Title
Design of a Digital Guitar Amplifier

Author
Wahal, Moon

Publication Date
2022

UNIVERSITY OF CALIFORNIA SAN DIEGO

Design of a Digital Guitar Amplifier

A Thesis submitted in partial satisfaction of the requirements
for the degree Master of Science

in

Electrical Engineering (Signal & Image Processing)

by

Moon Wahal

Committee in charge:

Professor Pamela Cosman, Chair
Professor Truong Nguyen
Professor Tamara Smyth

2022

The Thesis of Moon Wahal is approved, and it is acceptable in quality and form for publication on microfilm and electronically.

University of California San Diego

2022

DEDICATION

I would like to thank my family for their unconditional love and support in my education and creative endeavors. I would not be the person I am today without their unending belief in me. I would like to thank my partner and closest friends for standing by me and giving me the strength to tackle every challenge I faced along the way.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF SUPPLEMENTAL SOUND RECORDINGS

ABSTRACT OF THE THESIS

Design of a Digital Guitar Amplifier

by

Moon Wahal

Master of Science in Electrical Engineering (Signal & Image Processing)

University of California San Diego, 2022

Professor Pamela Cosman, Chair

Digital amplifier modelling is an application of digital signal processing to recreate the sound of plugging an electric guitar into a traditional physical amplifier. Physical amplifiers can be impractical and unreliable for constant live use by professional musicians, and digital amplifiers have been developed to solve these problems. This paper explores my process of designing a digital guitar amplifier as well as an analysis of the final sound output.

CHAPTER 1: INTRODUCTION TO DIGITAL AMPLIFIER MODELLING

Traditional analog guitar amplifiers serve to increase the level of an incoming electric guitar signal and shape the tonal characteristics of the sound to the user's taste. This process usually adds intentional harmonic distortion to the signal, and electric guitarists take great care to decide on which specific amplifier matches their sonic needs. Despite having different sounds and circuit components, all analog amplifiers are constructed out of similar physical components: resistors, capacitors, transformers, transistors, and often vacuum tubes as well. The similarity of these designs lead to common problems across all physical amplifiers. Firstly, all the components add weight and bulk to the amplifier. Guitar amplifiers can range between 10 and 100 pounds, depending on the size of the amp and the loudness required from the output [7]. For professional musicians on the road, this can result in body strain over repeatedly setting up their amplifiers every night. Additionally, these amplifiers are not fully reliable due to the physical nature of their components. Over time, component values can drift due to the heat damage from regular amplifier operation, and constant travel on the road opens up avenues for many potential incidents that could damage the amp, from an accidental drop during a load or unload to components getting jostled loose while driving on bumpy highways over long hours.

Digital amplifiers were developed to combat these issues. Guitarists playing digital rigs do not have to worry about potentially damaging their bodies either, as the weight of these amps is simply the weight of whatever computer the program is running on. Digital amps do not suffer from the same reliability and damage concerns as physical ones as there are no physical components in the amplifier program itself that could get damaged from heat or physical accidents, and in the event of any computer hardware damage, the program can be transferred over to a new computer without issue.

The primary challenge with digital amplifiers is implementing them in a way that realistically sounds like a physical amplifier. Dempwolf et al. found that traditional amplifiers can be modelled by a nonlinear clipping stage, an equalizer stage, and another nonlinear clipping stage [3]. Pakarinen et al. studied different techniques for modelling these components, from black-box approaches that view an amplifier as an abstract system focused on finding coefficients to replicating a sound to white-box approaches such as virtual analog modelling that seek to digitally recreate discrete parts of amplifier circuits [9]. Yeh et al. expanded on the virtual analog modelling concept by studying the tonestack circuit of a 1959 Fender Bassman and deriving its transfer function [12]. Yeh furthered his research in virtual analog modelling of bipolar junction transistor (BJT) and vacuum tube amplification circuits to determine properties about their nonlinear behavior [11]. Cohen et al. explored various models to replicate the triode stage of a 12AX7 preamplifier tube, which is commonly used in amplifier designs [2]. Buffa et al. studied ways to model entire amplifiers using a multiband modelling approach [1]. Macak et al. also studied ways to implement amplifier-like nonlinear functions using differential equations [8]. Erbe explored different waveshaping functions to introduce non-linear clipping and subjectively analyzed the tonal qualities of different functions [5][6]. Eichas et al. developed grey-box methods to automate modelling of guitar amplifiers, based on combinations of linear time-invariant filtering blocks based on physical amplifier components and nonlinear distortion blocks to recreate the sound of amplifier stages [4]. Wright et al. applied machine learning techniques, using neural networks to make a black-box model of an amplifier [10].

CHAPTER 2: AMPLIFIER DESIGN

To develop my amplifier, I chose to use a mixed approach, based on virtual analog modelling of tonestack circuits combined with black-box modelling of nonlinear clipping stages to recreate the overall sound of a guitar amplifier. My amplifier structure is based off the work done by Dempwolf et al. and consists of a nonlinear preamplifier stage, a tonestack circuit to manipulate the frequency balance of the amplifier, and a nonlinear power amplifier to drive a speaker [3]. Having nonlinear sections in a guitar amplifier is important for a musician to control the timbre of the sound as they play. Typically, guitar amplifiers will compress and distort more as the player picks the strings harder and provide less compression and clipping when the musician plays lightly. This touch responsiveness gives the musician complete control over the sound of their instrument and emphasizes the range of dynamics present in a performance. The preamp usually includes multiple cascading gain stages that compress and distort the sound as the input signal is driven higher. Additionally, the lower frequencies are filtered out between each gain stage to stop the build up of unwanted rumble. The tonestack is an RC circuit with 3 potentiometers that adjust the amount of bass, middle, and treble frequencies present in the signal. The power amplifier serves to bring the level of the signal after the tonestack up to a suitable level for a speaker cabinet. In traditional amplifiers, this process is done with vacuum tubes that add a low amount of compression and harmonic distortion during their operation [2]. The compression and distortion increase as the input level increases. The controllable parameters available for the user are volume, gain, bass, middle, and treble (v, g, b, m, t, respectively), selectable between 0 and 10 and scaled down to internally range from 0 to 1, as well as a switch to select between 2 different tonestacks.

