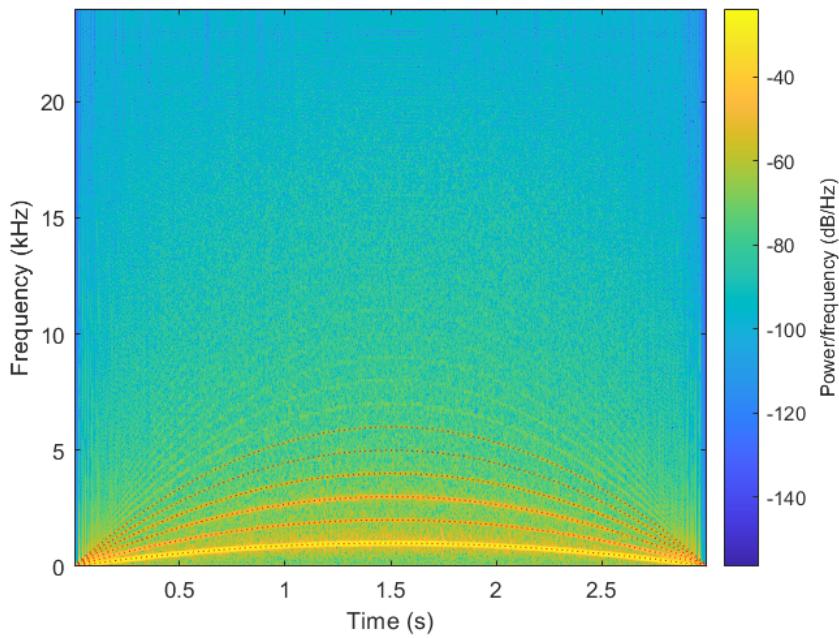


**Fig. 4.22** Output from Harmonic Resonator Bank test

### 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 4.23 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. 4.23** 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.

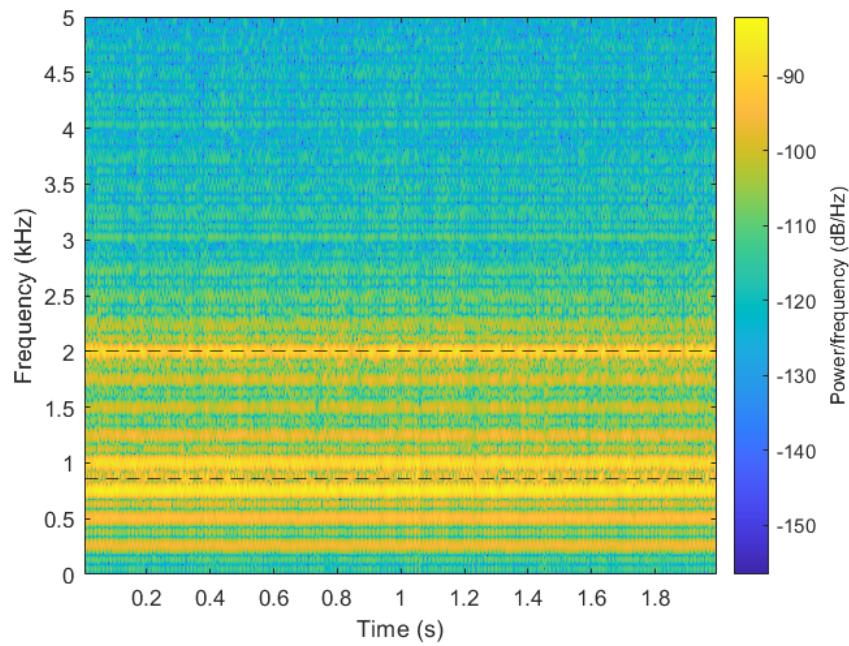
1. Longitudinal branch isolated
2. Harmonic branch isolated
3. Both branches combined

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.

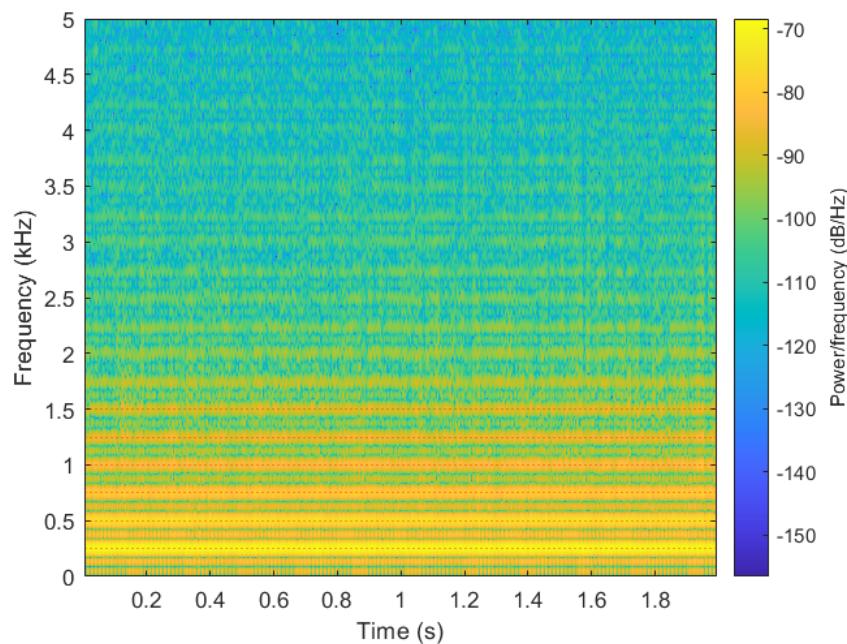
##### Static $f_c[n]$

The results for the constant slide velocity scenario are shown in figures 4.24, 4.25 and 4.26. Figure 4.27 illustrates how the original source stimuli produced by the Noise Pulse Generator does not

