Analog Emulation of the Black Arts Toneworks Pharaoh Fuzz Guitar Effect Pedal

- A wave simulation approach -

Master Thesis

Troels Lunde Hagensen

Aalborg University Copenhagen
Sound and Music Computing

AALBORG UNIVERSITY
STUDENT REPORT

Sound and Music Computing
Aalborg University Copenhagen
http://www.aau.dk

Title:
Analog Emulation of the Black Arts Toneworks Pharaoh Fuzz Guitar Effect Pedal

Project Period:
Spring Semester 2017

Participant(s):
Troels Lunde Hagensen

Supervisor(s):
Stefania Serafin

Copies: 1

Page Numbers: 61

Date of Completion:
June 2, 2017

Abstract:

In the course of the project a commercial available analog guitar effect pedal, the 'Pharaoh' fuzz by Black Art Toneworks, has been thoroughly analyzed and tested. Based on this a digital model has been developed and evaluated. The model has been based on numerous recorded analysis signals. The recordings were emulated by the use of a transfer function to simulate the waveforms. Insight of how the Pharaoh fuzz works were obtained through a circuit schematic and SPICE simulations thereof. The developed pedal is able to produce very similar waveforms in accordance with the analysis recordings. A qualitative evaluation of the simulated effect pedal was conducted with two guitarists. Both found the pedal considerably similar to the Pharaoh.

AALBORG UNIVERSITET
STUDENTERRAPPORT

Lyd og Musik
Aalborg Universitet København
http://www.aau.dk

Titel:
Analog Emulering af Black Arts Toneworks Pharaoh Fuzz Guitareffektpedal

Projektperiode:
Forårssemestret 2017

Deltager(e):
Troels Lunde Hagensen

Vejleder(e):
Stefania Serafin

Oplagstal: 1

Sidetal: 61

Afleveringsdato:
2. juni 2017

Abstract:

I løbet af dette projekt blev en kommerciel tilgængelig analog guitar effekt pedal, en 'Pharaoh' fuzz af Black Arts Toneworks, grundigt analyseret og testet. Baseret på dette, blev en digital model udviklet og evalueret. Modellen er blevet baseret på adskillige optagede analysesignaler. Optagelserne blev emuleret ved hjælp af en overføringsfunktion for at simulere bølgeformerne. Indsigt i hvordan Pharaoh'en fungerer blev opnået via et kredsløbsdiagram og SPICE simuleringer deraf. Den udviklede pedal er i stand til at gengive meget lignende bølgeformer i overensstemmelse med analyseoptagelserne. En kvalitativ evaluering af den simulerede effektpedal blev udført med to guitarister. Begge fandt pedalen betydeligt lig med Pharaoh'en.

Contents

Preface

This report and the project behind it has been carried out during the 4th and final semester of the Sound and Music Computing master's education at Aalborg University, Copenhagen. It presents the master's thesis of the education.

In the report, an approach to analog emulation of an electric guitar effect pedal, represented by the 'Pharaoh' fuzz pedal, created by 'Black Arts Toneworks', is presented. A focus on simplifying the emulation process was held throughout the implementation, and is reflected in the choice of used hardware and software. The idea behind this approach was to make analog emulation more approachable.

The project was carried out in the spring of 2017 under the supervision of Prof. Stefania Serafin.

I would like to thank Victor Milesen and Jorge Madrid Portillo for participating in the evaluation of the presented pedal emulation and providing feedback.

Aalborg University, June 2, 2017

Troels Lunde Hagensen

Troels Lunde Hagensen
thagen12@student.aau.dk

Chapter 1

Introduction - what is all the fuzz about?

Fuzz, in musical context, is a type of distortion used as an effect to change sound character. It is used to great extent with electric guitars. In 'Guitar Effect Pedals' Dave Hunter describes the sound as *"[...] slightly wooly, rounded warm but sparkly distortion all over the guitar signal [...which brings] more meat, girth, and sustain to the sound"* [1]. In the same category of distortion as fuzz are overdrive and distortion - the latter, a specific type of distortion holds the same name as the phenomenon. Dave Hunter describes how the different effects are produced as follows: *"Turn up a tube amp to where it is starting to break up and you have got gentle overdrive; crank it to the max and you have got heavy distortion. Pull out one of the pair of output tubes, use the wrong-value bias resistors on a preamp tube, or beat it senseless with a crowbar and you might just get it to sound like fuzz."* [1]. Before the introduction of the fuzz pedals in the '60s the sound was a result of, or achieved with, faulty gear. The guitarist from The Kinks is known to have sliced up a speaker membrane, which produces the distortion heard in the song 'You Really Got Me' from 1964. The Maestro Fuzz-Tone is generally regarded as the first commercially available fuzz effect pedal. The iconic fuzz was used on the song '(I Can't Get No) Satisfaction' in 1965 by 'The Rolling Stones'. The sound was meant as mimicking brass instruments [1]. Lastly worth mentioning and probably the most iconic fuzz effect pedal of all is the Fuzz Face from 1966 - whether the smiling face on the round box, Jimi Hendrix or the sound has contributed the most to the immortalization is not to say.

Great sounding analog effect pedals are issued all the time, some of which are innovative, some reissued classics and attempted clones, and a few plain silly ones are released from time to time as well. There is a great affection towards the analog pedals in the guitar community. The digital domain of guitar equipment is receiving less attention it seems. One could speculate why that is. One of the possible reasons for this could be the eternal feud between those who swear by analog equipment (tube equipment in particular) and those who are less picky. As a new guitarist it is easy to get the impression that the only equipment that works is the analog, in particular tube amplifiers, without forming your own personal experience. The solid state amplifiers are frowned upon by some - even more so is that the case with digital equipment. There is a claim that the solid state and digital amplifiers are not receiving the pedals as well as the tube models. Whether this claim is true or not is not in the scope of this project. Another reason for lack of interest in the digital equipment might have to do with the lack of physical aspect of digital plug-ins. This absence makes the plug-ins unfit or less than optimal for live performances. There are examples of physical, digital effect pedals, but many of these seem to try to capture every thinkable sound effect, equalizing, channel selection, and effect pedal in one unit. The Behringer V-Amp 2, Boss GT-10, and the DigiTech Vocal 300 are all examples of this concept; a concept that might be too niche for other than the guitarists looking for a sole piece of equipment to expand the sound of their guitar. More common is the electric guitarist taking advantage of the vast selection of available analog pedals to sort out a more personal sound. A digital option where people have the possibility to create their own sound and still have a physical pedal is the MOD Duo [1]. The device has around 200 free plug-ins it can use and a commercial selection reportedly coming soon. The concept at this point is mainly focused around an open-source community. Of the 200 plug-ins around 10% are in the category of distortion. Only a few of these are fuzz effects. One can speculate why that is; it might be linked to fuzz pedals being less popular than overdrive and distortion pedals. Another explanation could possibly be connected to the complexity of programming and emulating fuzz. More on this in chapter 2.

If done properly analog emulations have many great qualities to offer. The digital format is easy to distribute and share. This results in a limitless supply when an analog emulation is complete. Analog emulations can also ensure the preservation of items no longer in production and offer a more affordable alternative to expensive vintage gear. Wear and tear is not a concern; this can both be seen as a plus - less maintenance and expenses in the long run, and a minus - as the digital item in question is not developing 'character' over time from the missing wear and tear.

A number of key points associated with the emulation of a digital guitar effect pedal have been identified and are summarized here:

- A physical pedal is important for the usability of an analog emulation of an effect pedal.
- Quality appears to be more important than the quantity of effects when dealing with a digital platform for emulated guitar pedals.

[1] https://moddevices.com/

- The prospect of an open-source community, third-party developers, and future additional effects will help justify a higher price than a regular analog pedal.
- Digital fuzz effects pedals are somewhat neglected.

This project is an attempt at developing an embedded analog emulation of a guitar distortion effect pedal - the 'Black Arts Toneworks Pharaoh' fuzz pedal [2]. The emulated model is to be tested in a qualitative user-test to determine to what extent the model is sonically comparable and satisfying.

1.1 The Black Arts Toneworks Pharaoh fuzz effect pedal explained

Fuzz pedals are generally quite simple in the sense of how they are constructed [1]. The original Fuzz Face was constructed with the use of only eleven components; four resistors, three capacitors, two transistors, and two potentiometers. The simple circuitry of a fuzz pedal makes it a perfect fit for a novice DIY (Do It Yourself) effect pedal project. The simple circuitry however, does not result in a simple sound. The fuzz is the 'dirtiest' of the distortion effects. Overdrive pedals get their sound from soft symmetrical clipping of the waveform. A common approach to achieve soft clipping is by use of diodes in the feedback loop of an op-amp [2]; hard clipping, as used in distortion pedals, can be produced by placing the same parallel opposing diodes connected to ground after the op-amp [3]. In simple fuzz pedals a couple of transistors will take care of the amplification and clipping [4]. The amplification in most pedals today will be performed by op-amps, which distort the signal much less than the transistors. These were not invented at the time the fuzz pedals became popular. Furthermore, a clean signal is not the intention with fuzz pedals. Therefore they are still used today for this effect. The transistors produces hard asymmetric clipping resulting in what resemble square waves when the fuzz potentiometer is maxed.

Worth mentioning is that the distinction between these effects can be a bit blurry at times. Most fuzz pedal can also produce soft and hard clipping depending on how it is dialed in; the potentiometers of the fuzz pedal, the guitar and the amplifier all allow for a great deal of tweaking. The electrical components in *figure 1.1* are neither bound to only produce a specific type of clipping. The transistors can produce hard clipping, but can also produce soft clipping if combined in a circuit with the right components [5]. The Pharaoh is in a more complex group of fuzz pedals. It is based on the Ram's Head Big Muff pedal from 1973. The

[2]https://www.blackartstoneworks.com/pedal/pharaoh/

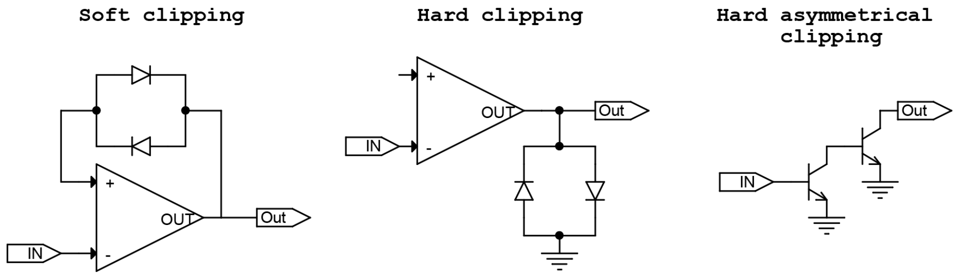


Figure 1.1: Examples of how different types of signal clipping can be achieved in electronic circuits.

following is a brief description, based on the circuit diagram of the Pharaoh fuzz, of how the effect pedal processes the guitar signal. The analysis will be narrowed down to the most characteristic elements of the circuit diagram. See *figure 1.2* for a

Figure 1.2: The Pharaoh fuzz pedal by the company 'Black Arts Toneworks'. The naming of the controls ('Fuzz', 'Tone', 'Hi', etc.) will be referred extensively throughout this report. *Image source: https://www.blackartstoneworks.com/pedal/pharaoh/*

picture of the Pharaoh pedal.

1.1.1 Input segment

In the input segment a switch between two resistors allow for a selection between to gain level - high or low. The switch however, also affects the low cut filtering. The 'high' side allows more treble to come through. The 'low' side has less gain and is therefore not as distorted as the 'high' side resulting in less of the fuzz character. But the pedal can be dialed back to a more fuzzy sound with the distortion potentiometer, marked R24 DIST in the schematic and 'Fuzz' on the pedal. This controls the signal strength, and therefore also when the signal is clipped, which controls the amount of distortion - fuzz. A coupling capacitor is placed after the switch to filter away DC current. The first of four transistors will amplify the signal before reaching the first clipping stage.

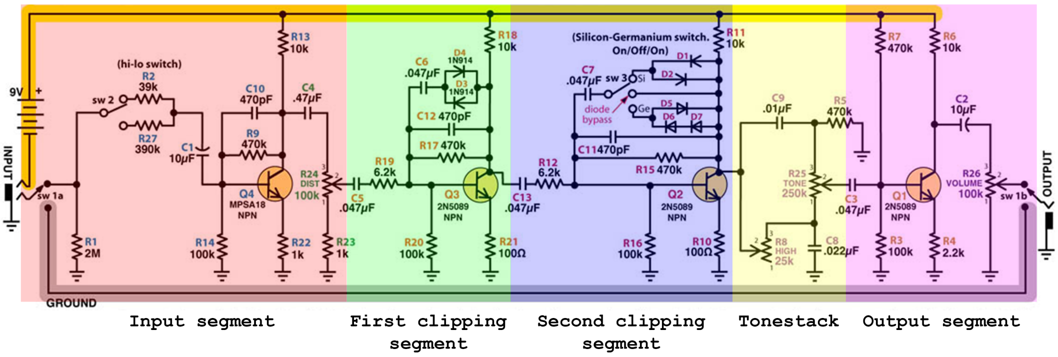


Figure 1.3: Black Arts Toneworks Pharaoh Fuzz circuit schematic, see *Appendix A* for a larger representation, with the different segments highlighted. *Original image source: http://www.bigmuffpage.com*

1.1.2 First clipping segment

The first and second clipping segment are very similar. The only difference is that the second segment has a three-way switch, which allows the user to switch between the type of clipping diode with each of their character. The first segment relies on silicon diodes only, more on that in the next segment description. Each of the two clipping segments have a transistor for amplification.

1.1.3 Second clipping segment

The Pharaoh has the addition of a three-way switch (unlike the original Big muff) with the option for two types of designated clipping diodes, silicon or germanium-based. These can be selected to alter the sound quality. Generally silicon diodes are characterized with a somewhat harsh quality, where the germanium diodes are a bit warmer and easier to listen to. The middle position of the switch will bypass this clipping stage and thereby rely on the more traditional use of transistors for clipping. When switching from the silicon diodes or the bypass option to the germanium diodes, there is a large volume drop. The germanium diodes

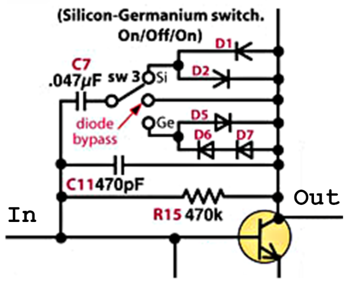


Figure 1.4: The selectable clipping diodes of the second clipping stage. The components are connected in feedback loops, which saturate the sound. A high signal level will result in more saturation because of the increased clipping. *Original image source: http://www.bigmuffpage.com*

start clipping at around half the voltage compared to silicon diodes. To compensate for the lower output signal a third germanium diode has been added in the germanium feedback loop to reduce the volume drop. *See figure* 1.4. The lower volume is, however, very noticeable. This makes switching between diodes in live performances a bit problematic.

1.1.4 Tonestack

The tonestack has two potentiometers. By manipulating the resistance the cutoff frequency is shifted, which will affect the sound. The 'tone' potentiometer can be used to give a bass boost or used as a high pass filter. When boosting the lows with the 'tone', the 'high' control can be used to add lost high end. The tonestack segment looses some mid range frequencies but the increased low and high makes sure that the sound can be dialed in and let the guitar come through in a mix. It is an essential part for the 'Big Muff' sound.

1.1.5 Output segment

A final transistor is used to amplify the sound to make up for the lost volume due to the clipping diodes. A potentiometer is used to control the volume of the output signal.

Chapter 2

Digital effect pedals, emulation methods and tools

The first section of this chapter explores recent work with digital effect pedals. The next section goes into detail with virtual analog emulation theory and approaches, and elaborates on a couple of practical emulation papers with focus on the Fuzz Face effect pedal [6, 7, 8]. In both sections attention will be payed to what hardware and, to some extent, software choices have been made for the implementations. Conclusively the chapter will be rounded off with some considerations, which are thought to be relevant for this project, based on what has been described.

