

Simulating Hexaphonic Distortion Using Parallel Comb Filters

Sebastian Laguerre
Schulich School of Music
McGill University, Montréal

August 2020

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

© 2020 Sebastian Laguerre

Abstract

This thesis introduces hexaphonic distortion as a way of achieving harmonically rich guitar distortion while minimizing intermodulation products regardless of playing style. The simulated hexaphonic distortion effect described in this thesis attempts to reproduce the characteristics of hexaphonic distortion for use with ordinary electric guitars with mono pickups. The proposed approach uses a parallel comb filter structure that separates a mono guitar signal into its harmonic components. This simulates the six individual string signals obtained from a hexaphonic pickup. Each of the signals are then individually distorted with oversampling used to avoid aliasing artifacts. A Stratocaster-style electric guitar was fitted with an Ubertar hexaphonic pickup in the middle position. Simultaneous recordings using both hexaphonic and mono pickups were made on an 8-channel preamplifier and A/D converter. Starting with the baseline of the distorted mono signal, the simulated distortion produces fewer intermodulation products with a result approaching that of hexaphonic distortion.

Résumé

Cette thèse présente la distorsion hexaphonique comme un moyen d'obtenir une distorsion de guitare harmoniquement riche tout en minimisant les produits d'intermodulation désagréables quel que soit le style de jeu. L'effet de distorsion hexaphonique simulé décrit dans cette thèse tente de reproduire les caractéristiques de la distorsion hexaphonique pour les guitares électriques ordinaires avec des micros mono réguliers. L'approche proposée utilise une structure de filtre en peigne parallèle qui sépare un signal de guitare mono en ses composants harmoniques. Cela simule les six signaux de cordes individuels obtenus à partir d'un capteur hexaphonique. Chacun des signaux est ensuite individuellement déformé avec un suréchantillonnage utilisé pour éviter les artefacts de crénelage. Une guitare électrique de style Stratocaster était équipée d'un micro hexaphonique Ubertar en position médiane. Des enregistrements simultanés à l'aide de micros hexaphoniques et mono ont été réalisés sur un préamplificateur à 8 canaux et un convertisseur A / N. En commençant par la ligne de base du signal mono déformé, la distorsion simulée produit moins de produits d'intermodulation avec un résultat proche de celui de la distorsion hexaphonique.

Acknowledgements

Thanks to all the Music Technology faculty for your feedback on the thesis proposal. Countless thanks to my supervisor Gary P. Scavone for his helpful comments and generous support. An additional thanks for the teaching assistant positions. It was experience integral to my graduate studies and I will forever be grateful to have had that opportunity. I would also like to thank the Computational Acoustic Modeling Laboratory (CAML) at McGill University for providing funding for the hexaphonic pickup hardware.

I am indebted to the Centre for Interdisciplinary Research in Music Media and Technology (CIRMMT) for the recording studio space and the recording equipment. Thank you, Electronics Coordinator Yves Méthot for your suggestions and for managing the lab bookings.

The chemical etching process of the printed circuit board was carried out at the McGill Department of Chemistry with the help of Harry Baker of the Kambhampati Ultrafast Spectroscopy Lab and Martin Otto of the Siwick Research Group. Special thanks to Chief Electronics Technician Richard Rossi for making the arrangements.

I would like to acknowledge Technical Coordinator Darryl Cameron for his constant improvements to the Music Technology facilities. I am especially thankful for all the times he helped me obtain the tools and equipment I needed.

Many thanks to my friends and colleagues in the Music Technology program, with whom I was able to share my most passionate interests. Thank you, Jordan Vladimir Lysenko, for introducing me to the sounds of Os Mutantes that inspired this research.

Contents

1	Introduction	1
1.1	Motivation	2
1.2	Objectives	2
1.3	Thesis Organization	2
2	Background and Literature Review	4
2.1	Intermodulation	4
2.2	Hexaphonic Distortion	5
2.3	Digital Distortion Effects	6
2.3.1	Aliasing Distortions	7
2.3.2	Reducing Intermodulation Distortion	7
2.4	Digital Processing with Comb Filters	8
2.4.1	Harmonic Sound Separation	8
2.4.2	Comb Filter-Based Harmonic Enhancement	9
2.4.3	Other Uses of Comb Filters	10
3	Hexaphonic Pickup Recordings	11
3.1	Recording Setup	11
3.2	Audio Analysis	12
3.3	Sympathetic Vibrations	14
4	Separation of Harmonic Signals	17
4.1	Comb Filters	17
4.1.1	FIR Comb Filter	17
4.1.2	IIR Comb Filter	19
4.1.3	Universal Comb Filter	21
4.1.4	Comb Filter Parameters	22
4.1.5	Impulse Response	24

4.2	Parallel Comb Filters	27
4.2.1	Simulating Sympathetic String Effects	27
4.2.2	Filter Bank	27
4.2.3	Microtonal Pitch Variations	31
4.3	String Detection	32
5	Distortion Emulation	33
5.1	Reference Hardware	33
5.1.1	Circuit Schematic and Operation	34
5.1.2	Printed Circuit Board	34
5.1.3	Assembly	34
5.1.4	Testing	35
5.2	Static Approximation of the Diode Clipper	38
5.3	Comparison with SPICE Simulation	39
5.4	Oversampling	41
5.4.1	Argument for Sixteen-times Oversampling	41
5.4.2	Implementation	42
6	Results	43
6.1	Simulated Distortion Evaluation	43
6.1.1	Harmonic Entropy	43
6.2	Comparison of Distortion Structures	45
6.3	Magnitude Response Comparison	46
6.4	Spectrogram Comparison	48
6.5	Algorithm Performance	51
6.6	Tools	52
7	Conclusion and Future Work	54
7.1	Circuit Emulation Preview	54
7.2	Vectorized Operations	55
7.3	Summary	56
A	Drawings	57
	Bibliography	60

List of Figures

2.1	Aliasing suppression by oversampling	7
3.1	Hexaphonic recording setup	12
3.2	Open strings hexaphonic pickup recording	13
3.3	Open strings mono pickup recording	14
3.4	Spectrum of plucked open A ₂ and G ₃ strings	15
3.5	Hexaphonic pickup amplitude measurements and sympathetic resonance . .	16
4.1	FIR comb filter block diagram	18
4.2	FIR comb filter magnitude response	19
4.3	IIR comb filter block diagram	20
4.4	IIR comb filter magnitude response	21
4.5	Universal comb filter block diagram	22
4.6	Comb filter magnitude and phase responses for different choices of $\Delta\omega$. . .	23
4.7	Comb filter impulse response	25
4.8	Comb filter t_{60} decay time for different choices of Q	26
4.9	Open string frequencies recovered by comb filtering	28
4.10	Parallel comb filter bank magnitude response	30
4.11	Summed magnitude response	31
5.1	Distorter R VIII circuit schematic	35
5.2	Distorter R VIII printed circuit board	36
5.3	Distorter R VIII unit interior	36
5.4	Distorter R VIII oscilloscope capture	37
5.5	Distortion input filter waveform and magnitude spectrum	37
5.6	Exponential distortion static characteristic curve	39
5.7	Comparison of SPICE model and exponential distortion	40
6.1	Harmonic entropy	44
6.2	Mono distortion block diagram	45

6.3	Simulated distortion block diagram	46
6.4	Hexaphonic distortion block diagram	46
6.5	Comparison of distortion output spectrums	47
6.6	Open strings distortion spectrograms	49
6.7	Open and first fret distortion spectrograms	50

List of Tables

3.1	Pitch intervals between open strings	16
4.1	Parallel comb filter bank fundamental frequencies	29
4.2	Guitar fretboard notes in standard tuning	32
5.1	Aliasing distortions for different oversampling rates	42
5.2	Resampler parameters	42
6.1	Note pair intervals used in distortion comparisons	45
6.2	Comparison of intermodulation product magnitudes	48
6.3	Simulated distortion computational expense	52

Chapter 1

Introduction

When Cláudio César Dias Baptista invented the Regulus guitar in the 1970s, he introduced the world to a marvellous unique new sound. Baptista's interest and motivation in creating this new type of electric guitar centered on reducing inharmonic distortion, which he found to be a potentially negative aspect of distortion in traditional electric guitars with a single pickup for all strings. While a perfect technical solution, it was too impractical for the average guitarist and so the sound has since been little heard. Distortion effects have been around since the inception of the electric guitar. They add sustain and harmonic overtones to create a richer, warmer sound. In the seminal Nova Eletrônica article [1], Baptista describes the effect as such (translated from Brazilian Portuguese):

The main modification that the distorter produces is in the timbre of the sound, that is, it modifies, amplifying, the harmonic content of that sound. This means that, in practice, it makes the sound of a guitar more or less pure, into a more vibrant, richer, more aspirated sound, more like a violin sound now than a guitar sound.

The distorted sounds that characterize many rock genres are the result of a non-linear process that produces frequencies not present in the original signal. When the input signal comprises a single sinusoidal frequency, the output of a non-linear system will contain integer multiples of the input frequency. However, if multiple frequencies are present in the input, the output of the system will also contain intermodulation products that may not be harmonically related to each other [2]. The resulting spectrum can be so dense that it can sound harsh and undefined. Quoting again from the article:

The wrapping or excessive intermodulation between the notes emitted by the high strings and those emitted by the bass strings of a guitar, when played

simultaneously, or in chords, [...] creating a hoarse and undefined sound that, being useful for many purposes, is not always what is desired.

For musicians who endeavour to achieve a cleaner and richer type of distortion, there is a need for distortion algorithms which do not suffer the harmonic limitations caused by intermodulation.

1.1 Motivation

This research comes at the request of a contemporary musician who wished to recreate the iconic distortion effects used by the group Os Mutantes. This seemingly simple challenge turned out to be much more involved. Faithfully recreating the sound requires a specialized hexaphonic pickup. Since this was unavailable to the musician and likewise for many others, it became of interest to research a method by which hexaphonic distortion could be simulated through digital signal processing techniques, allowing guitarists with regular mono pickups to achieve this iconic sound.

1.2 Objectives

The proposed approach used in this thesis involves a parallel bank of comb filters. This structure, which was informed by analysis of signals from a hexaphonic pickup, separates a mono signal into harmonically related signals. The individually distorted signals produce fewer intermodulation products with a result approaching that of hexaphonic distortion.

1.3 Thesis Organization

Chapter 2 provides the necessary background on intermodulation and hexaphonic distortion. It then provides a comprehensive review of previous work on the topic of intermodulation distortion and the literature on the digital signal processing techniques employed in this thesis.

Chapter 3 describes the process by which recordings of an actual hexaphonic pickup were acquired. A frequency domain analysis of the signals reveal important characteristics of the hexaphonic pickup which lead to the main hypothesis for the thesis. Additional findings include insights on sympathetic vibrations in electric guitar strings and cross-talk in hexaphonic pickups.

The hexaphonic distortion simulation algorithm involves two main development challenges. The first is the separation of harmonic signals and is the topic of Chapter 4. The second is development of a digital emulation of an analog distortion circuit and is described in Chapter 5.

In Chapter 6, the simulated distortion structure is evaluated from the perspective of intermodulation distortion, with comparisons to simple mono and actual hexaphonic distortion. The performance of the algorithm is also evaluated in terms of computational expense.

Chapter 7 concludes the thesis by revisiting the main outcomes of the research and its implications while also presenting opportunities for future work.

Chapter 2

Background and Literature Review

The hexaphonic distortion simulation approach proposed in this thesis builds upon previous and related works on the subjects of digital distortion, intermodulation, and the use of comb filters to analyze and process harmonic signals. This chapter provides an overview of the topics and consolidates the literature consulted in the development of the system.

2.1 Intermodulation

To illustrate the mechanics of intermodulation, consider the non-linear system corresponding to the 2nd-order Chebyshev polynomial of the 1st kind [3],

$$T_2(x) = 2x^2 - 1. \tag{2.1}$$

When the input is a single sinusoid, this transfer function has the useful property that it only generates frequencies up to the 2nd harmonic. For the single cosine input signal,

$$T_2(\cos(2\pi ft)) = 2 \cos(2\pi ft)^2 - 1 = \cos(2 \cdot 2\pi ft), \tag{2.2}$$

generating the 2nd harmonic as expected. For an input consisting of f_1 and f_2 , expanding and combining terms reveals

$$\begin{aligned}
 T_2(\cos(2\pi f_1 t) + \cos(2\pi f_2 t)) &= 2(\cos(2\pi f_1 t) + \cos(2\pi f_2 t))^2 - 1 \\
 &= 2 \cos(2\pi f_1 t - 2\pi f_2 t) \\
 &\quad + 2 \cos(2\pi f_1 t + 2\pi f_2 t) \\
 &\quad + \cos(2 \cdot 2\pi f_1 t) \\
 &\quad + \cos(2 \cdot 2\pi f_2 t) \\
 &\quad + 1.
 \end{aligned} \tag{2.3}$$

The output of the non-linear function includes the 2nd harmonic frequencies but also the sum and difference frequencies $f_1 - f_2$, $f_1 + f_2$, along with a DC component. The non-linear systems used in guitar distortion algorithms generate an infinite number of harmonics. Extending from this idealized example, it is easy to see how the output frequency spectrum becomes densely filled when multiple input frequencies are present. In any guitar pickup, vibrations in the strings are converted to an electrical signal that is then amplified. The output of a standard mono guitar pickup is the combined signal of all six strings. Because of this, guitarists will sometimes limit their use of distortion to single notes or power chords consisting of fifths and octave intervals to prevent the sound from becoming muddled. On occasion, musicians have worked around this limitation in the studio with multi-track recording by dubbing single notes on top of each other [4].

2.2 Hexaphonic Distortion

Baptista’s secret for avoiding intermodulation was to use an independent pickup for each string and one distorter for each pickup, for a total of six fuzz distorters per guitar. The guitars Baptista built by hand for his brothers Arnaldo and Sérgio of the psychedelic rock group Os Mutantes had many onboard effects, capable of creating very unique sounds [5]. A hexaphonic pickup, as it is commonly known, provides individual signals for each string. Distorting these signals separately results in fewer intermodulation products while retaining the harmonic enrichment. This unique effect gave the guitar an organ-like singing quality pioneered by Os Mutantes. The first and most famous of these guitars used gold foil shielding and was thus named Regulus – ‘little king’.

The hexaphonic pickup has many uses beyond distortion. Unsurprisingly, it found its most widespread use in guitar synthesizers [6]. Early guitar synthesizers such as the EMS Synthi Hi-Fli (1973) were nothing more than multi-effects processors without actual syn-

thesis circuitry. The late 1970s saw the introduction of three guitar synthesizers; the ARP Avatar, the 360 Systems Spectre, and notably, the Roland GR-500 [7]. Unlike earlier guitar synthesizers, these three comprised a controller guitar fitted with a hexaphonic pickup and a separate synthesizer module connected by a multi-conductor patch cable. The central element in each of these synthesizers was a pitch-to-voltage converter that tracked the pitch of the individual strings to control the polyphonic synthesizer parameters. While the ARP Avatar and the Spectre proved to be prohibitively expensive and commercial failures, Roland continued development with the GR-300. Beginning in 1984 with the GR-700, the controller can output MIDI digital control signals that allow each voice of a polyphonic synthesizer to be controlled by the different strings. Since then, Roland has continually developed a variety of guitar synthesizer products and introduced the widely adopted 13-pin interface for hexaphonic pickups. There are numerous other companies now offering hexaphonic guitars and aftermarket conversion kits. Hexaphonic distortion effects have occasionally appeared on the market, such as the Roland GR-100, and most recently, the Spicetone 6Appeal [8]. Paul Rubenstein, maker of the modern Ubertar hexaphonic guitar pickup [9] has this to say about hexaphonic distortion:

Hexaphonic fuzz (applying fuzz to each string separately) is a way to avoid intermodulation distortion. This means you get the fuzz distortion sound without muddiness. The individual notes are distinct, so the chords are clear yet still fuzzed.

Unfortunately, these effects all require an expensive hexaphonic pickup guitar and for this reason hexaphonic effects have remained niche. This thesis expounds upon a method that aims to simulate hexaphonic distortion using digital signal processing techniques that will approximate the characteristics of the effect when using a regular mono pickup guitar.

2.3 Digital Distortion Effects

The effect now known as distortion was initially the result of overdriving guitar amplifiers past their linear operating range. Tube amplifiers saturate the output at high volumes, effectively clipping the output waveform. This non-linear operation gave the electric guitar its harmonically rich tone. Soon after, transistor-based circuits were used to simulate the effect, offering a wider range of distortion types [7]. Since the advent of digital signal processing, several approaches for the implementation of digital distortion effects have been proposed [10]. The simplest ones employ static non-linear waveshaping functions that clip the amplified input. Conversely, digital emulations based on physical modeling approaches can offer

faithful reproductions of analog audio circuits [11].

2.3.1 Aliasing Distortions

An important consideration in any digital implementation of distortion is aliasing distortion. The non-linear distortion function produces harmonics that extend beyond the $f_s/2$ Nyquist limit of the input signal sampling frequency. High frequency harmonics are folded onto the audio band, resulting in audible inharmonic distortions [2]. This is illustrated in Fig. 2.1, where a normalized 1760 Hz sinusoid was amplified by 40 dB and clipped using an exponential distortion function. This behaviour has been the subject of many papers and is adequately mitigated by oversampling the input signal [10]. Oversampling is a way of suppressing aliasing by first upsampling the input signal then applying the non-linear distortion. The distortion products have amplitudes inversely proportional to the frequency. Increasing the sampling frequency by a factor of N ensures that distortion products beyond $Nf_s/2$ are sufficiently attenuated and thus inaudible when folded over the audio band. A steep lowpass filter attenuates frequencies above the $f_s/2$ Nyquist limit prior to downsampling back to the original sampling frequency.

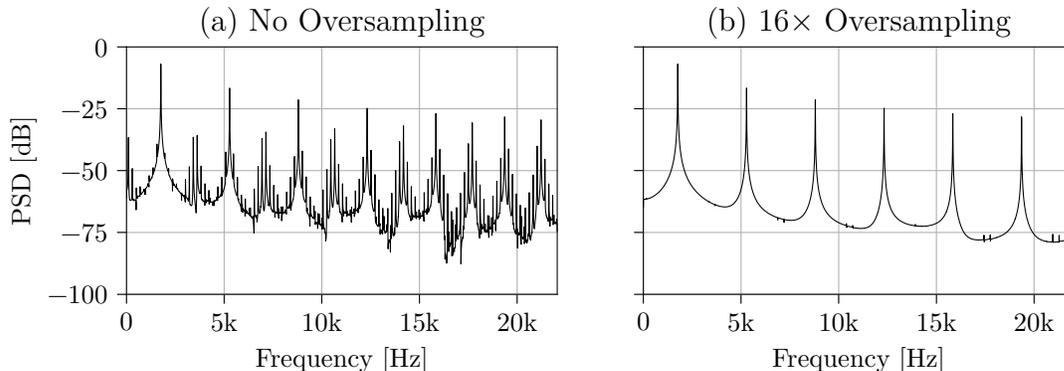


Figure 2.1: *Magnitude spectrum of aliasing distortions of a 1760 Hz sinusoid.*

2.3.2 Reducing Intermodulation Distortion

While there is much literature on digital distortion effects, the role of intermodulation is not often addressed, and only a few algorithms attempt to minimize intermodulation distortion in their design. Such approaches are based on multiband distortion units, the most popular of which is the Quadrafuzz [12]. By splitting the input signal into separate frequency bands

and applying a clipping distortion to each band independently, multiband distortion limits the intermodulation components to those in a given band.