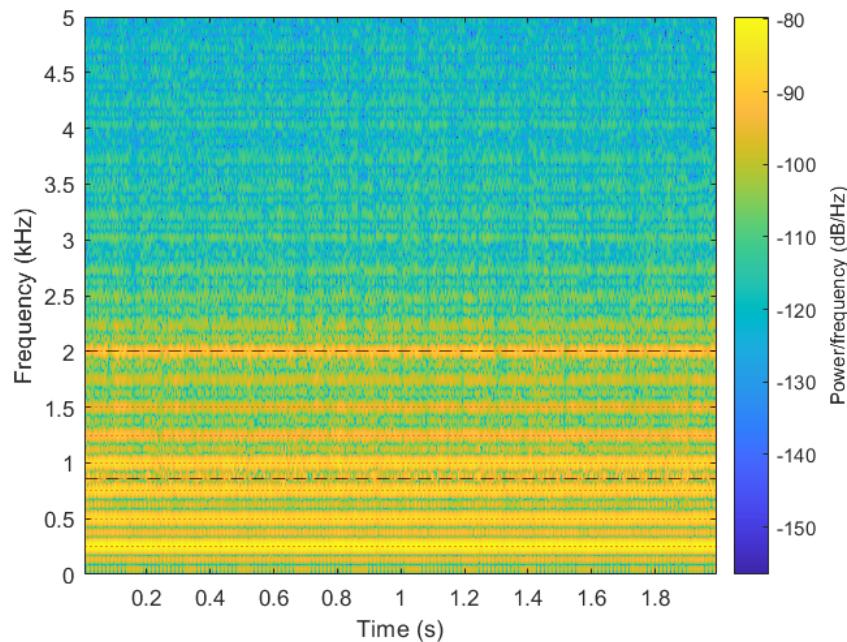
contain strong frequency components at the 1st longitudinal mode frequency. Also illustrated is that the harmonic branch extracts and emphasizes the fundamental while retaining many of the upper harmonics in decreasing strength. The corresponding audio can be heard in *CSG-Wound-Static-Long.wav*, *CSG-Wound-Static-Harm.wav* and *CSG-Wound-Static-Both.wav*.



**Fig. 4.24** Wound CSG and Static Slide Velocity - Longitudinal Branch



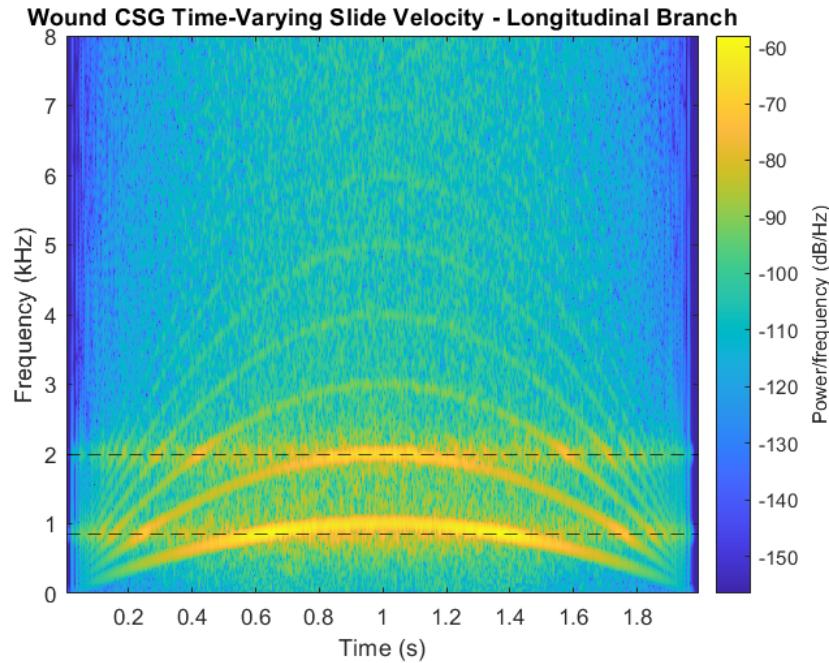
**Fig. 4.25** Wound CSG and Static Slide Velocity - Harmonic Branch



**Fig. 4.26** Wound CSG and Static Slide Velocity - Both Branches

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

The results for the time-varying scenario are shown in figures 4.27, 4.28 and 4.29. Dashed red lines indicate the theoretical harmonic trajectories. Dashed black lines correspond to the static longitudinal modes. As is clearly shown in Figure 4.27, 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 seep 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. 4.27** Dashed black lines = Longitudinal modes