2.1 Work in the field of digital effect pedals

As mentioned earlier many guitar players and the community are quite conservative about their equipment. A broad consensus deems only tube amplifiers and analog effect pedals worthy of use. The attitude is softening on amplifiers, but that does not seem to happen for effect pedals - at least not yet or to the same extent. One way to challenge this attitude can be accomplished by presenting a product, which forces the user to question his belief; perhaps because of additional features and emulated sound, which can not be discerned from its analog original. One of the intriguing potentials of a digital platform is the aspect of modularity. New scripts can be embedded for different guitar effects, and knobs and switches can potentially be assigned as desired. The following examples have adopted some of these possibilities. The OWL [9] and OpenStomp (*see subsection 2.1.2*) are both open source digital effect pedals with some interesting similarities and differences, which will be elaborated on in this section. Combining the microcontroller unit 'Bela'[1] and the open source graphical audio programming environ-

[1] http://bela.io/

ment 'Pure Data'[2] for embedded instrumental purposes will also be elaborated on in this section.

2.1.1 The OWL

The Open Ware Laboratory, or OWL, is a physical, Open source, programmable, platform intended in particular for guitarist as a digital effect pedal [9]. The idea behind the pedal is to offer a platform for musicians with experience with DSP and programming. The device is based around an ARM Cortex M4 microcontroller With a clock speed of 168 MHz the micro controller is capable of performing low latency real time audio signal processing. On the outside the pedal box has five 1/4" jack sockets; two for stereo inputs and outputs and a socket for an expression pedal. It is powered by a 9V DC source like the majority of effect pedals. Unlike most effect pedal the OWL has a USB socket for communication with a PC and transfer of new effect 'patches'. On top of the pedal are four potentiometers, which can be programmed to control desired parameters. The box has a push button to switch between two memory slots, which potentially can be used to have different effects or modes stored. Last but not least, also on the top, is the classic footswitch to engage or bypass the pedal. The effects for the pedal are programmed in C++ in a desired IDE. The framework JUCE [3] is based on C++ and is a powerful tool for the purpose of programming audio processing applications, and therefore suited for developing effects for the pedal. A few effects can be found in a library including overdrive modulation effects, delays, reverbs, and some synthesis effects. The idea is to have a community with people developing and sharing their work.

2.1.2 OpenStomp

Another digital effect pedal worth mentioning is the OpenStomp. It received a lot of attention by numerous online medias (Create Digital Media [4],Premier Guitar [5], Parallax [6]) back in 2008 and 09, but the pedal is nowhere to find as of today, it does not seem that it got any further than being a promising prototype. Regardless, the pedal did possess some compelling and interesting qualities. Of the examples described so far, the OpenStomp reminds a lot about the MOD Duo described in the introduction. The pedal had 15 'modules' for the users to play around with. The idea behind the pedal was to stand out from the rest and invite to play and experiment with the modules. The modules could be chained together as desired. One of the supplied modules, a LFO, could be chained together and set to control

[2]https://puredata.info/
[3]https://www.juce.com/
[4]http://cdm.link/
[5]https://www.premierguitar.com/
[6]http://learn.parallax.com/

a distortion unit e.g. The OpenStomp seemed to focus more on what can be done in the digital domain with a unified system, and less of trying to replicate the sound of specific pedals. Creating emulations of pedals was still an opportunity in a more low-level programming environment for the technically skilled. The MOD Duo seems to be quite the opposite. The available effects seems very inspired of how pedals are approached in the analog world. In that regard, the pedals are very much the same, but with different focuses. The MOD Duo is possible to obtain unlike the OpenStomp, and seems to have a thriving community of users. One can speculate why one of the pedals, the MOD Duo, appears to be a success and the other not. One possible explanation could be the focus. The fun and experimental focus of the OpenStomp offers some creative possibilities, but what matters in the long run is proven and playable sound effects. This is only speculations.

2.1.3 Bela and Pure Data

In an article by Moro et. al. the possibility of creating 'high-performance' embedded instruments with a combination of Bela and Pure Data is explored [10]. By 'high-performance instruments' the writers refer to the computational load on the CPU. A digital system is embedded when it is able to operate a dedicated function within the constraints of its physical boundaries. This means that for the Pure Data instruments to be embedded on the Bela, the micro controller needs to be able to run the PD patch without use of external hardware, like a PC doing the DSP. This excludes a number of micro controllers, the Arduino Uno[7] a popular MCU for creating hardware prototypes with use of different sensors, has a clock speed of 16 MHz. A typical PC has multiple CPU cores clocking in at around 2 GHz. However, the Bela has been designed with audio processing in mind and therefore been build around a 1 GHz ARM Cortex A8 processor. The increased performance of the micro controllers allow for more high level programming such as Pure Data. A trade off is that low-level programming languages certainly are more performance efficient, but have a tendency of being much more technically challenging. A generative audio patch using libpd (a standard PD library) used 53% of the Bela computational power [10]. This is a very inefficient way of running PD patches on the Bela. A tool from Enzien Audio called 'Heavy Audio Tools'[8] will generated optimized C code based on a PD patch. Running the same generative code example using the Heavy Audio Tools decreased the CPU usage to 26% [10] a 50% CPU load reduction.

[7]https://www.arduino.cc/en/main/arduinoBoardUno
[8]https://enzienaudio.com/

2.2 Virtual analog related methods and tools

In recent years, the field of analog emulations, or 'virtual analog' - both terms are used interchangeably in this writing, has received increasing attention. The available computational power at hand and affordability are two decisive factors evidently. Many of today's smartphones have more powerful CPU's than PCs had a few years ago. This development allows for emulations of more complex systems. One of the challenges of working with virtual analog is to achieve high resemblance of what is being emulated while maintaining a low latency. This is done by keeping the number of required computations to a minimum and/or by optimizing for less computationally demanding processes. It is not uncommon that analog emulations rely on complex mathematical expressions. Coming up with new emulation models can sometimes cut the required CPU power to a half [11] and is therefore a great way to improve on performance. Extensive research is carried out on improving triodes - or vacuum tubes [11, 12, 13]. Of the electrical analog components associated with guitar equipment triodes seems to take most of the attention. Triodes are used to great extent in guitar amplifiers and are loved for what they add to the sound. Some electrical components retain great complexity and form the basis of studies on their own. A common practice, however, is to isolate not just one component from a circuit, but smaller segments and analyze how they function when combined. In [14] Yeh goes into detail with some common simple circuit designs, low- and high pass filters and distortion, based on resistors, capacitors and diodes. Theses common circuit designs can be found in almost every effect pedal, and are often constructed by rearranging the three above-mentioned components - very often just a combination of resistors and capacitors if clipping is not the intention.

2.2.1 Fuzz Face emulations

The Fuzz Face and its impact on fuzz effects was mentioned in the introduction. The Fuzz Face is loved for its simplicity and great sound. Adding to that, the fact that the circuitry is well documented, makes it a popular DIY project for people wanting to get into effect pedal building. Being one of the most recognized and popular fuzz pedals also means that numerous emulated versions can be found, some of which are well documented academically. In [6] a design flow meant for integration of a physically modeled sound is presented, *see figure 2.1*. The design flow is presented with an emulation of the Fuzz Face. It follows a number of steps to get to the desired embedded emulation.

Having access to the item, which one seeks to emulate is always favorable, but not always an option. This allows for comparison, both by listening and comparing of quantitative data obtained from the original and the emulated version.

The design flow starts with a circuit model created in a SPICE tool. This is used

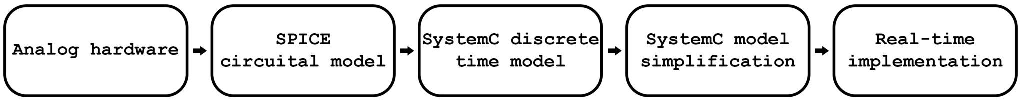


Figure 2.1: An example of a design flow for emulating physical systems with steps from analog hardware analysis to real-time implementation. *The original figure has been modified and can be found in:* [6].

to identify the filter coefficients, which is then used to create a discrete time model using SystemC-WMS, (Wave Mixed Signal Simulator) a class library in C++ meant for modeling and simulation of analog components[9]. This model is adapted to run on PC only [6]. The model is then simplified preparing for the implementation of a real-time embedded system. In [6] an ARM7 microcontroller serves as the platform for the embedded system. Interestingly, a preamplifier is used together with the ARM7 MCU, but very little is mentioned about the component. It seems that it handles, what previously has been referred to as, the input section, in this project; here the 'pick-up module'. Looking at the circuit schematic for the fuzz face, the preamplifier does not directly reflect this. It is mentioned that the pickup module and the 'filter module' are the same as in the schematic, but whether this is in the sense of analog or emulated without simplification is unfortunately not stated. The distortion section had to be simplified as the ARM7 MCU was not able to handle the more precise algorithm within a time frame allowing real-time processing while maintaining a sample rate of 44.1 kHz.

A master degree thesis in electrical engineering presents another attempt at modeling the Fuzz Face digitally [7]. Again SPICE software is used to simulate the circuit. The thesis presents extensive work carried out in an effort to get a better understanding of the components used to build the Fuzz Face. Resistors are analyzed to determine their true resistance. This is also done with six germanium transistors. Two transistors take care of most of the distortion produced by the Fuzz Face. The gain values were identified for the six transistors - values that suggested poor consistency, which is common for germanium transistors and diodes. A proper build of the Fuzz Face would have a gain matched pair of transistors [8]. This produces the desired harmonic content. This is of course a matter of subjective preference. The two most suitable pairs were selected according to the ideal gain. These pairs were used to analyze how the gain affected the circuit. It was later observed that the transistors can take up to five minutes to stabilize due to temperature changes (whether this is perceivable by the human ear was not investigated).

It was found that transistor Q1, the leftmost transistor in the circuit schematic,

[9]https://sourceforge.net/projects/systemc-wms/

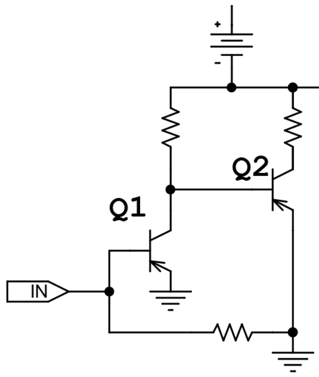


Figure 2.2: Simplified transistor circuit of the Fuzz Face intended for analysis.

see figure 2.2, greatly affects the clipping. When the gain factor was increased the hard clipping became more prominent producing higher odd-numbered harmonics resulting in a harsher sound. Q2, the rightmost transistor in the circuit schematic, showed overall decreased gain when the gain factor of the transistor was increased and vice-versa. A digital model was created using matlab. The intention was not to create a real-time implementation. Whether this approach is suitable for a real-time implementation remains unclear at this point. A number of different models were established. None were deemed perfect, but they yielded some useful insight, indicating how small changes in transistor parameters have a significant impact on the output. Some of the SPICE model values and measured values did not match and had to be corrected manually.

2.2.2 Wave Digital Filters

Only a brief introduction to Wave Digital Filters (WDF's), introduced by Fettweis [15], will be given, as it was deducted that WDF's were not well suited for this project. However, WDF's are used widely for modeling physical systems.

In the field of audio, WDF's is a digital approach to modeling of electric circuits and potentially be applied in virtual analog work. WDF's are considered to have great coefficient accuracy, dynamic range and stability [15].

The modular method is based on a number of WDF components representing electrical components. Each of the components are designed and functions as simple digital filters. WDF components are connecting through a number of 'ports'. Each port has two terminals, an input and an output. Basic components such as resistors, capacitors, and inductors are all 'one-port' elements. A transformer is an example of a two port component. Additionally all ports have a parameter of resistance, which is used to implement impedance coupling between components.

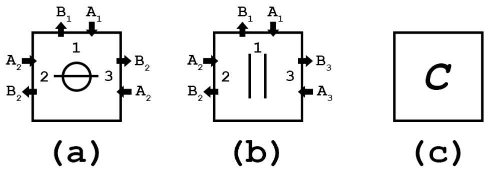


Figure 2.3: Graphical representation of three WDF components, including two adaptors and a one-port element. Three-port serial (a), and three-port parallel (b). (c) represents a one-port capacitor.

The port resistance is only used for calculation of this, and is not to be confused with electrical resistance. The modules connecting the components are referred to as adaptors and include series and parallel adaptors - *see figure 2.3*. Usually 3-way adaptors are used [16]. This approach creates a binary tree-like structure, which is based on inspection of the modeled circuit [17]. However, the method has its limitations. One of them being that the WDF binary tree structure does not handle multiple nonlinearities well [16]. However, multiple nonlinearities can be integrated by doing a global simulation of the affected subtree, which connects the nonlinearities [16]. This leads to a related problem, and a critical one for the usability in regards of this project, the WDF approach is cumbersome at realizing circuitry including loop-like structures [16, 18] and therefore does not seem like a viable approach for a highly nonlinear system such as a fuzz pedal.

2.2.3 Black, white and gray box testing

When performing an emulation of a system there are different approaches. These are sometimes predetermined by what information can be obtained about the system. Some companies share complete circuit diagrams for anyone interested in copying the product themselves. This can be very helpful when doing analog emulations. Other companies are very protective of their products. The company behind the sought after (and expensive) 'Klon Centaur' overdrive pedal is notoriously known for covering the circuit of the pedal in black epoxy to protect the layout of the circuitry from curious 'cloners'.
The *black box* approach requires access to the system for use and analysis. Dealing with an analog effect pedal common practice is to excite the pedal with different test signals [14]. Practical signals can be signals covering the frequency spectrum of interest, either via noise or a sine wave sweep. Another practice is to excite the pedal with a signal with a fixed frequency and inspect the composition of the overtones [19]. This procedure is repeated a number of times with different settings for single parameter (knobs and switches e.g.). The output signal is captured digitally for analysis. Based on the analysis appropriate filters can be designed to create a fitting model.

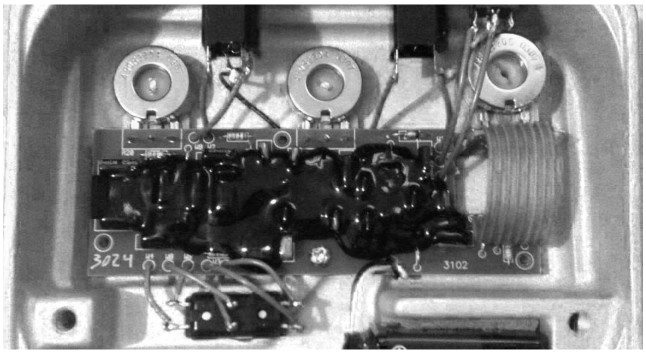


Figure 2.4: Picture showing the black epoxy - a measure taken to prevent 'cloners' from copying the effect pedal circuit. *Image source: https://soundvacuum.wordpress.com/*

The *white box* approach relies on a deeper insight of how a device is actually put together. Unlike the black box approach, the white box approach calls for opening up the piece of hardware in question to investigate how it has been build. This knowledge can also come from reliable circuit diagrams. The design of the circuit and the individual components are mapped digitally according to accurate mathematical models. SPICE software (Simulation Program with Integrated Circuit Emphasis) can be used to investigate behavior of circuit segments and are used extensively in the field of digital audio emulation. In [20] Paiva et. al use SPICE software to analyze new WDF models of op-amp circuits including diodes. By comparing the emulated signal with a SPICE simulation a rapid indication of accuracy can be established. A more time consuming alternative would be to build the circuit the emulation is based upon and compare the performance of those.

A *gray box* approach, which, as the name suggest, is a combination of the two approaches described above, can be conducted with some degree of insight of the circuit design. This approach is inclined to involve additional filters when compared to the black box approach. The insight of the circuitry can be used to devise and arrange filters according to segments of circuit components. As it was the case with the white box approach, SPICE software can prove very useful with the gray box approach.

To summarize, the black box approach does not require knowledge of how a given item works. It relies on skills at transforming data obtained from analyzing how the item reacts when manipulated, e.g. feeding a test signal to an effect pedal, into a system emulating the behavior. The white box approach relies on technical information about how the item has been constructed and how it works. This information is used to emulate the item. The gray box approach is a combination of the two.

