

# DIGITAL CIRCUIT-LEVEL EMULATION OF TRANSISTOR-BASED GUITAR DISTORTION EFFECTS

A Thesis  
Presented to  
The Academic Faculty

By

William E. Overton

In Partial Fulfillment  
Of the Requirements for the Degree  
Master of Science in Electrical Engineering

Georgia Institute of Technology

May 2006

**DIGITAL CIRCUIT-LEVEL EMULATION OF  
TRANSISTOR-BASED GUITAR DISTORTION EFFECTS**

Approved by:

Dr. Aaron Lanterman, Advisor  
College of Electrical and Computer Engineering  
*Georgia Institute of Technology*

Dr. W. Marshall Leach  
College of Electrical and Computer Engineering  
*Georgia Institute of Technology*

Dr. Paul Hasler  
College of Electrical and Computer Engineering  
*Georgia Institute of Technology*

Date Approved: May 10, 2006

## ACKNOWLEDGEMENTS

This thesis would not have been possible without the guidance and support of many people. First, I would like to thank my advisor Aaron Lanterman for his guidance and support. Whenever I was having difficulty, he was there to help me through it. His criticism of this document was also greatly appreciated. I would also like to thank Marshall Leach and Paul Hasler for serving on my thesis reading committee and helping out whenever I had a question. I also thank my parents Bill and Dolores Overton who encouraged me throughout this whole process. Finally, many thanks are owed to my fiancée Alison Hancock for being supportive of my studies and understanding about the time necessary to complete this thesis.

## TABLE OF CONTENTS

ACKNOWLEDGEMENTS.....	iii
LIST OF TABLES.....	v
LIST OF FIGURES .....	vi
LIST OF SYMBOLS, ABBREVIATIONS, AND TERMS.....	vii
SUMMARY .....	viii
CHAPTER 1: INTRODUCTION.....	1
CHAPTER 2: LITERATURE REVIEW .....	3
2.1 Origin and History .....	3
2.2 Common Circuit Modifications .....	5
2.3 The Ebers-Moll Transistor Model .....	8
CHAPTER 3: METHODOLOGY .....	11
3.1 Circuit Simulation.....	11
3.2 Measurements from a Real Circuit .....	13
3.3 Digital Circuit Design.....	17
3.3.1 Filter Effects.....	17
3.3.2 Transistor Modeling.....	18
CHAPTER 4: RESULTS .....	25
4.1 Simulated Output from PSPICE .....	25
4.2 Analog Circuit Parameters .....	28
4.3 Algorithm Design.....	30
4.4 MATLAB Algorithm Performance.....	33
CHAPTER 5: CONCLUSIONS .....	39
CHAPTER 6: DIRECTION FOR FUTURE STUDY .....	42
APPENDIX A: SIMULATION DATA.....	44
APPENDIX B: PSPICE BIPOLAR TRANSISTOR MODEL PARAMETERS .....	51
APPENDIX C: MATLAB CODE AND DATA FILES.....	55
REFERENCES .....	64

## LIST OF TABLES

Table 3.1	Resistance measurements.....	13
Table 3.2	Values for computing the true $\beta$ .....	14
Table 3.3	Nodal voltages, fuzz control at 0 .....	16
Table 3.4	Nodal voltages, fuzz control at 1K .....	16
Table 3.5	Transistor parameters.....	17
Table 4.1	Effects of changing transistor gains .....	27
Table 4.2	Maximum percent error compared to mean.....	28
Table B.1	Bipolar transistor parameters .....	52

## LIST OF FIGURES

Figure 1.1	Basic fuzz face .....	1
Figure 2.1	A modern example of the Fuzz Face circuit and its housing.....	3
Figure 2.2	Sample output of a Fuzz Face.....	5
Figure 2.3	Fuzz Face with “Roger Mayer” or “Jimi Hendrix” mods.....	6
Figure 2.4	Fuzz Face with Fuller mods.....	7
Figure 2.5	Vox Tone Bender 5/67.....	8
Figure 2.6	PNP Ebers-Moll transistor model.....	9
Figure 3.1	PSPICE schematic of the Fuzz Face.....	11
Figure 3.2	PSPICE AC128 model parameters .....	12
Figure 3.3	Germanium transistor testing.....	14
Figure 3.4	Fuzz Face circuit, fuzz control at 0.....	15
Figure 3.5	Fuzz Face circuit, fuzz control at 1K.....	16
Figure 3.6	Circuit in terms of transistor currents .....	19
Figure 4.1	PSPICE simulation of the Fuzz Face .....	25
Figure 4.2	Comparison of theoretical clipping to the output .....	25
Figure 4.3	FFT of soft (top) and hard (bottom) clipped outputs .....	26
Figure 4.4	DC bias voltages before and after parameter modifications.....	27
Figure 4.5	Comparison of MATLAB and PSPICE transfer functions.....	34
Figure 4.6	Processed guitar output, Ebers-Moll modeling.....	35
Figure 4.7	Processed guitar output, PSPICE modeling.....	36
Figure 4.8	Processed guitar output, Simplified Ebers-Moll modeling.....	37

Figure 4.9	Clean and distorted guitar signals .....	38
Figure 5.1	PSPICE DC and transient analysis .....	40
Figure A.1	Simulated output for $\beta_1=70$ , $\beta_2=120$ ; Inputs of 0.5, 1, 1.5, and 2 mV....	45
Figure A.2	Simulated output for $\beta_1=70$ , $\beta_2=70$ ; Inputs of 0.5, 1, 1.5, and 2 mV.....	46
Figure A.3	Simulated output for $\beta_1=120$ , $\beta_2=70$ ; Inputs of 0.5, 1, 1.5, and 2 mV....	47
Figure A.4	Simulated output for $\beta_1=120$ , $\beta_2=120$ ; Inputs of 0.5, 1, 1.5, and 2 mV..	48
Figure A.5	Simulated output for $\beta_1=40$ , $\beta_2=120$ ; Inputs of 0.5, 1, 1.5, and 2 mV....	49
Figure A.6	Simulated output for $\beta_1=70$ , $\beta_2=180$ ; Inputs of 0.5, 1, 1.5, and 2 mV....	50

## LIST OF SYMBOLS, ABBREVIATIONS, AND TERMS

**$\beta$  (Beta)** – Represents the gain of a transistor. (Used interchangeably)

**BJT** – Bipolar junction transistor

**Distortion** – In guitar effects this refers to harmonic distortion

**FFT** – Fast Fourier Transform, used to determine the frequency content of a signal

**Fuzz Pot** – The potentiometer in the Fuzz Face between the emitter of Q2 and ground

**Fuzz Face** – The proprietary name of the transistor circuit being explored in this paper

**MATLAB** – A numerical computing environment and programming language

**Mod** – Term for the modification of one of more components of a circuit

**Mono** – Monaural, or one channel audio

**Pot** – Potentiometer, a variable resistor

**PSPICE** – A SPICE program that runs on a personal computer.

**SPICE** – Simulated Program with Integrated Circuit Emphasis; used to simulate circuits

**Stereo** – Two channel audio, usually divided into left and right

**Volume Pot** – Potentiometer in the Fuzz Face from which the output is taken

**WAV** – Audio file format using pulse-code modulation for compression



## SUMMARY

The objective of this research was to model the Fuzz Face<sup>1</sup>, a transistor-based guitar distortion effect, digitally at the circuit level, and explore how changes in the discrete analog components change the digital model. The circuit was first simulated using SPICE simulation software. Typically outputs and how they changed based on transistor gains were documented. A test circuit was then constructed in lab to determine true transistor gains. An analog Fuzz Face circuit was then constructed, and physical parameters were recorded. A digital model was then created using MATLAB. Capacitive filtering effects were found to be negligible in terms of the guitar signal and were not modeled. The transistors were modeled using the Ebers-Moll equations. A MATLAB algorithm was written to produce Fuzz Face type distortion given an input guitar signal. The algorithm used numerical techniques to solve the nonlinear equations and stored them in a look-up table. This table was used to process the input clips. The sound of the Fuzz Face was not perfectly modeled, but the equations were found to provide a reasonable approximation of the circuit. Further study is needed to determine a more complete modeling equation for the circuit.

---

<sup>1</sup> Fuzz Face® is a registered trademark of Dunlop Guitar Accessories, USA

## CHAPTER 1: INTRODUCTION

Musicians have long been intentionally distorting guitar signals to create a wide variety of tones. The first “distortion effects” documented were created by removing electronics such as vacuum tubes from amplifiers and punching holes in speakers [1]. Electronics were later created to replicate these effects, first with vacuum tubes and then later with transistors. Currently, digital signal processors are also being used to create these distortion effects. This thesis studies a specific analog guitar distortion circuit, the Fuzz Face, and explores how to model it at the circuit level.