Fernández-Cid and Casajús Quirós [3] took this concept to the extreme, combining a waveshaping function with a 13-band filter bank. Their research was focused on achieving a highly customizable type of distortion through the use of Chebyshev polynomials that offer precise control over the harmonic partials. The algorithm requires a normalized input whose amplitude envelope is flattened. This is achieved by dividing each band of the input signal by an envelope estimation prior to waveshaping and restoring it afterwards. Timoney et al. [13] furthered this concept by showing that static waveshaping could be used to emulate analog distortion circuits. Abel and Werner [4] applied this technique to a modal reverberator architecture to produce distortion without intermodulation products.

However, no one has yet explored a method to simulate hexaphonic distortion. Guitar pickups exhibit their own non-linear behaviour [14]. With the addition of cross-talk between individual pickups and sympathetic vibrations [15], the output signal of a hexaphonic pickup will always contain some unrelated harmonics, which when distorted, results in limited amounts of intermodulation thereby giving it its distinctive tone.

2.4 Digital Processing with Comb Filters

The algorithm proposed in this thesis makes use of a bank of comb filters to extract harmonic content from electric guitar signals. This section highlights some of the important previous audio research that also made use of comb filters.

2.4.1 Harmonic Sound Separation

Many instruments, including the guitar, produce predominantly harmonic sounds that consist of integer multiples of a fundamental frequency f_1 . Auditory source separation techniques can take advantage of the harmonic structure of many natural acoustical signals to significantly increase the robustness of frequency tracking [16]. A comb filter also exhibits peaks at integer multiples of a fundamental frequency and can therefore be used to reduce harmonic interference or to enhance a harmonic signal buried in noise, provided the “teeth” of the comb coincide with the harmonics of the harmonic signal. Välimäki et al. [17] demonstrated several such approaches. A comb filter is used to attenuate the harmonic components of a signal thereby extracting the background noise component. The authors go on to describe a “harmonic extraction filter”; by cascading the comb filter with a 2nd-order resonance filter, it is possible to extract a single harmonic component. Since the calculation f_s/f_1 rarely

yields an integer number sample delay M , in practice, the delay line of the comb filter is implemented as a fractional delay line to avoid such inaccuracies.

When used in parallel, comb filters can separate harmonic signals. A comb filter with peaks located at the multiples of the second harmonic f_2 will extract even harmonics and a complementary comb filter with peaks shifted by $f_2/2$ will extract odd harmonics. This method is used in digital color TV systems to separate the interlaced luminance and chrominance signals in composite video signals [18].

Comb filters have been employed in sound source separation and polyphonic pitch detection, where the main application is automatic music transcription [19]. Traditional pitch detection algorithms based on the extraction of the fundamental frequency struggle with the dynamically changing harmonic components produced by musical instruments. Taking advantage of the fact that musical instrument sounds are composed of a fundamental frequency corresponding to each tone and its harmonic frequencies, Miwa et al. [20] developed a method of detecting the pitch of polyphonic music played by several instruments by observing the intermediate outputs of a series combination of twelve comb filters each attenuating a tone from C to B. Detecting a zero output reveals the presence of the tones corresponding to one or more of the comb filters. By process of elimination, successive reshuffling of the comb filter cascade order eventually reveals the individual tones that make up the signal.

Gainza et al. [19] revisited the use of comb filters for musical sound source separation, this time relying on multi-pitch estimation (MPE) to detect the comprising signal pitches, followed by frequency-domain filtering. In this short-time Fourier transform (STFT)-based approach, the FIR comb filter provides the magnitude response that extracts a harmonic sound source when multiplied with the signal in the frequency domain. Since the peaks of an FIR comb filter are very wide, a configurable number of bins on either side of the spectral peaks are set to zero. While this method is effective in the separation of harmonic signals, the high resolution of the STFT required for MPE and frequency-domain filtering imposes large latencies in the system making it unsuitable for real-time applications. Furthermore, the authors express frustration with the inaccuracies in the pitch detection system.

2.4.2 Comb Filter-Based Harmonic Enhancement

The multiband distortion and modal reverberator architectures discussed in the previous section are preprocessing approaches to harmonic-enhancing distortion, whereby the signal is separated prior to distortion. In applications where the desired harmonics are predetermined, such as in classical waveform synthesis, an IIR comb filter postprocessing approach to alias reduction has been proposed by J. Pekonen and V. Välimäki [21]. In the same way that digital

implementations of distortion produce aliasing, sampling of continuous-time waveforms with frequencies above the Nyquist limit introduces aliasing distortions. The authors proposed a combination of IIR comb filter and DC blocking filter to attenuate the aliased components that lie between the harmonics.

A postprocessing approach as such may also prove to be an effective method of removing intermodulation distortions, provided that the individual pitches of the original signal are accurately tracked. Because of the numerous difficulties involved with polyphonic pitch detection mentioned previously, the hexaphonic distortion simulation proposed in this thesis is designed to avoid intermodulation distortions from being generated in the first place.

2.4.3 Other Uses of Comb Filters

Traube and Depalle [22] used an FIR comb filter structure in a simple digital physical model of a guitar string. The notches in the spectrum correspond to the harmonics that have a node at the plucking point. With this model, the plucking position can be extracted from a recording by fitting the comb filter delay value to the measured spectral envelope.

The extraction of pitch from a musical excerpt is closely related to the extraction of tempo, with the difference being that frequencies of interest are greatly reduced – typically corresponding to the range 30 to 240 beats per minute. Scheirer [23] describes an algorithm for beat-tracking musical signals of several genres. Drawing from a psychoacoustic model of rhythmic perception, the signal is divided into six bands from which amplitude envelopes are calculated. This vastly simplifies the analysis data. Each processed envelope channel is passed to a filter bank of comb filters for which the delays cover a range of pulse frequencies. The filter with response peaks matching the signal will have larger output. The filter with maximum energy output is selected as the tempo of the signal.

Comb filters are often found in echo simulation and reverberation effects. The influential Schroeder and Moorer reverberation architectures employ parallel feedback comb filters and a cascaded all-pass structure [24, 25]. Both of these types of filters will be seen in the development of the harmonic separation structure proposed in this thesis.

Chapter 3

Hexaphonic Pickup Recordings

The first step in development was the recording of an actual hexaphonic pickup and is the topic of this chapter. The recordings provided vital insights into hexaphonic audio signal characteristics and later served as the reference for comparison in the evaluation of the simulated hexaphonic distortion. Additionally, this chapter explores the implications of sympathetic vibration and cross-talk in hexaphonic pickups.

3.1 Recording Setup

The electric guitar used in the audio recordings for this research was a Peavey Raptor Plus fitted with an Ubertar Hex Plus pickup. The Peavey Raptor series of Stratocaster-style electric guitars have three pickups. The neck and middle pickups are single coil pickups and the bridge pickup is a dual coil humbucker. A five-way switch selects either one of the neck, middle, or bridge pickups, or a combination of the neck-middle or middle-bridge pickups.

The Ubertar hexaphonic pickup [9] takes the shape of a single coil pickup but consists of six low-noise HC coils. A six-conductor wire lead connects to a 7-pin output jack, allowing for simultaneous output of both hexaphonic and mono signals¹. The length of the wire lead on the hexaphonic pickup was too short to install it in the neck position, making the middle position the most straightforward option. The hexaphonic pickup was also slightly wider than the pickup it was replacing and the slot in the pickguard had to be filed down slightly.

The Ubertar breakout box provides 1/4" outputs for each string. These were connected to the high impedance instrument inputs of an RME Micstasy 8-channel preamplifier and A/D converter. The mono neck pickup was simultaneously recorded on the seventh channel.

¹Here mono refers to the signal captured from a single or dual coil humbucker pickup. This is a summed signal consisting of the vibrations of all six strings.

A photo of the recording equipment setup at the Centre for Interdisciplinary Research in Music Media and Technology (CIRMMT) is shown in Fig. 3.1.



Figure 3.1: *Hexaphonic recording setup. The hexaphonic pickup is the middle pickup. The color patch cables connect the breakout box to channels 1 through 6. An additional patch cable connects the mono pickup to input channel 7.*

The height of the hexaphonic pickup was adjusted for even volumes across each string and each string was carefully tuned using an electronic tuner prior to each recording. The audio was recorded using Adobe Audition, whose multitrack audio editing and frequency display features proved very useful during the analysis.

3.2 Audio Analysis

To capture a full set of analysis data, the recordings consisted of single pluck and strum recordings of the open strings and the first twelve frets with the use of a capo — a clamp fastened across all the strings. Plucking each string with the capo at each fret position creates a complete set of recordings covering the entire chromatic scale over three octaves from E_2 to E_5 . As an example, Fig. 3.2 shows the six-channel hexaphonic pickup recording and Fig. 3.3 the mono neck pickup recording of the open strings.

Up until this recording experience, it was unclear what digital signal processing methods should be pursued in the formulation of the hexaphonic distortion simulation. Many approaches were in consideration, most of which relied on some incarnation of a filter bank. However, upon initial observations of the magnitude spectrums of the recordings, which looked much like the familiar comb filter magnitude response, the use of a bank of parallel comb filters rapidly became the most intuitive solution. As seen in Fig. 3.4, the separate outputs of the hexaphonic pickup are predominantly harmonic with a magnitude response

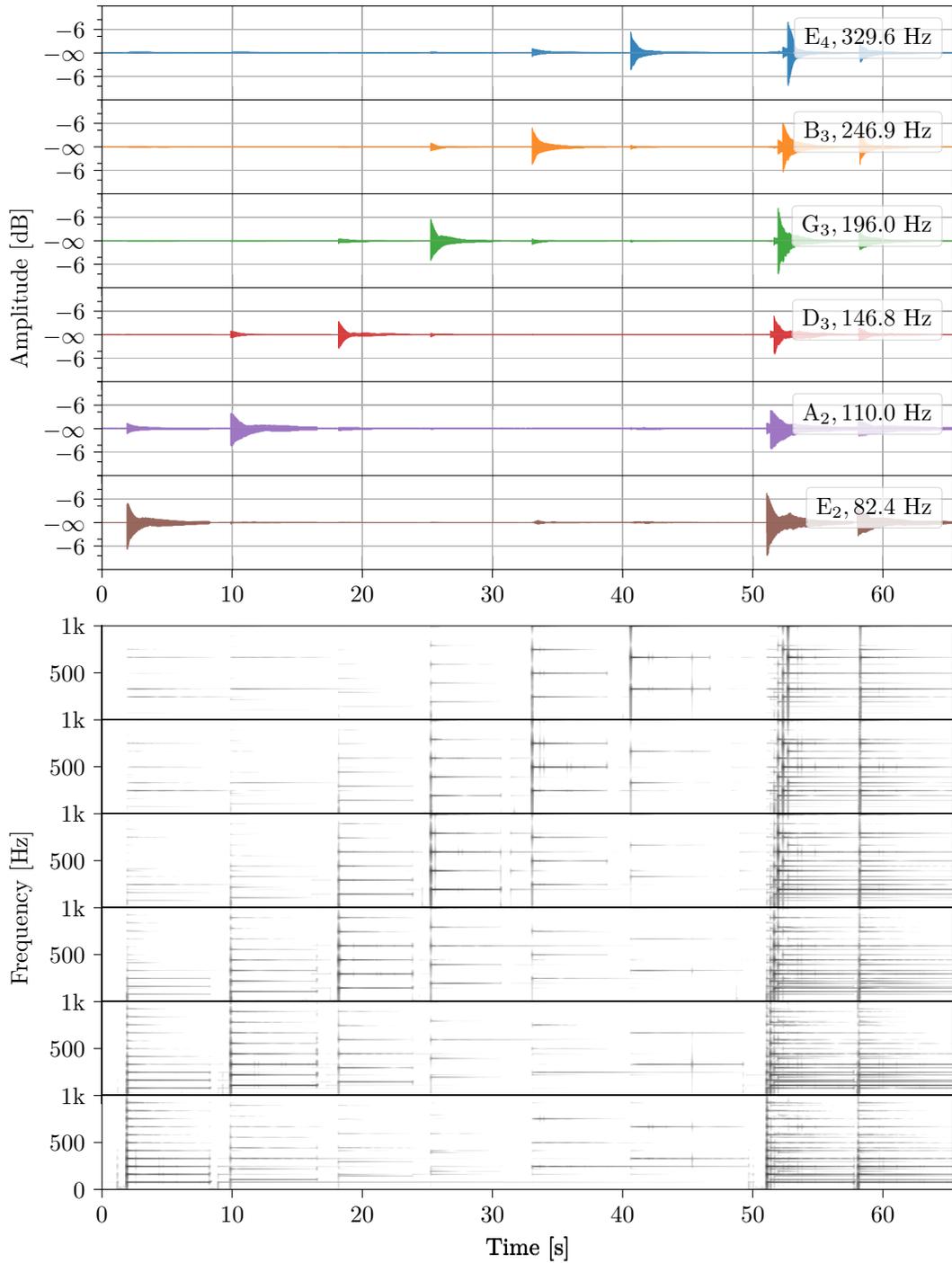


Figure 3.2: *Open strings hexaphonic pickup recording. There is a small amount of signal cross-talk (bleed) to adjacent pickups. The presence of a signal in the low E₂ pickup when plucking high E₄ is due to sympathetic vibrations.*

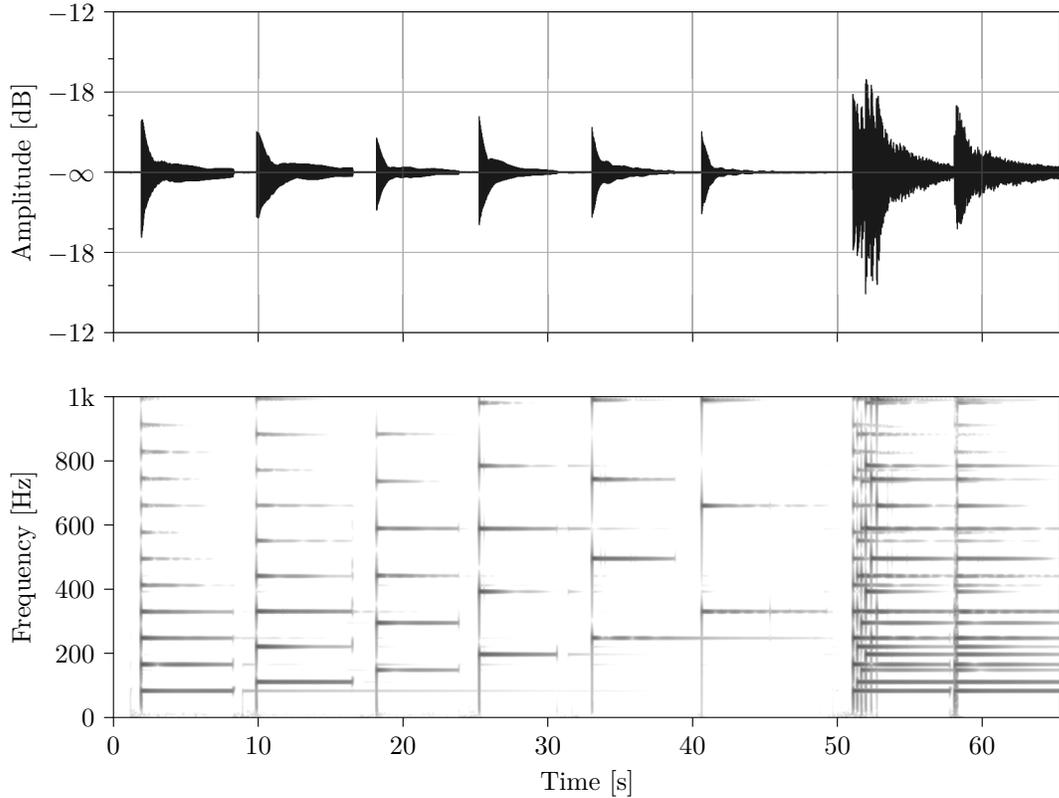


Figure 3.3: *Open strings mono pickup recording. Plucks of low E_2 to high E_4 followed by two strums.*

resembling the teeth of a comb filter. This lead to the theory that a signal composed of several of these harmonic signals could be decomposed into harmonic signals using comb filters.

3.3 Sympathetic Vibrations

Measurements with a hexaphonic pickup provide a unique opportunity to compare and analyze sympathetic vibrations in guitar strings versus cross-talk from adjacent pickups. Sympathetic vibration is the transmission of energy of a plucked string to the other strings through the bridge [15]. Cross-talk, on the other hand, is the contamination of a pickup signal due to vibrations of an adjacent string. The measurements shown in Fig. 3.2 demonstrate both cross-talk and sympathetic vibrations, though it requires some careful analysis to distinguish the two. Cross-talk is readily apparent in the signals from adjacent strings for

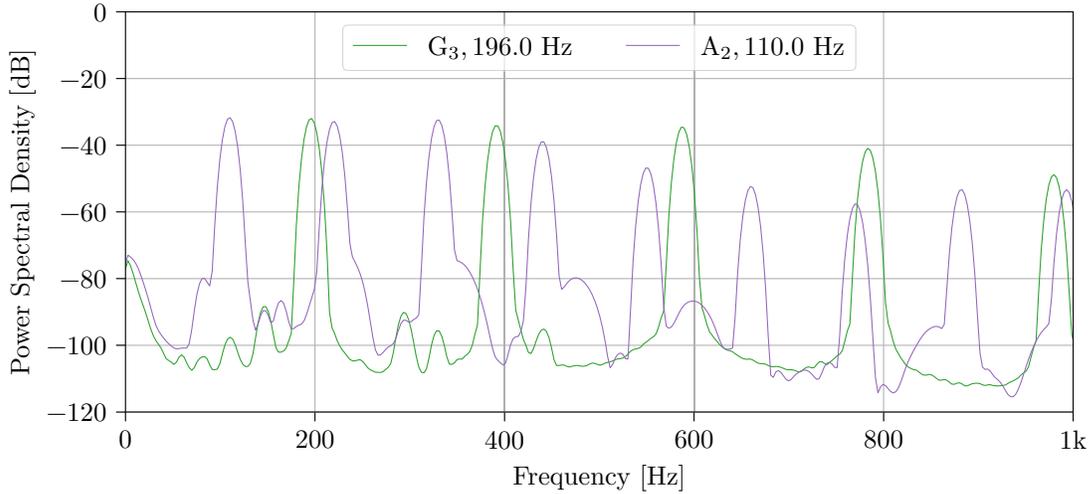


Figure 3.4: *Magnitude spectrum of plucked open A_2 and G_3 strings recorded using a hexaphonic pickup.*

each individually plucked open string. For example, when the D_3 string is plucked (red), a bit of signal is seen in both the A_2 and G_3 pickup signals. At the same time, sympathetic vibrations might appear in other pickup signals if any of the plucked string frequency partials match any partials in the other strings. For example, when the high E_4 is plucked, the next highest signal amplitude is found at the low E_2 pickup, which is the farthest string from the one plucked. The third highest signal amplitude is in the A_2 string. The partials of the E strings line up exactly and the second partial of the A_2 string (330.0 Hz) is very close to the first partial of the high E_4 string (329.6 Hz). If the frequency ratio between strings is a simple integer fraction, the coincident partials that occur at the common multiples are numerous and the resonance of the string increases.