2.3 Fourier series

In the examination of the Pharaoh pedal in chapter 1.1, the concept of clipping was introduced. The following will elaborate on clipping and what it adds to sound by relating it to the theory behind Fourier series. The theory in this section is mainly from [21].

Fourier series can be used to expand a piecewise continuous periodic function, ***f(x)***, and express said function by an infinite sum of sines and cosines. Fourier series can be used to split an arbitrary periodic function into a group of simple terms, which can be solved individually. The sum of the solved terms is a solution to the problem, where the accuracy depends on the number of terms included.

For $\boldsymbol{f : R \to R}$, with period $\boldsymbol{T} = \frac{2\pi}{\omega}$, we have:

$$f(t) = \frac{1}{2}a_0 + \sum_{n=1}^{\infty} (a_n \cos n\omega t + b_n \sin n\omega t) \ . \tag{2.1}$$

For sufficient values of $\boldsymbol{N}$ it can be assumed that:

$$f(t) \sim \frac{1}{2}a_0 + \sum_{n=1}^{N} (a_n \cos n\omega t + b_n \sin n\omega t) \ , \tag{2.2}$$

where a_n and b_n are called Fourier coefficients, and are given by:

$$a_n = \frac{2}{T} \int_0^T f(t) \cos n\omega t \ \ dt \ , \tag{2.3}$$

and

$$b_n = \frac{2}{T} \int_0^T f(t) \sin n\omega t \ \ dt \ . \tag{2.4}$$

where $\frac{1}{2}a_0$ is the mean value (DC term), and is given by:

$$\frac{1}{2}a_0 = \frac{1}{T} \int_0^T f(t) \ \ dt \ . \tag{2.5}$$

The n-th term in a Fourier series can be expressed by:

$$f_n(t) = a_n \cos n\omega t + b_n \sin n\omega t \ . \tag{2.6}$$

This can be rewritten as:

$$f_n(t) = A_n \cos(n\omega t + \varphi_n) \ , \tag{2.7}$$

where φ is the phase and $\boldsymbol{A}$ is the amplitude. $\boldsymbol{A}$ can be expressed by:

$$A_n = \sqrt{{a_n}^2 + {b_n}^2} \ . \tag{2.8}$$

f_n(t) is called the 'n-th' harmonic for ***f(t)***. The power of the n-th harmonic is

$$\frac{1}{2}{A_n}^2 = \frac{1}{2}({a_n}^2 + {b_n}^2) \ . \tag{2.9}$$

The Fourier series for common waveforms, such as: square- , sawtooth- and triangle waves etc. can be derived from *eq. 2.2 - 2.4*.

2.3.1 Example 1 - The square wave

The square wave, approximates signals in fuzz pedals with pronounced hard clipping. This will become apparent in the next chapter. The following derives the

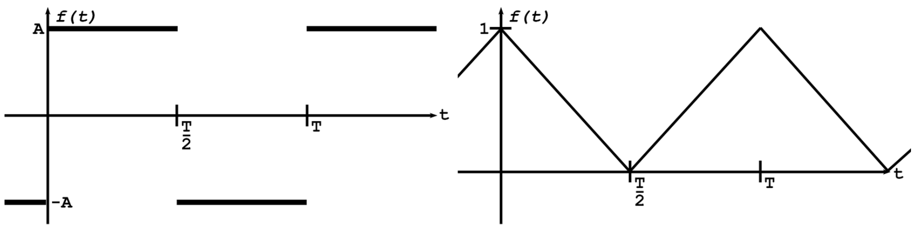


Figure 2.5: Sketches of wave functions. Left: Square wave function. Right: Triangle wave function.

Fourier series for a square wave, with period $\boldsymbol{T} = \dfrac{2\pi}{\omega}$ and amplitude $\boldsymbol{A}$, as shown in *figure 2.5* for a sketch of the square wave function (and triangle). We have:

$$a_n = \frac{2}{T}\int_0^T f(t)\cos n\omega t \ \ dt \quad = \quad \frac{2}{T}\int_{-\frac{T}{2}}^{\frac{T}{2}} f(t)\cos n\omega t \ \ dt = 0 \ , \tag{2.10}$$

because $\boldsymbol{f}$, and therefore the integrand, is odd.

$$b_n = \frac{2}{T}\int_{-\frac{T}{2}}^{\frac{T}{2}} f(t)\sin n\omega t \ \ dt \quad = \quad \frac{4}{T}\int_0^{\frac{T}{2}} f(t)\sin n\omega t \ \ dt \ , \tag{2.11}$$

because the integrand is even. Further derived:

$$b_n = \frac{4A}{T}\int_0^{\frac{T}{2}} \sin n\omega t \ \ dt \quad = \quad \frac{4A}{T}\frac{1}{n\omega}\Big[-\cos n\omega t\Big]_0^{\frac{T}{2}} \tag{2.12}$$

$$= \frac{4A}{2\pi n}\Big(1-\cos n\pi\Big) = \begin{cases} 0 & \text{for } n \text{ even} \\ \dfrac{4A}{\pi n} & \text{for } n \text{ odd.} \end{cases} \tag{2.13}$$

The Fourier series then becomes, from *eq. 2.2*:

$$f(t) \sim \frac{4A}{\pi}\Big(sin\omega t + \frac{1}{3}sin3\omega t + \frac{1}{5}sin5\omega t + \ldots\Big)\ . \tag{2.14}$$

The Fourier series, and the influence of 'N-included' harmonics can be visualized graphically. *Eq. 2.14* was applied in matlab to produce the following plot:

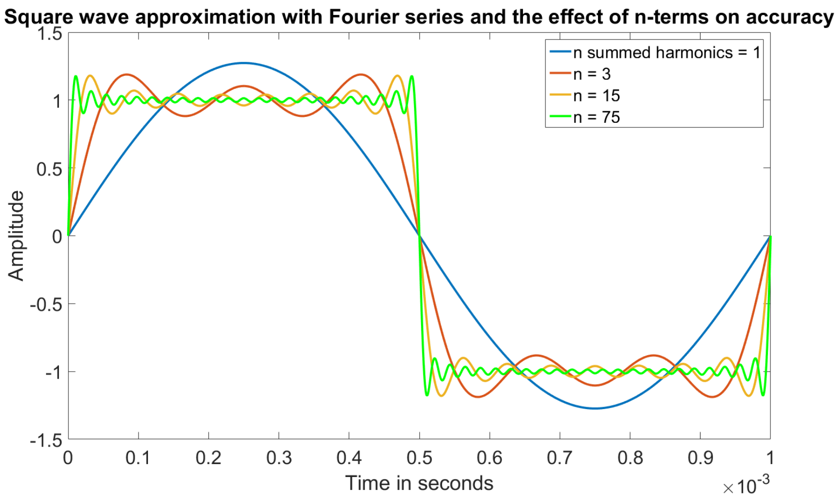


Figure 2.6: Matlab plot of how the number of 'N-included' harmonics affect the accuracy of the Fourier series of a square wave.

It can be seen in *figure 2.6* that it takes several harmonics to accurately approximate a square wave.

2.3.2 Example 2 - The triangle wave

Using the exact same procedure, the Fourier series for a triangle wave can be found. The triangle wave is defined in *figure 2.5*, here we get:

$$f(t) = \frac{1}{2} + \frac{4}{\pi^2}\left(\frac{1}{1^2}cos\omega t + \frac{1}{3^2}cos3\omega t + \frac{1}{5^2}cos5\omega t + \dots\right). \tag{2.15}$$

Like before the eq. (2.15) can be visualized using matlab. This is shown in *figure 2.7*.

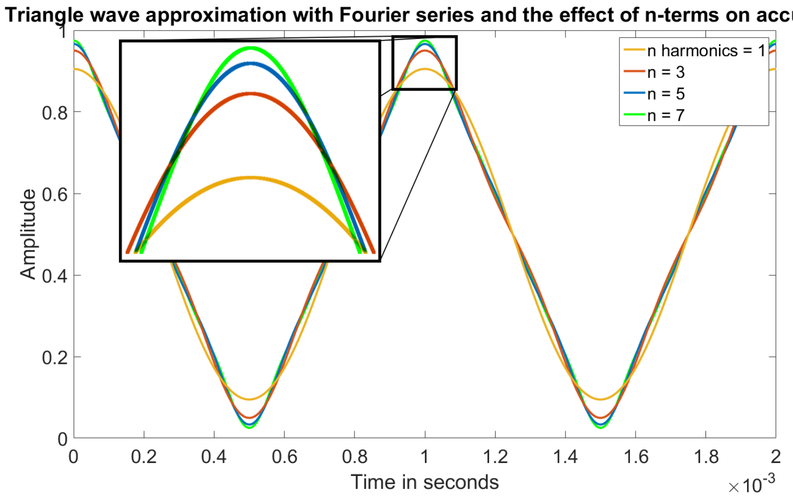


Figure 2.7: Matlab plot of how the number of 'N-included' harmonics affect the accuracy of the Fourier series of a triangle wave.

From *figure 2.7* it can be seen that it does not take a lot of harmonics to approximate a triangle wave with the Fourier series. Compared to the square wave, the triangle wave has a much lower content of harmonics.

By producing square wave signals with high content of harmonics, fuzz pedals are capable of producing rich saturated sound.

2.4 Choice of tools and methods

This section is a discussion based on what has been described in the previous chapter.

A number of the described physical platforms were intended for the user to keep developing and sharing work; work such as sound effects and analog emulations of varying complexity. Examples were given with the Open Stomp, the MOD Duo and the OWL. Knowledge of C++ programming and DSP is required to be able to develop new content for the OWL. The company behind the product has received feedback concerning this, for some, obstacle; *"[...] feedback from potential users of the OWL suggests that ideally they would like the API to be made easier* *still."* [9]. This is thought to be important for a platform where the intention is to enable the user to create and share work. One could speculate that platforms, such as the OWL, which requires a high learning effort, might limit the amount of work being shared, but in the process, ensuring a better overall quality. The OWL project seems successful and is now supported by a number of professional 3rd party developers.

The way the Pharaoh Fuzz is constructed makes it complex to emulate. It contains multiple clipping loops with different diode types, some of which are arranged asymmetrically. Wave digital filters were found to be less ideal for circuitry including multiple nonlinearities or loop-structures. The Pharaoh Fuzz does contain both. Further adding to that, in [18], the following was stated: *"For a standalone simulation of nonlinear circuits, [WDFs] may not be the best choice"*. The Pharaoh Fuzz does inherent nonlinear behavior (depends on its setting), in particular from the use of transistors and clipping diodes, used exactly for this reason - the nonlinear behavior causes the fuzz. WDFs are neither thought to help simplify the process of creating effects according to what was addressed in the above. Therefore WDFs are not considered to have any application value in the scope of this project.

In [6] it was found that running a Fuzz Face emulation algorithm on a ARM7 MCU posed a challenge as the intention was to do so in real-time. The ARM7 has a maximum clock speed of 60 MHz. Much faster CPU are found in many MCUs. Getting a faster CPU will open up for the possibility of running more complex algorithm and also put less stress on the performance efficiency. The intention in this project is also a real-time application. An example of a faster MCU was presented with the Bela [10], with a clock speed of 1 GHz it might be a good choice for the implementation of the fuzz emulation of this project.

Lastly, in [7] some interesting findings were presented: Inconsistency between SPICE simulations and lab measurements were experienced, which should be considered when using SPICE simulations. SPICE simulations are used to great extent for effect pedal emulations, but the findings suggest that the simulations will get you close but might not provide you with a perfect emulation model. The reason for this, among others, can be linked to inconsistency associated with some of the analog components used in many fuzz effect pedals and other circuits. Especially the germanium transistors were found to vary a lot when measuring their performance. Additional inconsistency connected to temperature changes can be added. The findings associated with germanium transistor inconsistency are by no means new knowledge. However, it stresses the importance of additional analysis, comparison, if possible, to the real pedal, both analytically and by listening.

This is important for certain pedals, which rely on these inconsistent transistors. There is going to be differences between pedals regardless of being the same maker and model. Some differences might only come through by the numbers, but some will come through as perceptible differences. This is also one of the incentives to create an analog emulation. A digital version will always sound the same regardless of temperature change. Some will argue that the unique behavior of analog equipment is part of the charm. Having to buy multiple examples of the same model to find the perfect sound or the one with minimal noise floor could potentially be eliminated with an analog emulation.

Chapter 3

Analysis

As mentioned in a previous chapter, the implementation of a virtual analog version of the Black Arts Toneworks Pharaoh fuzz effect pedal is going to be based on a combination of black box and white box analysis - or as coined: a 'gray box' analysis. 180 signals of open notes were recorded. An overview of the recordings can be found in *figure* 3.1. Additionally ten white noise recordings were used to excite the pedal and capture frequency response, meant for filter design. More on the implementation in chapter 4. Access to the complete circuit diagram (*Appendix*

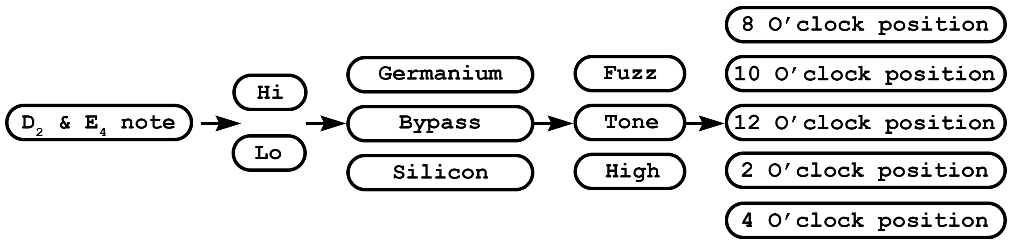


Figure 3.1: Overview of the recorded analysis signals. 90 signals were recordings of open D_2 notes and the same number of E_4 notes. The recordings were sampled in the 'Hi' and 'Lo' configuration for all three clipping settings with the 'Fuzz', 'Tone' and 'High' knobs in five different positions.

A) is a great source of information. It can reveal which parts of the circuit can be isolated by manipulating the knobs and switches to get proper recordings for analysis. It can also tell which of the parts are too merged and therefore better analyzed by SPICE simulation rather than using recordings.

An example of this can be given with the two clipping stages. The first clipping stage can not be manipulated with any knobs or switches directly as it is the case with the second clipping stage, which can be manipulated by switching between clipping diodes or be bypassed. The bypass option was utilized to obtain more isolated recordings, which will be explained further in the following subsections. Lastly, perhaps needless to remark, listening carefully while manipulating the pedal can in itself provide clues on how things are connected, how the parameters affect each other and ease reading a circuit diagram, if accessible.

3.1 Apparatus

A selection of hardware and software was used for the analysis of the Pharaoh fuzz pedal. The following is the most essential gear and software.

- Black Arts Toneworks Pharaoh fuzz effect pedal. The pedal is described in detail in section 1.1.
- Harley Benton SC-Custom Les Paul style guitar with humbucker pickups.
- Voodoo Lab Pedal Power 2 Plus. A cheap power supply for the pedal was initially used but it caused some very undesirable high pitched noise. This flaw could potentially have caused poor recordings and affect the accuracy of the emulation. The power supply was meant for guitar use, built with audio filters accordingly, but was flawed nonetheless. The Voodoo lab power supply eliminated the issue.
- Behringer U-Phoria UM2 48kHz Audio Interface. This item was used to convert the analog signals from the guitar to digital samples for recording. The interface has a maximum sampling rate of 48kHz.
- BeagleBone Black micro-controller + Bela cape. This combination was also used to produce white noise to excite the pedal for analysis of filtering characteristics.
- HP 971A multimeter for measuring signal strengths.
- Bugera G5 class-A tube head amplifier connected to an Orange 1x12" cabinet for listening.
- Ableton Live 9 Lite DAW. This digital audio workstation was used for recording guitar signals.
- Pure Data Extended. With Pure Data running on the BeagleBone-Bela combination, test signals can easily be produced and fed to the pedal. Pure Data has a object called 'noise' which generates white noise, which was used to obtain filtering characteristics.
- Adobe Audition CS6 was used for playback and normalization of the recordings.