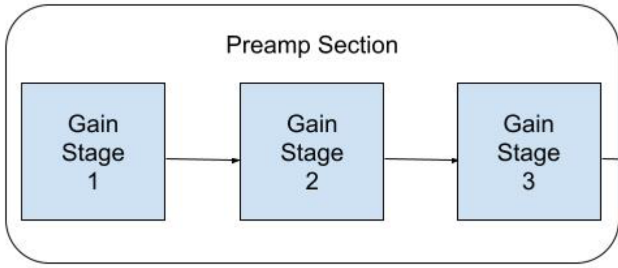

Figure 2.1: Block diagram of a guitar amplifier

For my preamplifier, I chose to design three different gain stages. Each gain stage applies compression, distortion, and mild filtering to the input signal. The amount of compression and distortion applied increases proportionally with the set volume and gain parameters. While each gain stage functionally serves the same purpose, they are all implemented with slight tweaks to parameters to create more harmonic variation across the entire preamplifier.

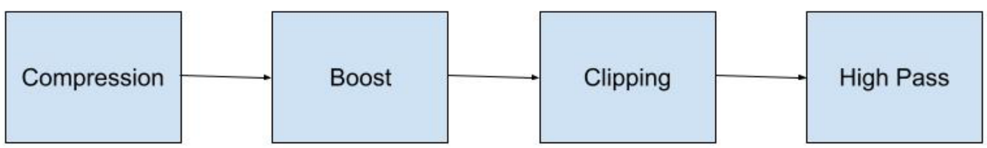

Figure 2.2: Block diagram of a gain stage

All the gain stages start with a polynomial compression to the input signal. The compressor serves to reduce the gain of the louder parts of the signal to make the overall signal amplitude more even before clipping. Functionally, the compressor takes in an input signal with magnitude ranging from 0 to 1 and multiplies values above a specific threshold, 0.5, by a ratio to reduce them. To create my compression polynomials, I decided on my desired compression ratios, shown in Table 2.1, and then used MATLAB's *polyfit* function to find third order polynomials that matched my desired curves. The compression curves I used are shown in Figure 2.3. Compressors 1, 2, and 3 were used for the preamplifier gain stages, and Compressor 4 will be used later in the power amp section. After applying the compression polynomial to the input signal, the output signal is then averaged with the original input depending on the user's volume parameter, with higher volume levels leading to a more compressed output. This process is detailed in Equation 2.1.

Equation 2.1: Compression algorithms

$$R_1 = \begin{cases} 1 - (-0.3544x^3 - 0.0581x^2 + 0.1141x + 1), & x \geq 0.5 \\ 0, & x < 0.5 \end{cases}$$

$$R_2 = \begin{cases} 1 - (0.2416x^3 - 1.1623x^2 + 0.5200x + 1), & x \geq 0.5 \\ 0, & x < 0.5 \end{cases}$$

$$R_3 = \begin{cases} 1 - (0.8406x^3 - 2.3437x^2 + 0.9600x + 1), & x \geq 0.5 \\ 0, & x < 0.5 \end{cases}$$

$$R_4 = \begin{cases} 1 - (0.6360x^3 - 1.4471x^2 + 0.5623x + 1), & x \geq 0.5 \\ 0, & x < 0.5 \end{cases}$$

$$C_n = x - R_n$$

$$y_1 = x * (1 - V) + C_1 * V$$

$$y_2 = x * (1 - V * 0.8) + C_2 * V * 0.8$$

$$y_3 = x * (1 - V * 0.7) + C_3 * V * 0.7$$

$$y_4 = x * (1 - V) + C_4 * V$$

The input gain is given by *x*, and the output gain is given by *y*. *C* is the output of the compressor, and *R* is the amount of gain reduction applied by the compressor. *V* is the user's volume parameter, ranging from 0 to 1. In each subsequent gain stage, the balance is shifted to favor more of the original signal entering the stage in the mix to avoid an unpleasant sounding overcompressed tone.

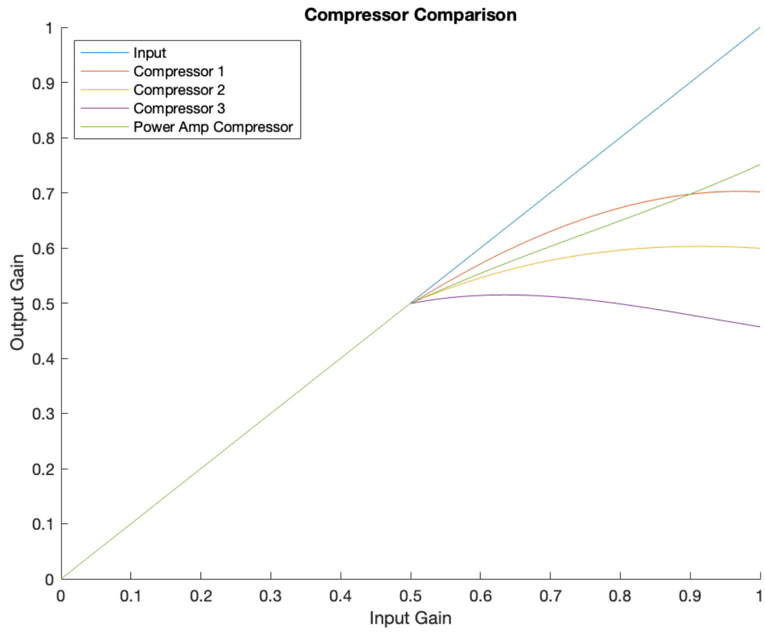

Figure 2.3: Comparison of the compression algorithms

Table 2.1: This table shows the ratios of the output gain against the input gain of all 4 of my compressors.