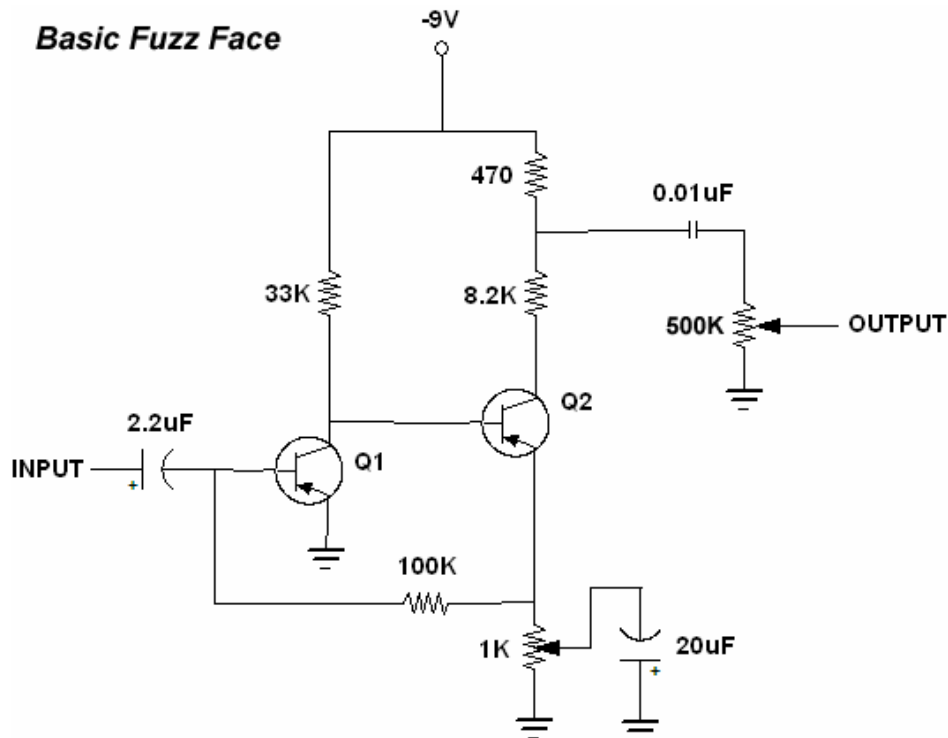


Figure 1.1 – Basic fuzz.face [2]

The Fuzz Face is well known as being the effect of choice for 60s rock musician James Marshall “Jimi” Hendrix. It is still popular today; however, modern day musicians have

found that these devices vary a lot from unit to unit. Much of this is because they used germanium transistors, which are far less consistent than silicon transistors. However, due to intrinsic properties of germanium and silicon transistors, the Fuzz Face is reported to not sound the same if the more consistent silicon components are used. Thus, we seek a digital model of “good” germanium transistors, so that we may implement a digital Fuzz Face recreation that combines the musical properties of a germanium device with the consistency and stability of a digital circuit.

We first explore the theoretical operation of the Fuzz Face will first be explored using PSPICE. We then extract transistor parameters and operating points from a real constructed Fuzz Face circuit. We also explore analytic mathematical models based on the Ebers-Moll transistor equations. Finally, this information is used to create a digital model, which should operate similarly to its analog counterpart.

## CHAPTER 2: LITERATURE REVIEW

### 2.1 Origin and History [3][4]

The Fuzz Face is a guitar distortion pedal manufactured by Dallas-Arbiter and first made available in 1966. It was reissued in the early 1990's by Dunlop Guitar Accessories, USA. It is an extremely simple circuit, containing two transistors, four resistors, three capacitors, and two potentiometers (pots). The pots control the volume and the amount of distortion. Early models used germanium transistors. They were argued to be the better sounding models, although some later models used silicon. PNP transistors are used in the germanium models, whereas NPN transistors are used in the silicon models.



*Figure 2.1 – A modern example of the Fuzz Face circuit and its housing [5][6].  
Photo used with permission.*

The original transistor used in the Fuzz Face was the AC128. Later, the NKT275 was used due to its similar performance but higher consistency. Most musicians say that the germanium transistors are more musical, but there are some who prefer to use the silicon

variety. Many musicians have changed the components to modify the circuit in order to change the tone, distortion, or responsiveness of the device. Among musicians, this is known as “modding.” Some popular mods include the Hendrix/Mayer Mod, the Fuller Mod, and the Vox Tone Bender Mod.

As described by Keen [3], the two transistors in the Fuzz Face make up a voltage feedback biasing circuit. The current flowing into the base of the first transistor is proportional to its collector voltage. This arrangement gets the highest gain out of the transistor, which is good for a distortion device. When biased correctly, there is a lot of headroom, which leads to soft clipping. However, it clips much earlier on the opposite polarity, resulting in asymmetrical behavior. This asymmetrical clipping is important for the musical quality of this device. The second transistor stage behaves more symmetrically, so when it is driven hard, the amount of distortion will increase as the upward swing becomes more heavily clipped. When not driven as hard, the asymmetrical shape will be preserved. This “touch sensitivity,” based on how hard the transistor is driven, is also an important musical quality of the Fuzz Face.

With the coming of the “digital age,” many musicians have “gone digital,” not only with their recording tools, but with their effects as well. “Virtual effects” software is becoming increasingly popular, and musicians are now starting to expect every analog effect to be available in a digital format. Our goal is to not only have these models sound like the originals, but for them to be able to be modified like their analog counterparts. A sample output of an actual Fuzz Face is found below.

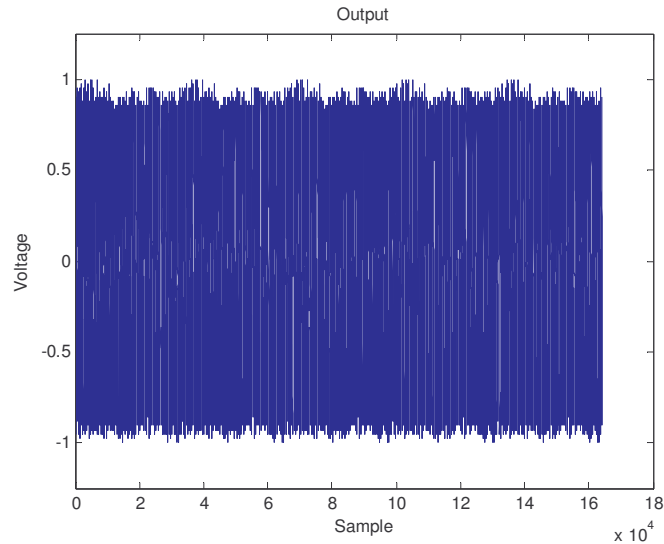


Figure 2.2 – Sample output of a Fuzz Face [7]  
 (overton\_william\_e\_200605\_mast\_fig22\_fuzzface.wav, 438K)

## **2.2 Common Circuit Modifications [3]**

In an effort to customize their sounds to their own style and preference, musicians have implemented many modifications, or “mods,” to the Fuzz Face circuit.

### Hendrix/Mayer Mod

Roger Mayer is a guitar effects guru who began creating effects in 1964 and began working with Jimi Hendrix in 1967. Mayer tweaked Jimi’s gear heavily, including his Fuzz Face. Mayer’s changes are commonly referred to as the “Hendrix” Mods or “Roger Mayer” Mods. Mayer’s changes were:

- Replacing the 470Ω output resistor with 1kΩ
- Replacing the 8.2kΩ resistor at the collector of Q2 with 18kΩ
- Replacing the 1kΩ pot at the emitter of Q2 with 2kΩ

These changes increase the resistance seen by the second transistor, and increase its output level and gain.