3.1.1 Setup

As mentioned in the previous section, the BeagleBone and Bela combination with Pure Data running on it was found convenient for analysis purposes of the Pharaoh fuzz pedal. *Figure 3.2* shows how the components of the recording setup and the excitation setup was connected.

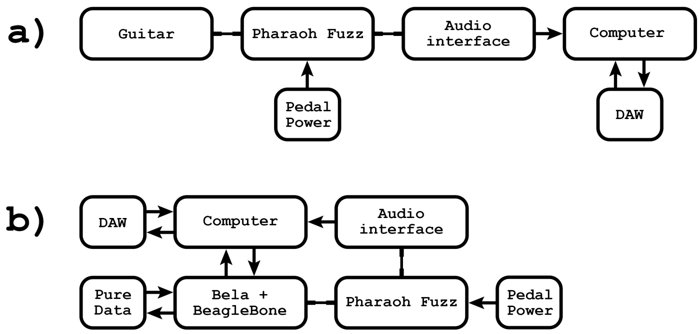


Figure 3.2: Setups used for recording analysis signals. a) Setup used to record notes played on various pedal configurations. b) Setup used to generate white noise, which was fed to the pedal to get frequency responses for various pedal configurations.

3.2 Input segment

The input segment has two components, which can be manipulated with the controls of the effect pedal; a 'Hi-Lo' switch and the 'Fuzz' knob. These will be investigated in the following.

3.2.1 Hi/Lo configuration

When switching back and forth between the two settings, 'Hi' and 'Lo', in the input segment, an easily audible difference is present. At the *bigmuffpage*, where the circuit of the Pharaoh fuzz was found (*figure 1.3*), the switch is referred to as the 'gain switch'. The switch selects between two resistors. It is written that the switch is affecting the low-cut filtering in the input stage. It was decided to investigate this using SPICE software - OrCAD Capture[1] to obtain more detailed data on how the sound is affected. The circuit design was structured according to the *bigmuffpage* circuit. The input segment can be seen in *figure 3.3*, see *Appendix B* for the full circuit schematic. Note that R7 was set to 1 Ohm during the simulation. A simulation for each of the two resistors were conducted with an AC sweep from 20 Hz to 22

[1]http://www.orcad.com/products/orcad-capture/overview

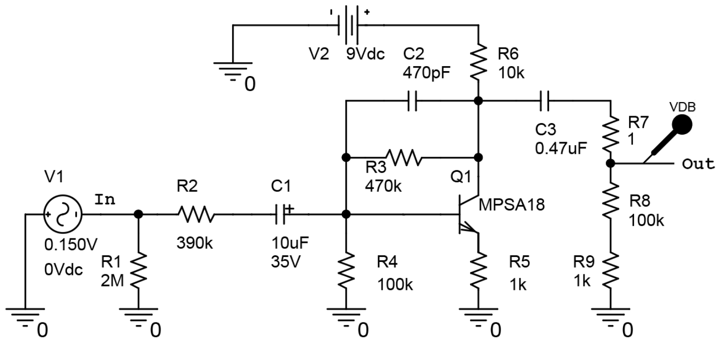


Figure 3.3: The input segment of the Pharaoh fuzz. The R2 resistor is the 'Lp' setting of the 'Hi-Lo' switch. R7 is the potentiometer connected to the 'Fuzz' knob, here set at minimum resistance. Probing for the filtering characteristics was performed at the output of the input segment.

kHz. The resulting frequency response graphs can be seen in *figure 3.4*. Noting the

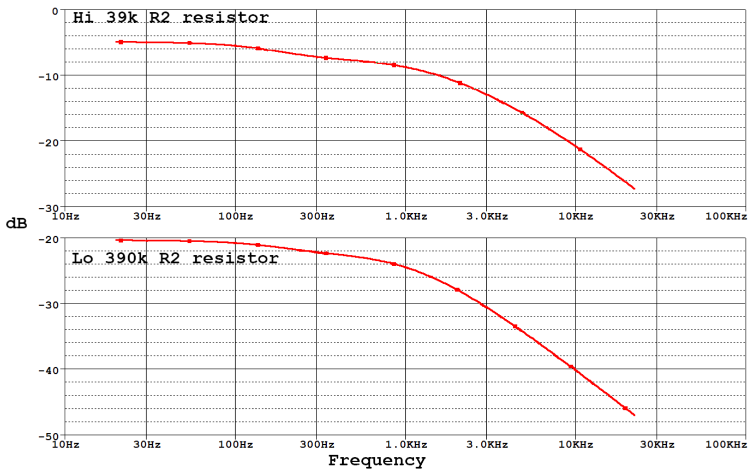


Figure 3.4: SPICE simulation of the input segment. The 'Fuzz' knob potentiometer (R7 in *figure 3.3*) is set to low resistance at 1 Ohm (high signal strength). The top graph is the 'Hi' setting, bottom is the 'Lo' setting. Note the different intervals along the y-axes.

different intervals on the y-axis makes it clear that a large drop in signal strength is happening when switching from 'Hi' to 'Lo'. Using the SPICE analysis tools the difference of the two graphs were found to be approximately 15dB. A quick note on volume; this will be disregarded for now; why, will be explained in **??**. From the graphs it can be seen that switching from 'Hi' to 'Lo' shifts the signal strength rather than changing filtering characteristics. The loss of treble is, however, thought to be connected to the lower signal strength and thereby change in harmonic content. It is important to remember that from here the signal will be amplified by the transistors and clipped by the transistors and diodes. This will produce harmonics and saturate the signal. The lower signal strength of the 'Lo' side will therefor be less compressed and saturated - preserving more of the original input signal. The 'Tone' and 'High' filters located in the 'Tonestack', *see Appendix A*, are closer to the output and are thought to be more defining for the tonality of the sound, as the signal will only pass through a single transistor from the 'Tonestack'. Whereas the 'Hi-Lo' switch will have a great impact on the harmonic content - the saturation of the signal. In that aspect the 'Fuzz' knob is quite similar to the 'Hi-Lo' switch as this is another resistor to adjust the signal strength, and thereby the clipping- and saturation level induced by the remaining part of the circuit. Proof of this can be found by inspecting recordings in appropriate software. *Figure 3.5* illustrates the

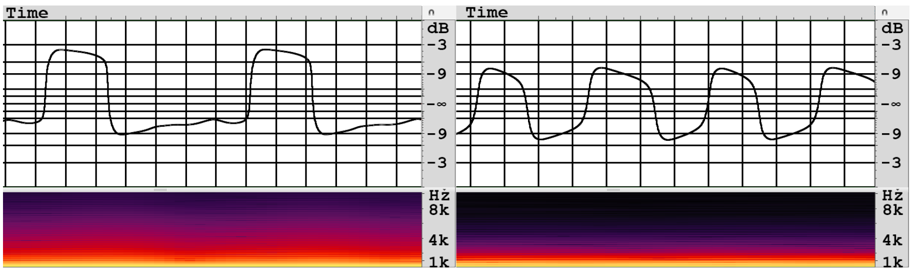


Figure 3.5: Two waveforms and spectrograms from recordings of D-notes played open at the same settings, all knobs at 12 O'Clock, with silicon diodes at about the same instance after the attack: 2 seconds in. The depicted signal in the left window was recorded with the 'Hi' configuration. The right was recorded with the 'Lo' configuration.

waveform of two different recordings, around the same instance, all settings are identical apart from the right recording being recorded at the 'Hi' setting and the left at 'Lo'. Taking a look at the dB scales reveals a signal strength drop but nothing like the simulated results, which makes sense as no transistor gain was included in the simulated results. The 'Lo' signal is less altered and bears a little more resemblance of a sine wave than the 'Hi' signal does. Both signals have similarities with a square wave. This is more pronounced with the 'Hi' signal. Taking a look at the spectrogram at the bottom reveals that the more square wave-like signal has a richer content of overtones.

3.2.2 Fuzz

The 'Fuzz' knob is also located in the input segment. Its functionality is very similar to that of the 'Hi-Lo' switch, but with added fine tuning of a variable resistor. The potentiometer can be used to in- or decrease the signal strength within an interval. This will affect the degree of clipping in the following clipping segments.

3.3 First clipping segment

The first clipping segment can not be manipulated directly with any of the external control knobs or switches. However, by selecting the 'Bypass' option of the second clipping segment, that clipping segment is skipped as the name implies. This setting will output a signal produced by the first clipping segment, the transistors and the tonestack combined.

3.3.1 Bypass

The 'Bypass' option is located in the second clipping segment, but as explained in the prior section, this setting is the best possible way to isolate the signal from the first clipping segment. The first clipping segment comprises a bipolar transistors (2N5089 NPN) in common emitter configuration and a diode clipper configuration based on two 1N914 silicon diodes. From simulation of the SPICE circuit layout (see *figure 3.6* for the first clipping segment), it was found that the diodes receive the same bias from the 9 volt DC source. The result is symmetric clipping of the

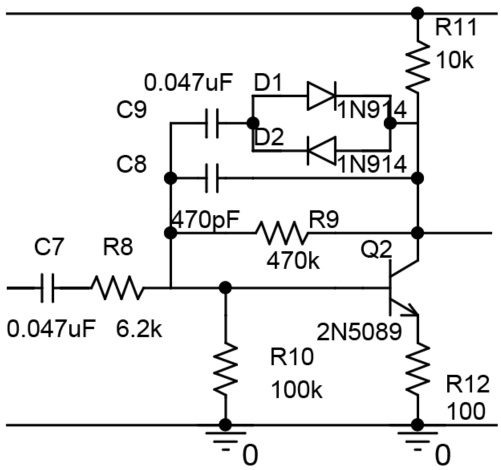


Figure 3.6: Section of the SPICE modeled first clipping segment. The top input is connected to a 9V DC source.

positive and negative wave cycle. An example of this can be seen in *figure 3.7*. The waveform from the recording is a product of the 1N914 diode clipper in the first clipping segment, the four transistors and the filtering of the pedal. When inspecting the progression of the waveform as the voltage decreases over time, the signal generated by the initial strum, the amount of clipping follows and fades. Eventually, approximately 30 seconds after the strum, the signal is no longer being clipped. The progression of the waveform was recorded with the control knobs set to 12 O'clock and the Hi-Lo switch at 'Hi'.

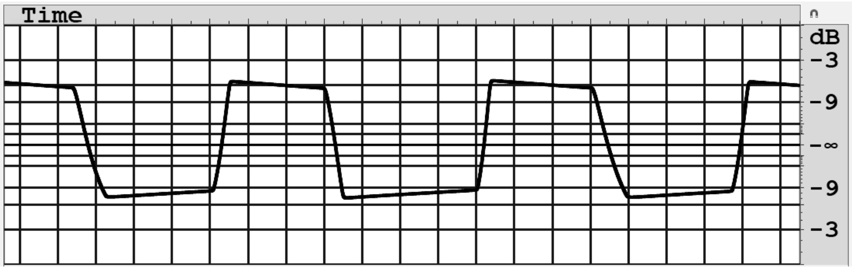


Figure 3.7: Recording showing the waveform of an open D_2 note with all knobs set to 12 O'clock, 'Hi', and second clipping segment bypassed. The illustrated section is approximately 1.5 sec. after the strum.

Unlike diodes, transistors are active and function as amplifiers. With a multimeter the signal strength of an electric guitar can be determined. For the open D_2, which produces the greatest signal strength of all the strings due to greater mass, with both pickups active, the signal strength was found to peak just below 130mV, measured my connecting the multimeter test leads to the guitar cable plug. Silicon diodes have a voltage drop, or forward voltage, more than double of that of germanium diodes (0.6-0.7 V as opposed to 0.2-0.4 V [3]). The unamplified guitar signal does not have the signal strength to pass the diodes. The amplification by the transistors resolve this. Using silicon diodes result in less clipping and therefore outputs more volume, as it takes greater signal strength for the diodes to turn on.

It is important to understand that the signal through the pedal is amplified and clipped multiple times, which causes the saturation of the signal. The gain of the transistors will be discussed in implementation chapter.

3.4 Second clipping segment

The second clipping stage is build around the same diode clipper feedback loop as the first clipping segment, but with the added option for the user to switch between germanium or silicon diodes or bypass these. The bypass option was analyzed in the previous section.

3.4.1 Germanium

An unbiased germanium diode will start clipping at around 0.3 volts resulting in approximately half the signal strength compared to silicon diodes. The different forward voltage ratings can be applied as voltage controlled switches with varying sensitivities. The germanium diode is known for having a transient activation phase, which is softer compared to that of silicon diodes, see *figure 3.8*. This is

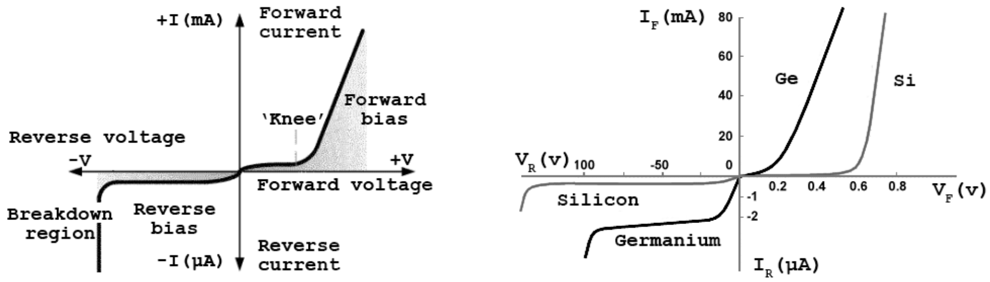


Figure 3.8: Left: Illustration of how diodes behave at different voltage and current levels. The reverse bias region is not interesting for the present application. In this region the diode is simply considered as a disconnection. The bend when the diode becomes forward biased is here referred to as the 'knee'. Right: The different behavior of germanium and silicon diodes. The knee is softer on the germanium diode when activated. *Original image sources left: http://www.gopracticals.com, Right: [22].*

considered an imperfection in many fields of electronics, which is why the silicon diode is often considered superior to the germanium. For some audio applications, such as effect pedals, this imperfection is cherished and exploited. Germanium diodes are often described as having a warmer, rounded tone compared to the harsher silicon diode. The clipping performed by the two types of diode can be seen in *figure 3.9*. In the second clipping segment the germanium diode clipper has

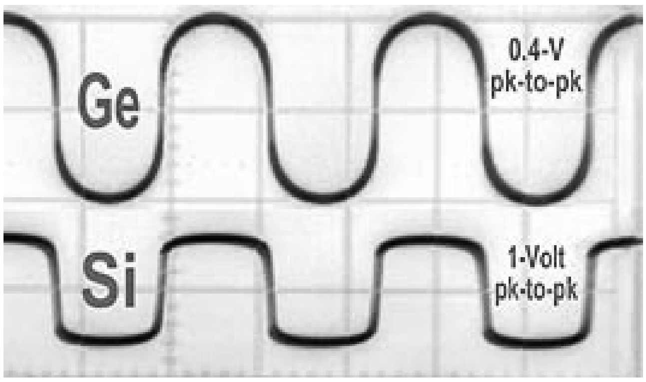


Figure 3.9: The different clipping characteristics of the softer clipping germanium (top) hand the harder clipping silicon (bottom) as seen on an oscilloscope.*Image source: http://www.mixonline.com/news/profiles/guitar-distortion/365765.*

been arranged to produce asymmetrical clipping. When the diodes are arranged in series the forward voltage are added together. In the Pharaoh pedal, three 1N34A germanium diodes are arranged in parallel, two of which are in series, oriented in the opposite direction of the last diode, see *figure 1.4*. The forward voltage of the series arrangement is therefore 2 times 0.3 volts = 0.6 forward voltage (approximately). This means that the positive half cycle of the waveform will be clipped at approximately half the forward voltage (0.3 V) of the negative wave cycle (0.6 V). The result can be seen in *figure 3.10*.