Input Gain	0.7	0.8	0.9	1
Compressor 1	0.925	0.875	0.8	0.7
Compressor 2	0.875	0.8	0.7	0.6
Compressor 3	0.815	0.7	0.58	0.46
Compressor 4	0.9	0.85	0.8	0.75

After compression, the output signal is then boosted or cut in a range of +6dB to -12dB, depending on the volume parameter. Then, the adjusted signal is sent into a clipping algorithm, where the shape of the clipping is determined by the gain parameter. For my design, I chose to apply two different soft clipping algorithms inspired by Erbe's research in waveshaping, based on a cubic function and a tangent function shown in Equation 2.2 and Figure 2.4 [5][6]. In these equations, x refers to the input level, y is the output level, and the parameter α is used to control the amount of clipping created by each function, ranging between 0 to 2.5 for the cubic function and 0 to 12 for the tangent function based on the user's selected gain value. These algorithms were selected for a two key reasons. Firstly, they both are odd functions that will clip the positive and negative sides of the input signal evenly. Secondly, I found the tonal characteristics of the clipping created by these functions and how the amount of clipping applied by increasing α within my set ranges very pleasant to listen to as well as helpful to design my gain stages around.

As α increases, the tangent function will clip the signal at lower levels and add more odd-order harmonics to the signal. When applied to a sinusoid, this has the effect of shaping it into a square wave. The cubic function will start introducing and strongly emphasizing the third order harmonic of the input signal, even making it louder than the fundamental frequency at high enough values for α. This adds an interesting sonic texture to the distortion that I found very pleasant, especially when cascaded into the tangent clipper. In my preamplifier, the first gain stage uses a cubic clipper, and the second and third gain stages use the tangent clipper. Figures 2.5 and 2.6 show the output of the clipping functions as applied to a sinusoidal input, and Figures 2.7 and 2.8 show the Fourier transformations of the output from the clippers.