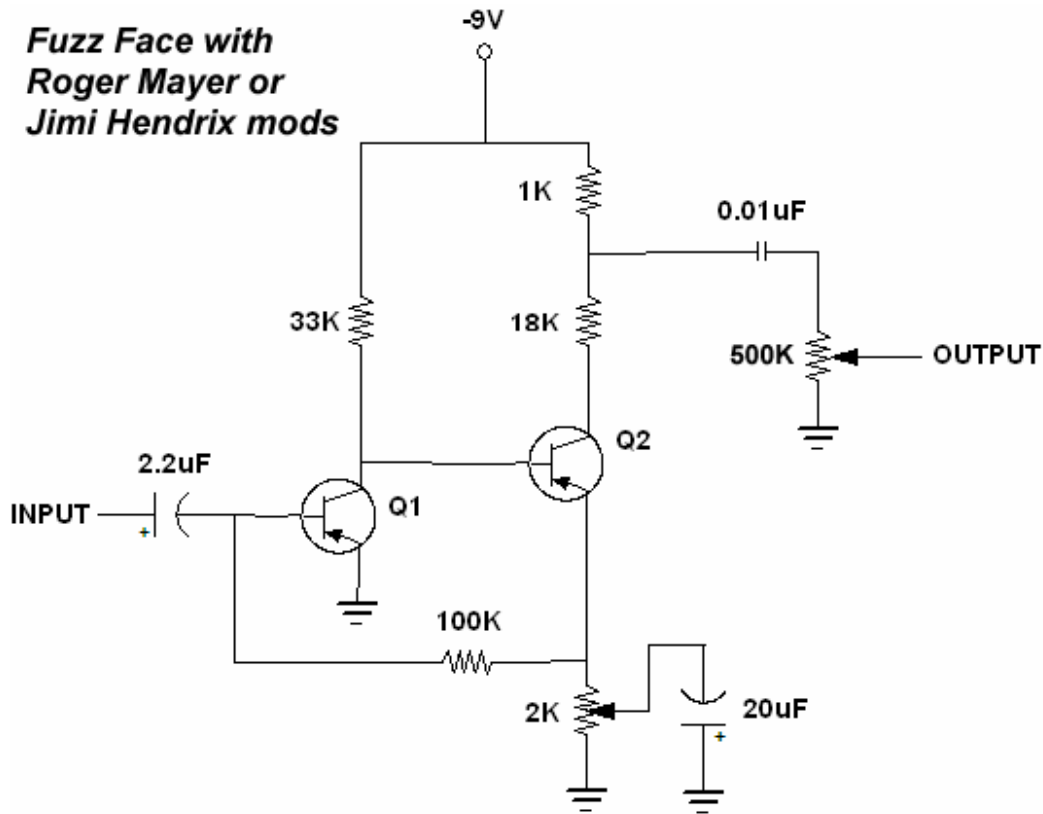


Figure 2.3 – Fuzz Face with “Roger Mayer” or “Jimi Hendrix” mods [8]

### Fuller Mod

Mike Fuller, a renowned creator of guitar effects and owner of Fulltone Custom Effects, also created a spin on the Fuzz Face. His changes were:

- Adding a 1 k $\Omega$  pot in series with the 470  $\Omega$  output resistor
- Adding a 50 k $\Omega$  pot in series with the input before the input capacitor

The 1k $\Omega$  pot acts as a variable resistor. As seen in the figure below, the control is shorted to one end lug. Therefore the output resistance can vary between 470 - 1.47 k $\Omega$ . This has a similar effect to the Hendrix Mod, increasing the output and gain of Q2. The 50 k $\Omega$  pot is set up in the same way. However, it has an opposite effect. By creating a higher source impedance, the guitar pickup acts in a more linear fashion, essentially lessening the amount of distortion.

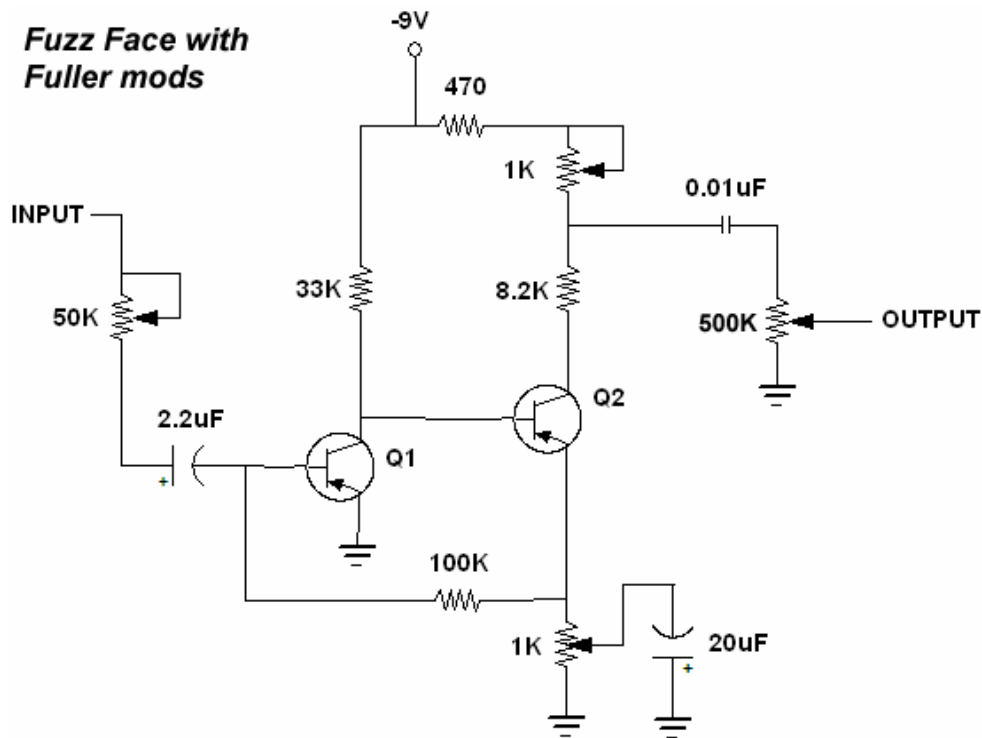


Figure 2.4 – Fuzz Face with Fuller mods [9]

### Vox Tone Bender<sup>2</sup> Mod

Vox Amplification, a leading guitar effects manufacturer, came out with their own take on the Fuzz Face. The changes are a bit more extensive, but definitely based on the same circuit. Notable changes include:

- Using NPN silicon transistors instead of PNP germanium
- Reducing the values of the coupling capacitors
- Adding a resistance in series with the pot at the emitter of Q2
- Adding a resistance in parallel with the output pot

Also, using silicone transistors makes it necessary to change the values of the biasing resistors. A complete schematic of the Vox Tone Bender is shown below.

<sup>2</sup> The Vox Tone Bender is a registered trademark of Vox Amplification, Ltd.



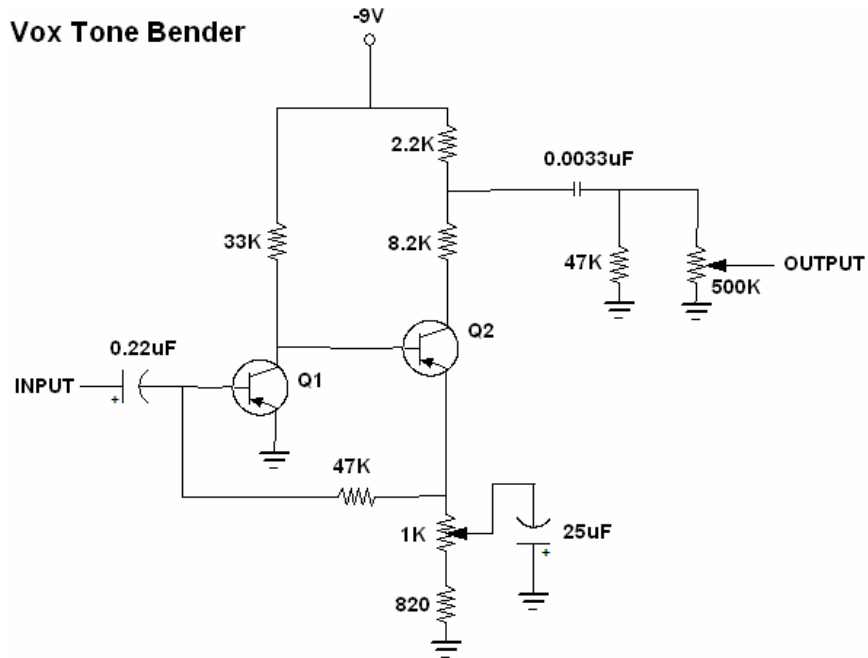


Figure 2.5 – Vox Tone Bender 5/67 [10]

In addition to these mods, R. G. Keen makes some other suggestions [3]. Increasing the values of the capacitors will increase the bass response of the circuit. From a filter standpoint, this effectively decreases the cutoff frequency of the high-pass filters at the input and the output. Also, high frequency taming capacitors can be added to soften the distortion. Adding a 100 – 680 pF capacitor across the collector resistor of Q1 or adding a 10 – 100 pF capacitor from the collector to the base on Q2 will accomplish this.