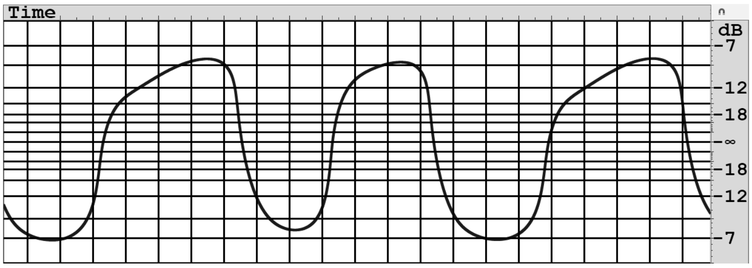


Figure 3.10: Example of the asymmetrical germanium diode clipping. The recording is of an open D_2 played with the 'Hi' setting and all knobs set to 12 O'clock. The positive half-wave cycle is more aggressively clipped than the negative wave cycle.

3.4.2 Silicon

The last option is a silicon diode clipper much like the one found in the first clipping segment, but with 1N4001 diodes instead of 1N914. Much of the behavior of diodes in general and that of the silicon type, has been described in the prior sections. The two silicon diodes have different 'knee' properties. The 1N914 is a fast switch type diode, but with the addition of a gold doped junction. When browsing for experience with these diodes in audio applications at various online communities, the claim is that the 1N914 supposedly has a softer 'knee' and therefore the clipping a little softer than that of the 1N4001.

Listening to the audio produced by the 1N914's at the bypass setting and the combination of the 1N914's and 1N4001's from the two clipping segments at the 'Si' setting results in the same impression. The 'Si' setting sounds harsher. Whether this is caused by the diode type or the additional clipping of the second clipping segment is unclear. When running a very simple SPICE simulation of the diode clippers with a 10k resistor load and a AC voltage source, the result is ambiguous. See *figure 3.11* for the SPICE output. The 1N4001 starts to clip a little earlier than the 1N914, but the knees (not to be confused with the diode knee) are very alike. If any difference had to be pointed out, the rounding of the 1N4001 waveform might be slightly softer, contrary to what was stated by a few effect pedal DIY'ers online. The harsher sound seems more likely to be caused by cascading the two clipping segments along with the transistors.

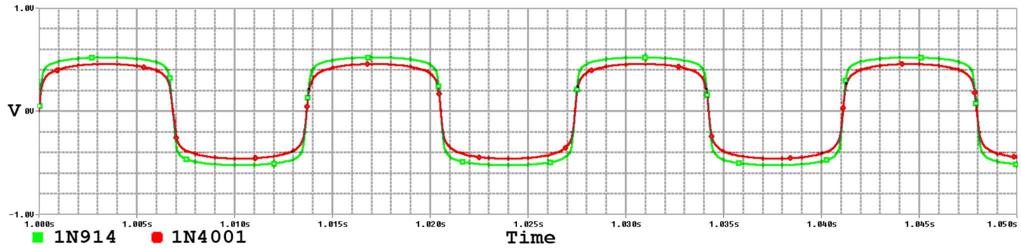


Figure 3.11: SPICE simulated waveform of the 1N914 silicon diode of the first clipping segment and the 1N4001 silicon diode of the second clipping segment.

3.5 Tonestack

The two filters of the tonestack - 'Tone' and 'High' *see Appendix A* were analyzed by exciting the pedal with white noise generated with Pure Data on the bela MCU. The reason why the bela was used for this was simply due to easy connection of guitar cables to the pedal. The 'Bypass' option was selected to skip the clipping diodes from the second clipping stage. The 'Fuzz' knob was set, which minimizes the signal strength to avoid clipping from the diodes in the first clipping stage and minimize saturation from the transistors. Playing the guitar in this configuration produces a very clean sound with no audible resemblance of additional saturation or distortion. The 'Hi-Lo' switch was set to 'Hi'.

3.5.1 'Tone' knob investigation

Five filtered white noise recordings of about eight seconds each were sampled using the behringer audio interface and recorded in Ableton. The 'High' knob was set to minimum as the 'Tone' knob seemed much more reactive in this setting. This gave the implication that the signal was cleanest, or richest, ergo less filtered with the 'High' set to minimum. The recordings were plotted in Matlab and assessed visually, see *figure 3.12* for the plot. It can be seen how the 'Tone' knob is able to let through ranges of frequencies. The 8 o'clock bass-heavy setting lets through 9dB more of the lowest frequency content compared to the treble-heavy 4 o'clock setting. The 'Tone' knob shows an inversely proportional connection between the bass and treble when manipulated. The frequencies around 10kHz are therefore 24dB lower in the 8 o'clock setting. The knob can be used to boost a specific range; like the bass-, mids- or treble-content. A dynamic filter will be designed accordingly with the plot in *figure 3.12*, more on this in *chapter* 4.

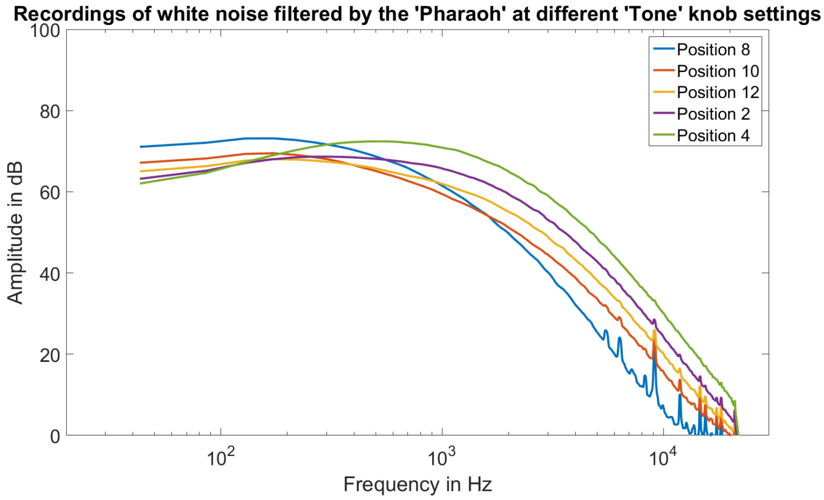


Figure 3.12: Frequency response of the five recordings from the different 'Tone' knob positions.

3.5.2 'High' knob investigation

The 'High' knob recordings were recorded in the same manner as to what has been described about the process of the 'Tone' recordings. The 'Tone' was set to minimum. This was done as the 'High' knob has little effect when the 'Tone' knob is set to boost the high frequencies content. The purpose of the 'High' knob is to boost, to some extent, the mids (easily audible) and to great extent the treble when the 'Tone' is in a more bass-heavy setting to avoid a muddy, flat or woolly sound. In *figure 3.13* it can also be seen that the filter actually boost the entire frequency range, but with much more effect around 1kHz and above because of different filter characteristics. The two knobs combined can be used to scoop the mids and boost the bass with the 'Tone' knob, and boost the treble content with the 'High' knob. This is done extensively in metal genres; Metallica guitar play is a great example of this. A dynamic filter will be designed according to the plot in *figure 3.13*, more on this in *chapter* 4.

3.5.3 RC-circuits

RC circuits, or Resistor-Capacitor circuits, are very common in analog effect pedals (and other applications, for that matter), and can be used to create simple one-pole low-pass and high-pass filters. *Figure 3.14* shows the component configuration. The cutoff frequency of the RC circuits is given by:

$$f_c = \frac{1}{2\pi RC} \quad . \tag{3.1}$$

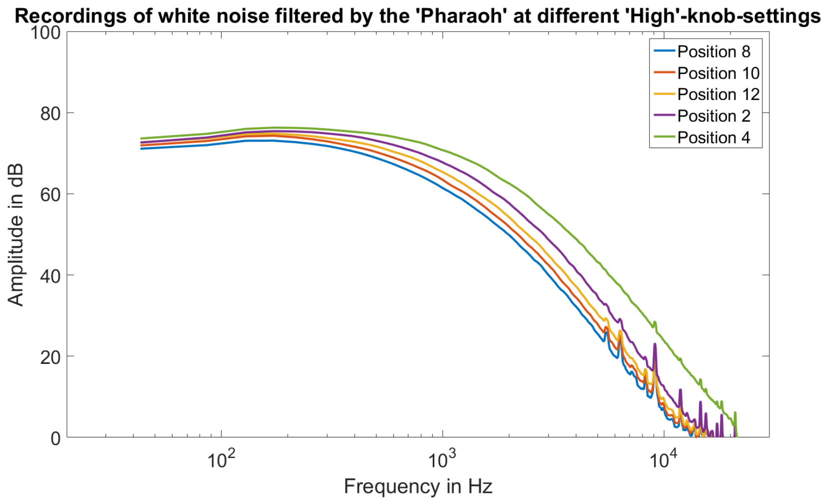


Figure 3.13: Frequency response of the five recordings from the different 'High' knob positions.

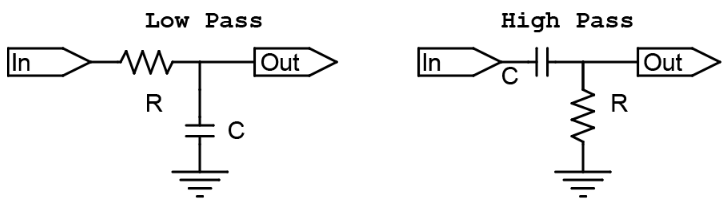


Figure 3.14: The two RC circuit configurations for the first order low-pass filter (left), and first order high-pass filter (right).

The configurations can be found several places in the circuit schematic (see *Appendix A*). The tonestack is build around RC circuit filters, the 'High' filter is a RC low-pass filter e.g. (component R8 and C8 in the tonestack, see *Appendix A*).

3.6 Output

The output segment is the simplest of the five segments, *see Appendix A*. Of the accessible exterior controls, only the volume knob is connected to this segment. The volume control is located at the output of the pedal. The signal is not manipulated further after this point. It controls the signal strength, which is sent to an amplifier from here.

The task now is to attempt to use the obtained data from the analysis (if not all of it, parts of it), to implement a virtual analog version of the Pharaoh fuzz in Pure Data. The task is described in the next chapter *Implementation*.

Chapter 4

Implementation

The implementation of the fuzz pedal was performed in Pure Data Extended. However, not all objects of PD are supported when emulating the script on the BeagleBone and bela. This will be elaborated on in the discussion. The full list of supported objects can by found via this link [1]. The decision behind this approach has been described in chapter 2. To recap; the approach is foremost an attempt at making analog emulations more approachable and in the process investigating to what extend virtual analog can in fact be accomplished with this software.

The implementation is primarily based on the analysis recordings, the black box approach, but with the added knowledge from the circuit diagram, SPICE simulations and how the individual components act under isolated conditions to a lesser extend. The implementation follows the signal path and will be described accordingly. The complete layout of the patch can be seen in *Appendix C*.

4.1 Input and first clipping segment

When using the bela; connecting with the digital and analog I/O (input/output) ports are fairly straight forward. The pure data object 'adc~2' (analog-to-digital), as used, will connect to an analog-in connection, which on the board has a male socket with three pins, stereo and ground. The socket is useful for connecting adaptors such as mini jack or 1/4" jack for instance. The guitar signal is mono, which is why only one of the stereo channels is used. The 'dac~' object (digital-to-analog) is used to reconvert the signal and direct it to the bela's analog-out connection. The digital connections of the bela need to be instantiated as either in or out, as they can be both. The message object with '11' written in it and connected to a 'send' object (s for send, r to receive) with the text 's bela_setDigital' will take care of this. The digital input can now be accessed with a 'receive' object with the text 'r

[1] https://enzienaudio.com/docs/pdobjects.html

belaDigitalIn11'. This will connect to bela's digital pin 0. A patch by forum[2] user 'LandonPD' was used as the base for the implementation. The patch included the

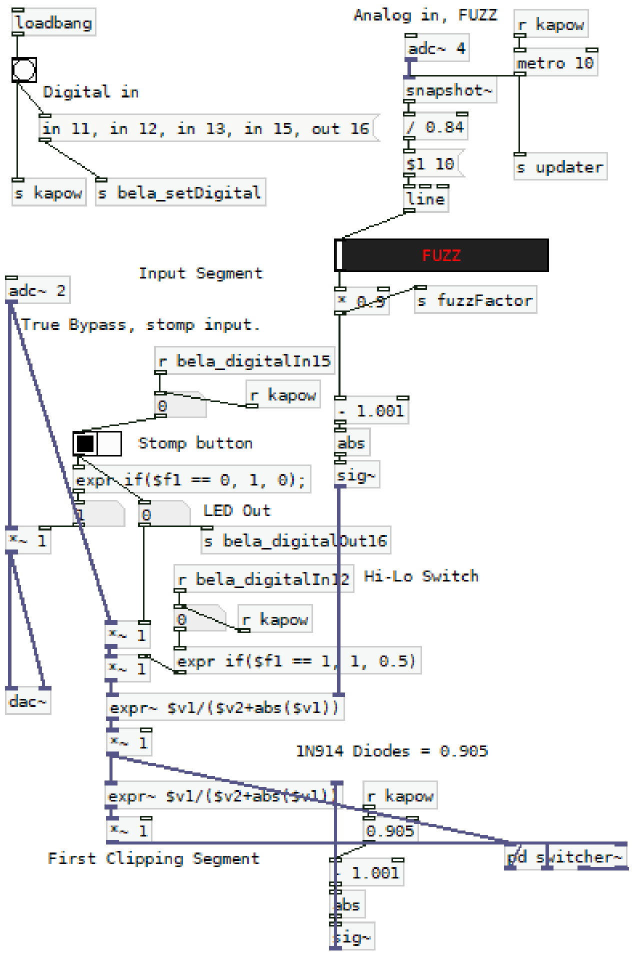


Figure 4.1: The input and first clipping segment of the fuzz pedal as it was implemented in Pure Data. Also included in this part of the patch are examples of how to deal with analog and digital I/O communication with the bela cape.

following transfer function;

$$\frac{V1}{V2+|V1|} \quad , \tag{4.1}$$

where *V1* is the original signal and *V2* is a factor in the range *0 < x < 1*. Throughout the implementation a sine-wave was used as a test signal. When $V2 \to 0$ the sine wave is distorted and eventually takes on the shape of a square wave. This

[2] https://forum.pdpatchrepo.info/

waveform is convenient, considering what was found in the *Analysis* chapter on how diodes clip AC signals. Additionally, the patch included a graphical feedback of the waveform, which was kept for quick referencing. The first time *eq. 4.1* is used is shortly after the input of the signal. It is connected to the 'FUZZ' slider showed in *figure 4.1* in black, which is controlled by one of the control knobs. This stage will saturate the signal according to the variable fuzz control input. The signal is then directed to the first clipping stage. A connection was created to skip this stage, this signal is more similar to the original, and will be used later on to attenuate the distortion of the silicon diodes of the second clipping segment. The transfer function is used once more to emulate the two 1N914 diodes of the first clipping stage with a static *V2* value.

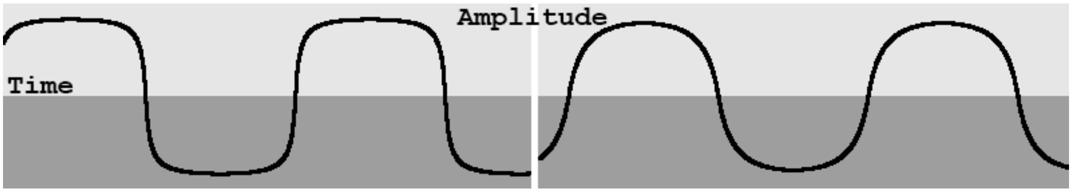


Figure 4.2: Bela simulations of silicon diodes: Left: 1N914, Right: 1N4001.

The diodes were modeled according to what was found from SPICE analysis *figure 3.11*. The PD-generated model output can be seen in *figure 4.2*. The slope of the 1N4001 (right in the figure) could be more accurate when compared to the SPICE analysis of *figure 3.11*. However, this is done intentionally to have a bit of the sine wave characteristics to work with through the signal path. The 1N914 signal is only sent to the silicon simulation, see *figure 4.1*.

4.2 Second clipping segment

The three different clipping options have all been based on analysis recordings of the pharaoh pedal. The PD simulation of the second clipping segment can be seen in *figure 4.3*.