In a parallel exploration of consonance and dissonance, early investigations by Helmholtz found that note intervals with small integer ratios were perceived as more consonant. Later psychoacoustic studies support the theory that coincident partials produce consonant intervals and a number of models have been suggested to provide a concrete definition of consonance [26]. While perceptual studies [27] show that human perception of consonance and dissonance is more nuanced than these models suggest, they will serve adequately in this exploration of sympathetic vibration.

Harmonic entropy is one such model. It is a measure of interval consonance, where a lower $HE(i)$ indicates a more consonant interval. A detailed definition of harmonic entropy is provided in Sec. 6.1.1. It turns out that this definition also provides a good model of

sympathetic vibration. Consider the intervals between a plucked high E_4 string and each of the other strings in Table 3.1.

Table 3.1: *The equal-tempered intervals i between E_4 and the other open strings notes and the corresponding harmonic entropy.*

Open string note	E_4	B_3	G_3	D_3	A_2	E_2
Frequency [Hz]	329.6	246.9	196.0	146.8	110.0	82.4
i [cents]	0	500	900	1400 (200)	1900 (700)	2400 (1200)
$HE(i)$	0.021	1.873	2.979	2.48	0.073	0.021

Unsurprisingly, the two-octave interval between the high E_4 and low E_2 string has an equally low harmonic entropy value. The sympathetic vibrations observed in the A_2 string are also confirmed by a low entropy value. Figure 3.5 shows the root mean square (RMS) amplitude for each string of the recording from Fig. 3.2 from 40 to 50 seconds in which the high E_4 string is plucked. The RMS amplitude illustrates the combination of sympathetic vibrations and cross-talk present in a given string caused by plucking the E_4 string. A scaled plot of the harmonic entropy values from Table 3.1 is overlaid. The plot is reflected about the horizontal axis such that complex intervals contribute to less sympathetic vibration. The close fit suggests that the harmonic entropy model could prove to be a useful tool in modeling or distinguishing sympathetic vibration. Such a model may be useful in instrument simulation algorithms [28] that utilize a coupling matrix to synthetically generate sympathetic vibration, providing parameter values at any tuning.

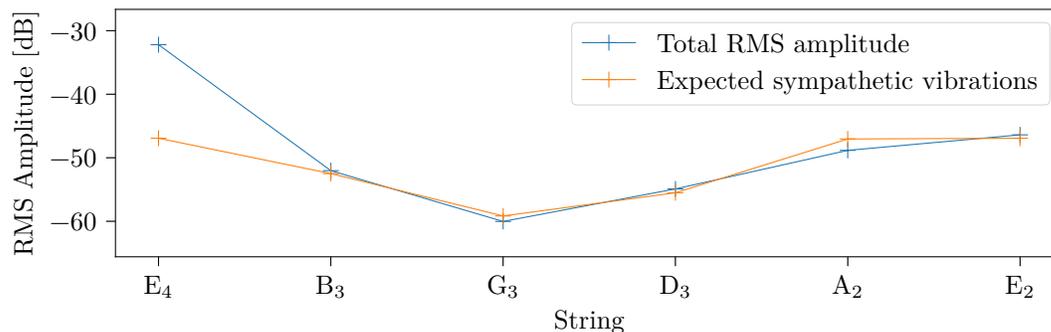


Figure 3.5: *Total RMS amplitude for each pickup when plucking a high E_4 overlaid with the expected propensity of each string to sympathetically resonate.*

Chapter 4

Separation of Harmonic Signals

This chapter describes the harmonic signal separation algorithm by introducing several digital signal processing concepts, discussing their implementation in the system, and describing the decisions made during the design process of the parallel comb filter bank.

4.1 Comb Filters

Vibrating strings generally exhibit harmonic spectral content. In the case of the electric guitar, a single guitar tone consists of frequency components at a fundamental plus all odd and even harmonics with magnitudes of the higher partials diminishing as the frequency increases [2]. This intrinsic characteristic of plucked strings is made evident by the strong presence of harmonics in the individual pickup signals. Frequency analysis of the hexaphonic pickup signals inspired a parallel comb filtering structure that separates a mono signal into harmonically related signals for subsequent effect processing.

4.1.1 FIR Comb Filter

The finite impulse response (FIR) feedforward comb filter is obtained by summing an input signal with the same signal delayed by M samples. From the block diagram in Fig 4.1, the difference equation of the feedforward comb filter is given by

$$y[n] = b_0x[n] + b_Mx[n - M], \quad (4.1)$$

where $x[n]$ is the input signal and $y[n]$ is the output signal. b_M is the feedforward gain coefficient and b_0 is the blend coefficient.

A system with a finite impulse response $h[n]$ is called an FIR system. The impulse

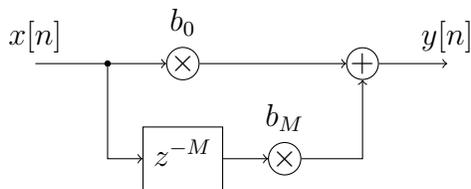


Figure 4.1: *FIR comb filter block diagram.*

response of the feedforward comb filter is given by

$$h[n] = b_0\delta[n] + b_M\delta[n - M], \quad (4.2)$$

where the unit impulse is defined as

$$\delta[n] = \begin{cases} 1, & n = 0 \\ 0, & n \neq 0. \end{cases} \quad (4.3)$$

The impulse response of the feedforward comb filter therefore simply consists of two non-zero terms, thus satisfying the finite duration of $M + 1$ samples.

The transfer function relates the Z-transforms of input signal and output signal of the described system,

$$H_{\text{FF}}(z) = \frac{Y(z)}{X(z)} = b_0 + b_M z^{-M}. \quad (4.4)$$

A plot of the frequency response is shown Fig. 4.2. The frequency is expressed in radians per sample, where

$$\omega = \frac{2\pi f}{f_s}. \quad (4.5)$$

The feedforward comb filter has spectral peaks at frequencies that are multiples of $2\pi/M$ for positive values of b_M and nulls at multiples of $2\pi/M$ for negative values of b_M . In other words, changing the sign of b_M results in a comb filter with peaks shifted by π/M . To extract a harmonic signal, the peaks must coincide with the integer multiple harmonics so only positive coefficients are of interest.

By evaluating the magnitude response at the peaks and nulls, it can be observed that the magnitude response has a minimum of $b_0 - b_M$ at the nulls and reaches a maximum of $b_0 + b_M$ at the peaks. The 3-dB width $\Delta\omega$ of the comb filter is defined as the width of the peaks at half the maximum of the magnitude squared response. This corresponds to the point at which the power of the output signal is reduced to half of the maximum. When the

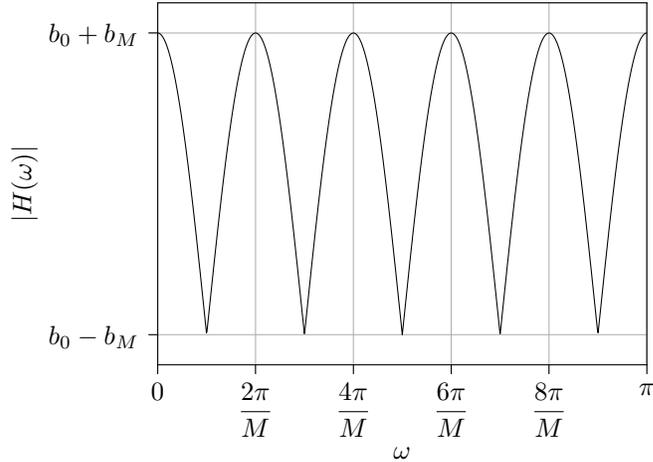


Figure 4.2: *FIR comb filter magnitude response where $M = 10$.*

filter has unity gain, it can be found by solving the equation

$$|H(\omega)|^2 = \frac{1}{2}. \quad (4.6)$$

Equivalently in decibels,

$$20 \log(|H(\omega)|) = 20 \log\left(\frac{1}{\sqrt{2}}\right) = -3.01 \text{ dB}. \quad (4.7)$$

For a normalized magnitude response between 0 and 1, $b_0 = b_M = 1/2$ [17]. The FIR transfer function becomes

$$H(z) = \frac{1}{2}(1 + z^{-M}). \quad (4.8)$$

This filter has the 3-dB width of the peaks $\Delta\omega = \pi/M$. This corresponds to the maximum possible peak width, equal to the separation between peaks. While the FIR comb filter has the advantage of having a short impulse response, the wide peaks of the FIR filter insufficiently attenuate neighbouring frequencies.

4.1.2 IIR Comb Filter

The infinite impulse response (IIR) feedback comb filter is obtained by summing an input signal with a delayed version of the output attenuated by a feedback gain a_M . This produces a magnitude response with the appearance of a comb where a larger gain factor produces

sharper peaks. The feedback comb filter in Fig. 4.3 has the difference equation

$$y[n] = x[n] + a_M y[n - M]. \quad (4.9)$$

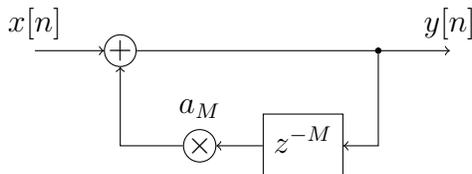


Figure 4.3: *IIR comb filter block diagram.*

The feedback comb filter has the infinite impulse response

$$h[n] = \sum_{k=0}^{\infty} a_M^k \delta[n - kM] \quad (4.10)$$

and zero elsewhere. The corresponding transfer function is given by

$$H_{\text{FB}}(z) = \frac{1}{1 - a_M z^{-M}}. \quad (4.11)$$

A plot of the frequency response is shown Fig. 4.4. In a similar fashion to the feedforward comb filter, the feedback comb filter has peaks at frequencies that are multiples of $2\pi/M$ for positive values of a_M and nulls at multiples of $2\pi/M$ for negative values of a_M . In other words, changing the sign of a_M results in a comb filter with peaks shifted by π/M . To extract a harmonic signal, the peaks must coincide with the harmonics so only positive coefficients are of interest.

The magnitude response has a minimum of $1/(1 + a_M)$ at the nulls and a maximum of $1/(1 - a_M)$ at the peaks, with $|a_M| < 1$ required for stability. Unlike with the FIR comb filter, a null magnitude of 0 is unattainable. However, the IIR comb filter has an adjustable peak width $\Delta\omega$ better suited to the separation of harmonic signals.

The ideal comb filter for harmonic separation would offer both an adjustable peak width and a normalized magnitude response with null values of 0 corresponding to a $-\infty$ dB attenuation between harmonic frequencies. This filter is achieved in the form of the universal comb filter constructed from the combination of the FIR and IIR comb filters [2].

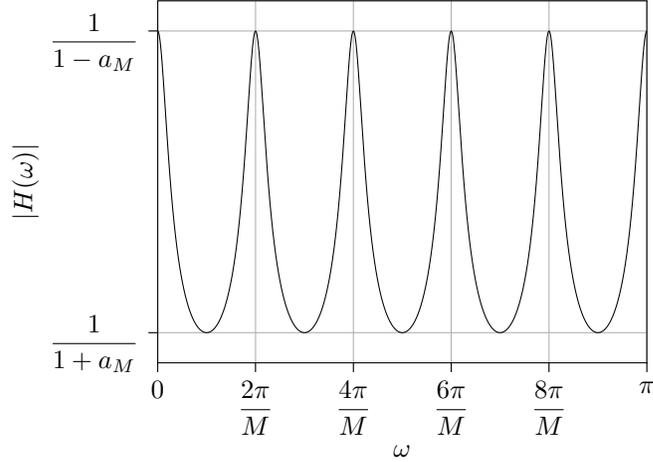


Figure 4.4: *IIR comb filter magnitude response where $M = 10$.*

4.1.3 Universal Comb Filter

The series combination of the feedforward (4.4) and feedback (4.11) comb filters leads to the universal comb filter with a new response given in the frequency-domain as

$$H_{\text{UNI}} = H_{\text{FF}}H_{\text{FB}}. \quad (4.12)$$

The direct-form I difference equation and transfer function of the universal comb filter is given by

$$y[n] = b_0x[n] + b_Mx[n - M] + a_My[n - M], \quad (4.13)$$

$$H_{\text{UNI}}(z) = \frac{b_0 + b_Mz^{-M}}{1 - a_Mz^{-M}}. \quad (4.14)$$

This implementation requires two delay lines. Alternatively, the canonical implementation shown in Fig. 4.5 requires a single delay line and is described by the difference equations

$$\begin{aligned} x_h[n] &= x[n] + a_Mx_h[n - M] \\ y[n] &= b_0x_h[n] + b_Mx_h[n - M]. \end{aligned} \quad (4.15)$$

In the special case where $b_0 = -a_M$ and $b_M = 1$, this filter structure reverts to a first-order allpass filter used in Schroeder reverberators [24]. The magnitude response of the FIR, IIR, and universal comb filters is summarized below. With the universal comb filter, the

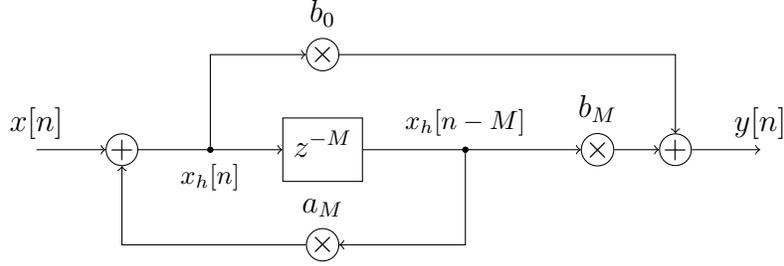


Figure 4.5: *The canonical universal comb filter.*

peak and null values along with the 3-dB bandwidth can be set to any desired value.

	$H(\omega)$	$ H(0) $	$ H(\frac{\pi}{M}) $
Feedforward	$\frac{b_0 + b_M e^{-j\omega M}}{1}$	$b_0 + b_M$	$b_0 - b_M$
Feedback	$\frac{1 - a_M e^{-j\omega M}}{1 - a_M}$	$1 - a_M$	$1 + a_M$
Universal	$\frac{b_0 + b_M e^{-j\omega M}}{1 - a_M e^{-j\omega M}}$	$\frac{b_0 + b_M}{1 - a_M}$	$\frac{b_0 - b_M}{1 + a_M}$

(4.16)

4.1.4 Comb Filter Parameters

The delay of the filter M is fixed according to the fundamental frequency of interest. To extract frequencies that are multiples of f_1 the length of the delay line is set to

$$M = \frac{2\pi}{\omega_1} = \frac{f_s}{f_1} \quad \text{where} \quad \omega_1 = \frac{2\pi f_1}{f_s}. \quad (4.17)$$

The comb filter has peaks at the multiples

$$\omega_k = k \frac{2\pi}{M} \quad \text{or} \quad f_k = k \frac{f_s}{M}, \quad k = 0, 1, \dots, M-1 \quad (4.18)$$

and nulls at the half-multiples

$$\omega_k = \frac{(2k+1)2\pi}{2M} \quad \text{or} \quad f_k = \frac{(2k+1)f_s}{2M}, \quad k = 0, 1, \dots, M-1. \quad (4.19)$$

The feedforward coefficients are chosen such that the peaks of the comb filter are normalized for unity-gain with the nulls having a value of 0. Solving $H(\omega) = 1$ for b_0 when $\omega = 0$ gives

$$b_0 = b_M = \frac{1 - a_M}{2}. \quad (4.20)$$

The choice of feedback coefficient places the nulls of the FIR comb filter between the peaks

of the IIR comb filter. As such,

$$a_M = \frac{1 - \beta}{1 + \beta}, \quad b_0 = b_M = \frac{\beta}{1 + \beta}, \quad \text{where } \beta = \tan\left(\frac{M\Delta\omega}{4}\right), \quad (4.21)$$

for a desired 3-dB width $\Delta\omega$ in the range $0 \leq \Delta\omega \leq \pi/M$. This maximum $\Delta\omega$ occurs when the 3-dB peak width is equal to the separation between peaks. This limit applies the following constraints on β and a_M :

$$0 \leq \beta \leq 1, \quad 0 \leq a_M \leq 1. \quad (4.22)$$

Figure 4.6 shows the comb filter magnitude response for different 3-dB widths.

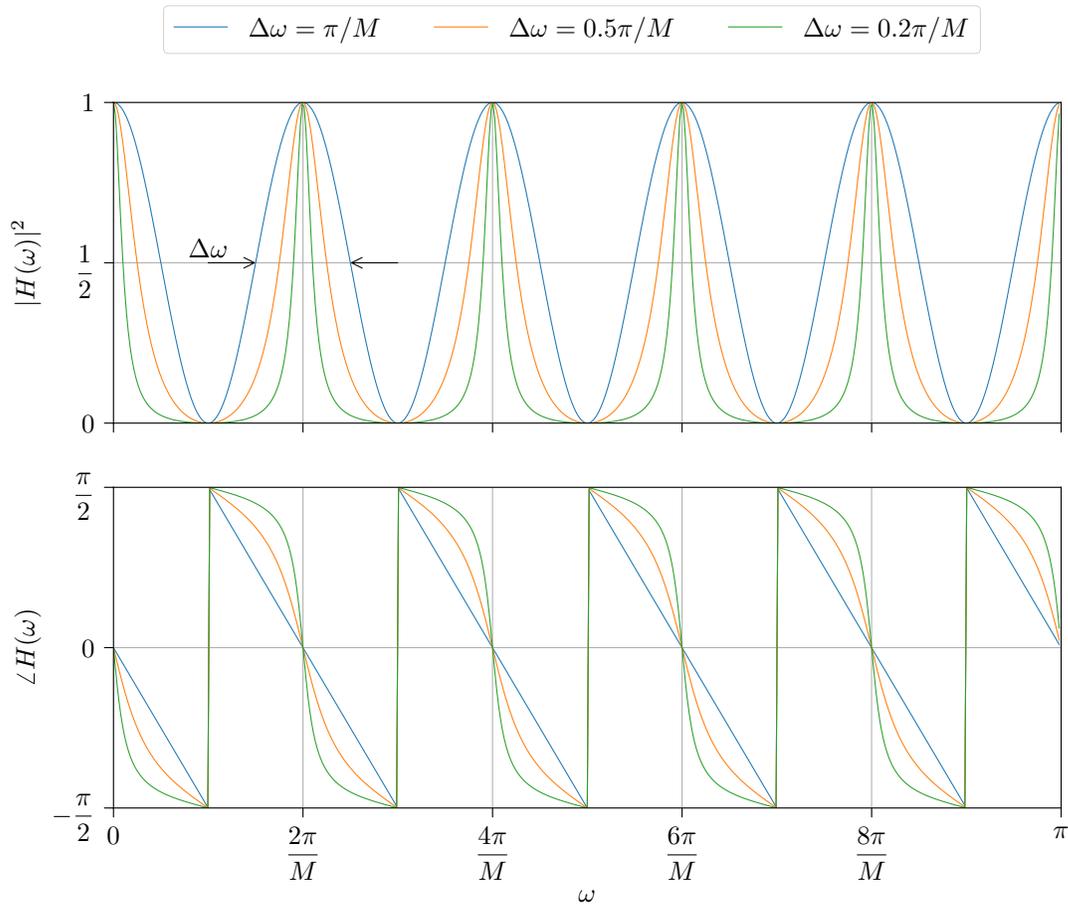


Figure 4.6: The magnitude squared and phase responses for different choices of $\Delta\omega$ where $M = 10$.

The corresponding quality factor is given by

$$Q = \frac{\omega_1}{\Delta\omega} = \frac{2\pi/M}{\Delta\omega}, \quad (4.23)$$

such that a narrower peak results in a higher Q . An equivalent expression for β is

$$\beta = \tan\left(\frac{\pi}{2Q}\right). \quad (4.24)$$

where $Q \geq 2$.