### 2.3 The Ebers-Moll Transistor Model

The Ebers-Moll model is an ideal model for a bipolar transistor. It consists of two diodes, described by the classic exponential voltage/current relationship for diodes, and two current sources. Early versions of SPICE used this model when simulating transistors [11]. Current SPICE programs use the Gummel-Poon transistor model. In addition to the diode currents, the Gummel-Poon model accounts for high bias level effects, such as

junction capacitances. The Ebers-Moll model is obtained in modern SPICE programs by leaving out these parameters. The model is typically drawn for NPN transistors, with current flowing into the base. By reversing the current directions and voltage notations, the model for PNP transistors is obtained, as shown below.

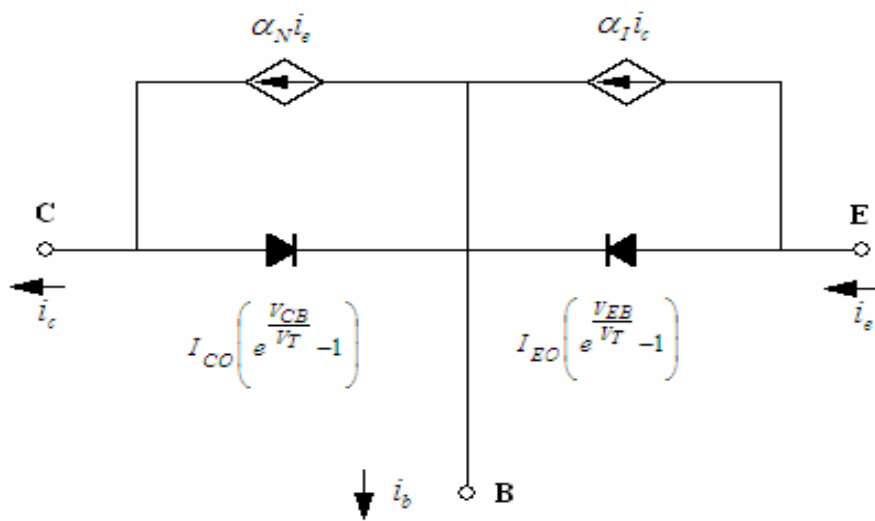


Figure 2.6 – PNP Ebers-Moll transistor model

For PNP transistors, the currents in the above figure are related by the following equations [12]:

$$\begin{aligned}
 I_E &= \alpha_I I_C + I_{EO} \left( e^{\frac{V_{EB}}{V_T}} - 1 \right), \\
 I_C &= \alpha_N I_E + I_{CO} \left( e^{\frac{V_{CB}}{V_T}} - 1 \right).
 \end{aligned}
 \tag{2.1}$$

where the parameters  $\alpha_N$ ,  $\alpha_I$ ,  $I_{EO}$ , and  $I_{CO}$  are given by

$$\begin{aligned}
\alpha_N &= \frac{I_C}{I_E} \Big|_{V_{CB}=0}, \\
\alpha_I &= \frac{I_E}{I_C} \Big|_{V_{EB}=0}, \\
I_C &= I_{CO}, \\
I_E &= -I_{EO}, \\
\alpha_I I_{CO} &= \alpha_N I_{EO}.
\end{aligned} \tag{2.2}$$

The parameters in (2.2) can be extracted in lab. At low bias levels, these equations give a good approximation of the nonlinear operation of BJTs. Since the Fuzz Face operates at low bias levels, the Ebers-Moll model was selected to model the transistors over the more complex Gummel-Poon model.

## CHAPTER 3: METHODOLOGY

The Fuzz Face was explored in three different ways. First, the circuit was simulated with PSPICE to gain an understanding of what the output should look like, and how changes in components would change the output. Second, an analog Fuzz Face circuit was built in lab to measure its physical properties, and characteristics of components were measured. Finally, a digital model was created to be used in a digital guitar effects processor.

### 3.1 Circuit Simulation

The Fuzz Face circuit was constructed and simulated using OrCAD PSPICE as shown below. It exhibits the asymmetrical clipping expected from initial circuit analysis. The circuit was driven with different signal strengths meant to approximate guitar signals.

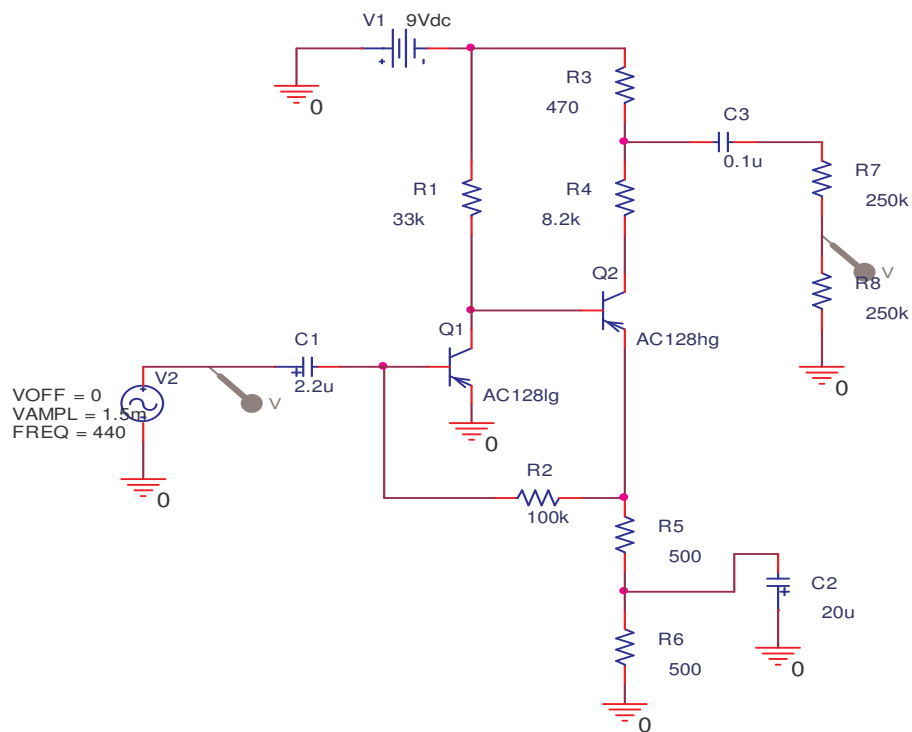


Figure 3.1 – PSPICE schematic of the Fuzz Face

The AC source was chosen to have an amplitude between 0.5 mV and 2 mV at a frequency of 440 Hz. This voltage range was chosen because it provided a range of outputs from no noticeable distortion to heavy distortion. This was also estimated to be a reasonable output voltage for a guitar. The transistor models AC128lg and AC128hg were created in PSPICE to model the germanium transistors used in these devices. The transistor models included with PSPICE, as well as virtually every transistor model available online, are silicon. One of the major differences between silicon transistors and germanium transistors is that the forward bias voltage for germanium is around 0.2 V, whereas it is around 0.7 V for silicon. Initially, gains were set at 70 and 120, which have been determined by musicians over the years to be the “sweet spot” for these devices [3]. Our first models were constructed by taking an existing PSPICE model for a National Semiconductor silicon transistor (with PID 66), changing the forward bias voltages ( $V_{je}$  and  $V_{jc}$ ) to 0.2 V, changing  $B_f$  to the desired gain, and removing the Early voltage (in effect making the Early voltage infinite). The only difference between the low  $\beta$  (AC128lg) and high  $\beta$  (AC128hg) are transistor models was the value of  $B_f$ .

```
.model AC128lg PNP(Bf=70 Vje=0.2 Is=1.41f Xti=3 Eg=1.11
Ne=1.5 Ise=0 Ikf=80m Xtb=1.5 Br=4.977 Nc=2
Isc=0 Ikr=0 Rc=2.5 Cjc=9.728p Mjc=0.5776
Vjc=0.2 Fc=0.5 Cje=8.063p Mje=0.3677 Tr=33.42n
Tf=179.3p Itf=0.4 Vtf=4 Xtf=6 Rb=10)
```

Figure 3.2 – PSPICE AC128 model parameters

A complete list of PSPICE bipolar transistor parameters can be found in Appendix A. We also created a second model that only defined  $B_f$ ,  $V_{je}$ , and  $V_{jc}$ , since the Ebers-Moll equations do not model the other parameters of these transistors. However, the difference

between the PSPICE runs with the two models was found to be negligible. Because these simulations were only for initial evaluation, further simulations were not conducted with the simplified models. Bf was also modified to see how much different gains would affect circuit operation.