4.2.1 Germanium simulation

Just like the simulation of the 1N914 silicon diode, the 1N34A germanium diode has a static distortion factor, but at 0.3 versus 0.905 of the silicon diode. Additionally a one-pole high pass filter and a one-pole low pass filter with cutoff frequencies at 10 and 70 Hz respectively were used to model sloping tilt of the waveform. The roll-off of the 10 Hz high pass filter will affect the sound despite the low cutoff value. The two filters are placed in parallel. Simulation results with recordings at corresponding settings can be seen in *figure 4.4*.

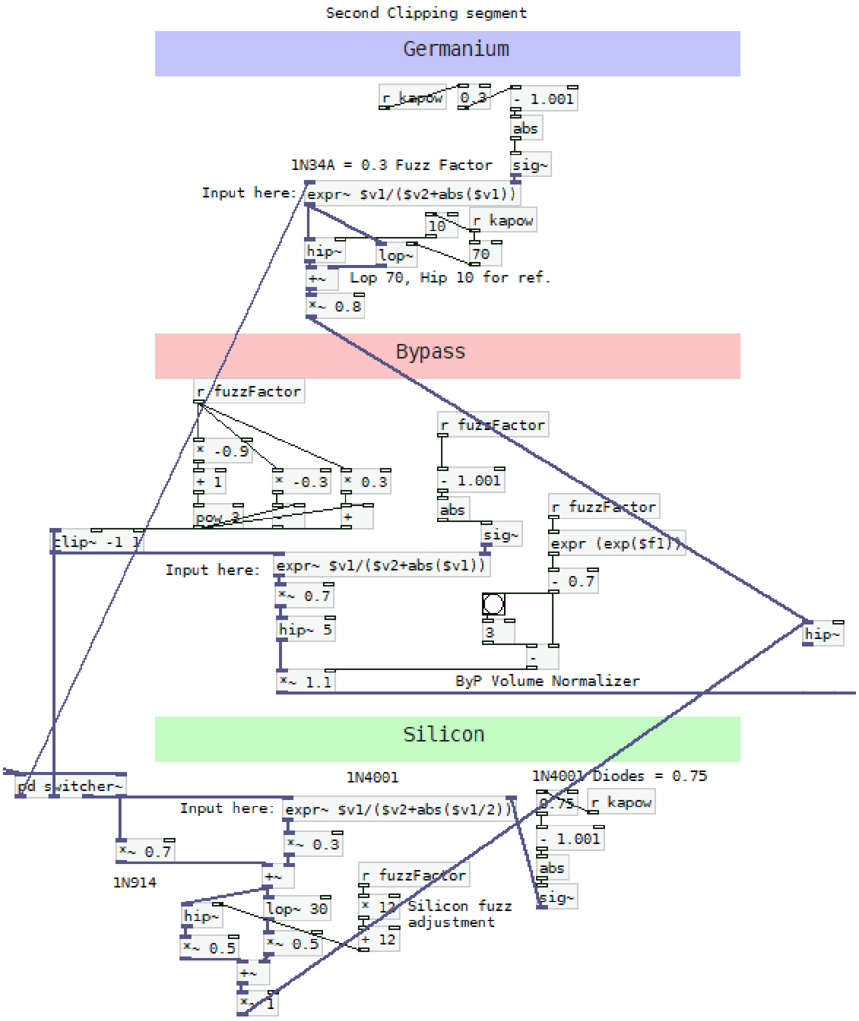


Figure 4.3: The implementation of the second clipping segment in Pure Data.

4.2.2 Bypass simulation

From the recordings of the different 'Fuzz' settings played in the bypass setting it was found that the clipping was very hard and sudden. For that to be captured the PD object 'clip~' was used. The object works as digital clipping. The limits were organized to be inversely connected to the fuzz control: When the fuzz is set low, the clipping limit is high. This parameter was structured to follow an exponential progression, which made it possible to have the distortion kick in at the right time and escalate rapidly. To keep the audio under control the fuzz control was connected to an exponential function, which minimize the output amplitude as the fuzz is increased. A low value high pass filter was used to give the waveform the

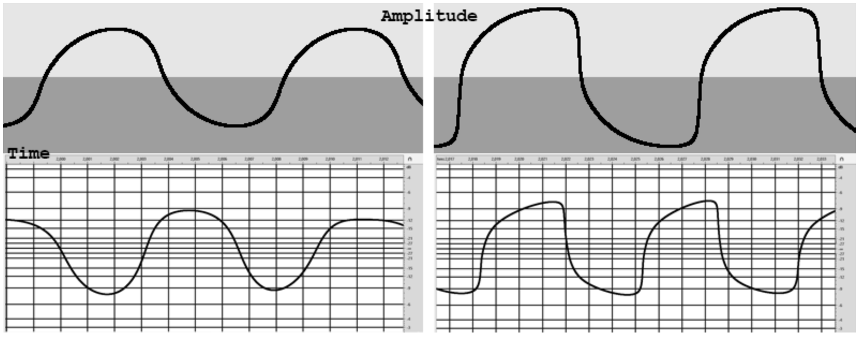


Figure 4.4: Top left: Simulated version of the germanium recording with fuzz at min. setting. Top right: Simulated with fuzz at max. Bottom left: Recording with fuzz at 8 O'Clock setting. Bottom right: Recording at 4 O'Clock setting.

top and bottom tilt (low frequency distortion [23]). The results of the simulation

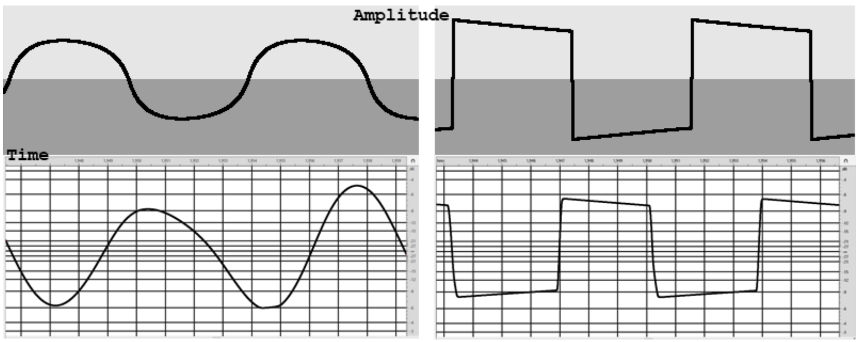


Figure 4.5: Top left: Simulated version of the 'Bypass' recording with fuzz at min. setting. Top right: Simulated with fuzz at max. Bottom left: Recording with fuzz at 8 O'clock setting. Bottom right: Recording at 4 O'clock setting.

can be seen in *figure 4.5*. Looking at the waveforms, it can be seen that the simulated signal with the lowest fuzz setting is more distorted than that of the recorded signal. This is regarded as a minor concern. The hard clipping was detected with the 'Fuzz' set to 10 O'clock at the Pharaoh fuzz. The simulated clipping starts at around 0.25 (0 - 1 interval) when the clipping becomes more prominent.

4.2.3 Silicon simulation

Remember, the silicon clipping is the combined result of the silicon diodes of the two clipping segments (1N914 and 1N4001). The bias by the DC source of the diodes results in a bit more complex implementation than cascading simulations of the unbiased theoretical clipping of the two diode types. An iterative process was used to manipulating different PD values to match the silicon recordings at different fuzz settings. The silicon simulation uses the same input signal as the rest of the simulations, the fuzz variable dependent signal with a static distortion factor of 0.75 to simulate the clipping og the 1N4001 diode. The 1N4001 signal is then added with the simulated 1N914 signal in a ratio of 3:7 (30% 1N4001). Likewise

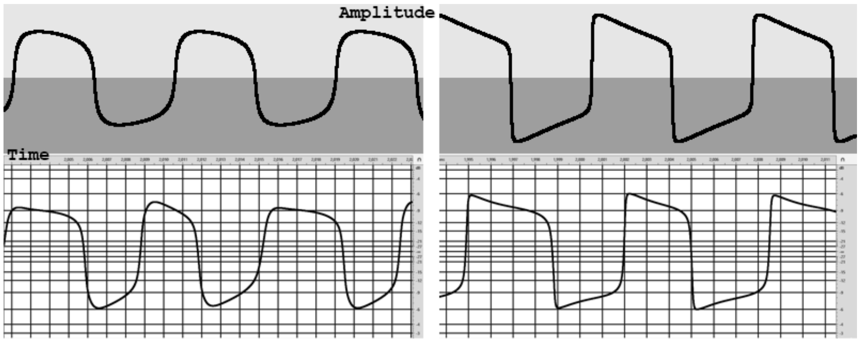


Figure 4.6: Top left: Simulated version of the 'Silicon' recording with fuzz at min. setting. Top right: Simulated with fuzz at max. Bottom left: Recording with fuzz at 8 O'clock setting. Bottom right: Recording at 4 O'clock setting.

with the other simulations, a low frequency high pass filter was needed to give the right tilt, low frequency distortion, of the top and bottom of the waveform. A dynamic high pass filter (10-20 Hz), controlled by the fuzz setting, is located after adding the signals of the two silicon simulations. A static low pass filter was used to get the correct tilt. The output waveform can be seen in *figure 4.6*. From each of the clipping simulations the signal is directed to the 'Tonestack'.

4.3 Tonestack and output

Initially, the 'Tone' and 'High' filters of the tonestack, see *Appendix A*, were implemented based upon the white noise excitation analysis of the Pharaoh pedal. Each of the filters were constructed by cascading a high pass filter, three low pass filters and a biquad filter. The curves followed the obtained data very well, but the timbre of the sound was inaccurate. This issue will be discussed later in the chapter *Discussion*.

Instead the same iterative approach by using the analysis recordings of different 'Tone' and 'High' settings were used as waveform references. The Pure Data tonestack implementation can be seen in *figure 4.7*. It was found that the germanium and silicon simulations could be filtered almost exactly the same way, but the bypass had to be filtered separately by the 'Tone' filter. The 'High' filter is the same for all three simulations.

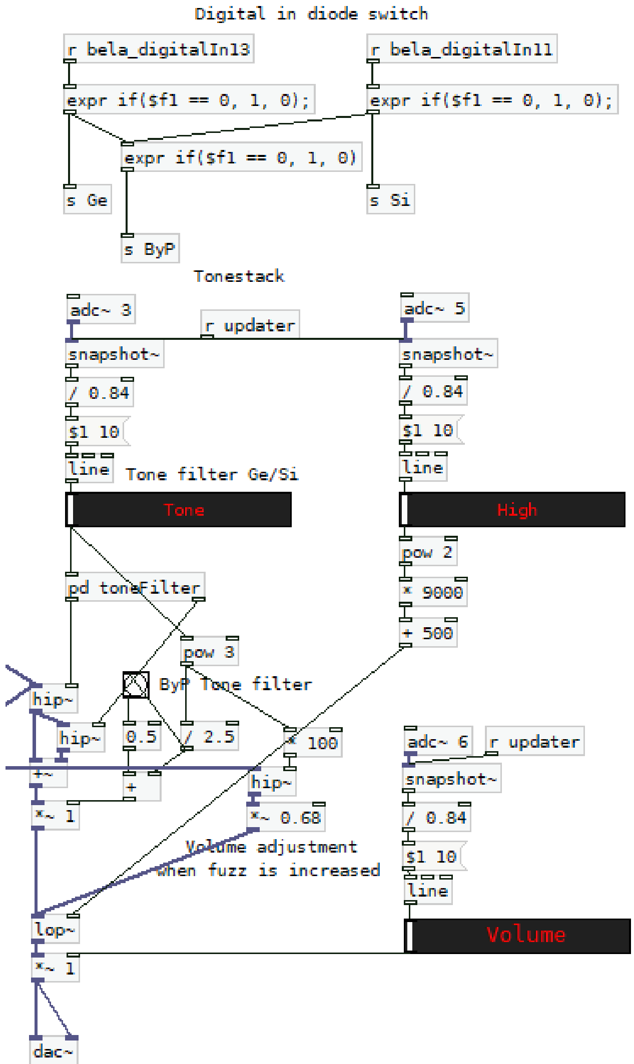


Figure 4.7: The implementation of the tonestack and output segment in Pure Data.

4.3.1 Tone filter simulation

The 'Tone' filter of the 'Ge/Si' setting receives the analog input from PD object 'adc~4' and controls the cutoff frequency of two high pass filters in parallel with different values exponentially. When selecting between germanium and silicon clipping, a multiply factor of the cutoff frequency is also selected; 3 and 2 respectively. Analytically, this was done to match the recorded waveforms, sonically this makes the germanium less 'bright'. The germanium high pass filter cutoff frequencies both start at 0 and range to approximately 110 and 360 Hz, for silicon these values are 70 and 240 Hz. When the cutoff is increased, the amplitude decreases. A countermeasure to reduce this has been applied with the addition of an exponentially increasing value also controlled by the 'Tone' potentiometer input.

When using the Pharaoh with the bypass option, it becomes apparent that the two filters have much less effect on the timbre than it is the case with both the germanium setting and silicon setting. This is also supported by the recorded bypass waveforms, see *Appendix D* this includes comparisons of simulated and recorded waveforms at different 'Tone' settings for both germanium, bypass and silicon. Little difference was detected between min. and max. setting. The waveform suggest marginal more high and low frequency distortion when the 'Tone' setting is at maximum. This can be seen as slightly longer rise time and increased tilt [23]. A more subtle filter was required, therefore a single high pass filter was used for the tone-filtering of the bypass simulation. The range of the bypass low pass filter cut off frequency is 0 to 35 Hz. From here all three simulations are sent to the same 'High' filtering.

4.3.2 High filter simulation

Listening to the effect of the 'High' filter of the Pharaoh pedal revealed that the filter is not simply a low pass filter with high frequency cutoff as first assumed. The 'High' filter when increased, will also filter away low frequency contents, an audible difference. *Figure 4.8* shows a graphical illustration of this. A decision

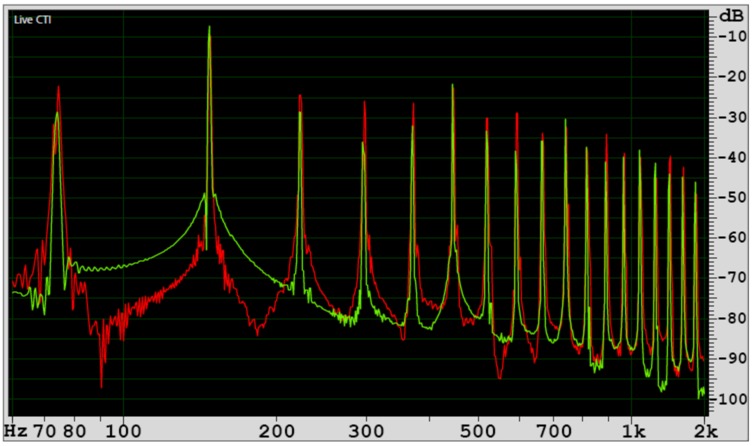


Figure 4.8: The frequency spectrum from recordings of two open D_2 notes at played at two different 'High' settings. Green played at 8 o'clock setting. Red played at 4 o'clock setting.

was taken not to simulate the described behavior, this will be elaborated on in the Discussion chapter. The same effect can be achieved with the tone filter. By not implementing this, the pedal is thought to offer a higher degree of precise fine tuning.

Instead a simple low pass filter was implemented with a range from 500 Hz to 10k Hz controlled by input from a potentiometer. This range will pass plenty high frequency contents when increased and give a 'muddy' scooped sound when turned down, this can also be done with the Pharaoh pedal.

4.3.3 Output

The output is a simple stage, which receives the input from a potentiometer controlling the output volume. The PD object 'line' has been used to avoid speaker clicks. It accomplishes this by performing a smooth transition between change of values over a duration of the 10 ms, as specified in *figure 4.7*.

4.4 Building the pedal

With the first version of the emulation ready, the attention was directed towards building the physical pedal. A diecast aluminum box such as the one used by every pedal builders was acquired.