4.1.5 Impulse Response

The filtering operation results in an inherent temporal smearing of the signal that has the effect of smoothing out sharp transitions. The amount of temporal smearing is dictated by the filter impulse response. The universal comb filter has the causal impulse response

$$h[n] = b_0\delta[n] + b_M\delta[n - M] + a_M h[n - M] \quad (4.25)$$

Iterating the first few multiples of M ,

$$\begin{aligned} h[0] &= b_0(1) + b_M(0) + a_M h[-M] = b_0, \\ h[M] &= b_0(0) + b_M(1) + a_M h[0] = a_M b_0 + b_M, \\ h[2M] &= b_0(0) + b_M(0) + a_M h[M] = a_M(a_M b_0 + b_M), \\ h[3M] &= b_0(0) + b_M(0) + a_M h[2M] = a_M^2(a_M b_0 + b_M). \end{aligned}$$

In the exponentially decaying form,

$$h[n] = \begin{cases} b_0, & n = 0 \\ a_M^{n/M-1}(a_M b_0 + b_M), & n = M, 2M, 3M, \dots, \end{cases} \quad (4.26)$$

and zero elsewhere. Ignoring for the moment the $n = 0$ case and substituting $n = t f_s$, the decay rate of the impulse response can be determined by

$$\frac{h(t f_s)}{h(0)} = \frac{a_M^{t f_s/M-1}(a_M b_0 + b_M)}{a_M^{0/M-1}(a_M b_0 + b_M)} = \frac{a_M^{t f_s/M-1}}{a_M^{0/M-1}} = a_M^{t f_s/M} \quad (4.27)$$

This decay rate is the same as that of the IIR comb filter. Since M is fixed, the a_M coefficient solely defines the impulse response decay time for a given sampling frequency. The decay time is typically represented by the t_{60} measure, defined as the time for the impulse response to decay by 60 dB,

$$a_M^{t_{60} f_s/M} = 10^{-60/20} = 0.001. \quad (4.28)$$

Taking the logarithm of both sides and rearranging gives the decay time:

$$\frac{t_{60}f_s}{M} \log(a_M) = -\frac{60}{20}, \quad (4.29)$$

$$t_{60} = \frac{-3M}{f_s \log a_M}. \quad (4.30)$$

Given the definition in (4.20), and the constraints of (4.22), it can be observed that the maximum amplitude of the impulse response occurs not at $n = 0$ but rather $n = M$:

$$h[M] = a_M b_0 + b_M = a_M b_0 + b_0, \quad (4.31)$$

which is greater than $h[0] = b_0$. This is illustrated in Fig. 4.7.

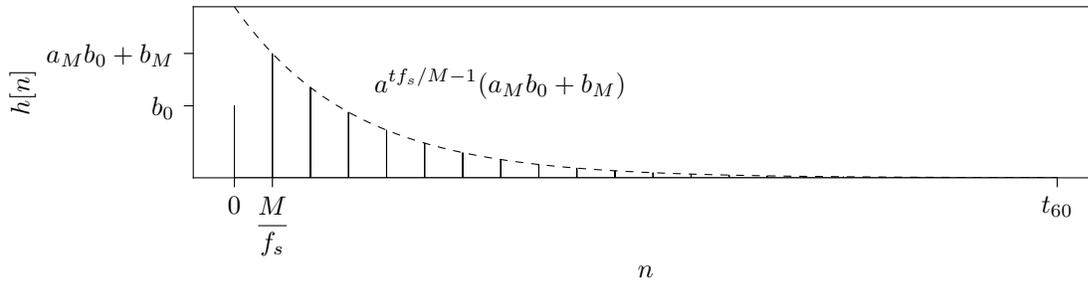


Figure 4.7: Typical impulse response of the comb filter with $h[0]$, $h[M]$, and t_{60} annotated. The dashed line shows the exponential decay function.

Another way to demonstrate this is by finding the causal inverse Z-transform of the transfer function by partial fraction expansion:

$$H_{\text{UNI}}(z) = b_0 + \frac{(a_M b_0 + b_M) z^{-M}}{1 - a_M z^{-M}} \quad (4.32)$$

$$h[n] = b_0 \delta[n] + (a_M b_0 + b_M) \sum_{k=0}^{\infty} a_M^k \delta[n - kM - M] \quad (4.33)$$

After sample M , the impulse response of a pole-zero filter with M zeros behaves like that of an all-pole filter [18]. By considering the start of the decay at $n = M$, the decay time becomes

$$t_{60} = \frac{M}{f_s} \left(\frac{-3}{\log a_M} + 1 \right), \quad (4.34)$$

increasing the previous result by M/f_s . This small penalty is incurred by the addition of the feedforward terms, with the benefit being $-\infty$ dB attenuation at the nulls.

Using the definition in equation (4.24), the a_M coefficient can be expressed as

$$a_M = \frac{1 - \tan(\pi/2Q)}{1 + \tan(\pi/2Q)}. \quad (4.35)$$

With the addition of definition (4.17), equation (4.34) can be expressed solely in terms of f_1 and Q factor:

$$t_{60} = \frac{1}{f_1} \left(\frac{-3}{\log\left(\frac{1 - \tan(\pi/2Q)}{1 + \tan(\pi/2Q)}\right)} + 1 \right). \quad (4.36)$$

This relationship between f_1 , Q , and t_{60} is illustrated in Fig. 4.8.

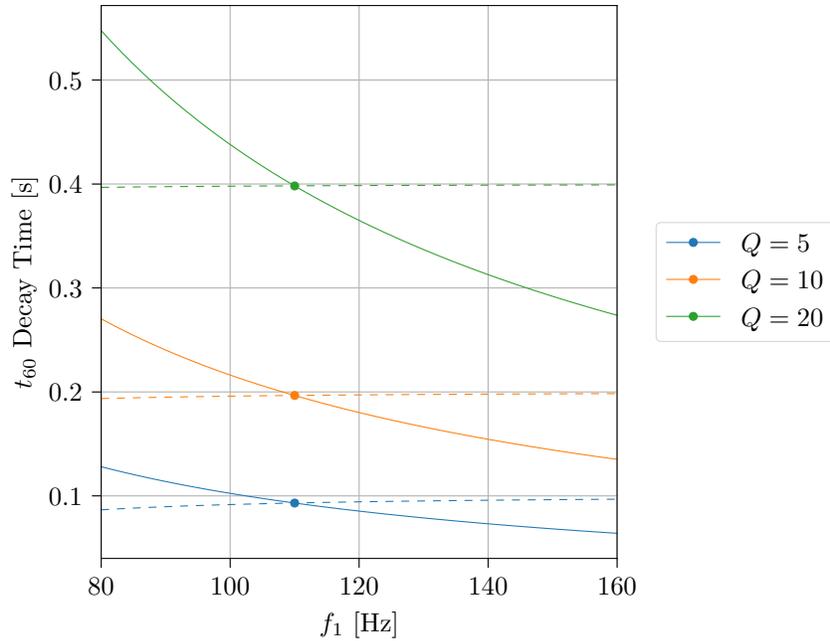


Figure 4.8: Three choices of Q are plotted with their corresponding t_{60} decay time for the frequency 110 Hz. t_{60} values are extended for constant Q (solid line) and constant $\Delta\omega$ (dashed line).

The pluck or attack part of a guitar note is severely attenuated for a very long decay time t_{60} . This is made less noticeable with distortion, but the individual onsets get blurred together when played in fast succession. At the same time, a higher Q results in a smaller peak width and is more effective in attenuating neighbouring frequencies. This creates a

trade-off between harmonic separation and sharp transitions. In the implementation, this is left as a user-adjustable parameter.

4.2 Parallel Comb Filters

The mono guitar pickup signal can be modeled as the sum of harmonic signals. A bank of parallel comb filters can separate a mono signal into separate signals for subsequent effect processing. The separation of harmonic signals is illustrated in the comparison of the original six-channel hexaphonic recording in Fig. 3.2, the mono recording in Fig. 3.3, and the six channels extracted from the mono signal in Fig. 4.9. The parallel comb filters recover each of the individual string plucks by separating each tone. When an arbitrary signal is separated into harmonic components, distortion can be applied in a way that only produces harmonic distortion products.

4.2.1 Simulating Sympathetic String Effects

As seen in Fig. 3.2, the hexaphonic pickup exhibits small amounts of cross-talk in which an individual pickup will capture the vibrations of an adjacent string. The Ubertar HC-series coils offer excellent separation. In addition to cross-talk, strings that are not directly interacted with also resonate due to sympathetic vibrations. Interestingly, the parallel filter bank exhibits a characteristic analogous to the sympathetic vibrations captured by a hexaphonic pickup. For example, as seen in Fig. 4.9, the comb filter with fundamental frequency 82.4 Hz has peaks that capture the partials of the B₃ string (246.9 Hz) and also those of the high E₄ string (329.6 Hz). In effect, it recaptures some of the sympathetic vibration signals found in hexaphonic recordings. The effect can be excessive in some cases, causing a note to appear more prominently in the output of comb filter of a different pitch, as seen in the plucking of the B₃ string (246.9 Hz) between 32 and 40 seconds.

4.2.2 Filter Bank

Guitar fundamental frequencies range from 82.4 Hz to a little over 1 kHz with the upper limit depending on the number of frets. A filter bank of twelve comb filters is sufficient to separate all harmonics belonging to the musical notes of the equal-tempered chromatic scale. The fundamental frequencies of these filters are set to the chromatic progression of the sixth string, being the lowest octave in standard tuning. Higher octaves of a given fundamental frequency are captured in the higher harmonic peaks. In this implementation, the filters conform to the equal temperament tuning of the guitar fretboard. The comb filters can be

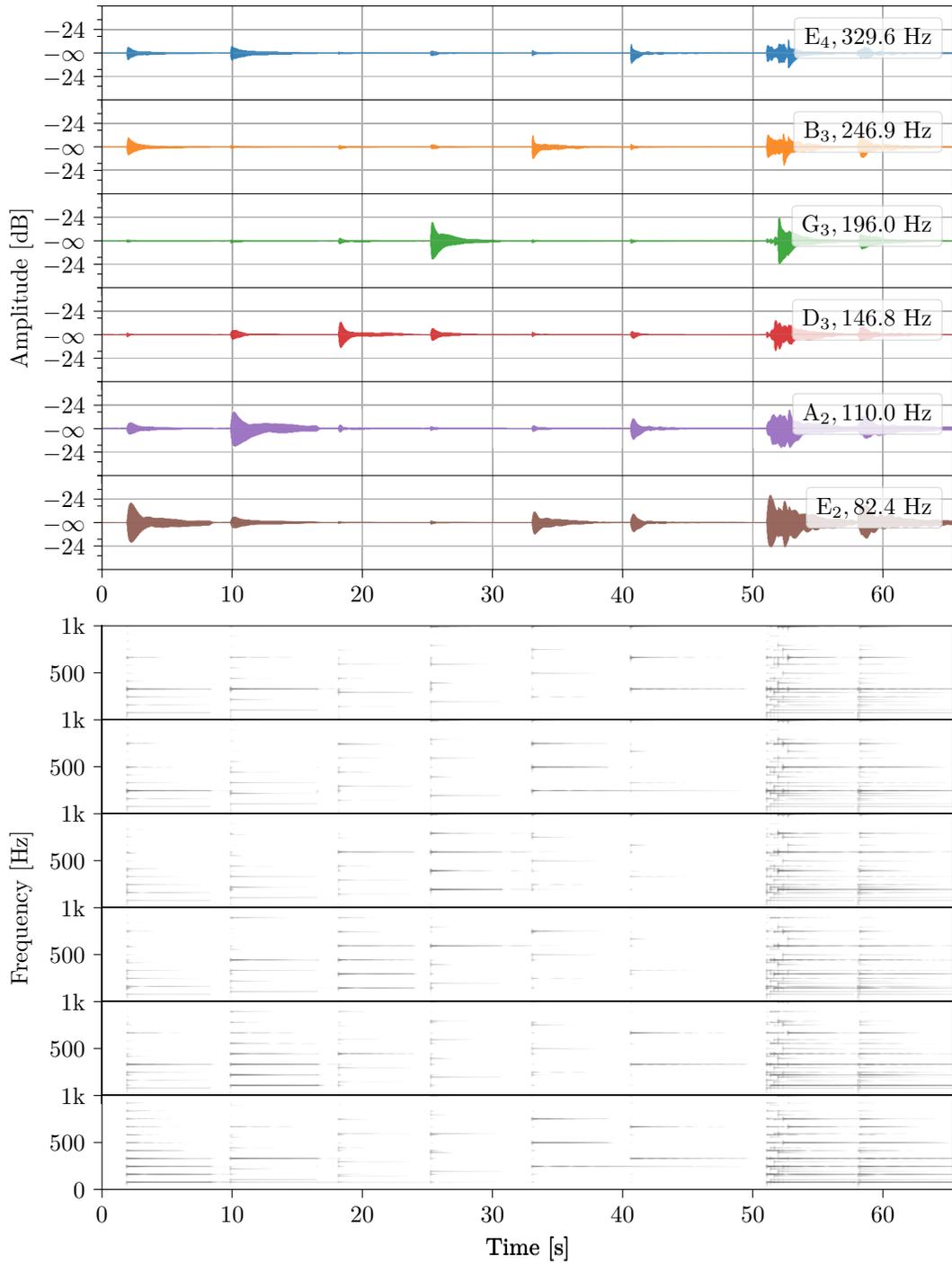


Figure 4.9: *Output signals of six comb filters with peaks aligned to the harmonics of the open string frequencies. Input signal is the mono pickup recording. Where the peaks of the comb filters overlap, the harmonics of the tones appear at the output of multiple comb filters.*

set for any tuning system when used on fretless instruments. With inspiration from Miwa et al. [20], the i th harmonic of tone p is notated $f_{i,p}$, with each harmonic being a multiple of the fundamental frequency,

$$f_{i,p} = i f_{1,p}, \quad p = \text{E, F, } \dots, \text{D}^\sharp. \quad (4.37)$$

The comb filter transfer function with peaks at the harmonics of tone p is denoted $H_p(z)$. The twelve parallel comb filters have delay line lengths given in Table 4.1. Without the use of oversampling, the rounding error caused by an integer sample delay, with a sample rate of 44.1 kHz, is at most 0.17%. However, with $16\times$ oversampling, which is necessary for the distortion processing to be discussed in Sec. 5.4, the rounding error is reduced to 0.01% in the worst case, avoiding the need for fractional delay line methods.

Table 4.1: *The twelve equal-tempered fundamental frequencies and the integer delays with a sampling rate of 16×44.1 kHz.*

Tone p	E	F	F $^\sharp$	G	G $^\sharp$	A	A $^\sharp$	B	C	C $^\sharp$	D	D $^\sharp$
$f_{1,p}$ [Hz]	82.4	87.3	92.5	98	103.8	110	116.5	123.5	130.8	138.6	146.8	155.6
Delay M	8562	8082	7628	7200	6796	6415	6055	5715	5394	5091	4805	4536

The magnitude response of the filter bank is shown in Fig. 4.10. Notes separated by consonant intervals will have overlapping harmonic peaks. Here the comb filters share a constant Q factor so that the overlaps are made visible.

The combined transfer function of the filter bank is obtained by summing the frequency response of the parallel filters,

$$H_{\text{sum}}(z) = \sum_p H_p(z) = H_{\text{E}}(z) + H_{\text{F}}(z) + \dots + H_{\text{D}^\sharp}(z). \quad (4.38)$$

The combined magnitude response, $|H_{\text{sum}}(z)|$, is shown in Fig. 4.11 for different Q factors and $\Delta\omega$.

The first null of each comb filter combine to create a large single null at 55.4 Hz. With a high Q factor, the overlapping harmonic peaks and nulls of the parallel filters create an irregular magnitude response at high frequencies. A smaller Q factor minimizes the irregularities but is less effective at separating the signal into harmonic components. A good compromise exists at $Q = 10$ or $\Delta\omega = 0.2\pi/M$, where the magnitude response is nearly linear in the range $f_{1,\text{E}} = 82.4$ Hz to $f_{1,\text{D}^\sharp} = 155.6$ Hz, and the irregularities are within 20 dB for frequencies up to 1 kHz, then within 35 dB for frequencies up to $f_s/2$.

The parallel filters can be designed with either a constant Q factor or a constant $\Delta\omega$. When the comb filters have constant Q the peak width increases with f_1 , which makes the

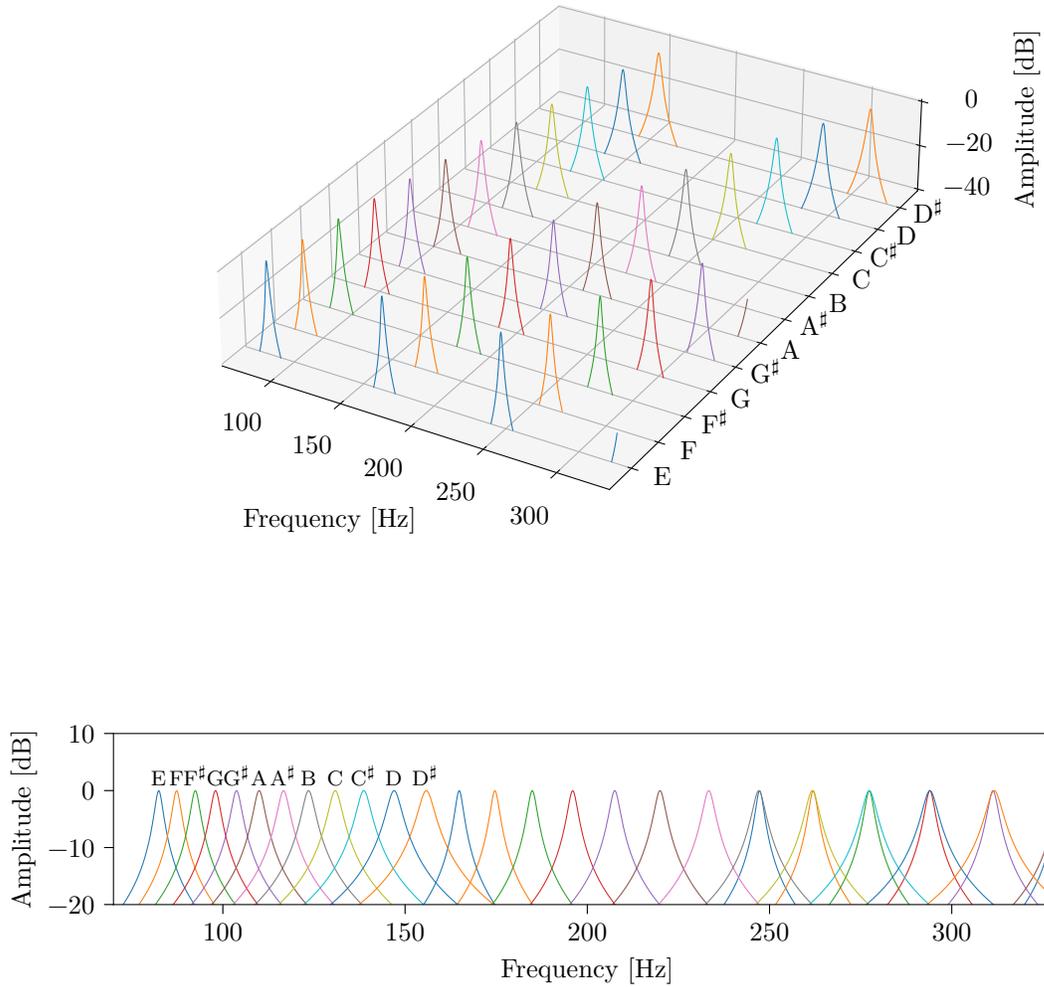


Figure 4.10: *Magnitude response of the twelve parallel comb filters where $\omega = 0.05\pi/M$. Annotated are the $f_{1,p}$ fundamental frequencies. Overlapping peaks occur at the coincident partials, e.g. $f_{2,B} = 246.9$ Hz and $f_{3,E} = 247.2$ Hz.*