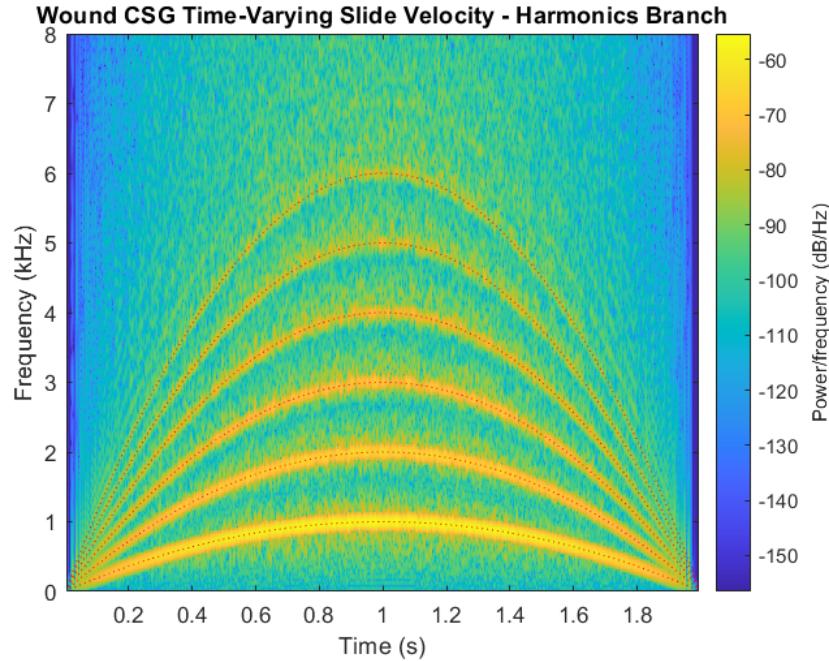


Fig. 4.28 Dotted red lines = harmonic trajectories

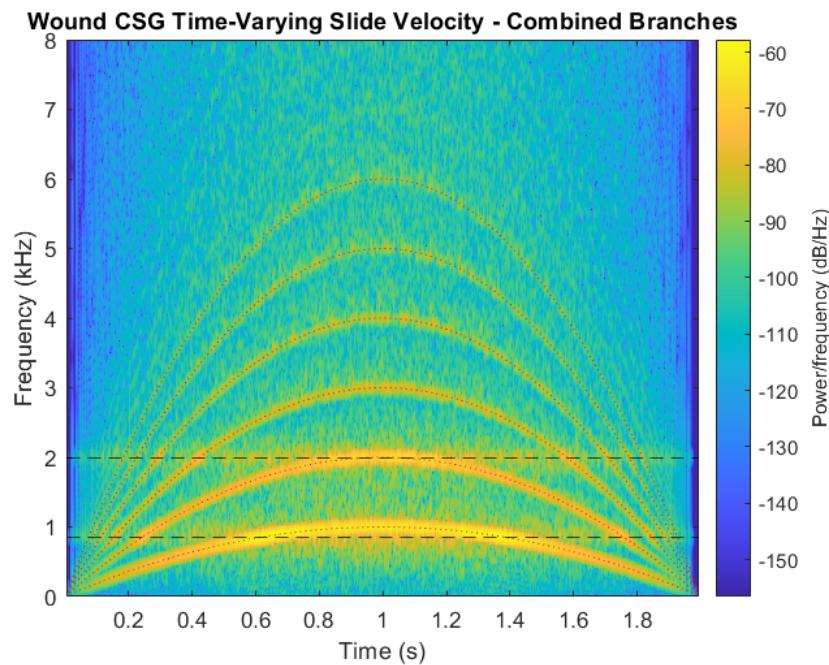
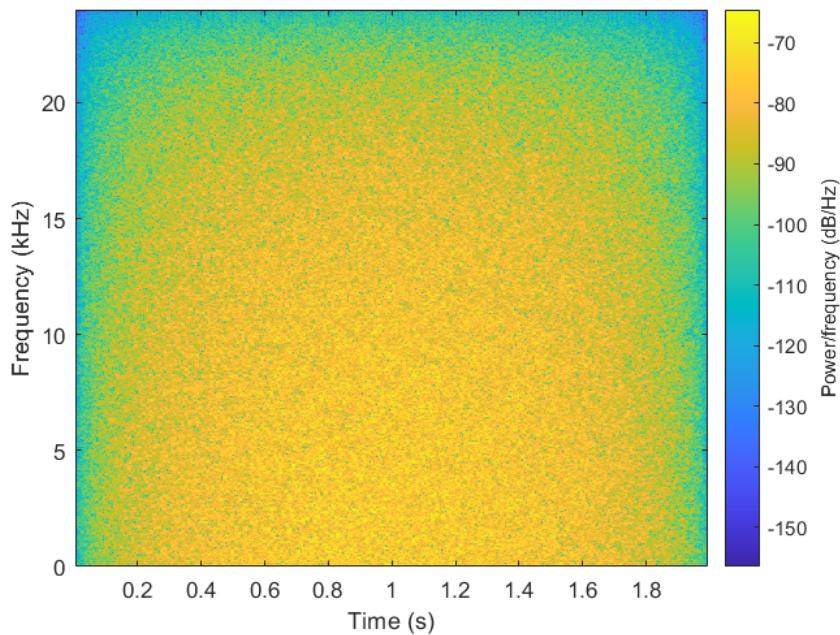


Fig. 4.29 Dashed black lines = Longitudinal modes, Dotted red lines = harmonic trajectories

#### 4.5.2 Unwound Variant

Figure 4.30 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. 4.30** Output from unwound branch for time-varying slide velocity

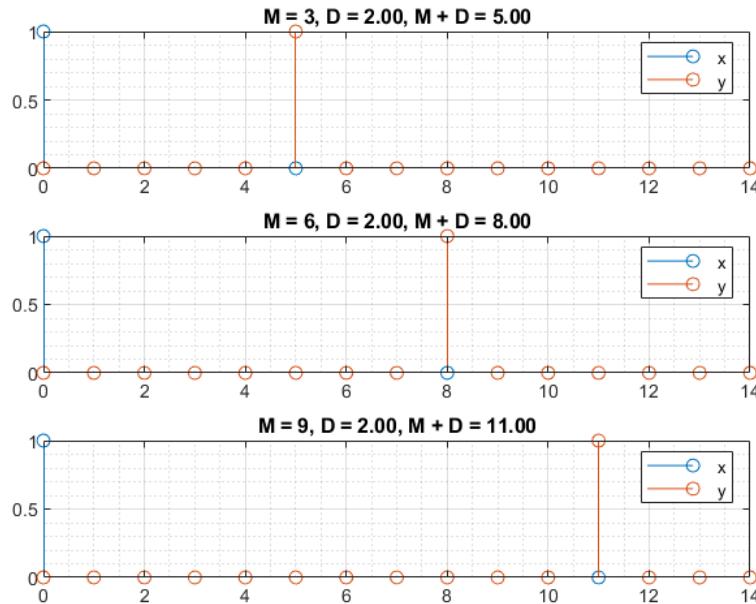
### 4.6 String DWG and Components

#### 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.

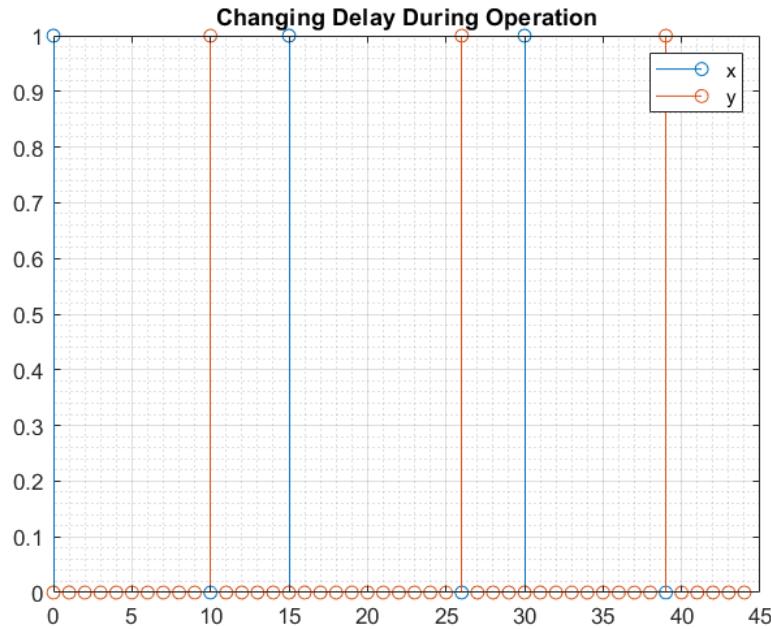
The first test which was performed was a test to ensure that it could indeed operate as an integer delay. Various Lagrange objects were constructed, ensuring that no fractional delay would be required. The output of this test is shown in figure 4.31 where  $x$  is the input signal and  $y$  is the output. As this figure clearly shows, the interpolated delay line clearly implements this functionality.



**Fig. 4.31** Test of integer values for fractional delay object

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 4.32 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.



**Fig. 4.32** Test of run-time updates for integer delay values

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 4.33. 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).