### **3.2 Measurements from a Real Circuit**

To accurately interpret our laboratory results, the characteristics of real transistors and resistors used in our experiments were determined. The Agilent 34401A Digital Multimeter was used to determine the exact resistance of the resistors. We selected resistors that were as close to ideal values as possible. The ideal and exact resistances of all resistors used in the Fuzz Face circuit and the separate transistor testing circuit described below are listed in Table 3.1.

*Table 3.1 – Resistance measurements*

<b>Ideal (<math>\Omega</math>)</b>	<b>Actual (<math>\Omega</math>)</b>
470	465.8
1 K	0.99 K
2.472 K	2.39 K
8.2 K	8.2585 K
33 K	32.76 K
100 K	98.65 K
2.2 M	2.1998 M

A simple circuit (shown below) was constructed to determine the true  $\beta$  of each transistor [3]. These exact resistor values were selected so that the gain could be found simply by the equation  $V_{csc} - V_{cso} = \beta$ .

## Germanium Transistor Testing

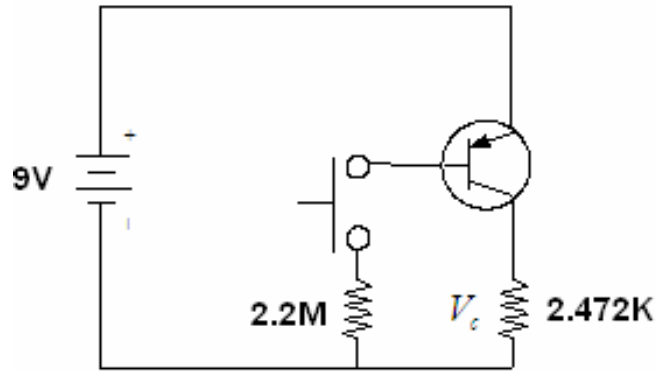


Figure 3.3 – Germanium transistor testing

A voltage  $V_{csc}$  (collector, switch closed) was read across the 2.472K collector resistor with the switch closed, and  $V_{cso}$  (collector, switch open) was read across the same resistor with the switch open. To ensure accuracy, the Hewlett-Packard E3630A DC Power Supply was used to set the voltage at 9V in lieu of a 9V battery. The Agilent Multimeter was used for the voltage measurements. With this set up,  $V_{cso}$  will display 2.472 V for every milliamp of leakage. Therefore, the leakage (in microamps) can be found via the

expression  $leakage = \frac{V_{cso}}{0.002472}$  [3]. The gain is found by the expression  $\beta = \frac{V_{csc} - V_{cso}}{V_{cso}}$ .

Table 3.2 – Values for computing the true  $\beta$

	$V_{csc}$ (V)	$V_{cso}$ (V)	Leakage ( $\mu$ A)	$\beta$
<b>AC128-1</b>	1.394	0.61	247	78.4
<b>AC128-2</b>	2.07	1.25	506	82
<b>AC128-3</b>	1.285	0.659	267	62.6
<b>AC128-4</b>	1.73	0.789	319	94.1
<b>AC128-5</b>	1.64	0.702	284	93.8
<b>AC128-6</b>	1.4	0.627	254	77.3
<b>AC125</b>	3.59	2.29	926	130

Six of the transistors were AC128s purchased off of ebay. The seventh, an AC125, was inherited by Prof. Aaron Lanterman from his grandfather, James Lanterman.

No transistors were found to be ideal for the purposes of the Fuzz Face ( $\beta$  of 70 or 120, and leakage less than  $200 \mu\text{A}$  [3]), so two pairs were selected whose characteristics were close to the ideal values. These different pairs would also help determine how much the gains or leakage currents affected the circuit. Transistor pair AC128-3/AC128-4 (Q1/Q2) was selected as the first pair and AC128-6/AC125 was selected as the second pair.

Next the DC parameters of the Fuzz Face circuit were determined. For easier measurements, a four-node simplification of the Fuzz Face was constructed.

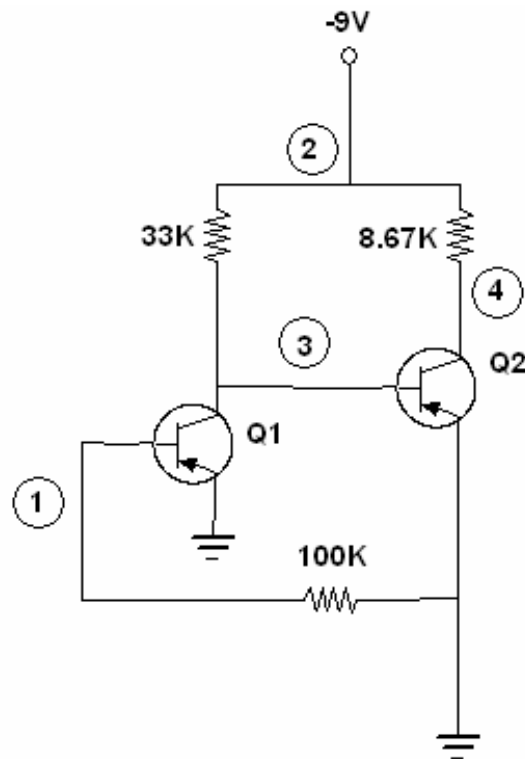


Figure 3.4 – Fuzz Face circuit, fuzz control at 0

The resistances at the output were lumped together to form the  $8.67 \text{ k}\Omega$  resistor, and the fuzz pot was eliminated, essentially setting it to zero. The node voltages shown in Table 3.3 were measured.



Table 3.3 – Nodal voltages, fuzz control at zero

Transistor Pair (Q1 / Q2)	Node 1 (mV)	Node 2 (V)	Node 3 (mV)	Node 4 (mV)
AC128-3 / AC128-4	-48.55	-8.992	-126.1	-16.2
AC128-6 / AC125	-51.22	-8.992	-112.1	-17.3
AC128-4 / AC128-3	-52.73	-8.992	-121.2	-19.2

To determine how the circuit would react with the fuzz control set at the other end, the fuzz pot was reintroduced. The resistances at the output, however, remained lumped. This yielded a five-node version of the fuzz face circuit. The node voltages were again measured, yielding Table 3.4.

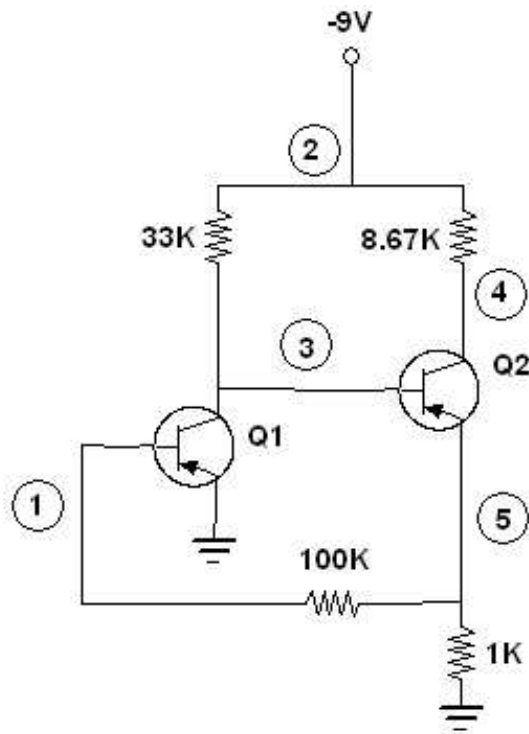


Figure 3.5 – Fuzz Face circuit, fuzz control at 1K

Table 3.4 – Nodal voltages, fuzz control at 1K

Transistor Pair (Q1 / Q2)	Node 1 (mV)	Node 2 (V)	Node 3 (mV)	Node 4 (mV)	Node 5 (mV)
AC128-3 / AC128-4	-47.16	-8.997	-136.6	-14.3	-0.477
AC128-6 / AC125	-49.76	-8.996	-121.5	-15.76	-0.5
AC128-4 / AC128-3	-50.95	-8.998	-132.6	-16.19	-0.515

Finally, some more selected parameters of the transistors were measured. The parameters of interest were the normal active collector-base current gain  $\alpha_N$ , the inverse active collector-base current gain  $\alpha_I$ , and the saturation leakage currents  $I_{EO}$  and  $I_{CO}$ . These parameters were extracted according to the methods detailed by David Perlman [11]. However, as his models are for NPN transistors, the currents and voltages in his exposition must be reversed. The relationships of these parameters to circuit characteristics are shown in (2.2).