Equation 2.2: Clipping algorithms

$$y_{cubic} = x - \alpha * x^3$$

$$y_{tan^{-1}} = \frac{2}{\pi} * \tan^{-1}(\alpha * x)$$

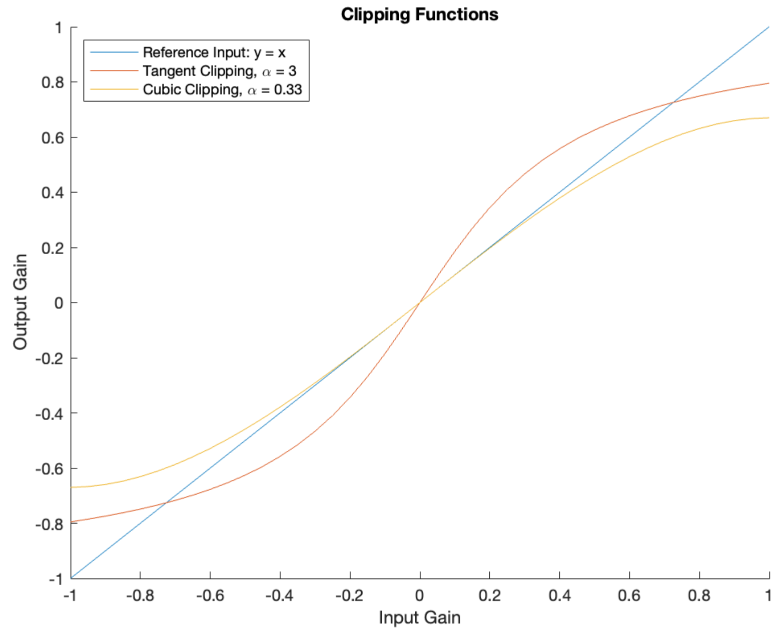

Figure 2.4: The clipping functions

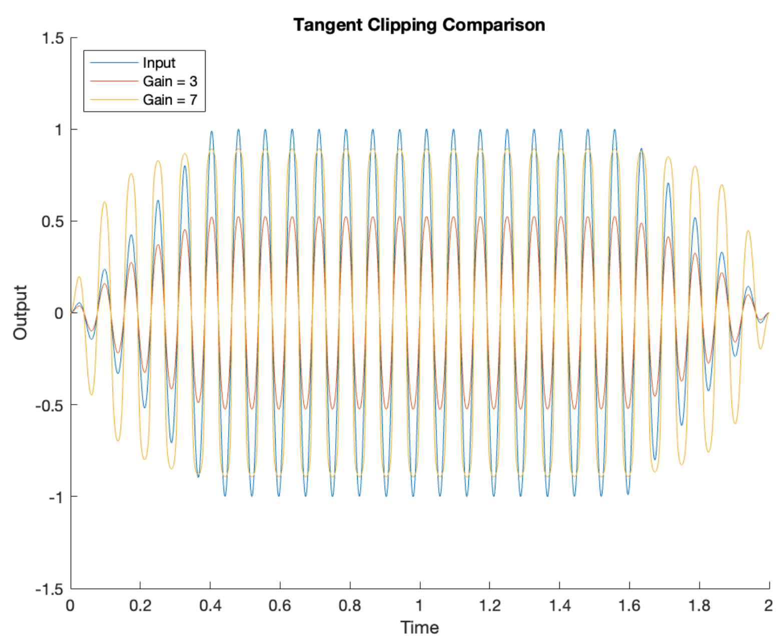

Figure 2.5: The tangent clipper applied to a sinusoid

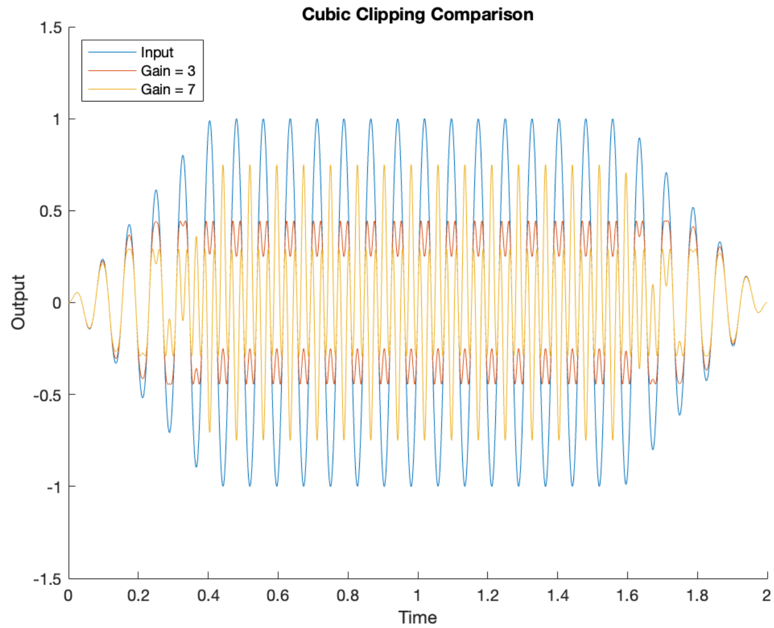

Figure 2.6: The cubic clipper applied to a sinusoid

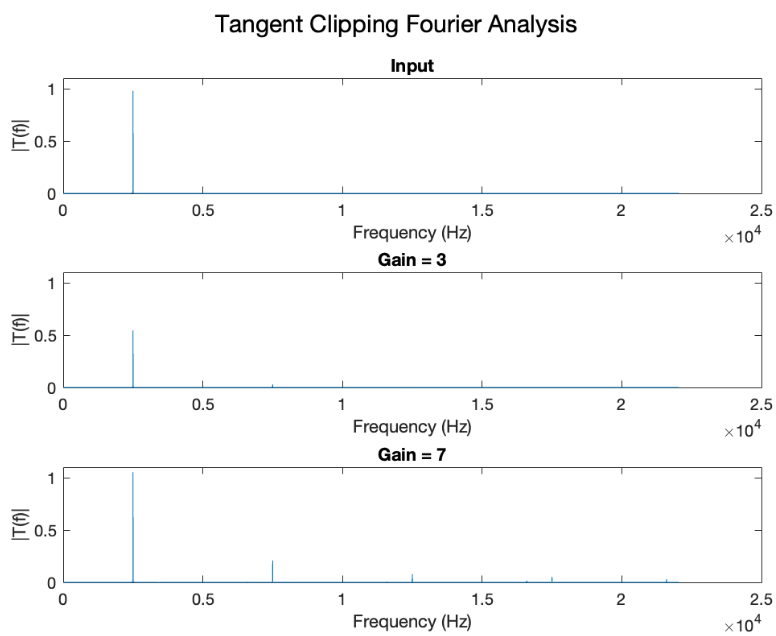

Figure 2.7: Fourier transform of a sine wave through the tangent clipper

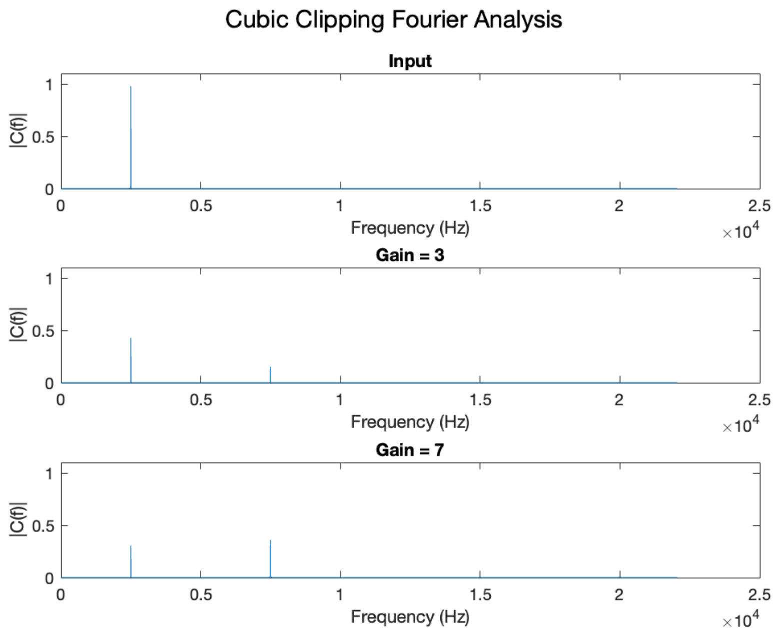

Figure 2.8: Fourier transform of a sine wave through the cubic clipper

Lastly, the signal is highpass filtered above 80 Hz. The lowest note on an electric guitar in standard tuning is E_2, which has a fundamental frequency around 82 Hz. If the preamp compressed and distorted the frequencies lower than this through cascading gain stages, it would only be amplifying any low frequency noise present in the signal and create an unpleasant audible rumbling in the final output of the amplifier. For my design, I chose to use a first order highpass filter modelling the simple RC circuits implemented in traditional analog amplifiers, and then used the bilinear transformation with prewarping at the cutoff frequency of 80 Hz to arrive at my final digital filter shown in Equation 2.3. All operations are done with a sampling frequency of 44.1 kHz. In this equation, ω_0 refers to the cutoff frequency of the filter in rad/sec, f_c is the cutoff frequency in Hertz, and ω_p is the prewarped cutoff frequency.

Equation 2.3: High Pass Filter

$$H(s) = \frac{s}{s + \omega_0}$$

$$\omega_0 = f_c * 2\pi = 502.6548\, rad/sec$$

$$\omega_p = \frac{2}{T}\tan\left(\frac{\omega_0 * T}{2}\right) = 502.6603\, rad/sec$$