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 Lagrange test #3

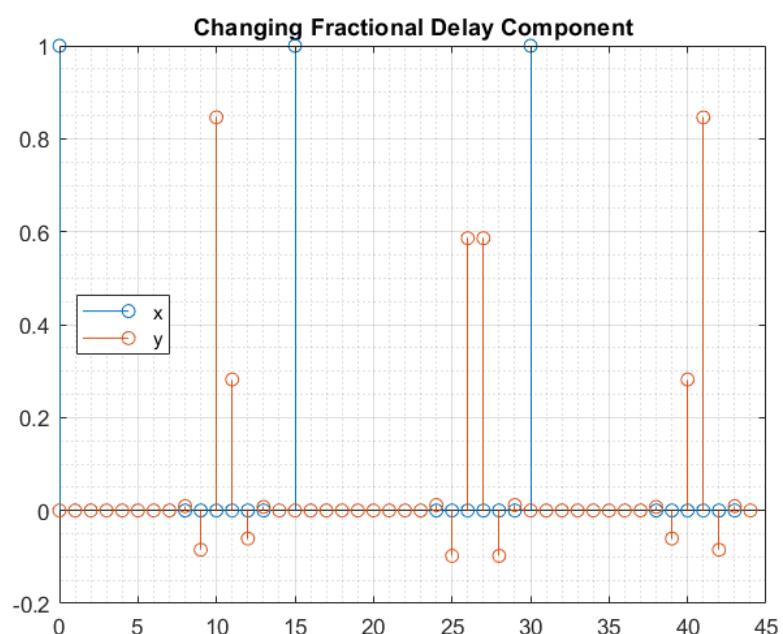


Fig. 4.33 Lagrange test #3

#### 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:

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

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

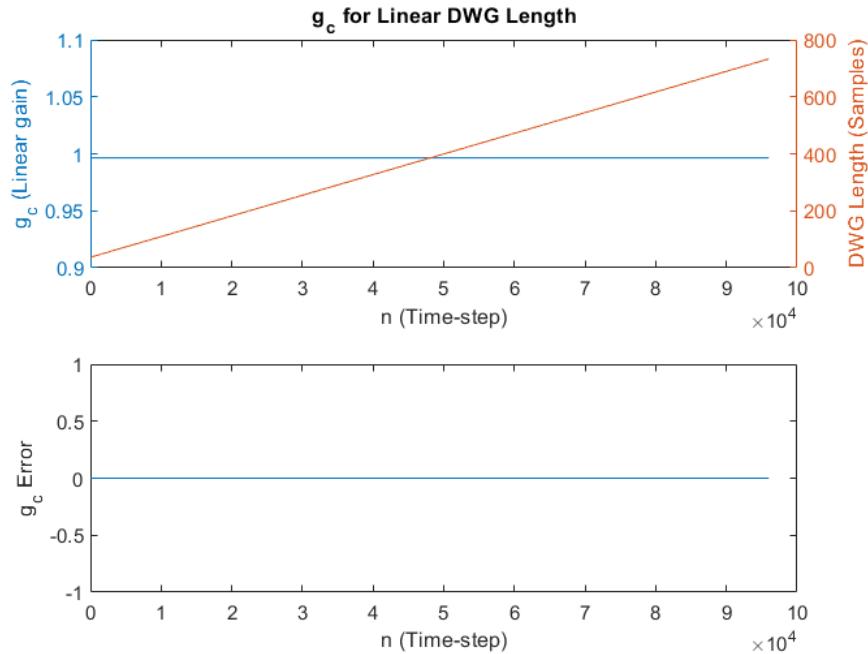
Given that  $\Delta x = DWGLength[n] - DWGLength[n - 1]$ , we can express  $g_c[n]$  as:

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

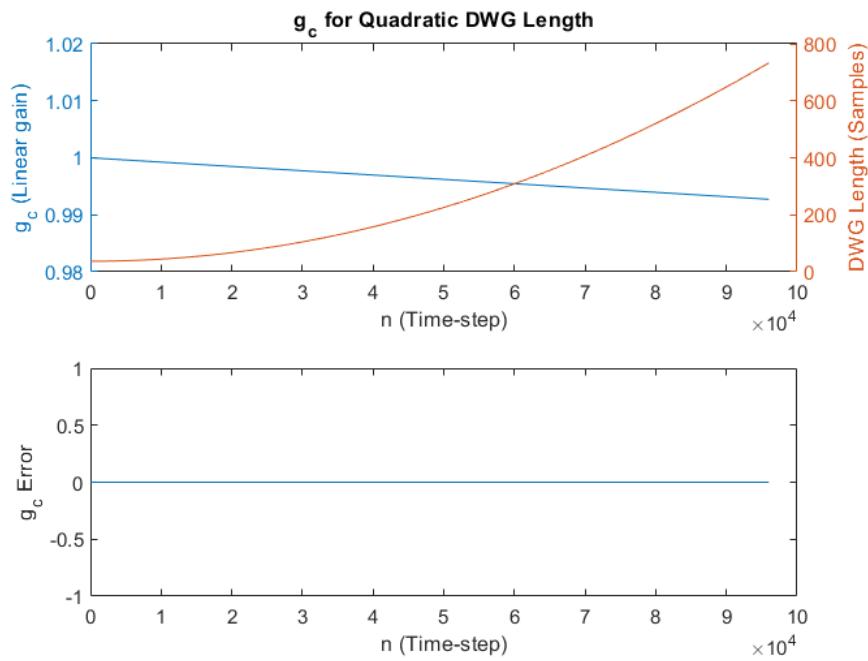
and use the expressions for  $a$  and  $b$  to develop a parameterized theoretical curve for the ideal output to a quadratic input. A similar procedure can be followed for the simpler linear case.

Figures 4.34 and 4.35 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.

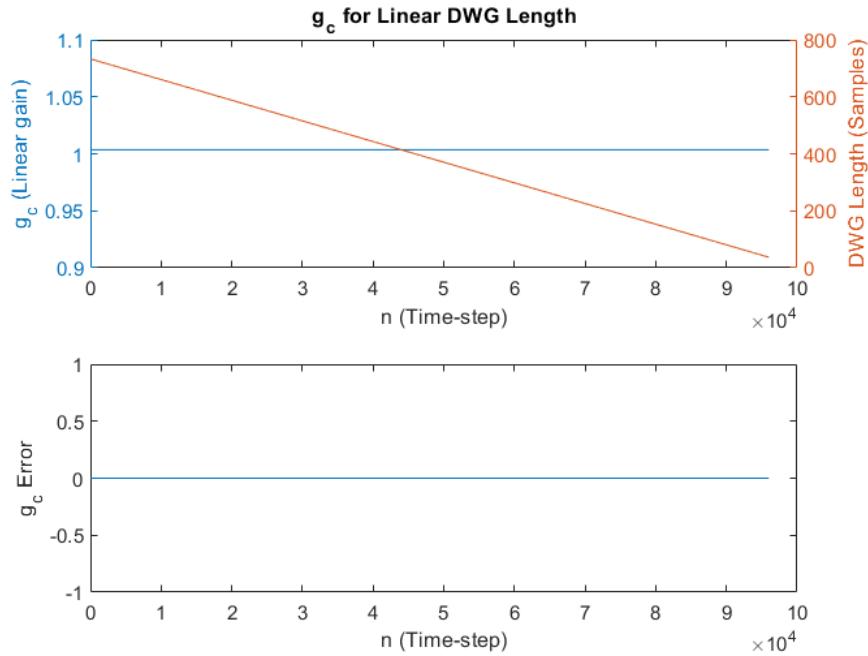
Figures 4.36 and 4.37 illustrate the output to the same functions which are now decreasing. As is illustrated, the error remains zero and the gain operates in the same pattern but applying amplification. Amplification is required in order to maintain the same perceptual loudness as samples are being discarded by the shortening length.