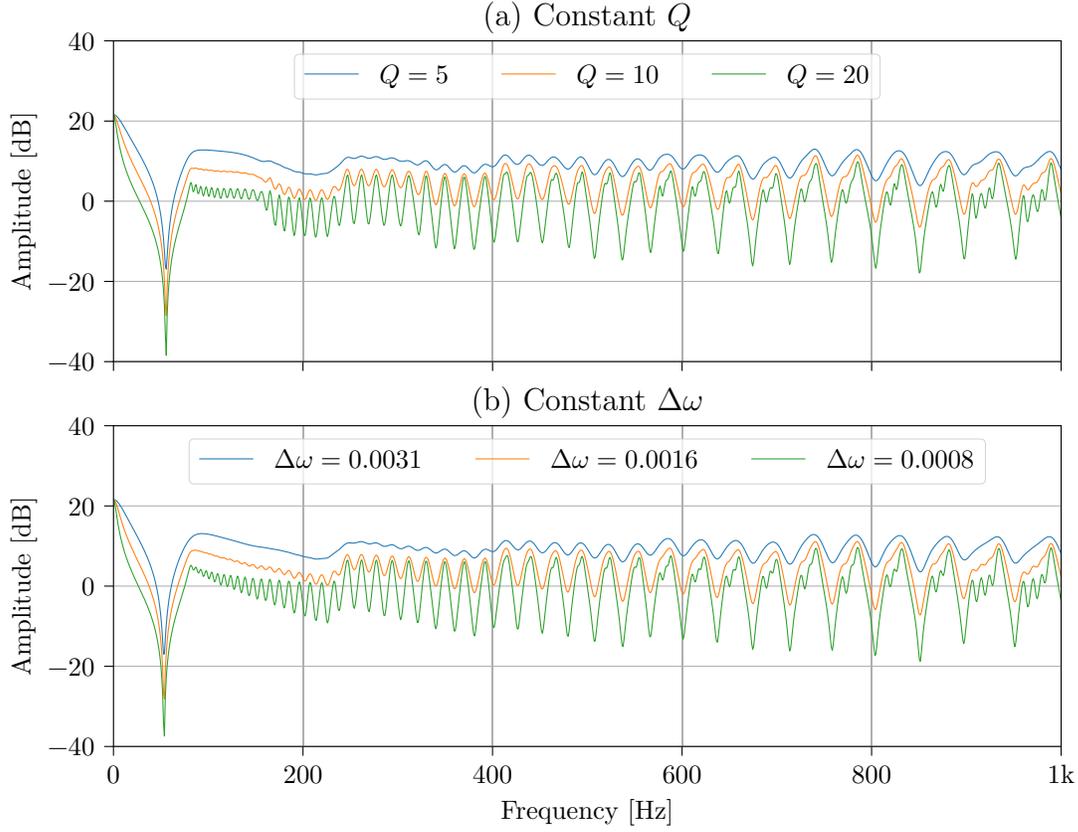


Figure 4.11: *Summed magnitude response of the parallel comb filters for different Q factors and $\Delta\omega$. Overlapping harmonic peaks and nulls of the parallel filters result in an irregular magnitude response at high frequencies.*

summed magnitude response of the parallel comb filter bank flatter. On the other hand, as was in seen Fig. 4.8, the impulse response decay times t_{60} vary widely over the frequencies that make up the filter bank. For this reason, a constant $\Delta\omega$, which provides nearly constant t_{60} across the filter bank frequencies, may be preferred. In this case, the average Q factor of the twelve filters is given by the geometric mean,

$$\left(\prod_p Q_p \right)^{1/12} = \frac{2\pi}{\Delta\omega f_s} \left(\prod_p f_{1,p} \right)^{1/12}. \quad (4.39)$$

4.2.3 Microtonal Pitch Variations

Since the peaks of the filter bank are tuned to a chromatic scale, pitches that lie between semitones are attenuated. The magnitude response of Fig. 4.11 provides some insight into

the behaviour of the filter during string bending, slides, or the use of a whammy bar. These playing techniques are all forms of glissandi — a continuous frequency increase or decrease from one note to another. As the pitch slides across harmonic peaks, it experiences amplitude modulations with an envelope resembling that of the magnitude response. This is yet another tradeoff in the choice of peak width and a further motivation for making the parameter user-adjustable.

4.3 String Detection

Some characteristics cannot be recovered from a mono signal. In standard tuning, notes with identical pitch and octave can be played on different strings. For example, the open note G_3 is repeated on the adjacent strings on the fifth and tenth frets (Table 4.2). Without the added complexity of fret fingering heuristics, even a sophisticated polyphonic pitch detection algorithm could not trace the individual notes that make up a mono signal back to its originating string.

Table 4.2: *Guitar fretboard in standard tuning. The highlighted G_3 notes are identical in pitch and octave and can be played on different strings.*

Open	1	2	3	4	5	6	7	8	9	10	11	12
E_4	F_4	F_4^\sharp	G_4	G_4^\sharp	A_4	A_4^\sharp	B_4	C_5	C_5^\sharp	D_5	D_5^\sharp	E_5
B_3	C_4	C_4^\sharp	D_4	D_4^\sharp	E_4	F_4	F_4^\sharp	G_4	G_4^\sharp	A_4	A_4^\sharp	B_4
G_3	G_3^\sharp	A_3	A_3^\sharp	B_3	C_4	C_4^\sharp	D_4	D_4^\sharp	E_4	F_4	F_4^\sharp	G_4
D_3	D_3^\sharp	E_3	F_3	F_3^\sharp	G_3	G_3^\sharp	A_3	A_3^\sharp	B_3	C_4	C_4^\sharp	D_4
A_2	A_2^\sharp	B_2	C_3	C_3^\sharp	D_3	D_3^\sharp	E_3	F_3	F_3^\sharp	G_3	G_3^\sharp	A_3
E_2	F_2	F_2^\sharp	G_2	G_2^\sharp	A_2	A_2^\sharp	B_2	C_3	C_3^\sharp	D_3	D_3^\sharp	E_3

Chapter 5

Distortion Emulation

The simulated hexaphonic distortion effect requires an efficient digital emulation of a non-linear distortion circuit which will run in twelve parallel instances. The distortion used in the evaluation of the proposed hexaphonic distortion simulation is a static approximation of the Distorter R VIII fuzz distortion circuit. Baptista designed the Distorter R VIII specifically for hexaphonic distortion in the Regulus guitars. As luck would have it, Baptista published detailed building instructions for the circuit in the Brazilian electronics magazine *Nova Eletrônica*.

This chapter describes the process of building the analog distortion circuit which serves as the reference hardware unit. It then describes a static non-linear function which emulates the diode clipper portion of the circuit. A comparison between the model and the SPICE simulation of the circuit will reveal opportunities for improvements to the model. Lastly, details of the oversampling implementation for alias suppression are provided.

5.1 Reference Hardware

In analog emulation, the “golden unit” is the original reference hardware. Given the deteriorating nature of electronics, obtaining a working and good-sounding historical unit of the 1970s is difficult at best. Instead, a replica faithfully reproducing the design from part two of the *Nova Eletrônica* article “Distortedor R VIII” [29] was built. Great care was taken to match the design to exacting detail, using electronic components of the era such as Mullard “tropical fish” capacitors and TO-18 metal can package transistors. The resistors, although new, are metal film resistors with 1% tolerance for accurate operation. The guitars built by Baptista contained six of these distortion circuits. Small-scale production of a printed circuit board is labor intensive and vintage components are costly. This single distortion circuit will serve as the reference hardware that will be modeled digitally in multiple instances.

5.1.1 Circuit Schematic and Operation

Fuzz distortion is achieved by brutally amplifying the electric guitar signal by injecting the signal into a bipolar junction transistor (BJT) driven above its linear operation range. This also clips the signal, producing a flattening of the top of the waveform. Following this, two diodes arranged in parallel but in opposite direction clip the top and bottom of the signal into a more symmetrical waveform. With reference to the schematic in Fig. 5.1, the basic operation of the device is described as such (translated from Brazilian Portuguese):

When connected, the signal goes to switch S1, which puts C3 in series with the signal or shorted, this signal going, filtered or not, by C3 to the base of Q1, via C1. Q1 amplifies the signal and delivers it to Q2, already partially distorted. This signal is further amplified by Q2, which distorts it even more, being then delivered to D1 and D2 that complete the service, drastically clipping the signal already quite distorted by Q1 and Q2. After D1 and D2, the signal is as it will be delivered at the output of the distorter, leaving R7, the potentiometer, the volume dosage or the amplitude of the signal that will reach the output jack.

Capacitor C3 acts as a low-cut filter, reducing intermodulation:

In our distorter, R VIII, this capacitor can be switched on and off by you, using a switch, giving you the deepest and most wrapped sound (but never too wrapped!) Or the highest and purest sound.

5.1.2 Printed Circuit Board

The printed circuit board (PCB) layout was redrawn from an illustration in the article. Circuit board dimensions were not provided and had to be inferred from the component lead spacing. Four copies of the layout fit on a 3" × 5" FR-4 copper-clad board. A 4-up transparency print was used to expose the light-sensitive surface of the board, which once developed, created a pattern in the photo-resist [30]. The etched PCB is shown in Fig. 5.2. After etching and drilling, the four boards were cut to a final size of 1.50" × 1.86". Two of the boards were populated with components, with two left over for future assemblies.

5.1.3 Assembly

The article only includes three pictures of the distortion unit. The dimensions of the utility enclosure had to be inferred from these pictures. A prototype panel was assembled using

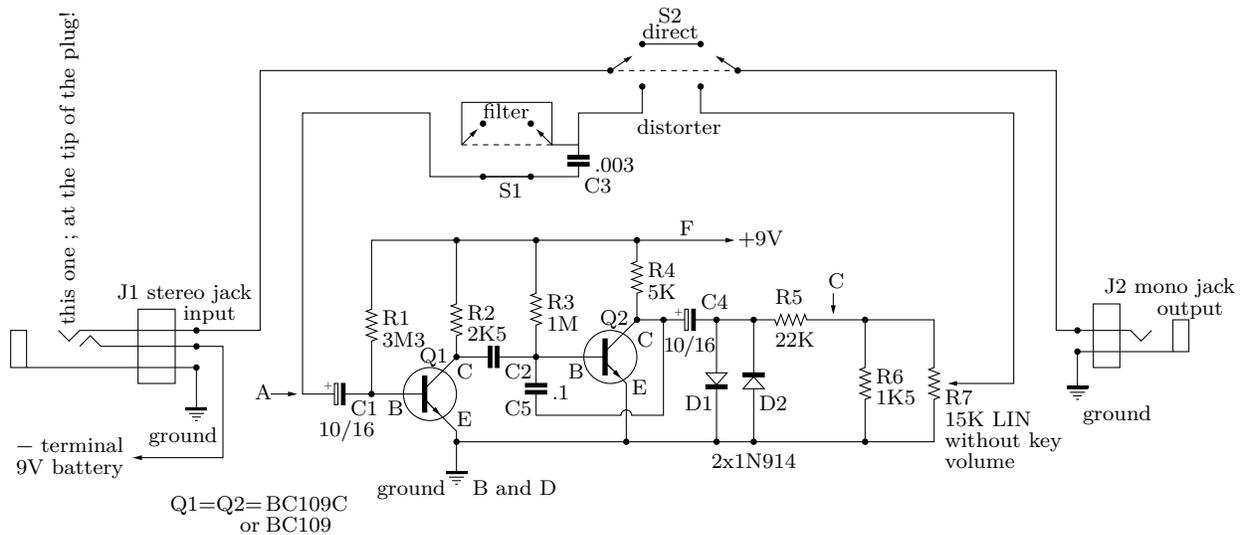


Figure 5.1: *Distorter R VIII circuit schematic.*

corrugated plastic to verify the spacing requirements of the panel mounted switches, jacks and potentiometers. Once hardware locations were determined, a computer-aided design (CAD) model of the assembly was used to determine the various lengths of wiring. This ensured that the total length of wiring did not exceed the amount specified in the original list of components as any excess wiring will contribute to noise in the circuit. To further minimize noise, the wiring was completed with shielded instrument cable as suggested in the article. Even with this precaution, the Distorter R VIII is a very sensitive device and will easily pick up radio signals if a cable is left disconnected. The CAD model drawing in Appendix A also served as the drill hole template. The finished unit is shown in Fig. 5.3.

5.1.4 Testing

The basic operation of the unit was verified with a function generator and oscilloscope. Consistent with the circuit's description, as shown in Fig. 5.4, with 64 mV at the input, the signal is clipped at the top of the wave.

The effect of the input filter capacitor is demonstrated for a pluck of the G₃ (196 Hz) string as captured by the middle mono pickup. Figure 5.5 shows the waveform and magnitude spectrum of the distorted signal. With the switch in the off position, capacitor C3 is shorted and the input signal is fed directly to the distortion circuit. With the filter switch in the on position, the input signal goes through capacitor C3. The series capacitor blocks DC and



Figure 5.2: *The etched board being drilled prior to being cut into four Distorter R VIII circuits.*

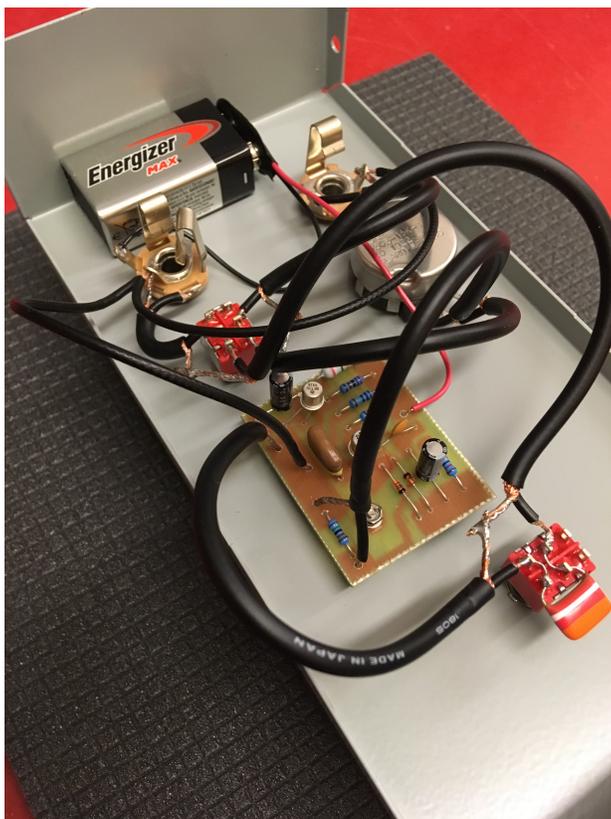


Figure 5.3: *Hardware unit interior.*

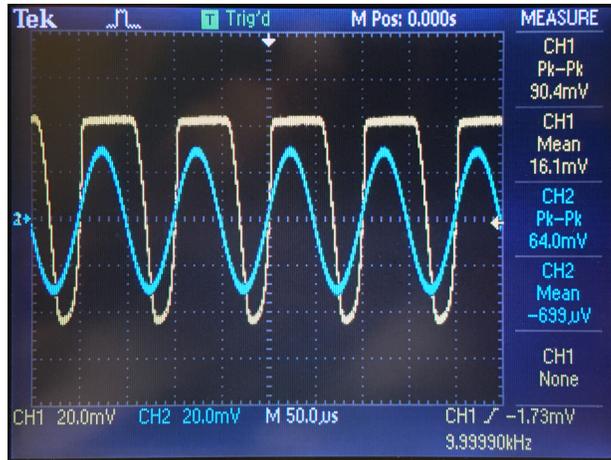


Figure 5.4: Oscilloscope capture of the output signal (CH1) of the distortion unit with a 10 kHz sinusoidal input signal (CH2).

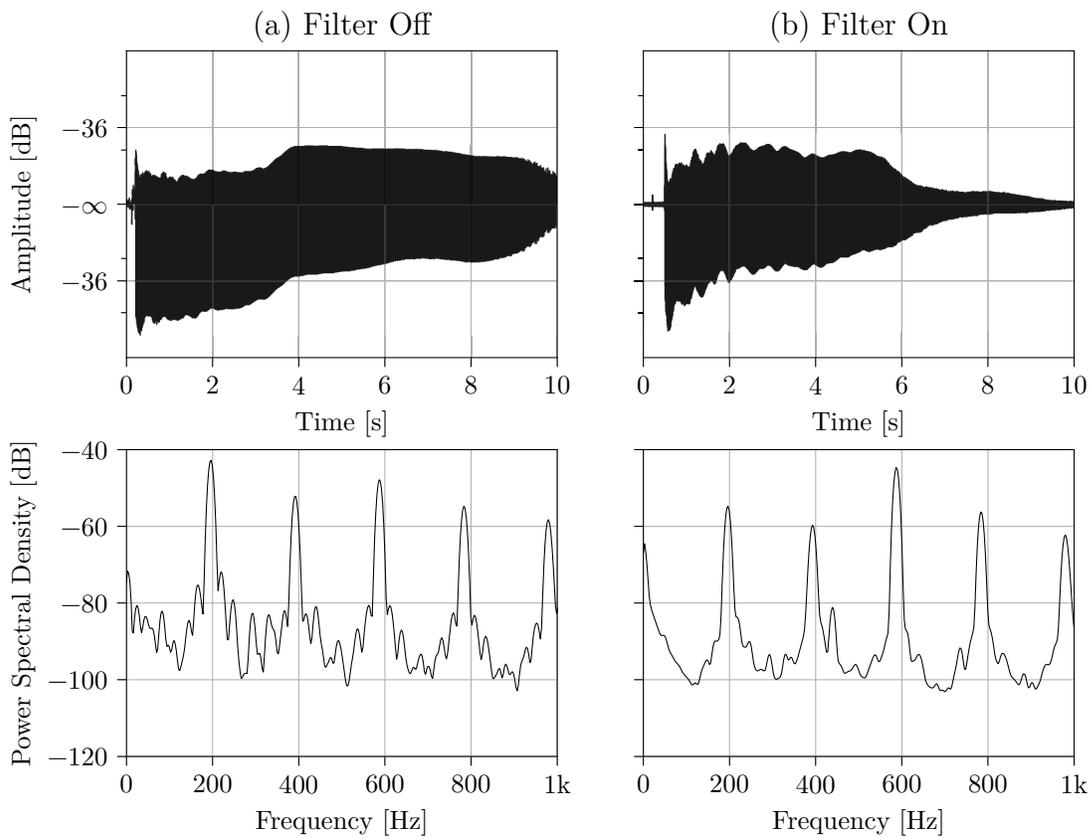


Figure 5.5: Difference in distortion waveform and magnitude spectrum with the input filter capacitor shorted (off) or in series (on).

acts as a low-cut filter with the measured cut-off frequency $f_c = 4.4$ kHz and a slope of 20 dB per decade. As evidenced by the magnitude spectrum plots, the purpose of the input filter is to reduce intermodulation distortion. It does this by attenuating low-frequency harmonics, thereby attenuating the sum and difference products generated by the lower harmonics. While the DC component of the input signal is attenuated by the filter, the subsequent distortion reintroduces energy at the zeroth harmonic (0 Hz). This explains the higher DC component magnitude seen in the magnitude spectrum plot.