$$s = \frac{2 * (1 - z^{-1})}{T * (1 + z^{-1})}$$

$$H(z) = \frac{0.9943 - 0.9943z^{-1}}{1 - 0.9887z^{-1}}$$

After the preamplifier, the signal is sent into a tonestack circuit that allows the user to adjust the levels of the bass, midband, and treble frequencies present in the signal. In traditional analog amplifiers, this is usually implemented through a set of RC circuits. For my amplifier, I chose to digitally recreate the tonestack from two common amplifier circuits, a Fender and a Marshall, based on the virtual analog research done by Yeh and Smith [10]. While these circuits use the same overall architecture, the component values used vary in each design, resulting in a subtle shift in the overall character of the amplifier's sound. The user is able to switch between which tonestack circuit is implemented in the signal path. The differences in component values between the two circuits are shown in Table 2.2. The schematic of the tonestack is shown in Figure 2.9. Equation 2.4 shows the transfer function symbolically, following the derivation in Yeh and Smith's paper [10]. In this schematic, RB, RM, and RT are potentiometers that control the level of the bass, middle, and treble frequencies respectively. To solve the transfer function of this tonestack, it is useful to think of the potentiometers instead as two parameterized resistors, as shown in Figure 2.10. Here, *b, m,* and *t* range from 0 to 1 and represent the user's bass, middle, and treble settings respectively. Through circuit analysis, I found the analog domain transfer function of this circuit, and then used the bilinear transformation to map the function to the digital domain as shown in Equations 2.4. The frequency response of each tonestack circuit is available in Figure 2.11.