*Table 3.5 – Transistor parameters*

	AC128-3	AC128-4
$\alpha_N$	0.99	0.992
$\alpha_I$	0.914	0.918
$I_{EO}$	3.4 $\mu$ A	3.9 $\mu$ A
$I_{CO}$	4.2 $\mu$ A	3 $\mu$ A

Interestingly, these parameters were quite close for these two transistors, whereas their gains and leakage currents were not as similar. Also,  $I_{EO}$  was less than  $I_{CO}$  for AC128-3 whereas  $I_{EO}$  was more than  $I_{CO}$  for AC128-4.

### **3.3 Digital Circuit Design**

#### *3.3.1 Filter Effects*

The capacitors in the Fuzz Face circuit were originally included to decouple the circuit. They were only intended for blocking DC only and not for filtering, but musicians claim the effects on the tone of the guitar cannot be ignored. The capacitors in the fuzz face

form high-pass filters. The cutoff frequency of a single-pole RC high-pass filter is given by

$$f_c = \frac{1}{2\pi RC}, \quad (3.1)$$

where C is the capacitance in the signal path and R is the resistance connecting to ground.

At the input, the resistance is  $100K + R_{fuzz}$ . However, since  $R_{fuzz}$  is at most 1K, its effects may be ignored. Therefore, the cutoff frequency at the input is given by

$$f_c = \frac{1}{2\pi(100K)(2.2\mu)} = 0.72Hz. \quad (3.2)$$

Since the lowest frequency produced by a guitar is approximately 82.4 Hz, the effects of this capacitor were deemed negligible. Similarly, the cutoff induced by the output capacitor is

$$f_c = \frac{1}{2\pi(500K)(0.01\mu)} = 31.8Hz. \quad (3.3)$$

Though higher than the cutoff of the input capacitor, this frequency is still much lower than 82.4 Hz. Hence, this simple analysis suggests that the capacitors do not have a significant effect on the guitar signal. Since the digital signal in our recreation does not need to be decoupled, we decided that the capacitors and their filtering effects did not need to be modeled to accurately model the Fuzz Face, at least at this stage.

### 3.3.2 Transistor Modeling

To aim for a mathematical Fuzz Face model that would not require PSPICE runs, the Ebers-Moll equations were selected to model the transistors. Using the parameters in Table 3.5, these equations give the transistor currents  $I_E$  and  $I_C$  for particular values of

$V_{EB}$  and  $V_{CB}$ . To create a model of the Fuzz Face, the entire input/output relationship must be obtained. We seek an equation for  $V_{OUT}$  in terms of  $V_{IN}$ . The five-node circuit shown on the following page was used as a basis for our derivations.

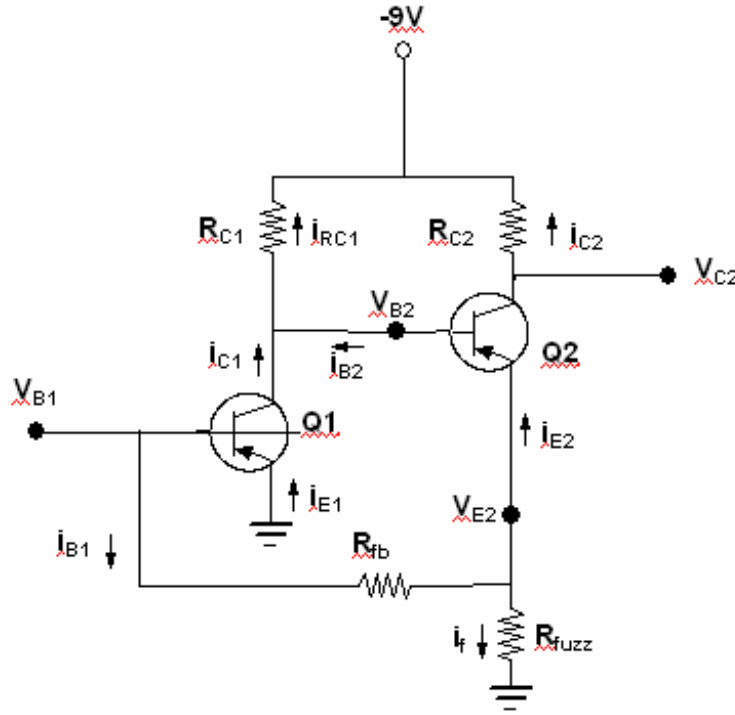


Figure 3.6 – Circuit in terms of transistor currents

Applying Kirchoff's Current Law and Ohm's Law to the circuit obtains the following equations:

$$\begin{aligned}
 i_{B1} &= \frac{V_{B1} - V_{E2}}{R_{fb}}, & i_{C2} &= \frac{V_{C2} - V_{DC}}{R_{C2}}, \\
 i_{C1} + i_{B2} &= i_{RC1}, & i_{B1} &= i_{E2} + i_f, \\
 i_{RC1} &= \frac{V_{B2} - V_{DC}}{R_{C1}}, & i_f &= \frac{V_{E2}}{R_{fuzz}}.
 \end{aligned} \tag{3.4}$$

Since we desire the output  $V_{OUT} = V_{C2}$  in terms of the input  $V_{IN} = V_{B1}$ , the currents  $i_{B1}$ ,  $i_{C1}$ ,  $i_{RC1}$ ,  $i_{B2}$ , and  $i_{C2}$  need to be eliminated. Also, the intermediate voltages  $V_{B2}$  and  $V_{E2}$

should be eliminated. After solving for the currents  $i_{C1}$ ,  $i_{E1}$ ,  $i_{C2}$ , and  $i_{E2}$  in terms of voltages and resistances, the currents may be plugged into the Ebers-Moll equations (2.1). The first step is to substitute within (3.4) where possible, yielding

$$\begin{aligned} i_{B1} &= \frac{V_{B1} - V_{E2}}{R_{fb}}, & i_{C2} &= \frac{V_{C2} - V_{DC}}{R_{C2}}, \\ i_{C1} + i_{B2} &= \frac{V_{B2} - V_{DC}}{R_{C1}}, & i_{B1} &= i_{E2} + \frac{V_{E2}}{R_{fuzz}}. \end{aligned} \quad (3.5)$$

As seen above,  $i_{C2}$  is already in terms of a voltage divided by a resistance. Also, by setting the two expressions for  $i_{B1}$  equal to each other,  $i_{E2}$  is obtained in voltage/resistance form. To proceed further, we use the relationship  $i_B = i_E - i_C$  was to eliminate  $i_{B2}$ . In summary, we now have

$$\begin{aligned} i_{E2} &= \frac{V_{B1} - V_{E2}}{R_{fb}} - \frac{V_{E2}}{R_{fuzz}}, \\ i_{C1} + i_{E2} - i_{C2} &= \frac{V_{B2} - V_{DC}}{R_{C1}}, \\ i_{E1} - i_{C1} &= \frac{V_{B1} - V_{E2}}{R_{fb}}. \end{aligned} \quad (3.6)$$

The equation for  $i_{E2}$  can be simplified, and then it and  $i_{C2}$  may be plugged into the second equation in (3.6) to get  $i_{C1}$  in its voltage/resistance form:

$$\begin{aligned} i_{E2} &= \frac{V_{B1}}{R_{fb}} - V_{E2} \left( \frac{1}{R_{fb}} + \frac{1}{R_{fuzz}} \right), \\ i_{C1} &= \frac{V_{B2} - V_{DC}}{R_{C1}} + \frac{V_{C2} - V_{DC}}{R_{C2}} - \frac{V_{B1}}{R_{fb}} + V_{E2} \left( \frac{1}{R_{fb}} + \frac{1}{R_{fuzz}} \right). \end{aligned} \quad (3.7)$$

Finally, the last current  $i_{E1}$  is solved for by substituting this new expression for  $i_{C1}$  into the last equation (3.6). The final expressions for the four currents are

$$i_{E1} = \frac{V_{B2} - V_{DC}}{R_{C1}} + \frac{V_{C2} - V_{DC}}{R_{C2}} + \frac{V_{E2}}{R_{fuzz}}, \quad (3.8)$$

$$i_{C1} = \frac{V_{B2} - V_{DC}}{R_{C1}} + \frac{V_{C2} - V_{DC}}{R_{C2}} - \frac{V_{B1}}{R_{fb}} + V_{E2} \left( \frac{1}{R_{fb}} + \frac{1}{R_{fuzz}} \right), \quad (3.9)$$

$$i_{E2} = \frac{V_{B1}}{R_{fb}} - V_{E2} \left( \frac{1}{R_{fb}} + \frac{1}{R_{fuzz}} \right), \quad (3.10)$$

$$i_{C2} = \frac{V_{C2} - V_{DC}}{R_{C2}}. \quad (3.11)$$

These expressions are now ready for substitution into the Ebers-Moll equations. To write the complete Ebers-Moll equations for this circuit, the emitter-base and collector-base voltages are needed. These are easily found by inspection of the circuit in Figure 3.6.