5.2 Static Approximation of the Diode Clipper

Real-time processing of the simulated distortion requires that the distortion process be sufficiently computationally efficient to process twelve channels in parallel. The Distorter R VIII circuit can be modeled as a preamplifier followed by a static waveshaping function which approximates the diode clipper. The static waveshaping technique was chosen for its computational simplicity and predictable behaviour. The clipping distortion used in the evaluation of the simulated hexaphonic distortion consists of a preamplifier gain followed by an exponential function [2] and is given by

$$f(x) = \text{sgn}(x)(1 - e^{-|gx|}), \quad (5.1)$$

where g is the distortion gain. The sign function acts as a comparator, driving the signal to positive and negative extremes. This generates high-order harmonics at high amplitudes.

$$\text{sgn}(x) = \begin{cases} 1, & x > 0, \\ 0, & x = 0, \\ -1, & x < 0. \end{cases} \quad (5.2)$$

The exponential function provides a slight improvement over the hard clipping sign function by rounding out the discontinuity, thereby reducing the amplitude of the high-order harmonics. However, as seen in the characteristic curve of the function in Fig. 5.6, as the gain increases, the non-linearity once again approaches a hard clip [11]. This characteristic curve is an approximation of the diode clipper non-linearity seen in the Distorter R VIII circuit that symmetrically clips the top and bottom of the input waveform.

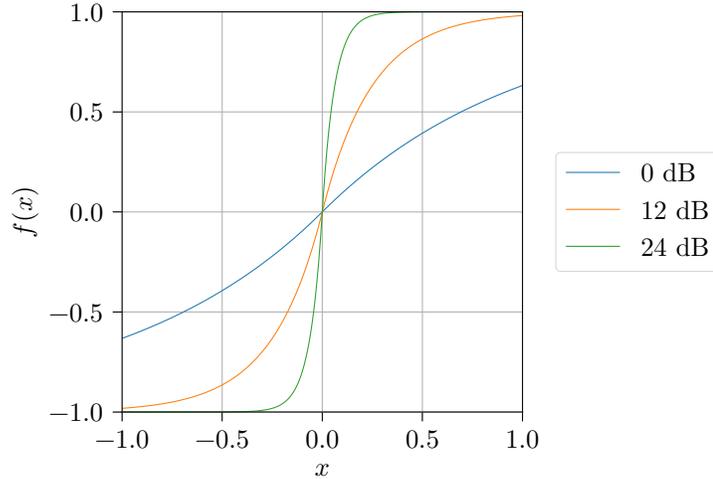


Figure 5.6: *Static characteristic curve of the symmetrical exponential distortion in equation (5.1) for different preamplifier gain levels g .*

5.3 Comparison with SPICE Simulation

The Distorter R VIII circuit was also simulated using SPICE (Simulation Program with Integrated Circuit Emphasis) to obtain an accurate model of the distortion in which memory elements are considered. Figure 5.7 shows the same 1760 Hz sine tone distorted by the LTspice model of the Distorter R VIII and the exponential function. Immediately apparent is the asymmetry of the waveform in the LTspice transient analysis. The signal is more clipped at the top than at the bottom. The importance of this characteristic is revealed in the magnitude spectrum, where the asymmetrical clipping produces both even-order and odd-order harmonics. This is in contrast to the odd symmetry of $f(x)$ which produces a magnitude spectrum that has only odd-order harmonics.

A more subtle distinction is in the envelope of this spectrum. In the exponential distortion, the spectral peaks are inversely proportional to the frequency,

$$\text{env}(|H(f)|) \propto \frac{1}{f}. \quad (5.3)$$

On the other hand, the LTspice model distortion has more variability in the peaks, producing a ragged spectral envelope, contributing to its characteristic tonalities. The exponential function distortion has predictable harmonic products and is thus favourable to the analysis and evaluation of the simulated hexaphonic distortion. However, the exponential function

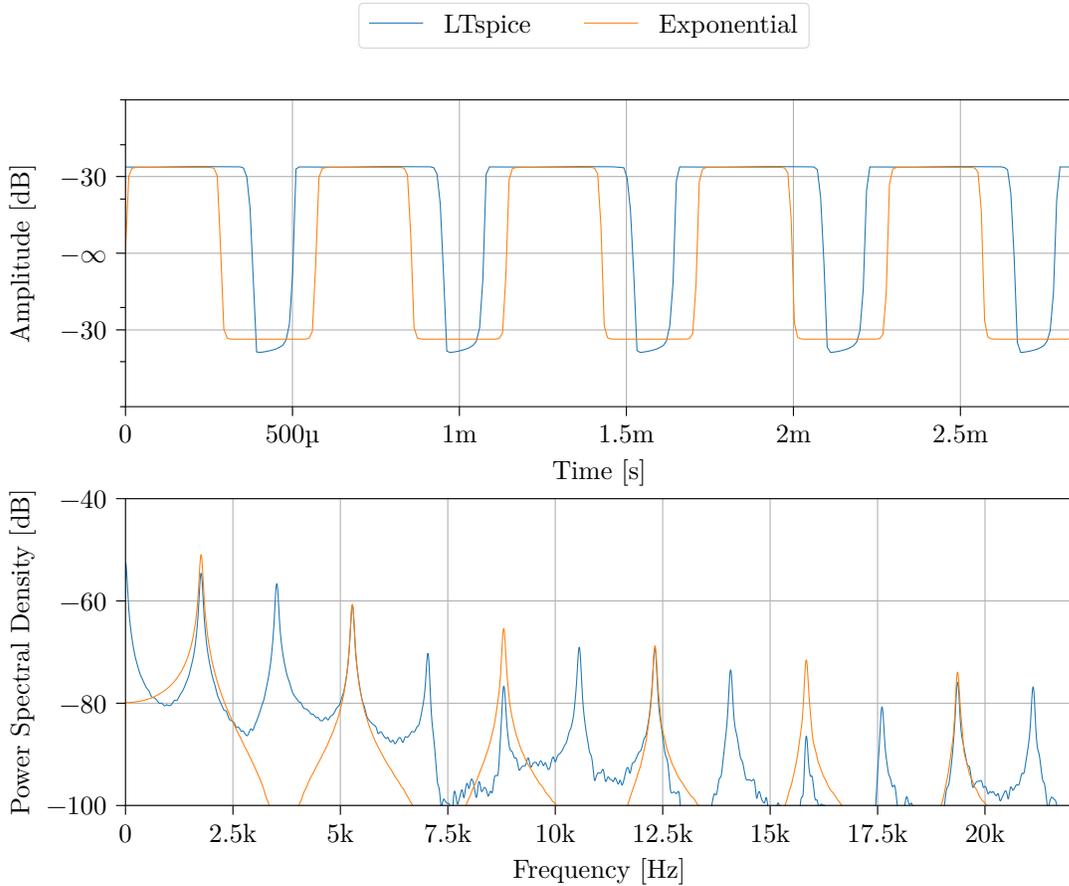


Figure 5.7: *Distorted 1760 Hz sine tone waveform and magnitude spectrum of the LTSpice model and the exponential distortion.*

distortion lacks the analog-sounding qualities desired in distortion effects.

The differences between the distortion implementations are even more pronounced when complex signals are considered. The combination of harmonic and inharmonic intermodulation products create a dense spectrum with greatly varying characteristics depending on the input signal.

Accurate emulation of the Distorter R VIII distortion circuit can not be accomplished with static waveshaping techniques. The capacitors in the circuit are memory elements that require solving non-linear ordinary differential equations (ODE). In addition, the bipolar junction transistors (BJT) in this circuit are far from ideal amplifiers. Circuit simulation programs such as LTSpice perform transient analysis using non-linear mathematical transistor models such as the Ebers-Moll model [31] and iterative methods to solve node voltage equations over a specified set of time steps.

LTSpice can output .wav audio files at any desired sample rate and can use .wav files

as input voltage sources. While the LTspice model presented here produces very accurate audio output, the non-linear transient analysis can not be done in real-time. However, circuit simulation techniques have been adapted and developed for the precise purpose of digitally emulating analog audio circuits [10, 11]. Future work will concentrate on implementing a proper real-time emulation of the Distorter R VIII circuit.

5.4 Oversampling

A well documented artifact of digital distortion is aliasing, whereby the harmonics produced by the non-linear effect extend beyond the Nyquist limit and wrap around to harmonically unrelated frequencies [2]. To accurately evaluate the spectral characteristics of the effect, the distortion must not introduce any aliased frequencies that would confound the analysis — with particular attention given to the intermodulation products. Alias-free distortion is achieved by the use of sixteen-times oversampling using the popular `r8brain-free-src` library to achieve alias-free distortion [32].

5.4.1 Argument for Sixteen-times Oversampling

The oversampling factor was chosen by observing the maximum alias magnitude of a distorted 1760 Hz sine tone. The magnitude spectrum of a distorted signal rolls off at higher frequencies and higher fundamental frequencies will produce larger aliasing products [11]. Given that the highest fundamental frequency of a 24-fret guitar is $E_6 = 1318.5$ Hz, the next whole number tone $A_6 = 1760$ Hz provides a convenient worst-case assessment.

The distortion products that exceed $f'_s/2$ extend into the bandlimiting filter and will be folded back onto the base band:

$$\frac{f'_s}{2} = Nf_s - \frac{f_s}{2}, \quad (5.4)$$

where N is the oversampling factor and $f_s = 44.1$ kHz. Since the exponential distortion produces only odd-order harmonics the first aliased distortion product exceeding $f'_s/2$ has frequency given by

$$f_k = f_1(2k + 1), \quad k = \left\lceil \frac{1}{2} \left(\frac{f'_s/2}{f_1} - 1 \right) \right\rceil. \quad (5.5)$$

When folded onto the base band it becomes

$$f'_k = Nf_s - f_k \quad (5.6)$$

and is the spectral alias with the largest magnitude. Table 5.1 shows the measured magnitude

of the f'_k aliased frequency component for increasing oversampling rates. Sixteen-times oversampling was judged sufficient since alias components below -75 dB will likely be masked by the distorted guitar signal.

Table 5.1: *Aliasing distortions for different oversampling rates for a normalized 1760 Hz sine tone amplified by 40 dB and clipped.*

N	Alias component	f_k [Hz]	Folded f'_k [Hz]	Magnitude [dB]
1		22 880	21 220	-22.44
2		68 640	19 560	-33.05
4		156 640	19 760	-44.05
8		332 640	20 160	-58.72
16		684 640	20 960	-75.95
32		1 392 160	19 040	-94.03

5.4.2 Implementation

Two instances of the `r8b::CDSPResampler24` class are initialized with the parameters given in Table 5.2.

Table 5.2: *Resampler parameters.*

	Source sample rate	Destination sample rate	Max input buffer length
Resampler 1	1	16	1024
Resampler 2	16	1	16×1024

The sampling rates are specified as a ratio rather than fixed sampling rates to ensure power-of-two optimizations and to avoid fractional interpolation. Successive input buffers are passed to the resampler process function, which returns a pointer to the resampled data. The upsampling is performed using four stages of $2\times$ oversampling. The first stage uses a steep reconstruction filter, implemented using the overlap-save method of block convolution. This method requires fewer complex multiplications than direct convolution of the lowpass FIR filter [33]. The three subsequent stages use more efficient half-band small sparse symmetric FIR filters. At this point, the upsampled audio is processed by the simulated hexaphonic distortion effect, including the parallel comb filter structure, and saved to a temporary buffer. The temporary buffer is passed to the second resampler in which the upsampling process is reversed: three half-band $2\times$ downsamplers followed by a steep reconstruction filter with a 2% transition band.

Chapter 6

Results

This chapter provides an evaluation of the hexaphonic distortion simulation in the context of intermodulation distortion reduction and computational performance. Harmonic entropy, which was introduced briefly in the previous chapter, gets a detailed description. Following this is a description of the various software tools developed for this thesis.

6.1 Simulated Distortion Evaluation

To evaluate the simulated hexaphonic distortion effect, the intermodulation products of a distorted hexaphonic recording are compared to the intermodulation products of a distorted mono recording. The type of pickup and pickup position used on an electric guitar will result in very different tones. This makes comparisons between the real hexaphonic distortion and the simulated effect when applied to a mono pickup difficult. For this reason, the subsequent analyses and comparisons make use of a mono signal created by averaging the individual signals of the hexaphonic pickup.

The algorithm process begins with upsampling the mono input signal. The upsampled signal is then separated into twelve harmonic signals by the parallel comb filter bank. Each individual signal is distorted using the exponential distortion given in equation (5.1). Once distorted, the signals are mixed down with the final output downsampled to the original sampling frequency.

6.1.1 Harmonic Entropy

Erlich's model of harmonic entropy provides a measure of interval tonality by comparing the interval to simple integer ratios. It also provides a general prediction of the amount of intermodulation distortion that a particular note pair is expected to produce. The Farey

sequence \mathcal{F}_n produces reduced fractions with denominators less than or equal to n . An important property of the Farey sequence is that simple fractions like $1/2$ and $1/3$ are spaced further apart than those of higher integers like $1/5$ and $1/6$. The closer a note interval i is to an element of the series, f_j , the more consonant it will sound. The region defined by the mediant (not to be confused with the median) below and the mediant above f_j is wide for simple ratios and small for complex ratios. Integrating a normal distribution over this region gives the probability $p_j(i)$ that an interval, i will be perceived as the ratio f_j . The harmonic entropy of the random variable i is

$$\text{HE}(i) = - \sum_j p_j(i) \log(p_j(i)). \quad (6.1)$$

When an interval is close to a simple fraction, the probability density function is concentrated on that fraction, resulting in a small harmonic entropy value. When it is far from a simple fraction, the probability density function is spread over multiple fractions, thus increasing its harmonic entropy. As a result, the harmonic entropy gives a measure of tonality over a continuous range of intervals as shown in Fig. 6.1.

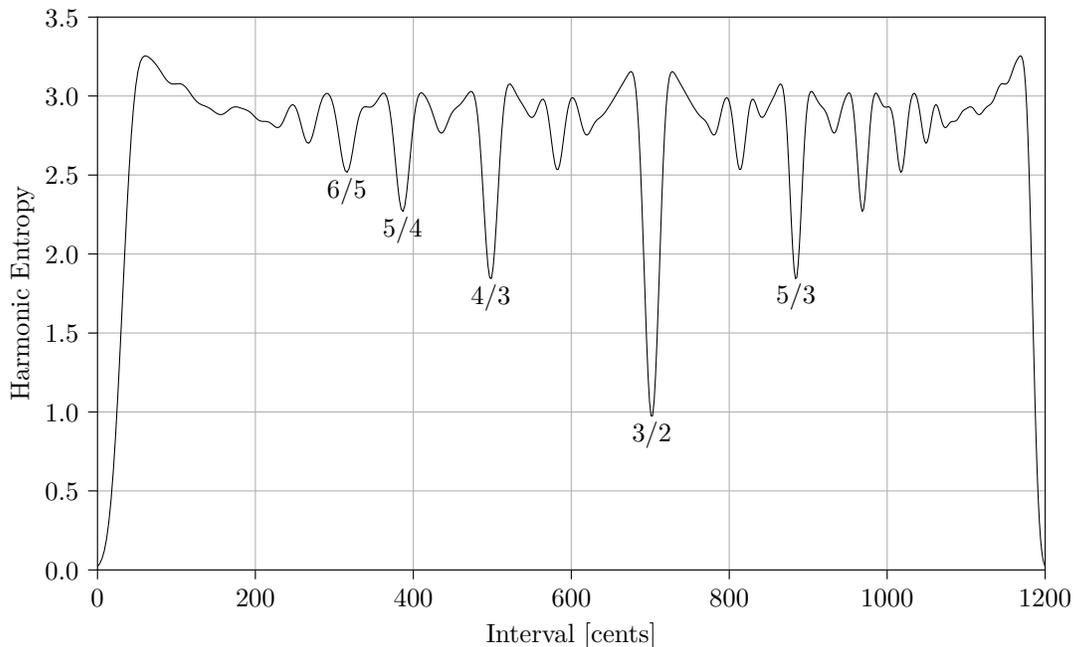


Figure 6.1: *Harmonic entropy for intervals over the range of an octave. \mathcal{F}_{50} , $\sigma = 0.007$.*

To best demonstrate the ability of the effect to reduce intermodulation, recordings of

the individual plucks of the open strings and the first fret are combined to create pairs of varying intervals. These pairwise combinations of notes include every simple interval in the chromatic scale. Table 6.1 lists the note pairs, the intervals, i , and the corresponding harmonic entropy, $\text{HE}(i)$.

Table 6.1: *Note pair intervals used in distortion comparisons.*

Note pair	i [cents]	$\text{HE}(i)$	Note pair	i [cents]	$\text{HE}(i)$
E ₄ -B ₃	500	1.873	E ₄ -C ₄	400	2.823
E ₄ -G ₃	900	2.979	E ₄ -G ₃ [#]	800	2.965
E ₄ -D ₃	1400 (200)	2.480	E ₄ -D ₃ [#]	1300 (100)	2.892
E ₄ -A ₂	1900 (700)	0.073	E ₄ -A ₂ [#]	1800 (600)	2.783
E ₄ -E ₂	2400 (1200)	0.021	E ₄ -F ₂	2300 (1100)	2.945
B ₃ -G ₃	400	2.823	B ₃ -G ₃ [#]	300	2.911
B ₃ -D ₃	900	2.979	B ₃ -D ₃ [#]	800	2.965
B ₃ -A ₂	1400 (200)	2.480	B ₃ -A ₂ [#]	1300 (100)	2.892
B ₃ -E ₂	1900 (700)	0.073	B ₃ -F ₂	1800 (600)	2.783
G ₃ -D ₃	500	1.873	G ₃ -D ₃ [#]	400	2.823
G ₃ -A ₂	1000	2.931	G ₃ -A ₂ [#]	900	2.979
G ₃ -E ₂	1500 (300)	2.900	G ₃ -F ₂	1400 (200)	2.480
D ₃ -A ₂	500	1.873	D ₃ -A ₂ [#]	400	2.823
D ₃ -E ₂	1000	2.931	D ₃ -F ₂	900	2.979
A ₂ -E ₂	500	1.873	A ₂ -F ₂	400	2.823

6.2 Comparison of Distortion Structures

In Fig. 6.2, the mono signal is distorted by the clipping function, $f(x)$. In the simulated distortion structure of Fig. 6.3, the same mono signal is separated into harmonic signals by the parallel comb filter bank prior to distortion. The comb filters are set to constant peak width of $\Delta\omega = 0.0015/16$ corresponding to an average Q of 10.75. In hexaphonic distortion, the clipping function is applied to each string separately, as shown in Fig. 6.4. In each case, the distortion gain is 100.0 or 40 dB and the post-distortion output signal is normalized to the root mean square (RMS) value of -12 dB. $N \times$ oversampling is employed throughout to reduce aliasing in the digital implementation of distortion.