Table 2.2: Component values for Fender and Marshall tonestack circuits

Component	Fender	Marshall
C_1	0.25nF	0.47nF
C_2	20nF	22nF
C_3	20nF	22nF
R_T	250kHz	220kHz
R_B	1MHz	1MHz
R_M	25kHz	25kHz
R_1	56kHz	33kHz

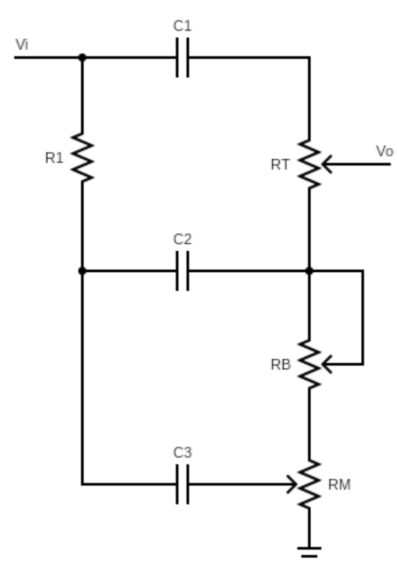

Figure 2.9: Tonestack schematic

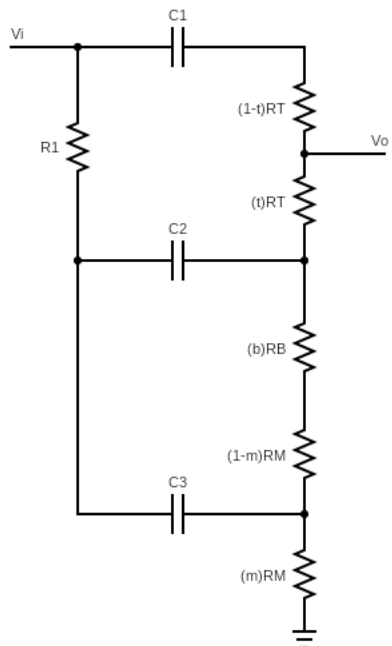

Figure 2.10: Tonestack with parameterized resistors

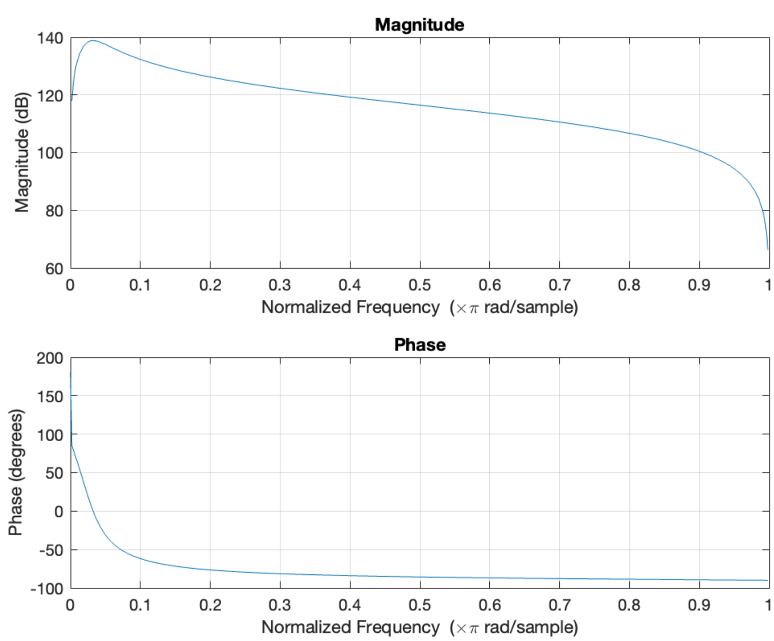

Figure 2.11a: Frequency response of the Fender tonestack

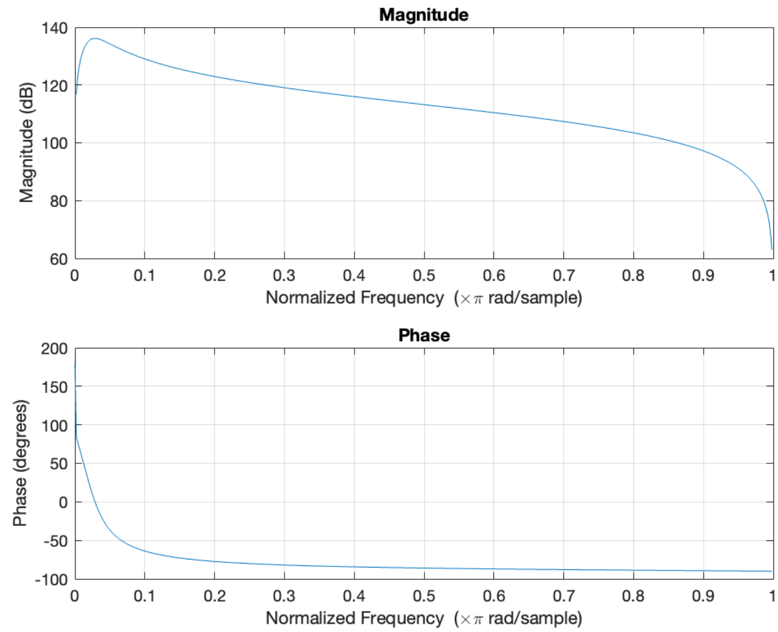

Figure 2.11b: Frequency response of the Marshall tonestack

After the tonestack, the signal is lastly sent to the power amp. The power amp serves to raise the level of the signal coming from the tonestack. As the user increases the volume of the amplifier, the power amp section tends to add subtle compression and clipping to the signal. For my design, I chose to use the cubic clipping function with a scaled down α parameter, ranging from 0 to 1.5. While the compression in this stage functionally happens the same way as in the preamplifier, the curve used for the power amp compressor is smoother as seen in Figure 2.3, leading to a subtler effect. Figure 2.12 shows an example soundfile processed through my system, and Figure 2.13 shows the output of a Fast Fourier Transform applied to a frame of 1024 samples from the steady state of a single note played on a guitar as it is processed through my amplifier. Figures 2.13b-d show how the harmonic spectrum of the note changes as the user's volume and gain controls increase. Figure 2.13b shows that at low gain levels, there is very little harmonic distortion occurring in the amplifier. The higher-order harmonics originally present in the signal are attenuated, and no new harmonics are added. This results in a tone that sounds very clear, with very little audible distortion in the output. This tone is useful for musicians who wish to play extremely cleanly and have the listeners' focus be on their musical phrasing. This type of sound is common in jazz and gospel music. In Figure 2.13c, which shows the response of my amplifier at a medium-gain setting, we can see that the higher-order harmonics that were attenuated at the lower gain setting are starting to be reintroduced back into the signal. This results in a tone that has some amount of audible distortion, which becomes more apparent as the user attacks the strings harder but still clears up when they play lighter. This tone would be useful for musicians who wish to perform with a very large dynamic range, as the added distortion when attacking the strings hard adds emphasis to loud portions while still retaining clarity for softer phrases. This type of sound is common in blues and country music. Figure 2.13d shows that at high-gain settings, the amplifier introduces many higher-order harmonics, beyond what was present in the input signal. This has the effect of audibly adding a lot of distortion to the signal, even if the musician plays lighter. This setting would be useful for musicians who are looking for a sound full of distortion and complex harmonics, such as those who play rock and heavy metal.

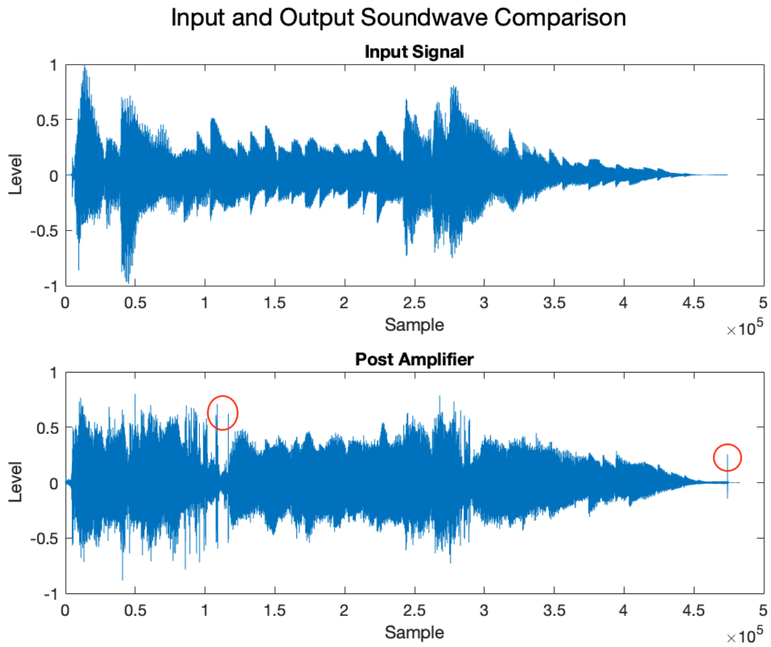

Figure 2.12: Comparison of a soundwave into and out of the amplifier. Clicks are circled in red.