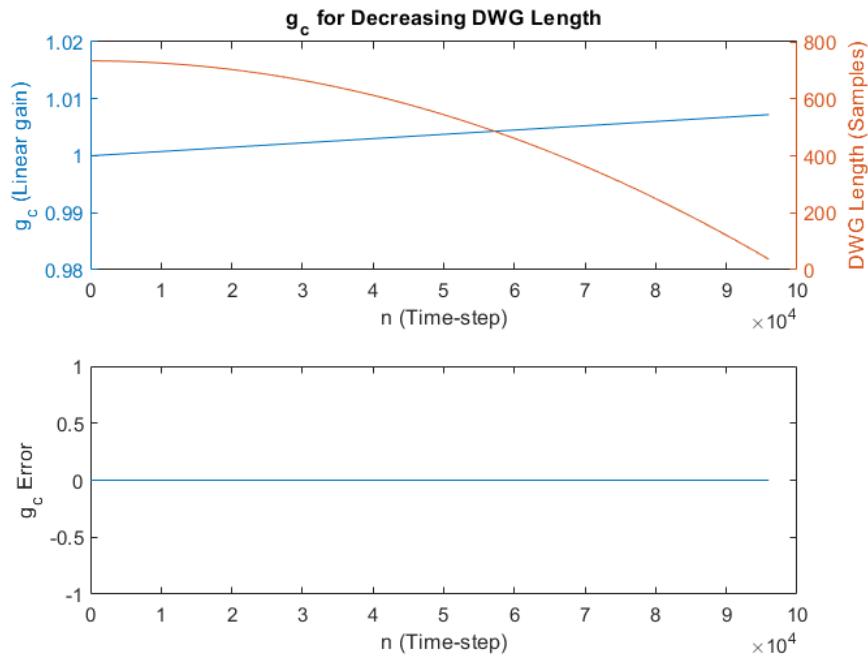
**Fig. 4.34** Energy scaler output for linearly increasing input



**Fig. 4.35** Energy scaler output for quadratically increasing output



**Fig. 4.36** Energy scaler output for linearly decreasing input



**Fig. 4.37** 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`spectral`nodate**). 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.

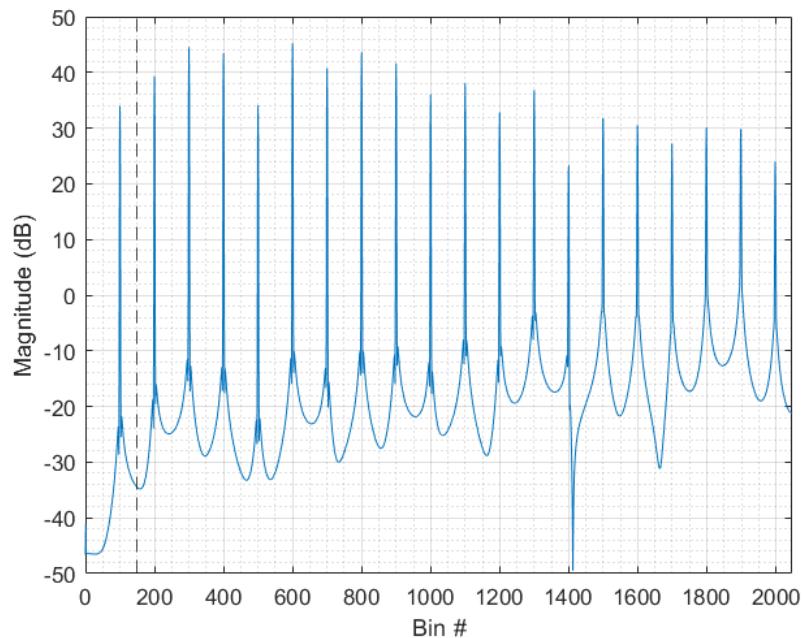
The initial assumption in the algorithm development was that the strongest resonance present in the signal would be the fundamental frequency. In practice this was not the case, likely due to the initial waveform being initialized by noise as well as the non-idealities in the Loop Filter's magnitude response. The assumption had to be modified and the search range of the algorithm is limited  $1.5 * F_{0bin}$ , where  $F_{0bin}$  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:

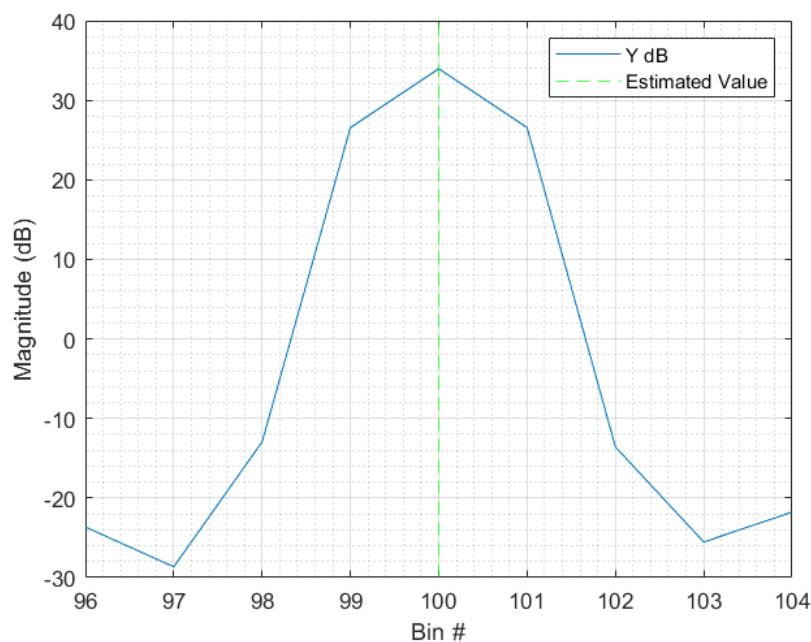
- Window Type = Hamming
- $F_s = 48,000$  hz
- $N = 4096$
- Overlap = 75