$$\begin{aligned} V_{EB1} &= -V_{B1}, \\ V_{CB1} &= V_{B2} - V_{B1}, \\ V_{EB2} &= V_{E2} - V_{B2}, \\ V_{CB2} &= V_{C2} - V_{B2}. \end{aligned} \quad (3.12)$$

The complete Ebers-Moll equations for this circuit may now be written for each transistor. Substituting in the emitter-base and collector-base voltages, the following equations are obtained:

$$\begin{aligned} i_{E1} &= \alpha_{I1} i_{C1} + I_{EO1} \left( e^{\frac{-V_{B1}}{V_T}} - 1 \right), \\ i_{C1} &= \alpha_{N1} i_{E1} + I_{CO1} \left( e^{\frac{V_{B2} - V_{B1}}{V_T}} - 1 \right), \\ i_{E2} &= \alpha_{I2} i_{C2} + I_{EO2} \left( e^{\frac{V_{E2} - V_{B2}}{V_T}} - 1 \right), \\ i_{C2} &= \alpha_{N2} i_{E2} + I_{CO2} \left( e^{\frac{V_{C2} - V_{B2}}{V_T}} - 1 \right). \end{aligned} \quad (3.13)$$

Now the values for  $i_{C1}$ ,  $i_{E1}$ ,  $i_{C2}$ , and  $i_{E2}$  from (3.8)-(3.11) must be substituted in (3.13) to eliminate all currents in the equation. Bringing all the terms to the same side yields

$$(1 - \alpha_{I1}) \left( \frac{V_{B2} - V_{DC}}{R_{C1}} + \frac{V_{C2} - V_{DC}}{R_{C2}} + \frac{V_{E2}}{R_{fuzz}} \right) + \alpha_{I1} \left( \frac{V_{B1} - V_{E2}}{R_{fb}} \right) - I_{EO1} \left( e^{\frac{-V_{B1}}{V_T}} - 1 \right) = 0, \quad (3.14)$$

$$(1 - \alpha_{N1}) \left( \frac{V_{B2} - V_{DC}}{R_{C1}} + \frac{V_{C2} - V_{DC}}{R_{C2}} + \frac{V_{E2}}{R_{fuzz}} \right) + \frac{V_{E2} - V_{B1}}{R_{fb}} - I_{CO1} \left( e^{\frac{V_{B2} - V_{B1}}{V_T}} - 1 \right) = 0, \quad (3.15)$$

$$\frac{V_{B1}}{R_{fb}} - V_{E2} \left( \frac{1}{R_{fb}} + \frac{1}{R_{fuzz}} \right) - \alpha_{I2} \frac{V_{C2} - V_{DC}}{R_{C2}} - I_{EO2} \left( e^{\frac{V_{E2} - V_{B2}}{V_T}} - 1 \right) = 0, \quad (3.16)$$

$$\frac{V_{C2} - V_{DC}}{R_{C2}} - \alpha_{N2} \left( \frac{V_{B1}}{R_{fb}} - V_{E2} \left( \frac{1}{R_{fb}} + \frac{1}{R_{fuzz}} \right) \right) - I_{CO2} \left( e^{\frac{V_{C2} - V_{B2}}{V_T}} - 1 \right) = 0. \quad (3.17)$$

These four equations and four variables form a nonlinear system of equations. Because of the exponential components, it is not possible to get  $V_{C2}$  directly in terms of  $V_{B1}$ .

However through substitution,  $V_{E2}$  may be eliminated, and an equation giving  $V_{C2}$  in terms of only  $V_{B1}$  and  $V_{B2}$  may be obtained. Intermediate variables, in terms of  $V_{B1}$  and  $V_{B2}$ , help make the equations more manageable:

$$\mathbf{K}_1 = \frac{I_{EO1} \left( e^{\frac{-V_B}{V_T}} - 1 \right) - \alpha_{I1} \frac{V_{B1}}{R_{fb}} - (1 - \alpha_{I1}) \left( \frac{V_{B2} - V_{DC}}{R_{C1}} - \frac{V_{DC}}{R_{C2}} \right)}{\left( \frac{1 - \alpha_{I1}}{R_{fuzz}} - \frac{\alpha_{I1}}{R_{fb}} \right)}, \quad (3.18)$$

$$\mathbf{K}_2 = \frac{I_{CO1} \left( e^{\frac{V_{B2} - V_{B1}}{V_T}} - 1 \right) + \frac{V_{B1}}{R_{fb}}}{(1 - \alpha_{N1})} - \frac{V_{B2} - V_{DC}}{R_{C1}} + \frac{V_{DC}}{R_{C2}}, \quad (3.19)$$

$$V_{C2} = \frac{\mathbf{K}_2 - \mathbf{K}_1 \left( \frac{1}{R_{fuzz}} - \frac{1}{R_{fb}(1 - \alpha_{N1})} \right)}{\left( \frac{1}{R_{C2}} - \frac{1 + \alpha_{I1}}{R_{C2} \left( \frac{1 - \alpha_{I1}}{R_{fuzz}} - \frac{\alpha_{I1}}{R_{fb}} \right)} \right) \left( \frac{1}{R_{fuzz}} + \frac{1}{R_{fb}(1 - \alpha_{N1})} \right)}, \quad (3.20)$$

$$\frac{\mathbf{K}_3}{R_{C2}} - \alpha_{N2} \frac{V_{B1}}{R_{fb}} - \left( \mathbf{K}_1 - \frac{(1 - \alpha_{I1}) \left( \frac{\mathbf{K}_3}{R_{C2}} \right)}{\frac{1 - \alpha_{I1}}{R_{fuzz}} - \frac{\alpha_{I1}}{R_{fb}}} \right) \left( \frac{1}{R_{fb}} - \frac{1}{R_{fuzz}} \right) - I_{CO2} \left( e^{\frac{\mathbf{K}_3 - V_{B2}}{V_T}} - 1 \right) = 0. \quad (3.21)$$

Substituting (3.20) into (3.21) gives an expression relating  $V_{B1}$  and  $V_{B2}$ . By reducing the system of four equations to one equation with only one unknown value, iterative nonlinear solution techniques may be used to find the unknown value  $V_{B2}$ .

Our attempts to simplify the above equations did not lead anywhere insightful, and since numerical techniques are needed to solve the equations anyway, they were left in these cumbersome forms. Since  $V_{B2}$  can be determined given  $V_{B1}$ , and since  $\mathbf{K}_1$  and  $\mathbf{K}_2$  are functions of  $V_{B1}$  and  $V_{B2}$ ,  $V_{C2}$  may now be found as a function of the input voltage  $V_{B1}$ . Equations 3.20 and 3.21 form the foundation of our MATLAB-based digital model.



Since these equations are so complex, further simplifications were made to the circuit to see if a more straightforward set of equations could be found. By assuming that the flow of current out of the base of each transistor is negligible,  $i_{B1}$  and  $i_{B2}$  may be removed. A new, simpler set of equations may be obtained using techniques similar to those used above:

$$\begin{aligned}
V_{B2} &= R_{C1} I_{CO1} \left( e^{\frac{-V_{B1}}{V_T}} - 1 \right) + V_{DC}, \\
V_{C2} &= R_{C2} \left[ \frac{V_{B1}}{R_f} - V_{E2} \left( \frac{1}{R_f} + \frac{1}{R_{fuzz}} \right) \right] + V_{DC}, \\
V_{E2} &= \left( \frac{1}{\frac{1}{R_f} + \frac{1}{R_{fuzz}}} \right) \left( \frac{V_{B1}}{R_f} - I_{CO2} \left( e^{\frac{-\left[ R_{C1} I_{CO1} \left( e^{\frac{-V_{B1}}{V_T}} - 1 \right) + V_{DC} \right]}{V_T}} e^{\frac{V_{E2}}{V_T}} - 1 \right) \right).
\end{aligned} \tag{3.22}$$

$V_{C2}$  can be found if we know the voltage  $V_{E2}$ . The last equation in (3.22) can be solved using nonlinear solution techniques.  $V_{E2}$  may be approximated with a first-order Taylor expansion of the exponential term with  $V_{E2}$  in the numerator, allowing  $V_{E2}$  to be solved explicitly, although approximately:

$$\begin{aligned}
V_{E2} &= \frac{\Lambda_1 \left( \frac{V_{B1}}{R_{fb}} - I_{CO2} (\Lambda_2 - 1) \right)}{1 + \frac{I_{CO2} \Lambda_1 \Lambda_2}{V_T}}, \\
\Lambda_1 &= \frac{1}{\frac{1}{R_f} + \frac{1}{R_{fuzz}}}, \\
\Lambda_2 &= e^{\frac{-\left[ R_{C1} I_{CO1} \left( e^{\frac{-V_{B1}}{V_T}} - 1 \right) + V_{DC} \right]}{V_T}}.
\end{aligned} \tag{3.23}$$

This approximation may provide a good first guess for numerical solution algorithms.