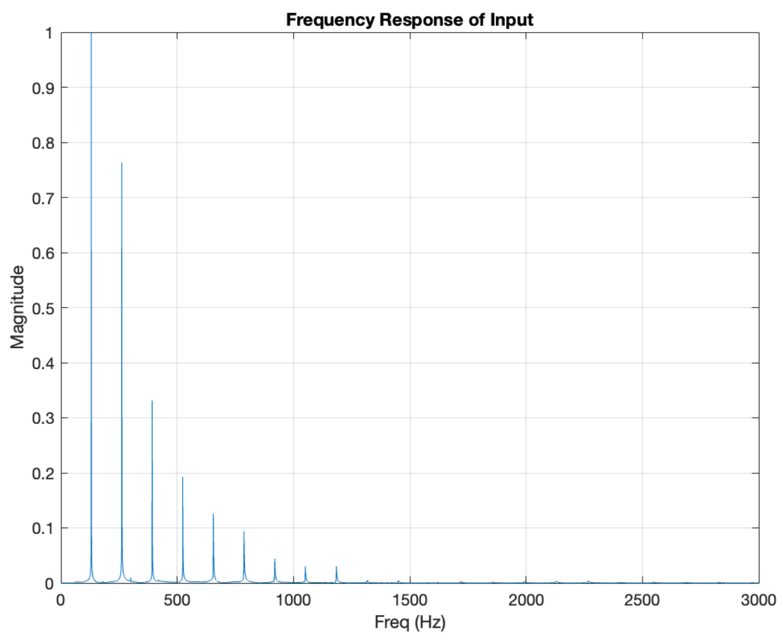

Figure 2.13a: Fourier analysis of the amplifier input

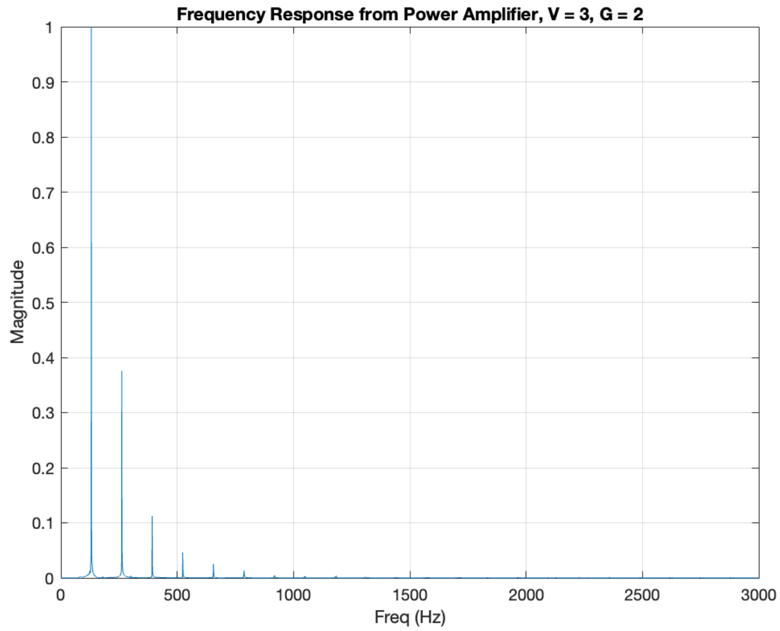

Figure 2.13b: Fourier analysis of low-gain amplifier output

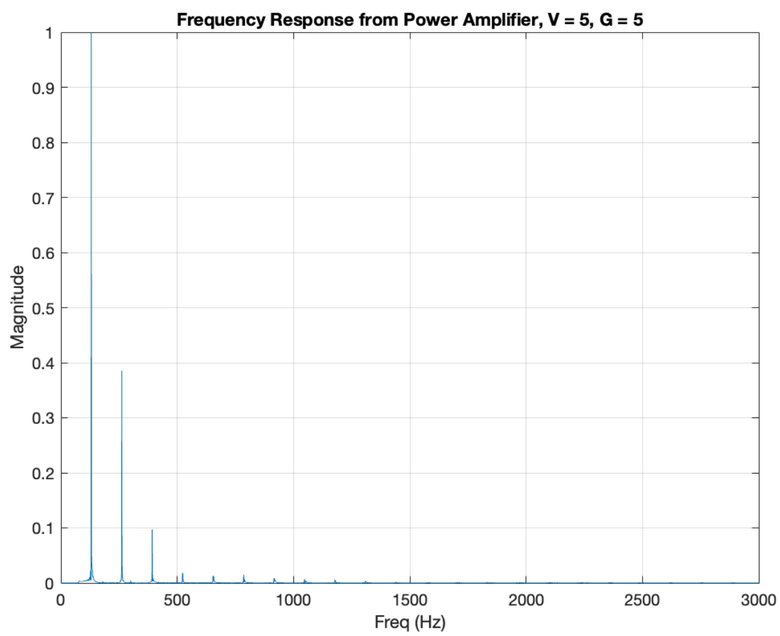

Figure 2.13c: Fourier analysis of medium-gain amplifier output

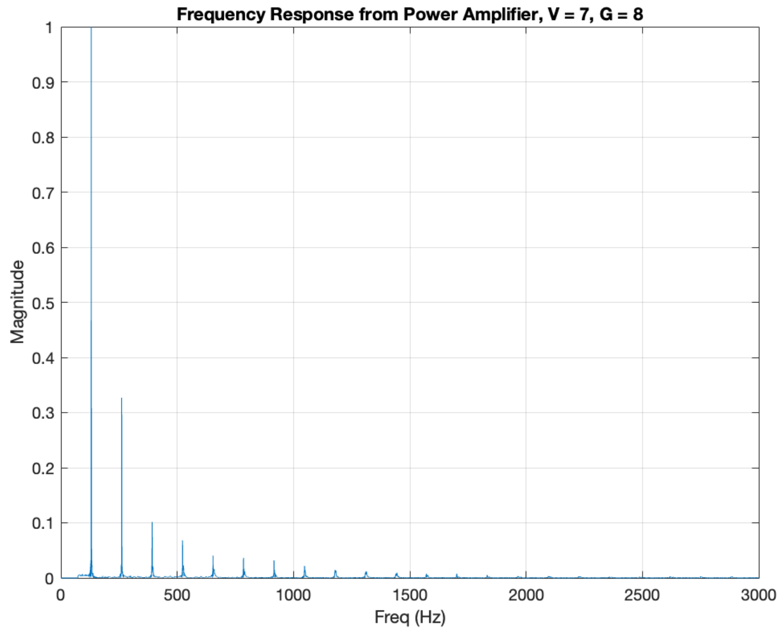

Figure 2.13d: Fourier analysis of high-gain amplifier output

Equation 2.4: Tonestack transfer function

$$
\begin{aligned}
H(s) &= \frac{b_1 s^3 + b_2 s^2 + b_3 s}{a_0 s^3 + a_1 s^2 + a_2 s + a_3} \\
b_3 &= tC_1R_T + mC_3R_M + (C_1 + C_2)(bR_B + R_M) \\
b_2 &= mC_3R_M(C_1R_T + (C_1 + C_2)((1-m)R_M + bR_B)) \\
&\quad + C_1(tR_TR_1(C_2 + C_3) + bR_B(C_2(R_T + R_1) + C_3R_1) \\
&\quad + R_M(C_2R_T + C_2R_1 + C_3R_1)) \\
b_1 &= C_1C_2C_3(R_M(R_T + R_1)(bmR_B - (1-m)R_M) + tR_TR_1(R_M(1-m) + bR_B)) \\
a_3 &= 1 \\
a_2 &= C_1R_T + C_1C_3 + C_2C_3 + C_2R_1 + C_3R_1 + mC_3R_M + bR_B(C_1 + C_2) \\
a_1 &= mC_3R_M(C_1R_T - C_2R_1 + R_M(C_1 + C_2)) + bmR_TR_M(C_1 + C_2) \\
&\quad - m^2C_3R_M^2(C_1 + C_2) + b(C_1C_2R_BR_1 + C_1C_2R_TR_B + C_1C_3R_BR_1 + C_2C_3R_BR_1) \\
&\quad + (C_1C_2R_TR_1 + C_1C_3R_TR_1 + C_1C_2R_MR_1 + C_1C_2R_TR_M + C_1C_3R_MR_1 + C_2C_3R_MR_1) \\
a_0 &= bmC_1C_2C_3R_BR_M(R_T + R_1) - m^2C_1C_2C_3R_M^2(R_T + R_1) \\
&\quad + mC_1C_2C_3R_M(R_MR_1 + R_TR_M + R_TR_1) + C_1C_2C_3R_TR_1(bR_B + R_M)
\end{aligned}
$$