- Window length = 12 ms

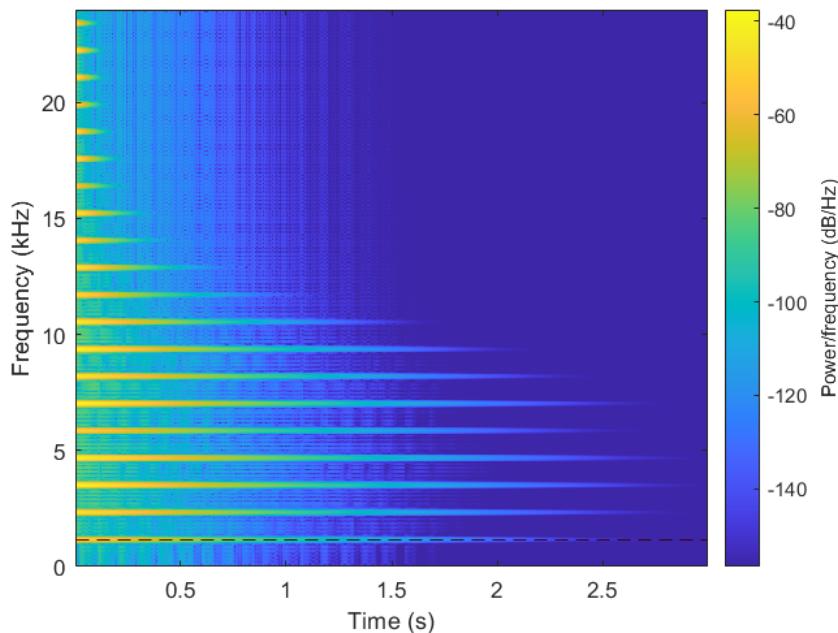
The DWG was configured to generate a signal with a fundamental frequency corresponding with  $F_{0bin} = 100$ , or 1,171.9 Hz. Following the method described in (**smith'spectral'nodate**), the calculated bin error was  $8.3499 \times 10^{-5}$ . In hertz this is  $9.7850 \times 10^{-4}$ . As this is extremely small, the tuning was considered to be accurate and verified. Figures 4.38, 4.39 and 4.40 illustrate this process.



**Fig. 4.38** 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. 4.39** Zoom of FFT for tuning verification.



**Fig. 4.40** 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-Test-#-direction.wav*, where # is replaced with the corresponding test number and *direction* is either up or down.

Figures 4.41 and 4.42 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.

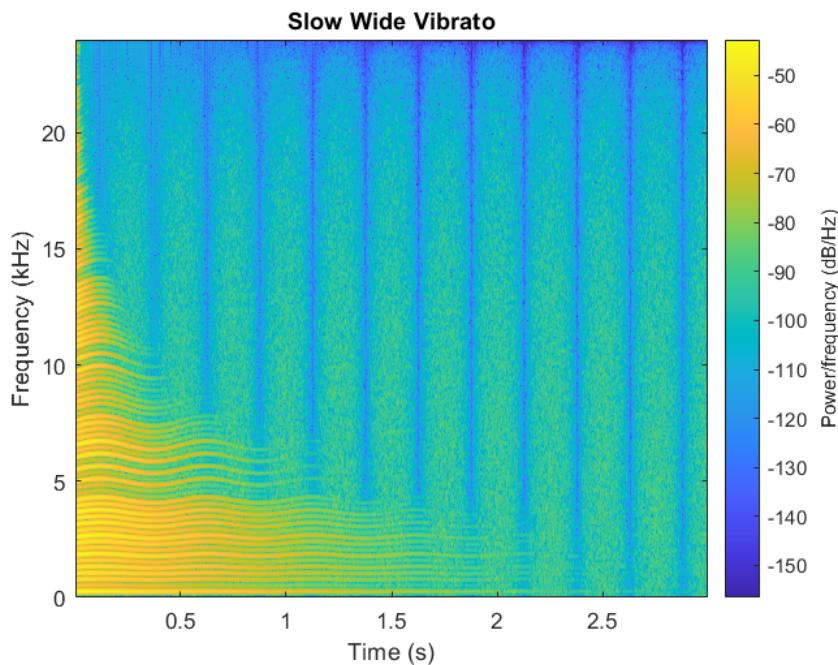


Fig. 4.41

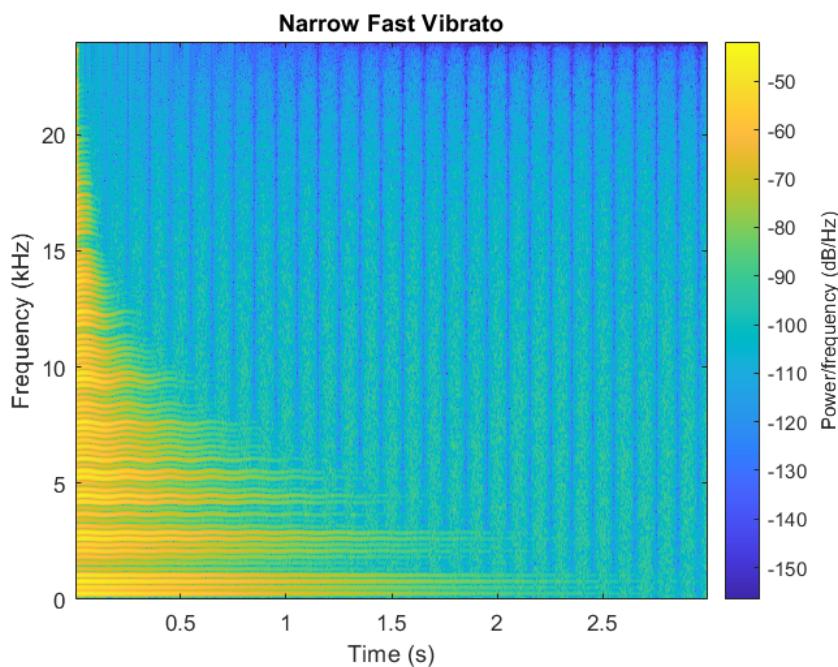


Fig. 4.42

## Chapter 5

# Physical Measurements of Slide Sounds

In this section, the physical measurement experiments which were used to confirm the theory of the design as well as attempt to refine the model by measuring parameters which were previous unmeasured.

### 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 as 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 and their masses are listed in table 5.1.

Brass	Chrome	Glass
125	50.0	17.5

**Table 5.1** Mass of various slides used in grams



**Fig. 5.1** Slides used in the measurements

The audio interface used was an RME Fireface UC. The microphones used were an AKG C480B and a DPA 4011-TL. The AKG was selected to be able to compare against the original recordings if need be. 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 interface with 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.

## 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. 5.2 provides a

summary of these measurements. Figure 5.2 illustrates the method for the E-string, while the appendix will contain the photos for all the strings measured.