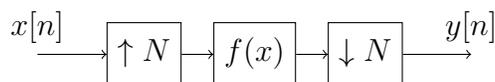


Figure 6.2: *The mono distortion structure.*

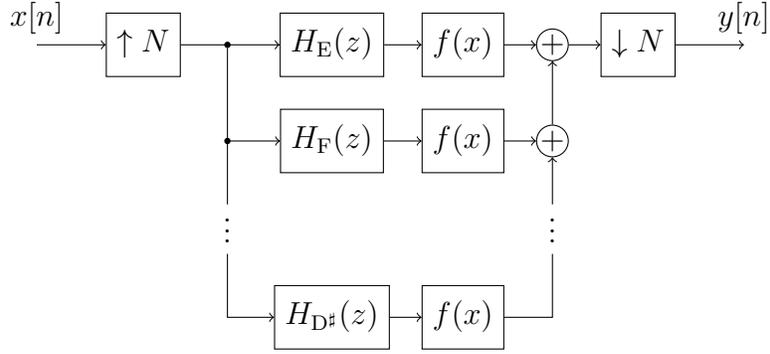


Figure 6.3: *The simulated distortion structure.*

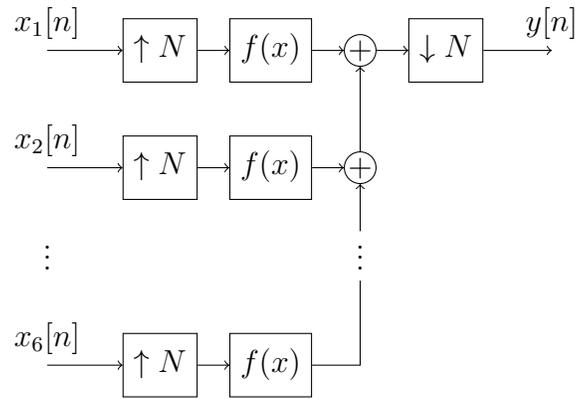


Figure 6.4: *The hexaphonic distortion structure.*

6.3 Magnitude Response Comparison

Two note pairs are examined by comparing the magnitude response of the different distortion outputs. Consider the minor seventh interval consisting of the tones G_3 (196 Hz) and A_2 (110 Hz) played on the open strings. Shown in Fig. 6.5a, when mono distortion is applied to the averaged hexaphonic signal, the three largest intermodulation products have the following sum and difference frequencies:

$$\begin{aligned}
 f_{1,G} + f_{1,A} &= 196 + 110 = 306 \text{ Hz}, \\
 f_{1,G} + f_{2,A} &= 196 + 2 \cdot 110 = 416 \text{ Hz}, \\
 f_{1,G} + f_{4,A} &= 196 + 4 \cdot 110 = 636 \text{ Hz}.
 \end{aligned}$$

The difference between the peak amplitude at 110 Hz and each of these intermodulation components is summarized in Table 6.2. On average, the comb filtering method in the simulated distortion attenuates the intermodulation products seen in the mono distortion by

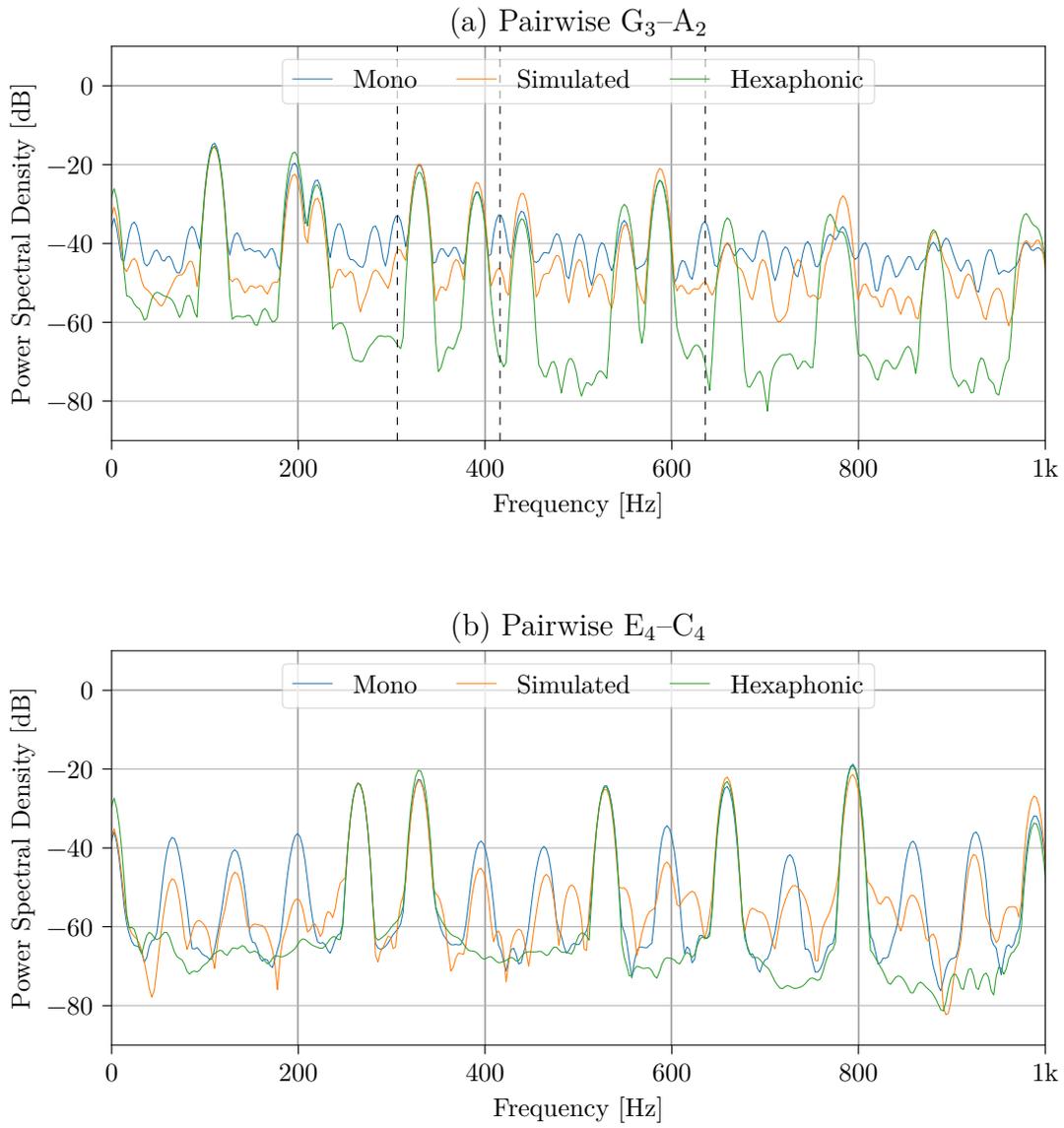


Figure 6.5: Magnitude spectrum of each distortion output for the note pairs (a) G_3-A_2 and (b) E_4-C_4 . The dashed lines show the three largest intermodulation components. Symmetrical soft clipping with gain 100.0. Signals are normalized to -12 dB RMS post-distortion.

11.6 dB, while hexaphonic distortion achieves 34.4 dB. This suggests the algorithm is about a third as effective at reducing intermodulation distortion in this particular note pair when compared to hexaphonic distortion.

Table 6.2: *Comparison of intermodulation product magnitudes for note pair G₃-A₂*

Intermodulation component [Hz]	306	416	636
Mono distortion [dB]	-18.3	-18.3	-19.8
Simulated distortion [dB]	-26.0	-31.2	-34.1
Hexaphonic distortion [dB]	-50.7	-54.4	-54.6

The tones E₄ (329.6 Hz) and C₄ (261.6 Hz) belonging to the C chord, produce very distinct intermodulation products when distorted in mono, as seen in Fig. 6.5b. The intermodulation product are sum and difference products:

$$\begin{aligned}
 f_{1,E} - f_{1,C} &= 329.6 - 261.6 = 68 \text{ Hz}, \\
 f_{2,E} - f_{2,C} &= 2 \cdot 329.6 - 2 \cdot 261.6 = 136 \text{ Hz}, \\
 f_{3,E} - f_{3,C} &= 3 \cdot 329.6 - 3 \cdot 261.6 = 204 \text{ Hz} \dots
 \end{aligned}$$

In mono distortion, the energy of the intermodulation distortions is concentrated at the peaks. The use of comb filters in the simulated distortion attenuates the intermodulation components by distributing the energy over the broader spectrum. In doing so, the harmonic distortion is made more prominent.

6.4 Spectrogram Comparison

Figures 6.6 and 6.7 show spectrograms for each distorted pairwise note combinations for frequencies up to 1 kHz. In the hexaphonic distortion, the two note pairs recorded from the hexaphonic pickup are distorted separately and the resulting spectrogram shows almost no intermodulation distortion. This translates to clear distinct notes. The double octave interval E₄-E₂ in Fig. 6.6 has a small harmonic entropy value and shows no intermodulation products in any of the distortion outputs. Conversely, the interval G₃-A₂ has few coincident partials and thus, a large harmonic entropy value. The resulting spectrogram of the mono distortion output is very dense. In the simulated distortion, the intermodulation products are partly attenuated.

None of the intervals in Fig. 6.7 have small harmonic entropy value. As expected, each of the note pairs produce intermodulation distortion in the mono distortion output and again, distorting each band of the filter bank separately helps reduce the intermodulation products.

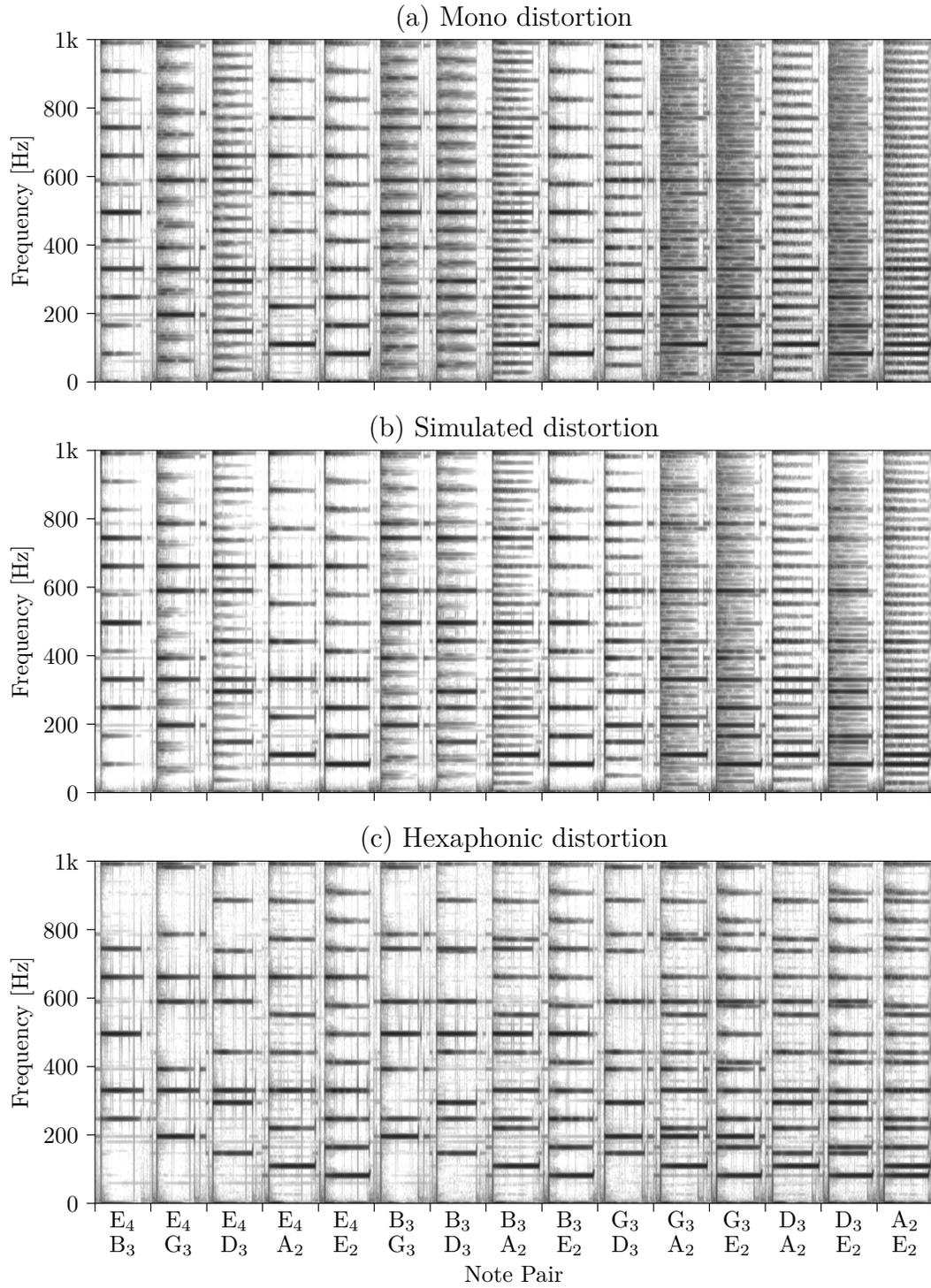


Figure 6.6: *Distortion spectrograms for pairwise combinations of open strings notes. Each signal is normalized to -12 dB RMS post-distortion.*

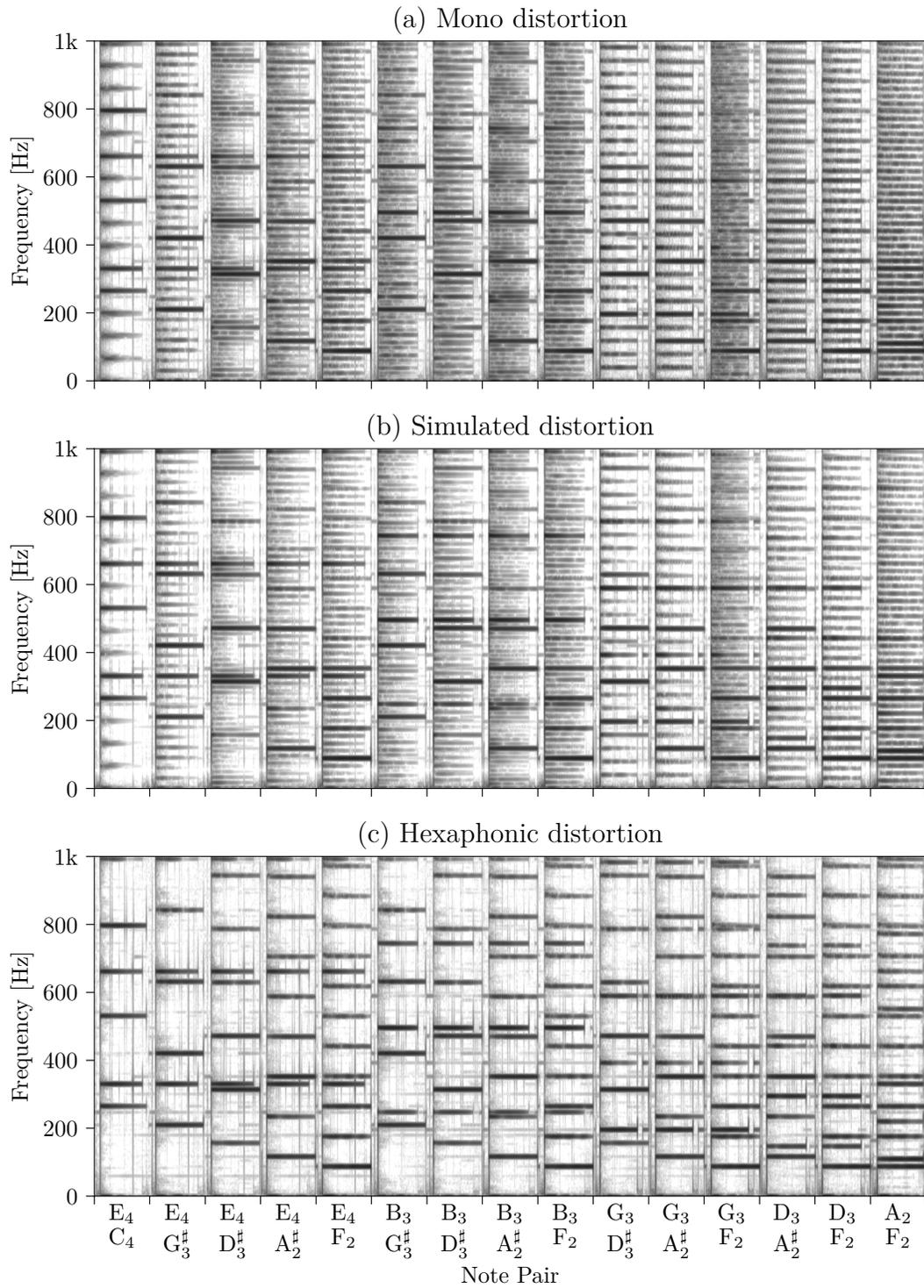


Figure 6.7: *Distortion spectrograms for pairwise combinations of open and first fret notes. Each signal is normalized to -12 dB RMS post-distortion.*

The analyzed sound files were made available online as part of the paper submission to the International Conference on Digital Audio Effects 2020 (DAFx-20) [34].

6.5 Algorithm Performance

The parallel comb filtering and distortion process illustrated in Fig. 6.3 begins by upsampling the mono signal by a factor of sixteen. The signal is then processed by twelve parallel comb filters and a non-linear distortion function. The resulting twelve channels of audio are averaged to obtain a mono sum. This structure employs only two resampling steps, with the tradeoff being that the comb filters also process oversampled audio. Consequently, the delay line buffers are proportionally increased by the oversampling factor. The memory requirements are therefore significantly increased. This is deemed a worthwhile compromise since memory is plentiful in modern computers.

The filter bank and distortion structures were implemented using the Synthesis ToolKit in C++ (STK) [35]. The parallel comb filters utilize twelve STK non-interpolating `Delay` objects in the canonical structure of Fig. 4.5. A loop performs sample-based processing of each resampler buffer. While it was initially assumed that the oversampling process would be the largest computational expense, runtime measurements reveal otherwise. A C++ class for the purpose of performance measurement was built using STK conventions. The class provides a convenient method for placing markers at different sections of the runtime loop with a string label. Once execution completes, a print function outputs the elapsed time of each labeled section of code. Since there is an inherent computational expense in the timing functions, the results are best understood by comparing the relative expense of each process. This is summarized in Table 6.3. The largest expense are the twelve parallel comb filters and the exponential function-based distortion. While the individual comb filtering operations are cheap, computing twelve parallel comb filters at sixteen times the sample rate quickly adds up. The upsampling and downsampling processes are comparatively cheap. The `r8brain-free-src` library implements optimizations for power-of-two resampling ratios in which case the largest limiting factor is the FFT function [32]. The results also show that the implementation can in fact run in real-time, utilizing $1.324 \text{ s}/8 \text{ s} = 0.17$ real-time blocks. Further optimizations of the parallel filtering blocks could be achieved using single-instruction multiple-data (SIMD) capabilities of modern processors.