$$
\begin{aligned}
H(z) &= \frac{B_0 + B_1z^{-1} + B_2z^{-2} + B_3z^{-3}}{A_0 + A_1z^{-1} + A_2z^{-2} + A_3z^{-3}} \\
c &= \frac{2}{T} \\
B_0 &= c(-b_3 - b_2c - b_1c^2) \\
B_1 &= c(-b_3 + b_2c + 3b_1c^2) \\
B_2 &= c(b_3 + b_2c - 3b_1c^2) \\
B_3 &= c(b_3 - b_2c + b_1c^2) \\
A_0 &= -a_3 - c(a_2 + a_1c + a_0c^2) \\
A_1 &= -3a_3 - c(a_2 - a_1c - 3a_0c^2) \\
A_2 &= -3a_3 - c(-a_2 - a_1c + 3a_0c^2) \\
A_3 &= -a_3 - c(-a_2 + a_1c - a_0c^2)
\end{aligned}
$$

CHAPTER 3: RESULTS & DISCUSSION

Overall, I was satisfied with the results of my amplifier. I successfully achieved my goals of digitally processing an electric guitar signal to make it sound like the output from a guitar amplifier. To judge my results, I had 15 undergraduate musicians from Musicians' Club at UCSD listen to sound files of recordings generated by my amplifier mixed in with recordings from traditional analog amplifiers. Most of the listeners could guess with good accuracy which sound files came from my amplifier and remarked that it sounded different and interesting compared to the analog amps they had been used to hearing. While they were able to pick out my amplifier, the listeners stated that they did enjoy the timbral content of my design and were interested in one day being able to play it themselves. Based on this feedback, I feel that I was successful in achieve my core goal of designing a digital amplifier.

However, there are a few areas where I believe I could improve this project further. Firstly, my algorithm will only work to process prerecorded signals. I plan on expanding my design to allow for real-time operation for live performance. Furthermore, I would like to modify the equalizer section and allow the user to have more varied sounds available from the tonestack section. With my current design, the two circuits sound fairly similar to each other, and I would like to implement more virtual analog models of different circuits to shape the frequency curve of the amplifier to a user's specific taste. Additionally, my current implementation of the tonestack exhibits a problem where it will sometimes add unwanted clicks and pops to the input signal, occasionally resulting in a "broken" sounding tone. These clicks can be seen as occasional discontinuous jumps in the soundfile, circled in Figure A.8. Despite my testing, I could not narrow down the cause of this bug, and I would like to study this more to figure out what part of the tonestack algorithm is causing them and how to control it further. In addition, I would like to develop the power amp section further by adding in a secondary equalization stage. Some traditional amplifier designs have a secondary equalization circuit placed in the power amp to fine tune the deep bass and extreme high frequencies, which tend to affect the compression and distortion characteristics of the power amp stage as well. I am interested in researching this part of amplifier design further and implementing these concepts into my own amp. Lastly, I would love to explore more ways of implementing clipping and harmonic control into my amplifier. I really enjoyed the effect of the cubic clipper creating a harmonic overtone stronger than the fundamental frequency, and I would love to explore more ways of modifying the harmonic spectrum of the input signal and see what sorts of timbres I can create. While I only used two distinct clipping algorithms in this design, I want to explore more ways of creating clipping and hear how different algorithms can interact with each other in cascaded designs.

In conclusion, I believe I have successfully made a strong amplifier core to process my guitar audio. The concepts I learned by approaching this project gave me a fundamental understanding of how an amplifier works and how to effectively shape the timbre of an electric guitar signal, and I plan on using what I learned from this project to create more of my original amplifier designs in the future.

REFERENCES

[1] Buffa, M., and Lebrun, J. "Real time tube guitar amplifier simulation using WebAudio." Web Audio Conference 2017–Collaborative Audio# WAC2017. 2017.

[2] Cohen, I., and Helie, T. "Simulation of a guitar amplifier stage for several triode models: examination of some relevant phenomena and choice of adapted numerical schemes." Audio Engineering Society Convention 127. Audio Engineering Society, 2009.

[3] Dempwolf, K., Holters, M., Möller, S., Zölzer, U. "The influence of small variations in a simplified guitar amplifier model." Proceedings of the 12th International Conference on Digital Audio Effects (DAFx'09). 2009.

[4] Eichas, F., and Zölzer, U. "Gray-box modeling of guitar amplifiers." Journal of the Audio Engineering Society 66.12 (2018): 1006-1015.

[5] Erbe, T. "Clipping Methods." Synth Notes: Practical Synthesis and Computer Music Programming. 22 Oct. 2019, http://synthnotes.ucsd.edu/wp4/index.php/2019/10/22/clipping-methods.

[6] Erbe, T. "Saturation and Waveshaping." Synth Notes: Practical Synthesis and Computer Music Programming. 30 Oct. 2019, http://synthnotes.ucsd.edu/wp4/index.php/2019/10/30/saturation-and-waveshaping.

[7] Laz, D. "How Much Does a Guitar Amp Weigh (With Examples)." Way More Sound, 24 Nov. 2020, waymoresound.com/how-much-does-a-guitar-amp-weigh-with-examples.

[8] Macak, J., and Schimmel, J. "Real-time guitar tube amplifier simulation using an approximation of differential equations." Proceedings of the 13th International Conference on Digital Audio Effects (DAFx'10). 2010.

[9] Pakarinen, J., and Yeh, D.T.. "A review of digital techniques for modeling vacuum-tube guitar amplifiers." Computer Music Journal 33.2 (2009): 85-100.

[10] Wright, A., Damskägg, E., Juvela, L., Välimäki, V. "Real-time guitar amplifier emulation with deep learning." Applied Sciences 10.3 (2020): 766.

[11] Yeh, D.T. "Automated physical modeling of nonlinear audio circuits for real-time audio effects—Part II: BJT and vacuum tube examples." IEEE Transactions on Audio, Speech, and Language Processing 20.4 (2011): 1207-1216.

[12] Yeh, D.T., and Smith, J.O. "Discretization of the'59 Fender Bassman tone stack." Proc. of the Int. Conf. on Digital Audio Effects (DAFx-06). 2006.