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

Table 5.2 String Winding Densities

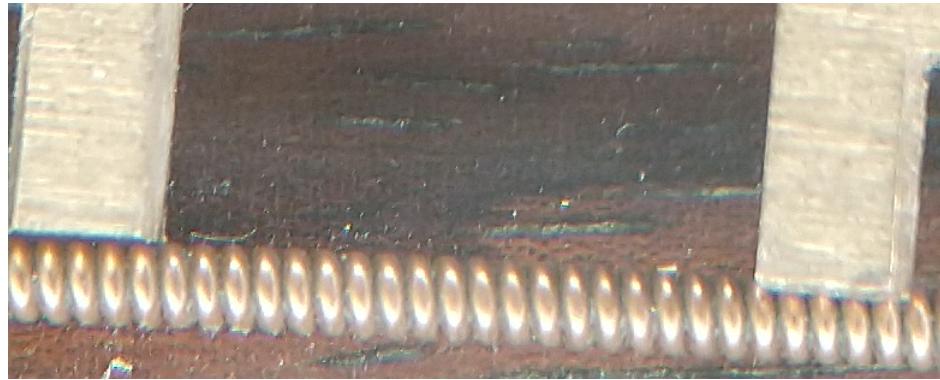


Fig. 5.2 E-string winding density measurement. Distance between inner edges of caliper is 1 cm

On this particular set of strings, the G string is wound. This differs from that of the guitar in the original paper (**pakarinen'virtual'2008**). When the synthesis model was developed as part of this thesis, the G-string was considered unwound as the procedure for generating the longitudinal mode filters was not clearly described and no filter specifications were present. The other strings use the  $n_w$  values which are specified above.

### 5.3 Transverse to Longitudinal Coupling

Existence of the coupling between the longitudinal motion of the slide and transverse vibrations of the strings was measured to confirm the injection of the CSG output into the string DWG model which is depicted in figure 8 in (**pakarinen'virtual'2008**). 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). These measurements were performed in A818.

### 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 slide 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. This was done perceptually by listening to the recordings and identifying the pitches corresponding with the slides fret location were present. An attempt was made to objectively confirm the existence by examining a spectrogram of the captured sounds

but this unfortunately failed as the signal-to-noise ratio was too low. Given that the sound under question is rather quiet and difficult to measure, the pre-amps on the audio-interface had to be set at extremely high settings to capture the noise with the mics. This in turn amplified the noise already present in the system, causing the signal degradation.

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.

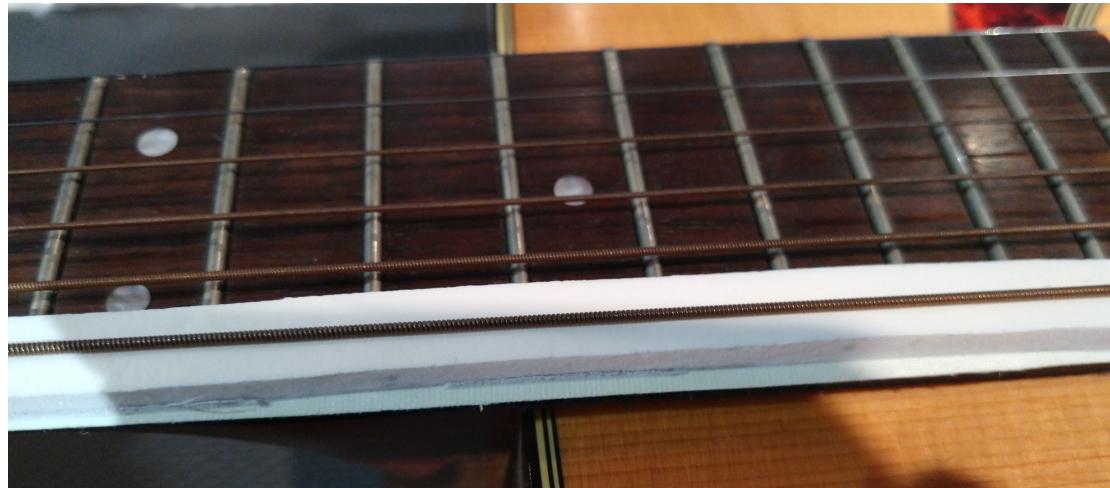
When comparing the coupling across different strings for the same slide type, the pattern observed was that the strings produced coupling from greatest to least in the following order: E, A, D, G, B and e. The first four are wound strings, and in order of increasing winding size. The pattern could be 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 5.4 illustrates foam muting technique.



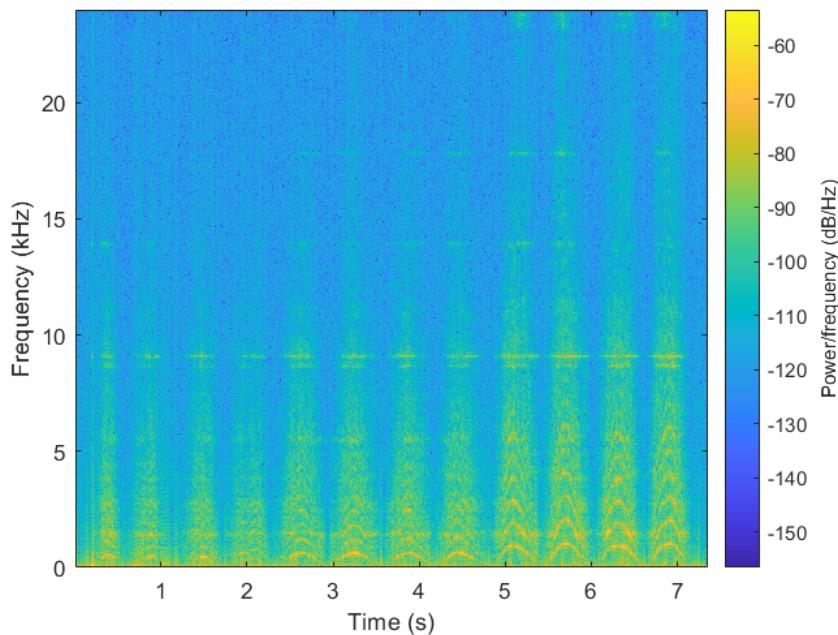
**Fig. 5.4** A foam strip placed underneath the E string to dampen its transverse vibrations. Note how it protrudes nicely without sinking too far into the foam.

In order to mitigate the less controllable human aspect, as no deterministic mechanical system was available to perform the measurements, 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 each measure. The first distance was from between frets 5-7 and frets 7-5, followed by a measure moving between frets 5-9 and 9-5, with the last measure having the slide move from frets 5-12 and back. These different distances were selected to have the top speed reached by the slide increase on each measure.