It was decided to copy the layout of the Pharaoh pedal (knobs, switches, stomp button and I/O connections). It has a nice amount of knobs and switches (some fuzz pedals come with one knob and the stomp button only). The Pharaoh have four potentiometers, a two- and three-way switch and a stomp button. Because of the large size of the BeagleBone and Bela a somewhat larger box than the original had to be used. The BeagleBone and bela was fixed to the bottom of the aluminum

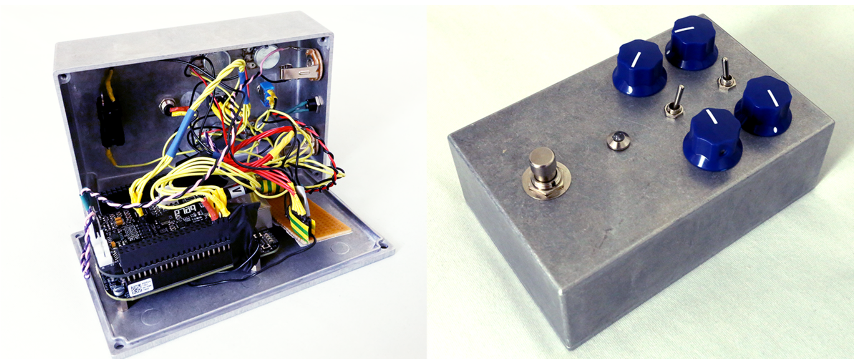


Figure 4.9: Left: The internal of the created pedal with microcontroller, wiring to the switches, potentiometers, diode, DC source connection and audio in/out. Right: The assembled pedal box.

box with screws and four spacers to lift it a little. The internals of the pedal and the assembled pedal can be seen in 4.9. Next step is to evaluate the pedal.

Chapter 5

Evaluation

To evaluate the established emulation model, two skilled guitarists were invited to a qualitative semi-structured in-depth interview [24]. The evaluation took place at Aalborg University, Copenhagen. Sound and video were recorded during the sessions.

5.1 The participants

Both test participants were final semester master students from Aalborg University, Copenhagen, also in the progress of writing their master's theses.

5.1.1 First participant

First participant was a male, 42 years of age, from the Medialogy master's programme. He has played guitar for 10 years before putting it away for a period of years before taking it up again five years ago. When asked about his experience with fuzz effects he replied that he has multiple fuzz effects and appreciates the sound a lot.

5.1.2 Second participant

Second participant was also a male, 25 years of age, from the Sound and Music Computing master's programme. He has been playing guitar for the last eight or nine years. When asked about his experience with fuzz he replied that he likes the effect, but finds it a bit too harsh when playing higher notes. He does not own a fuzz pedal, and is more into time dependent effects such as delay and echo units.

5.2 The setup

Much of the same gear as described in *section 3.1: Apparatus*, were also used for the evaluation, with the addition of a Zoom H6 audio recorder to obtain proper audio quality of the guitar playing. The camera used was a Canon EOS 6D. The day prior to the evaluation both participants were invited to bring their own guitars, but were informed that one would be at their disposal if preferred. The room had some acoustic absorption panels along the walls to reduce reverb. *Figure 5.1* shows

Figure 5.1: Setup with test participant no. II playing the developed pedal emulation of the Pharaoh fuzz.

the setup. The evaluation was a great opportunity to increase the volume of the pedal and amplifier a bit further than what has otherwise gone beyond good practice in terms of being a tolerable neighbor. Personal experience of the undersigned implies that greater amplitudes reveal different timbral characteristics, which can be difficult to detect with subtle speaker membrane movement (this holds in particular for bass).

5.3 Procedure

A semi-structured format was thought to be ideal for getting some feedback on the pedal simulation. The participants were in charge of how long they wanted to continue playing. A few questions were prepared beforehand and asked if the participants did not note on the subject by themselves or at appropriate occasions. The question topics were as follows:

- Age and years of experience playing guitar?

- Prior experience with fuzz effects and whether or not they like fuzz in general.
- How the two pedals are different/similar?
- Whether or not they felt like they would be able to distinguish the sound of the two pedals.

The topics were meant to open up for brief discussions and elaboration. The participants were instructed in the controls of the pedals and a volume setting was dialed according to their liking. The participants played with the original pedal first; both of them for about seven min. before changing to the simulated pedal. The first participant continued playing the emulated pedal for 23 min. before he wanted to stop. The second participant played the emulated version for about 20 min. Initially the idea was to conduct an A/B blind test, but this was deemed meaningless as the participants would have had minutes of experience with the original pedal, and would have to try to recall the sound from 20 min. ago. The scenario would have been different if the participants both owned the Pharaoh pedal and were very acquainted with its sound.

5.4 Evaluation feedback

This section will be a rundown of statements and replies for the prepared questions asked during the sessions. During the two evaluation sessions, a research assistant (R.A.) with focus on audio dropped by (heard the noise), showed great interest for the project and asked if he could try the pedal. He also had some remarks, which will be included in the following. Being a left-handed bass player he did not have the best of prerequisites, but he did not seem to care and was more interested in the technical side of the project.

Overall, both guitarist enjoyed playing the pedal, similarities and a few differences were identified. Both guitar players and the R.A. remarked that the Pharaoh has way more gain than the emulated version. Most of the time the participants were playing the emulated pedal at the highest volume setting, to try to match lower settings on the Pharaoh. Care has been taken to ensure that the digital signal would not be clipped digitally in Pure Data at any configuration. The loudest setting on the pedal has a little bit of headroom, but is quite close to its maximum output. To improve the gain of the emulated pedal a gain stage could be placed right after the signal is manipulated by the Bela, just before it is sent to the 1/4" jack output. A simple amplifier based on an operation amplifier could be used here.

The first participant contacted me later that day with a similar idea. Instead of using the more modern op-amp (a component not used a lot in fuzz pedals) he suggested using a triode (amplifying vacuum tube, a rare component in guitar pedals in general, but it has been done, Vox has a series of tube pedals [1]). While deviating from the concept of virtual analog, the idea is interesting. To the issue of gain the R.A. linked his impression of a cleaner fuzz signal with the emulated pedal, but noted that the emulated signal was much more distorted at the lowest fuzz setting than that of the Pharaoh. This is certainly true, *see figure 4.4 and 4.5*. The second participant also noted the emulated version as overall being cleaner, less fuzzy, than the Pharaoh. Another thing he stated was the Pharaoh as being more 'punchy' - to this he was asked to elaborate, and characterized that as being a bit more aggressive in the attack. This was noted as the most significant difference apart from the gain when asked into the difference between the two pedals.

Both of the invited guitar players responded to the question whether or not they thought they would be able to tell them apart, with a 'no'. To that they both added they thought they would be able get the same sound from the different pedals by dialing in the control knobs.

At the end they both asked whether the emulated pedal was digital or analog, out of curiosity the question was reflected back at them to get their guess. The first participant guessed correctly, that it was digital, which he based on the cleaner fuzz. The second participant guessed it analog.

After the evaluation, a few samples were recorded of the different clipping options, while trying to match the sound of the two pedals. The bypass recording turned out very similar. The other recordings were less accurate, both the germanium and the silicon Pharaoh recordings had more high frequency content. Differences, which are though reducible if a more meticulously recording and tuning process had been applied. But finding similar sounds is not difficult, it is getting the extremes right. Below are the spectrograms of the two bypass recordings.

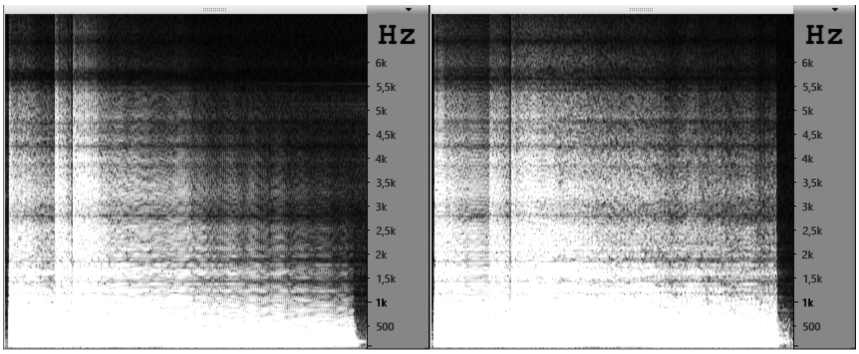


Figure 5.2: Spectrograms of a power cord played twice followed by an open strum of the same strings. Left: The emulation pedal. Right: The pharaoh. Both recordings are about 14 sec. in length, note the difference in sustain, which can be seen as more brightness along the X-axis.

[1] http://www.voxamps.com/trfz

Chapter 6

Discussion

The following chapter picks up some of the issues and concerns identified throughout the project.

6.1 Wave shaping and clipping

To implement the different diode simulation options, different approaches were applied, such as wave shaping with a transfer function, based on recordings and SPICE simulations. The bypass option showed the hardest clipping, which is why controlled non-linear digital clipping was utilized in the form of the PD object 'clip' in combination with the transfer function *eq. 4.1*. The germanium and silicon diode simulations relied on the transfer function only (no clipping). There is a clear difference between clipping and wave shaping, which needs to be addressed.

Clipping, whether it be digital or analog, will greatly affect the sustain and amplitude of a note. A clipped signal will be limited at a specific amplitude. In fuzz pedals the clipping starts at low signal values, clipping much of the transistor amplified signal. Therefore the signal will stay at this amplitude until its amplitude drops below the clipping amplitude. The effect is level dependent. This results in notes with very long sustain. With the Pharaoh, the sustain phase can be upwards of 25 seconds with only 3 dB signal drop, followed by a release over 5 seconds with 12 dB signal drop. The reason why the clipping was used for the bypass was because of its somewhat simplistic waveform, in particular when compared to the other diode options. Furthermore, the waveform was more consistent when manipulated by the control knobs. The waveform showed a high degree of resemblance with that of a square wave, but with some rise and fall time between the positive and negative wave cycle phase. This could be simulated with the above described combination of techniques. But when it came to the more complex waveforms of the silicon and germanium, the simplistic nature of controlled digital clipping was not suited for the simulation.

The transfer function *eq. 4.1* used for the germanium and silicon simulation (and bypass) on the contrary to clipping is not level dependent. It will roughly follow the amplitude of the affected signal, and therefore not have the same amount of sustain. The difference is going to be most prominent when playing slow and holding the notes for extended periods. The timbre of the sound will deviate between the two wave shaping techniques, as the overtones of the wave transform are going to fade to inaudible levels faster.

6.2 Accurate filter design - inaccurate sound

In the *Implementation* (chapter 4) it was described how the black box-designed 'Tone' and 'High' filters were not reproducing the Pharaoh sound properly, and were therefore replaced with some simpler filters. The reason for the inaccurate sound is thought, primarily, to be connected to the cascading of the two filters. Secondarily, to a lesser extent, the faster release in harmonic content due to use of wave shaping instead of clipping, could also be a possible explanation. The tonestack circuit of the Pharaoh fuzz is a bit more complex than cascading the 'Tone' and 'High' filter. A RC filter acts as a fixed high-pass filter, which bypasses the 'High' filter.

A more accurate virtual analog implementation of the tonestack, could be produced by; determining the frequency response of the RC- high-pass filter and low pass filter (the C9-R5 and R8-C8 configurations respectively, see *Appendix A*), implemented and connected in parallel. Both filters are connected to the 'Tone' potentiometer (R25), which weighs the balance between the two RC filters. It would be less responsive to the dynamic filtering, one of the reasons why a simpler tonestack, but less accurate was implemented. This will be discussed further in the following subsection.

6.2.1 Modifications and improvements

Most aspects of the Pharaoh fuzz was tried emulated as accurately as possible by the means of the Pure Data standard library objects. However, when it came to the filter design of the tonestack, the flexibility of digital filtering and prospect of improving on the design was sized when the first iteration proved inaccurate. The best sound of the Pharaoh fuzz (subjective preference) is achieved by dialing the 'Tone' all the way counterclockwise (producing pronounced lows), and the 'High' all the way clockwise (producing pronounced highs), to let through as much bass and treble as possible, for a rich sound. When designing analog effect pedals, many components will show different filtering characteristics, which can be altered by changing component value. Increasing the size of a capacitor will let through more bass for instance. The product is a complex filter formed by the many different components, which individually attenuates certain frequencies. With the digital implementation it was decided not to filter anything away before the 'tonestack' (apart from the filtering associated with the diode simulations). This is also were the emulation deviates the most, it was furthermore decided to allow the 'Tone' to let everything through at its most bass-influenced setting. Likewise, virtually all treble is let through by at the max. 'High' setting. The fuzz effect is thereby more dynamic and tunable than the Pharaoh in that regard. This is seen by the undersigned as an improvement from the original, which could benefit from more pronounced highs.

6.3 Patch performance and unsupported Pure Data objects

The efficiency of the Pure Data patch has not been a focus in this project. One property encountered during the implementation worth mentioning regarding performance, is the aspect of blocking a signal. Pure Data has some objects, which are not supported in the Bela framework. Curiously, the 'spigot' object is supported while the 'spigot~' object is not. The latter handles wave signals, which are either blocked or passed through depending on a control message (0 or 1 respectively). In the PD patch, the signal is directed to the three diode options. With the use of two 'spigot~' objects it would be possible to only send the signal to one at a time, possibly reducing computation power. As this is not possible, the signals not used are reduced to arrays of zeros, a less elegant solution, but it seems to reduced to computation load nonetheless. The simplest patch with an input obtained through the 'adc~' object and then directly connected to a 'dac~' output object results in a CPU load of around 22%. The presented patch uses around 33-34% of the available CPU computation power. Therefore, the actual DSP and sensor handling is accountable for a CPU load at around 11-12%. Leaving plenty of computational power for at least a handful additional effects of the same magnitude as the one presented in this project. In *chapter 2* an article was presented [10] including findings of computational load on the CPU when using the standard PD library versus 'Heavy Audio Tools'. By using the latter, they were able to reduce the CPU load by the patch with around 50%. Based on this, 'Heavy Audio Tool' seems like a decisive measure if CPU load is a concern.

Chapter 7

Conclusion

The number of digital platforms for electric guitar effects, capable of handling multiple effects, with a physical interface appears to be increasing in numbers. At this point in time they pose no threat to the market for analog pedals. Many of the digital platforms available are intended to allow the user to develop effects. However, knowledge of programming and digital signal processing is required to be able to exploit this feature. Based on feedback, this presents a challenge for some users.

The focus and intention of this project has been to try to make analog emulation more approachable. Especially through the use of a tangible, graphical programming environment - Pure Data.

In the course of the project a commercial available analog guitar effect pedal, the 'Pharaoh' fuzz by Black Art Toneworks, has been thoroughly analyzed and tested. Based on this a digital model has been developed and tested. The model has been based on numerous recorded analysis signals. The recordings were emulated by the use of a transfer function to simulate the waveforms. Insight of how the Pharaoh fuzz works were obtained through its circuit schematic and SPICE simulations thereof. The developed pedal is able to produce very similar waveforms in accordance with the analysis recordings.

The simulated effect pedal was made available for evaluation for two guitarists, both of which enjoyed playing the pedal and found it considerably close to the Pharaoh. The two effect pedals were found to carry some differences, most notably the gain, and therefore also sustain. The gain was found too low on the simulated model. With the experience the two evaluation participants had with the Pharaoh pedal by the end of the evaluation (none had played it before), they independently, expressed that they thought they would not be able to tell them apart in a listening test.

The knob settings of the two pedals did not match properly. Meaning that the pedals will not necessarily produce the same sound when the knobs are set identically. But the sound of the simulated pedal, can be dialed in to match that of the original to a reasonable degree at a different knob setting.

The Pharaoh pedal is implemented with three selectable distortion settings, where the signal can be subjected to clipping either by silicon diodes or germanium diodes. A bypass options exists. Simulation of the three diode options performs best when the 'Fuzz' (one of the control knobs) setting is not at its maximum. At that level, the germanium simulation sounds a bit too harsh, and the bypass and silicon option seems too identical. In general the germanium and bypass options seem more accurately emulated than the silicon option.