Table 6.3: *Computational expense averaged over three runs. Clang optimization level -Os (fastest, smallest). Running on a 1.6 GHz Intel Core i5.*

8 s of 24-bit audio at 44.1 kHz, 16× oversampling		
Process	Time [s]	Weight [%]
File Read	0.030	2.28
Upsample	0.042	3.14
Parallel Filters	$16 \times 12 \times 0.00311 = 0.598$	45.14
Distortion	$16 \times 12 \times 0.00265 = 0.510$	38.54
Mix	$16 \times 12 \times 0.00024 = 0.046$	3.47
Downsample	0.045	3.38
File Write	0.054	4.05
Total	1.324	100.00

6.6 Tools

The Synthesis ToolKit has support for multi-channel processing and audio files that make it ideal when working with hexaphonic signals. With the exception of `combeq` the following tools were all implemented using the STK. They were developed as command-line utilities for ease of use and batch processing. All the figures used in the thesis were created in the open source Python library, Matplotlib. Development also made substantial use of the NumPy and SciPy packages.

combeq

A Python script combining a parameterized comb filter and plotting functions to aid in the design and analysis of the individual comb filters.

harmonicfilter

Implements a bank of parallel comb filters. The output of each filter is saved to individual channels of a `.wav` file.

amerge

The `amerge` tool was created to combine the multitrack hexaphonic recordings to six-channel `.wav` files. It can also perform a summed-to-mono conversion by multi-channel averaging. It detects the input format and uses the same output format.

expdist

Utility that performs multi-channel distortion with oversampling based on an exponential function. This was used to create the mono and hexaphonic distortion reference signals used to sonically evaluate the simulated effect.

hexsim

The main hexaphonic distortion simulation effect, it takes a mono input signal and performs upsampling followed by parallel comb filtering and distortion prior to downsampling back to the original sampling frequency.

pairwise

This utility takes two six-channel hexaphonic recordings of string plucks and creates pairwise combinations of notes output to two channels. It detects note onsets and lines them up to generate simultaneous plucks.

Chapter 7

Conclusion and Future Work

This chapter provides a preview of the remaining challenges involved in building a physical model of the Distorter R VIII. It also discusses vectorization as an approach to improving the computational performance of the hexaphonic distortion simulation algorithm before finally concluding the thesis.

7.1 Circuit Emulation Preview

In Chapter 5, the diode clipper non-linearity was implemented as a static characteristic curve. As a preview of the physical modelling implementation, the diode is described in terms of its physical parameters. A commonly used and suitably accurate model of the diode is the Shockley diode equation [11]:

$$I_d = I_s \left(e^{\frac{V_d}{nV_T}} - 1 \right) \quad (7.1)$$

It describes the characteristic relationship between the voltage across the diode V_d and the current through the diode I_d under forward and reverse bias. The physical parameters for the 1N914 silicon diode are:

- the reverse bias saturation $I_s = 2.52 \text{ nA}$
- the ideality factor $n = 1.752$
- the thermal voltage evaluated at 25°C $V_T = \frac{kT}{q} = 25.69 \text{ mV}$

In the last expression, k is the Boltzmann constant, T is the temperature in Kelvins, and q is the elementary charge.

The similar equations that make up the Ebers-Moll model describe the BJT as a voltage-controlled current source. Using these equations and modified nodal analysis, the circuit can be represented as a system of non-linear differential-algebraic equations consisting of

memoryless elements (resistors), energy storage elements (capacitors), non-linear elements (transistors and diodes), and independent voltage sources. This system of equations is solved using numerical methods, with the backward Euler and Trapezoidal rule methods being most common approaches. The resulting non-linear equation is evaluated using Newton iteration yielding the node voltages for each specified time step. Applying an audio signal to the input node will generate a distorted signal at the output node. Various optimizations are then applied to the model to the point where the signal can be processed in real-time.

7.2 Vectorized Operations

In Chapter 6, it was suggested that the hexaphonic distortion simulation algorithm could be further optimized through the use of SIMD instructions. These are available on most CPU architectures, including Intel x86 in the form of the trademark MMX, Streaming SIMD Extensions (SSE), and the Advanced Vector Extensions (AVX). The ARM instruction set features the Neon SIMD architecture extension.

Using the ARM assembly language as an example, adding four pairs of registers typically requires four instructions:

```
ADD R0, R0, R4
ADD R1, R1, R5
ADD R2, R2, R6
ADD R3, R3, R7
```

where the destination register is followed by the operands. Neon SIMD can combine the operations into a single instruction and performs the operation simultaneously:

```
ADD V10.4S, V8.4S, V9.4S
```

The four values are packed into two pairs of 4-element registers, V8 and V9, and the result is stored in the vector register V10. The operand suffix `.4S` indicates four 32-bit words. Since digital signal processing algorithms rely on numerous repeated operations, SIMD can provide significant performance improvements.

There are numerous approaches to vectorization. Optimizing compilers like LLVM feature automatic vectorization of loops — transforming sequential operations into vector operations. However, vectorization will fail if the loop control flow is not understood by analyzer. Optimization diagnostics reveal that the parallel processing of the twelve comb filters and distortion is not automatically vectorized.

Guaranteed use of SIMD vectorized operations can be achieved through specialized math libraries, explicit use of intrinsic functions for the specific architecture, or using SIMD instructions in hand-coded assembler.

7.3 Summary

Despite the many advantages hexaphonic distortion has over traditional mono distortion, hexaphonic hardware is too expensive for the average guitarist and thus hexaphonic distortion has never been widely adopted. The simulated hexaphonic distortion effect described in this thesis attempts to reproduce the characteristics of hexaphonic distortion for use with ordinary electric guitars with mono pickups. A parallel comb filter structure has been shown to separate a mono guitar signal into predominantly harmonic signals akin to that of a hexaphonic pickup signal. In addition, the overlapping spectral peaks of the comb filters simulate the sympathetic vibrations captured by hexaphonic pickups. Individually distorting the harmonically separated signals results in clear, sustained, and harmonically rich distortion with fewer intermodulation products. The algorithm therefore provides guitarists with greater flexibility in their choice of chords when using distortion. The proposed structure may be used in real-time with any sufficiently optimized distortion algorithm. Physical modeling approaches present the most accurate distortion circuit emulations but are computationally expensive. Given the increased cost requirement of twelve separate distortion processes, the need for an efficient circuit emulation of the Distorter R VIII presents an interesting challenge for the future.

Appendix A

Drawings

The following pages include the Nova Eletrônica No. 3023 printed circuit board layout reproduced from the “Distorcedor R VIII” article [1] and by a drawing of the interior drill hole locations for the utility enclosure. The enclosure dimensions of $8'' \times 4'' \times 2''$ and the hole locations were determined from the pictures shown in the article.

4

3

2

1

F

F

E

E

D

D

C

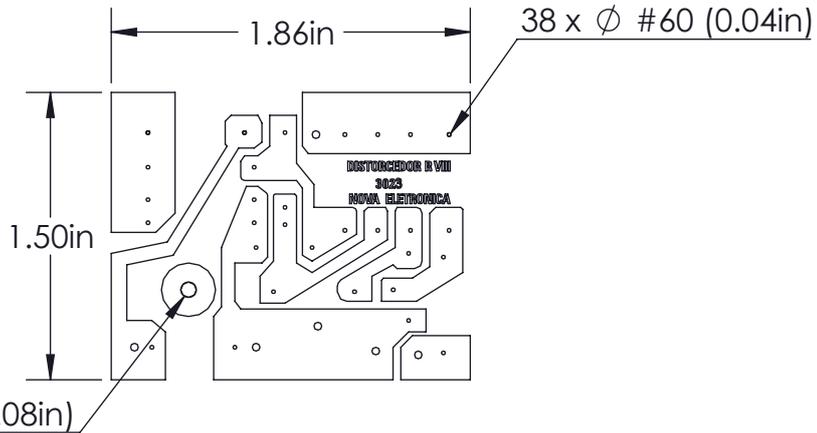
C

B

B

A

A



UNLESS OTHERWISE SPECIFIED:
 DIMENSIONS ARE IN MILLIMETERS
 SURFACE FINISH:
 TOLERANCES:
 LINEAR:
 ANGULAR:

FINISH:

DEBURR AND
 BREAK SHARP
 EDGES

DO NOT SCALE DRAWING

REVISION

	NAME	SIGNATURE	DATE
DRAWN	S.L.		2019-05-21
CHK'D			
APPV'D			
MFG			
Q.A			
MATERIAL:			
FR-4 Copper Clad Laminate			
WEIGHT:			

TITLE:

Distorcedor R VIII

DWG NO.

3023

A4

SCALE:1:1

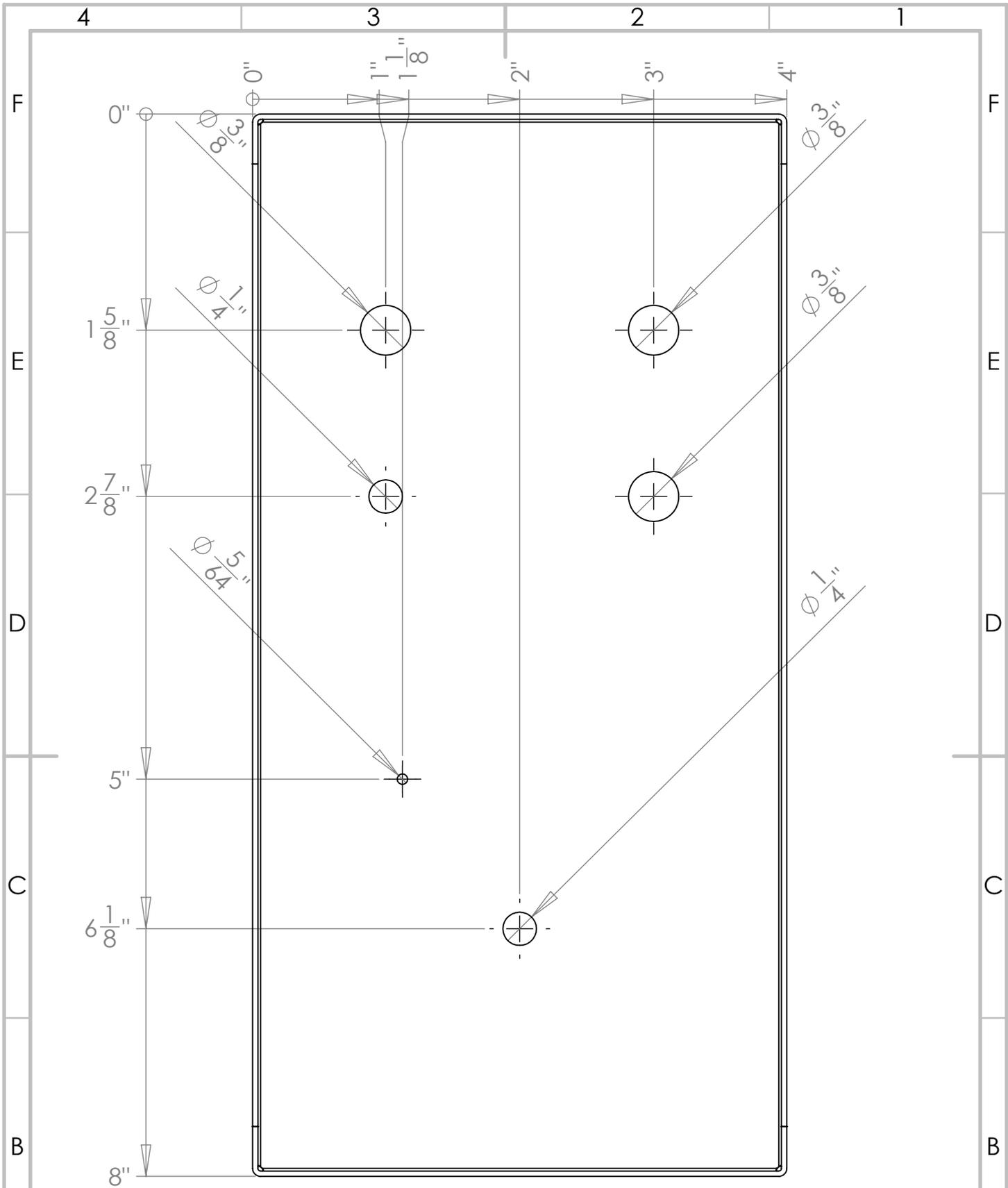
SHEET 1 OF 1

4

3

2

1



UNLESS OTHERWISE SPECIFIED:
 DIMENSIONS ARE IN MILLIMETERS
 SURFACE FINISH:
 TOLERANCES:
 LINEAR:
 ANGULAR:

FINISH:
ASA 61 Grey

DEBURR AND
 BREAK SHARP
 EDGES

DO NOT SCALE DRAWING

REVISION

NAME	SIGNATURE	DATE
DRAWN S.L.		2020-01-02
CHK'D		
APPV'D		
MFG		
Q.A		

TITLE:
Distorcedor RVIII

DWG NO.
1411SS - Cover

MATERIAL:
Aluminum

SCALE:1:1

SHEET 1 OF 1

Bibliography

- [1] C. C. D. Baptista, “Distorcedor R VIII – parte 1,” *Nova Eletrônica*, vol. 1, no. 4, pp. 422–427, May 1977.
- [2] U. Zölzer, Ed., *DAFX: Digital Audio Effects*, pp. 70–79, 101–106, 126, John Wiley & Sons, 2nd edition, 2011.
- [3] P. Fernández-Cid and J. Casajús Quirós, “Distortion of musical signals by means of multiband waveshaping,” *Journal of New Music Research*, vol. 30, no. 3, pp. 219–287, 2001.
- [4] J. S. Abel and K. J. Werner, “Distortion and pitch processing using a modal reverb architecture,” in *Proceedings of the 18th International Conference on Digital Audio Effects (DAFx-15)*, Trondheim, Norway, Nov. 30 – Dec. 3 2015.
- [5] J. E. R. de Paiva, “Os Mutantes: Hibridismo tecnológico na música popular brasileira dos anos 60/70,” *Revista Sonora*, vol. 4, no. 7, pp. 1–7, 2012.
- [6] W. Marx Jr., “Putting the hex on postmodern pickups,” *Premier Guitar*, Jan. 14 2010.
- [7] R. Dobson, *A Dictionary of Electronic and Computer Music Technology: Instruments, Terms, Techniques*, pp. 66, 89–91, 121, Oxford University Press, 1992.
- [8] R. Graham and J. Harding, “SEPTAR: Audio breakout circuit for multichannel guitar,” in *Proceedings of the International Conference on New Interfaces for Musical Expression (NIME)*, Baton Rouge, LA, USA, May 31 – June 3 2015, pp. 241–244.
- [9] P. Rubenstein, “Ubertar Hexaphonic Guitar Pickups,” Available at <http://www.ubertar.com/hexaphonic/>, accessed Dec. 30, 2019.
- [10] J. Pakarinen and D. T. Yeh, “A review of digital techniques for modeling vacuum-tube guitar amplifiers,” *Computer Music Journal*, vol. 33, no. 2, pp. 85–100, Summer 2009.

- [11] D. T.-M. Yeh, *Digital implementation of musical distortion circuits by analysis and simulation*, PhD dissertation, Stanford University, 2009.
- [12] C. Anderton, “Build the super-versatile Quadrafuzz: Four fuzzes in one with active EQ,” *Guitar Player Magazine*, pp. 37–46, June 1984.
- [13] J. Timoney, V. Lazzarini, A. Gibney, and J. Pekonen, “Digital emulation of distortion effects by wave and phase shaping methods,” in *Proceedings of the 13th International Conference on Digital Audio Effects (DAFx-10)*, Graz, Austria, Sept. 6–10 2010.
- [14] A. Novak, L. Guadagnin, B. Lihoreau, P. Lotton, E. Brasseur, and L. Simon, “Non-linear identification of an electric guitar pickup,” in *Proceedings of the 19th International Conference on Digital Audio Effects (DAFx-16)*, Brno, Czech Republic, Sept. 5–9 2016, pp. 241–246.
- [15] G. Evangelista and M. Raspaud, “Simplified guitar bridge model for the displacement wave representation in digital waveguides,” in *Proceedings of the 12th International Conference on Digital Audio Effects (DAFx-09)*, Como, Italy, Sept. 1–4 2009.
- [16] A. Wang, *Instantaneous and Frequency-Warped Signal Processing Techniques for Auditory Source Separation*, PhD dissertation, Stanford University, 1994.
- [17] V. Välimäki, H.-M. Lehtonen, and T. I. Laakso, “Musical signal analysis using fractional-delay inverse comb filters,” in *Proceedings of the 10th International Conference on Digital Audio Effects (DAFx-07)*, Bordeaux, France, Sept. 10–15 2007, pp. 261–268.
- [18] S. J. Orfanidis, *Introduction to Signal Processing*, pp. 202–210, 409–421, Prentice Hall, 2010.
- [19] M. Gainza, R. Lawlor, and E. Coyle, “Harmonic sound source separation using FIR comb filters,” in *Proceedings of the 117th Audio Engineering Society Convention*, San Francisco, CA, USA, Oct. 28–31 2004.
- [20] T. Miwa, Y. Tadokoro, and T. Saito, “Musical pitch estimation and discrimination of musical instruments using comb filters for transcription,” in *Proceedings of the 42nd Midwest Symposium on Circuits and Systems*, Las Cruces, NM, USA, Aug 8–11 1999, pp. 105–108.
- [21] J. Pekonen and V. Välimäki, “Filter-based alias reduction for digital classical waveform synthesis,” in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, Las Vegas, NV, USA, Mar. 2008, pp. 133–136.

- [22] C. Traube and P. Depalle, “Extraction of the excitation point location on a string using weighted least-square estimation of a comb filter delay,” in *Proceedings of the 6th International Conference on Digital Audio Effects (DAFx-03)*, London, UK, Sept. 8–11 2003.
- [23] E. Scheirer, “Tempo and beat analysis of acoustic musical signals,” *Journal of the Acoustical Society of America*, vol. 103, no. 1, pp. 588–601, 1998.
- [24] M. R. Schroeder, “Natural sounding artificial reverberation,” *Journal of the Audio Engineering Society*, vol. 10, no. 3, pp. 219–223, 1962.
- [25] J. A. Moorer, “About this reverberation business,” *Computer Music Journal*, vol. 3, no. 2, pp. 13–28, 1979.
- [26] W. A. Sethares, *Tuning, Timbre, Spectrum, Scale*, pp. 90–92, 371–373, Springer-Verlag, 2nd edition, 2005.
- [27] R. Plomp and W. J. M. Levelt, “Tonal consonance and critical bandwidth,” *Journal of the Acoustical Society of America*, vol. 38, no. 4, pp. 548–560, 1965.
- [28] D. A. Jaffe and J. O. Smith, “Extensions of the Karplus-Strong plucked-string algorithm,” *Computer Music Journal*, vol. 7, no. 2, pp. 56–69, 1983.
- [29] C. C. D. Baptista, “Distorcedor R VIII – conclusão,” *Nova Eletrônica*, vol. 1, no. 5, pp. 562–569, June 1977.
- [30] S. Daniels, “How To Make A PC Board By The Positive Photo Process,” Available at <http://diy.smallbearelec.com/HowTos/PhotoPCBoards/PhotoPCBoards.htm>, 2010.
- [31] J. J. Ebers and J. L. Moll, “Large-signal behavior of junction transistors,” *Proceedings of the IRE*, vol. 42, no. 12, pp. 1761–1772, 1954.
- [32] A. Vaneev, “r8brain-free-src: High-quality pro audio sample rate converter / resampler C++ library,” Available at <https://github.com/avaneev/r8brain-free-src>, 2019.
- [33] A. V. Oppenheim and R. W. Schaffer, *Digital Signal Processing*, chapter The Discrete Fourier Transform, p. 671, Prentice-Hall, 1975.
- [34] S. Laguerre, “DAFx20-Hex-Sim: Audio files accompanying the DAFx 2020 paper,” Available at <https://slaguerre2.github.io/DAFx20-Hex-Sim/>, 2020.
- [35] P. R. Cook and G. P. Scavone, “The Synthesis ToolKit in C++ (STK),” Available at <http://ccrma.stanford.edu/software/stk/>, 2019.