#### 5.4.2 Results

Figure 5.5 illustrates a spectrogram of measurements from the DPA microphone for the experiment being performed on the low E string with the brass slide. The results are extremely similar to the analysis performed in (**pakarinen'virtual'2008**) for the slide guitar and (**pakarinen'analysis'2007**) for finger noises. There are clearly both the time-varying harmonic as well as static modal com-

ponents 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. 5.5** 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 synthesis

model for unwound strings could not be as precisely confirmed.



**Fig. 5.6** A failed attempt to reduce the thickness of the foam.

## 5.5 Decay Rate of Single Winding Impact

Based on the material in (**pakarinen'virtual'2008**) and (**puputti'real-time'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. The foam used for muting would likely impede the longitudinal motion of the string slightly, however as it is also used in the noise characterization setup then the same effect would be seen in both experiments. Given this parameter 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.

### 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 instrument used to stimulate the strings was an exacto 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 5.7 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 toe oscillation process. As this system could be modeled as a damped oscillator with an initial displacement, the decay rate could be extracted from the recorded signal based on the relative changes of the amplitude over time.

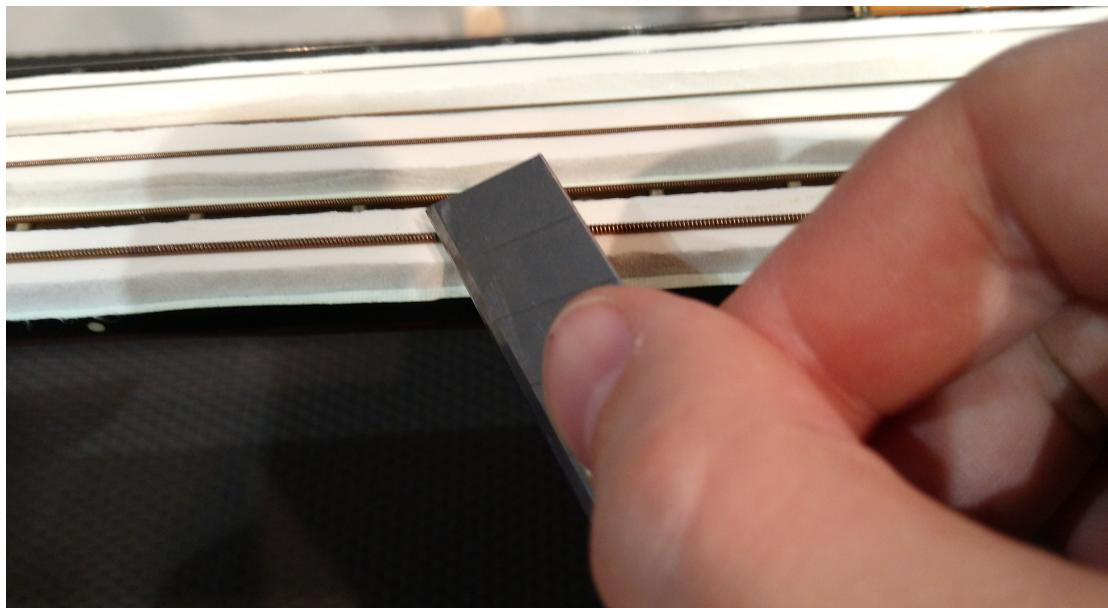


**Fig. 5.7** Knife edge wedged in between two string windings. This method was revised to be done with the blade out of the casing and on a string muted with foam.

### Strike Multiple Windings

Figure 5.8 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.



**Fig. 5.8** Knife striking setup. Alternating strings have muting foam as it was easier to prepare the setup for the next string measurements.

### 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 signal to make the articulations more realistic. This will be discussed more in the subsequent chapter on sound design.

## Chapter 6

# Sound Design and Controlling Parameters

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

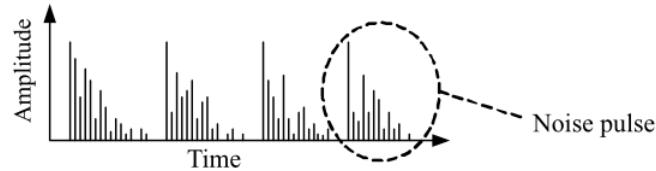
#### 6.2.1 $T_{60}$

$T_{60}$  is not fully described in the original (**pakarinen'virtual'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'real-time'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'model-based'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'vertual'2008**) and implemented in (**puputti'real-time'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'real-time'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'vertual'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'analysis'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 haromonicity to noise transition frequency for  $f_c[n]$ .

### 6.3 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**’real-time’**2010**) implementation while historically white noise has been used as illustrated by (**karplus**’digital’**1983**).

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

### 6.3.1 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

#### 6.4 Control Signal Parametrization

An attempt was made to synthesize the following musical segment.

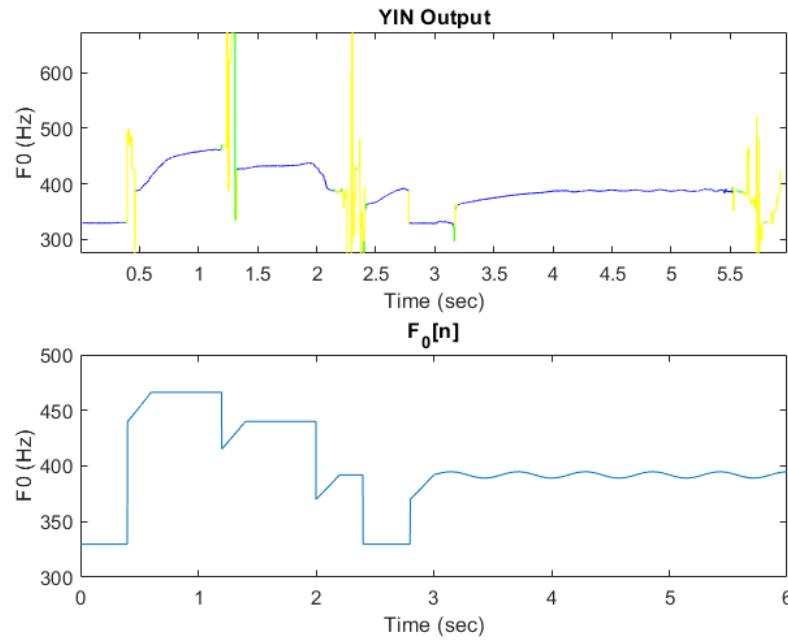
TODO: INSERT FIGURE OF THE SLIDE LICK HERE

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 6.2. 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*.

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.



**Fig. 6.2** Comparison of Recorded Example vs. Synthesized Signal F0

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.

## 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 wound strings.

Use a harder foam to mute the transverse vibrations?

Add more control over the dynamics of the plucked sound

## Appendix A

### String Winding Measurement Photos

## Appendix B

## Class Hierarchy

Fill me in later

## Appendix C

## Code

Fill me in later