Pure Data was found capable of going a long way in the sense of being able to replicate the recorded analysis waveforms, just by using the standard library objects. The simulated effect pedal model is not based on physical modeling. That is not to say that, it is not possible in Pure Data. Separating the circuit schematic into smaller sections to capture the effect of the individual RC filters seems feasible and intriguing as a future effort in making the filter design more accurate. Overall Pure Data performed better than expected. The bela and Pure Data combination is tangible and works very well. The performance was impressive too: At least a handful of PD patches with the same computational load as the simulated model would be able to run simultaneously.

Bibliography

[1] Dave Hunter. *Guitar Effects Pedals - The Practical Handbook*. Ed. by John Morrish. 2013. Chap. 2, pp. 19–42.

[2] Michael Alexander Jenkins. *Analog Electric Guitar Distortion Effects and Headphone Amplification*. Senior project. California Polytechnic State Univers, June 2010.

[3] Paul Scherz and Simon Monk. *Practical Electronics For Inventors*. 3rd ed. 2013. Chap. 4, pp. 401–493.

[4] Pierre Dutilleux et al. *DAFX: Digital Audio Effects*. Ed. by Udo Zölzer. 2nd ed. 2011. Chap. 4, pp. 101–138.

[5] Kendall Castor-Perry. *Just Add A Transistor: Clipping Me Softly*. Aug. 2014. URL: `http://www.planetanalog.com/author.asp?section_id=3066&doc_id=563331` (visited on 03/19/2017).

[6] Massimo Conti et al. *Intelligent Technical Systems. Real Time Implementation of Fuzz-Face Electric Guitar Effecte*. Vol. 38. 2009. Chap. 7, pp. 89–100.

[7] William E. Overton. "Digital Circuit-Level Emulation of Transistor-Based Guitar Distortion Effects". Master Thesis. Georgia Institute of Technology, May 2006.

[8] R.G. Keen. *The Technology of the Fuzz Face*. 1998. URL: `http://www.geofex.com/article_folders/fuzzface/fffram.htm` (visited on 03/19/2017).

[9] Thomas Webster, Guillaume LeNost, and Martin Klang. "The OWL: An Open Source, Programmable Stage Effects Pedal". In: *KES Transactions on Innovation in Music*. Vol. 1.1. 2013, pp. 147–157.

[10] Giulio Moro et al. "Making High-Performance Embedded Instruments with Bela and Pure Data". In: *Proceedings of the 2016 International Conference on Live Interfaces*. Experimental Music Technologies, 2016, pp. 339–343. ISBN: 978-0-9931996-8-4. URL: `http://www.liveinterfaces.org/proceedings2016.html`.

[11] Stefano D'Angelo, Jyri Pakarinen, and Vesa Välimäki. "New Family of Wave-Digital Triode Models". In: *IEEE Transactions on Audio, Speech, and Language Processing* 21.2 (Feb. 2013), pp. 313–321.

[12] Rafael Cauduro Dias de Paiva et al. "Real-Time Audio Transformer Emulation for Virtual Tube Amplifiers". In: *Journal on Advances in Signal Processing* 2011 (2011).

[13] Jyri Pakarinen and Mika Karjalainen. "Enhanced Wave Digital Triode Model for Real-Time Tube Amplifier Emulation". In: *IEEE Transactions on Audio, Speech, and Language Processing* 18.4 (May 2010), pp. 738–746.

[14] David Te-Mao Yeh. "Digital Implementation of Musical Distortion Circuits by Analysis and Simulation". Ph.D. Thesis. Department of Electrical Engineering of Standford University, June 2009. Chap. 2 and 3.

[15] Alfred Fettweis. "Wave Digital Filters: Theory and Practice". In: *Proceedings of The IEEE*. Vol. 74. 2. Feb. 1986, pp. 270–329.

[16] Vesa Välimäki et al. *DAFX: Digital Audio Effects*. Ed. by Udo Zölzer. 2nd ed. 2011. Chap. 12, pp. 473–522.

[17] Giovanni De Sanctis, Augusto Sarti, and Stefano Tubaro. "Automatic Synthesis Strategies for Object-Based Dynamical Physical Models in Musical Acoustics". In: *Digital Audio Effects*. 2003, 198–202s.

[18] David Yeh. *Tutorial on Wave Digital Filters*. Center for Computer Research in Music and Acoustics, Stanford University. Jan. 25, 2008. URL: `https://ccrma.stanford.edu/~dtyeh/papers/wdftutorial.pdf` (visited on 03/19/2017).

[19] Troels Lunde Hagensen et al. "Modelling and evaluation of the Roland Space echo 150". In: Currently being reviewed for the 14th Sound and Music Computing Conference. July 5, 2017.

[20] Rafael C. D. Paiva et al. "Emulation of Operational Amplifiers and Diodes in Audio Distortion Circuits". In: *IEEE Transactions on Circuits and Systems* 59.10 (Oct. 2012), pp. 688–692.

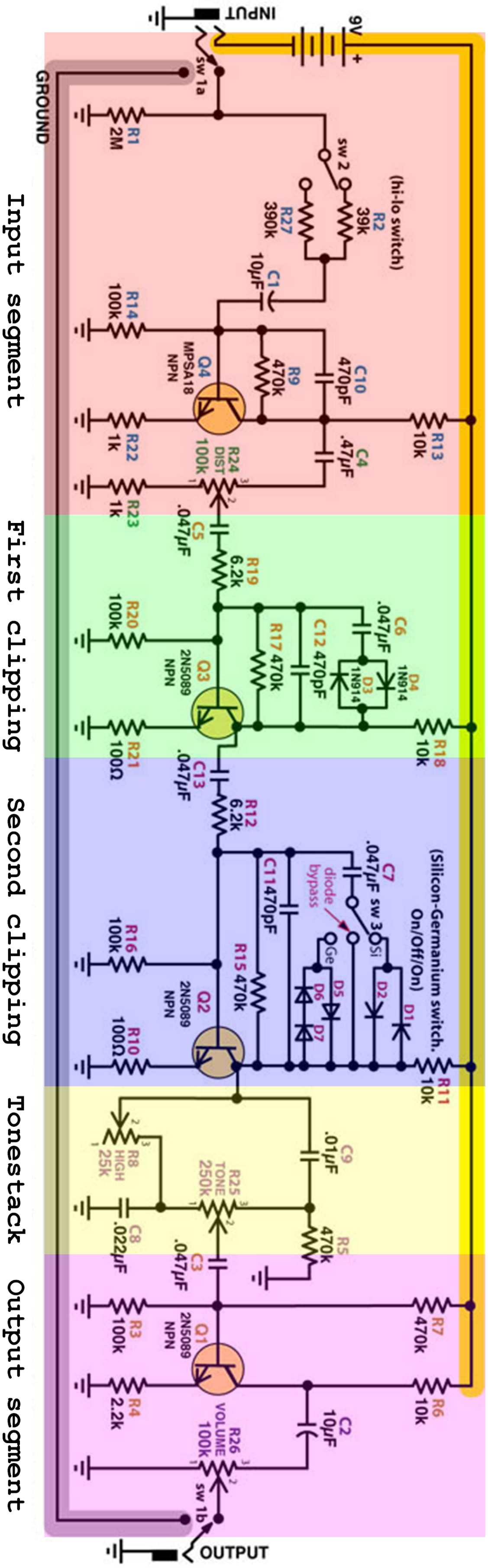
[21] Tom Høholdt, Helge Elbrønd Jensen, and Frank Nielsen. *Linear Differential Equations and Fourier Series*. Translated from Danish. Technical University of Denmark, 1982, pp. 80–95.

[22] Christopher J. Wells. *The Diode*. May 4, 2017. URL: `https://www.technologyuk.net/physics/electrical-principles/the-diode.shtml`.

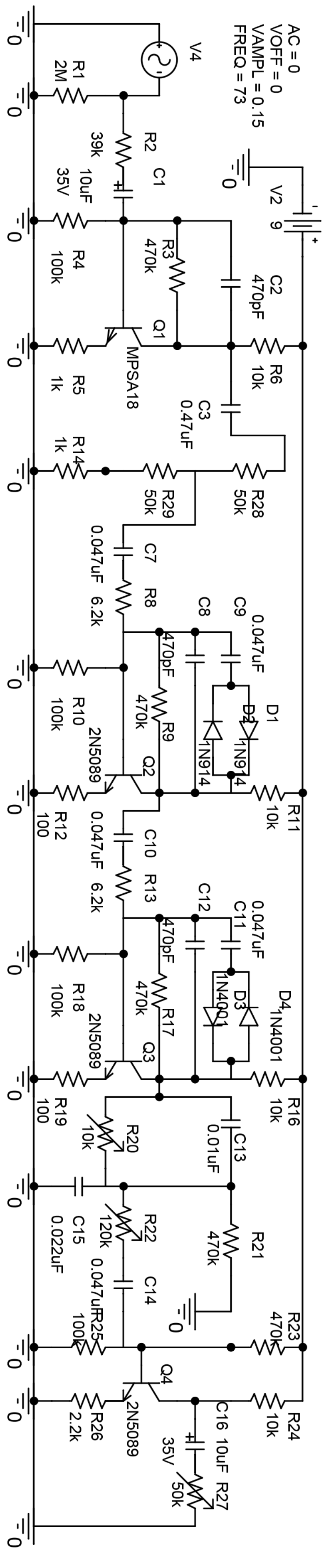
[23] David A. Bell. *Solid State Pule Circuits*. Reston Publishing Company, Inc, 1976. Chap. 1.

[24] Thomas Bjørner. *Qualitative Methods For Consumer Research*. 1st ed. 2015. Chap. 2, pp. 57–96.

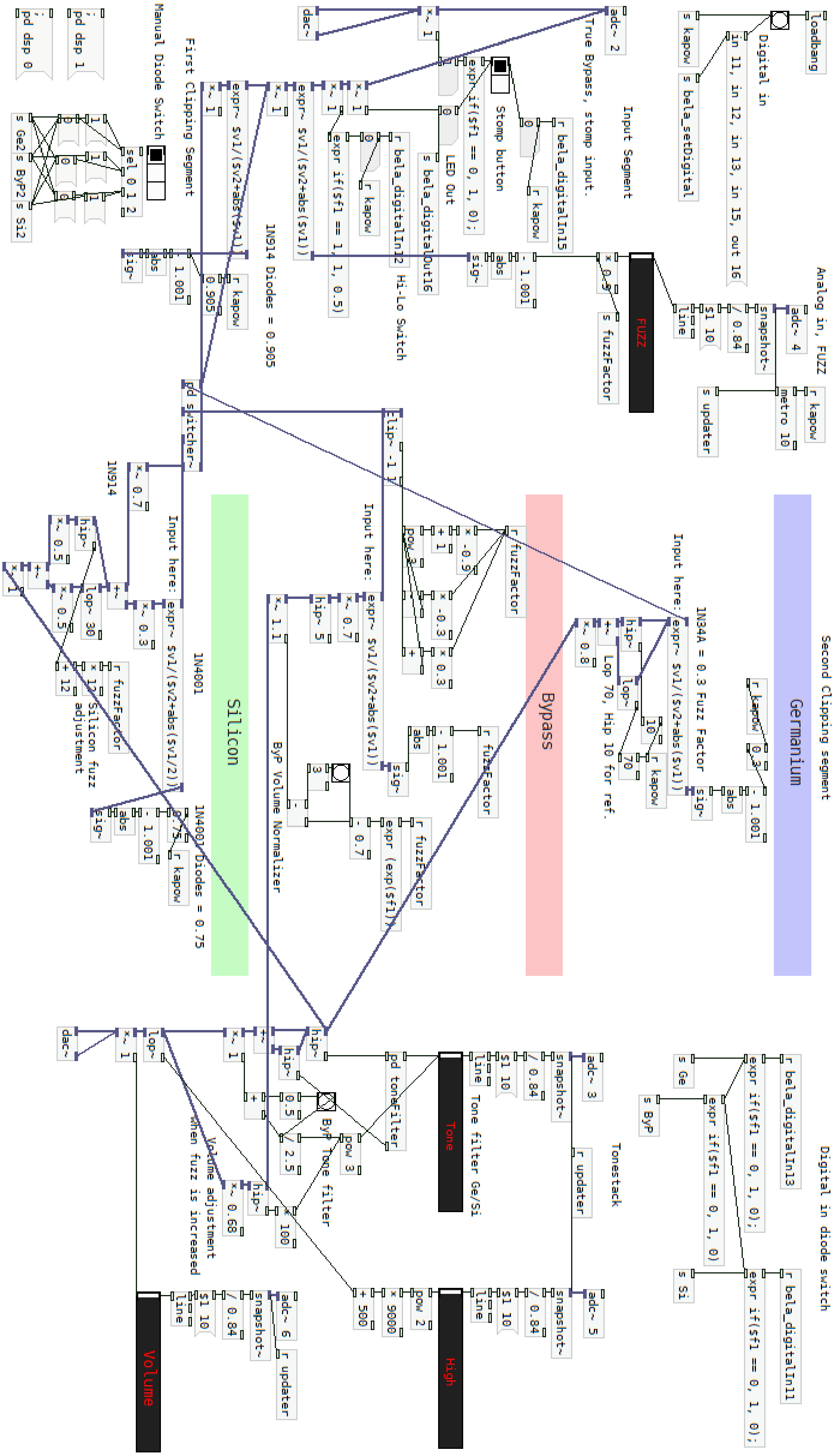
Appendix A: Pharaoh Circuit schematic.



Appendix B: SPICE schematic.

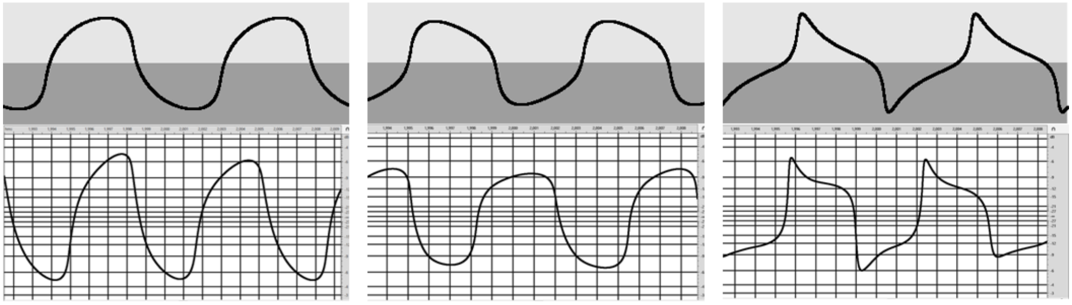


Appendix C: Complete Pure Data emulation patch.

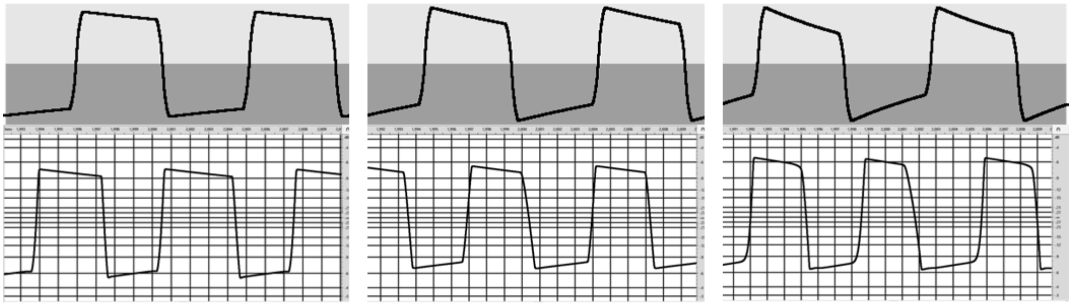


Appendix D: Comparison between simulated waveforms and the recorded analysis signals.

Germanium 'Tone' filtering at min., med. and max. settings. Top: simulated. Bottom: Recordings.



Bypass 'Tone' filtering at min., med. and max. settings. Top: simulated. Bottom: Recordings.



Silicon 'Tone' filtering at min., med. and max. settings. Top: simulated. Bottom: Recordings.