## CHAPTER 4: RESULTS

### 4.1 Simulated Output from PSPICE

Several PSPICE simulations were run, as stated in Chapter 3.1. The output waveforms seem reasonably close to the “ideal” output waveforms found in the literature.

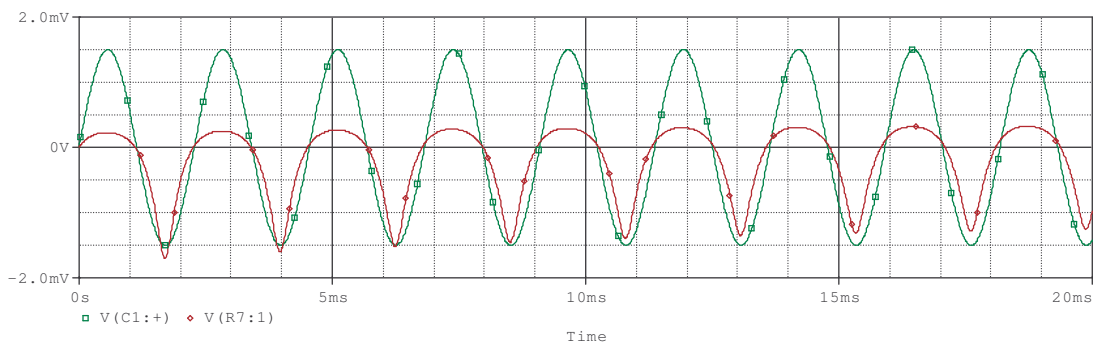


Figure 4.1 – PSPICE simulation of the Fuzz Face

Green indicates the input waveform (arbitrarily chosen at 440 Hz, concert A), and red is the output. The default gains used for the transistors were 70 for Q1 and 120 for Q2. The output was similar to the “ideal” asymmetrical clipping Keen presented [13].

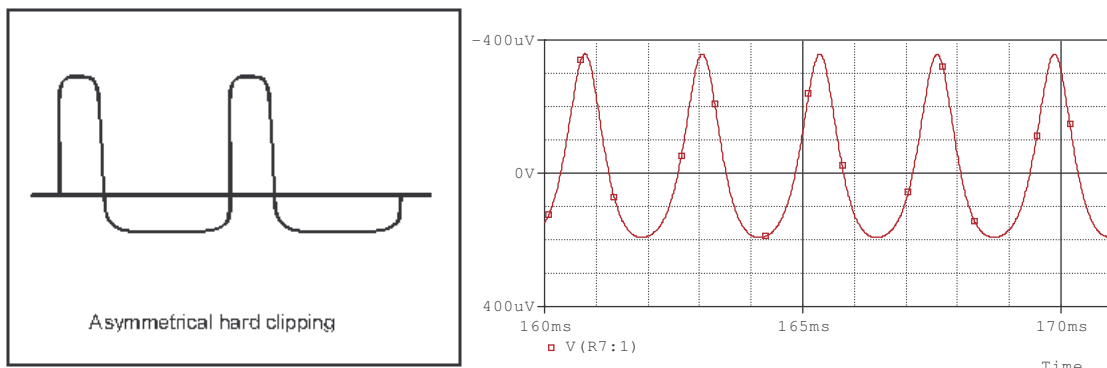


Figure 4.2 – Comparison of theoretical clipping to the output [14]; notice we flipped the vertical axis on the graph on the right to make the similarity more apparent.

We found that Q1 greatly affects the type of clipping. When Q1 was increased, it created more dramatic hard clipping. Hard clipping introduces more 7<sup>th</sup> order and higher harmonics, which are harsher sounding to the ear than the 3<sup>rd</sup> and 5<sup>th</sup> order harmonics introduced by soft clipping. The first figure below shows a Fast Fourier Transform of an output given an input of 1.5 mV, shown in Figure 4.1. By the 4<sup>th</sup> harmonic, the amplitude is almost zero. However, in the second figure below, with an input amplitude of 3 mV, the spectrum extends to the 7<sup>th</sup> harmonic and beyond. The amplitude is also increased over the first case. Thus, as the amount of hard clipping increases, so does the number of harmonics and their amplitudes.

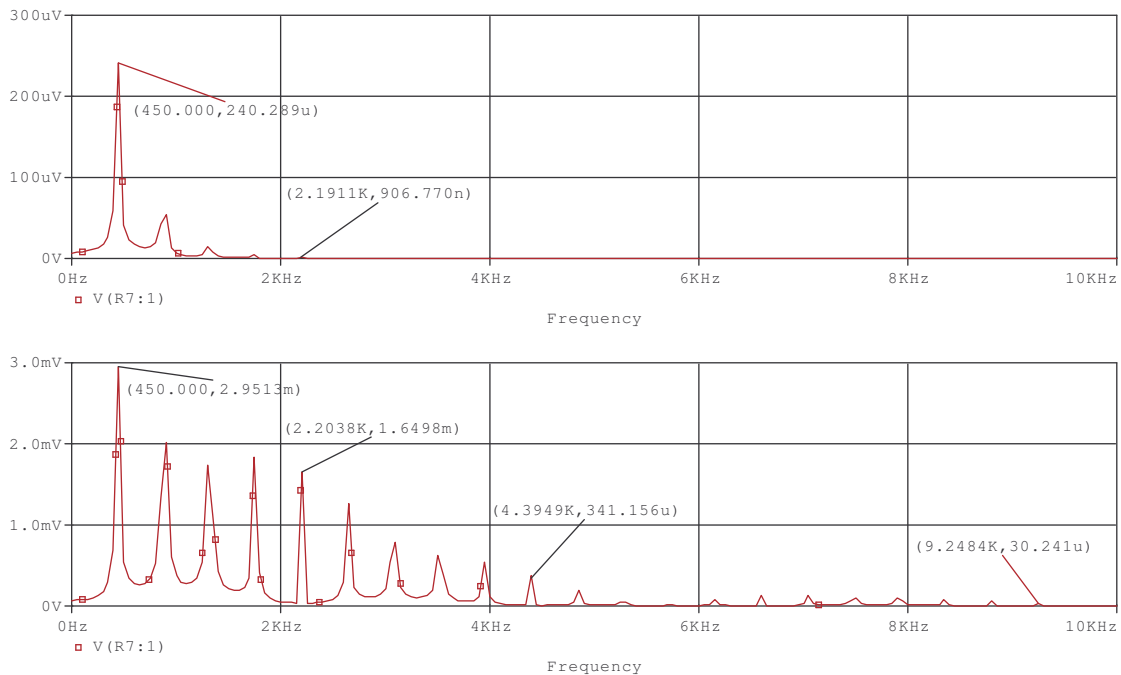


Figure 4.3 – FFT of soft (top) and hard (bottom) clipped outputs

Reducing the Q1 gain cause less clipping to be present in the output. We also found that Q2 mostly affected the overall gain of the circuit, and not so much the shape of the output wave. A higher Q2 gain actually decreased the overall gain, especially on the negative

swing. The effects of changing the  $\beta$  value for each transistor can be summarized in the following table.

Table 4.1 – Effects of changing transistor gains

	$\beta$ increased	$\beta$ decreased
Q1	Hard clipping increased	Clipping reduced
Q2	Overall gain decreased	Overall gain Increased

The bias voltages were also simulated. Using our original germanium transistor model shown in Figure 3.2, a bias voltage of -1.66 V was found at the collector of Q1, which is much higher than the -0.5 V bias that Keen [3] stated a “good” fuzz face should have. However, when the gain  $\beta$  and the saturation leakage currents  $I_{E0}$  and  $I_{C0}$ , obtained in lab, were added to the model, the circuit biased the collector of Q1 at -0.42 V, which is close to what was hoped for.

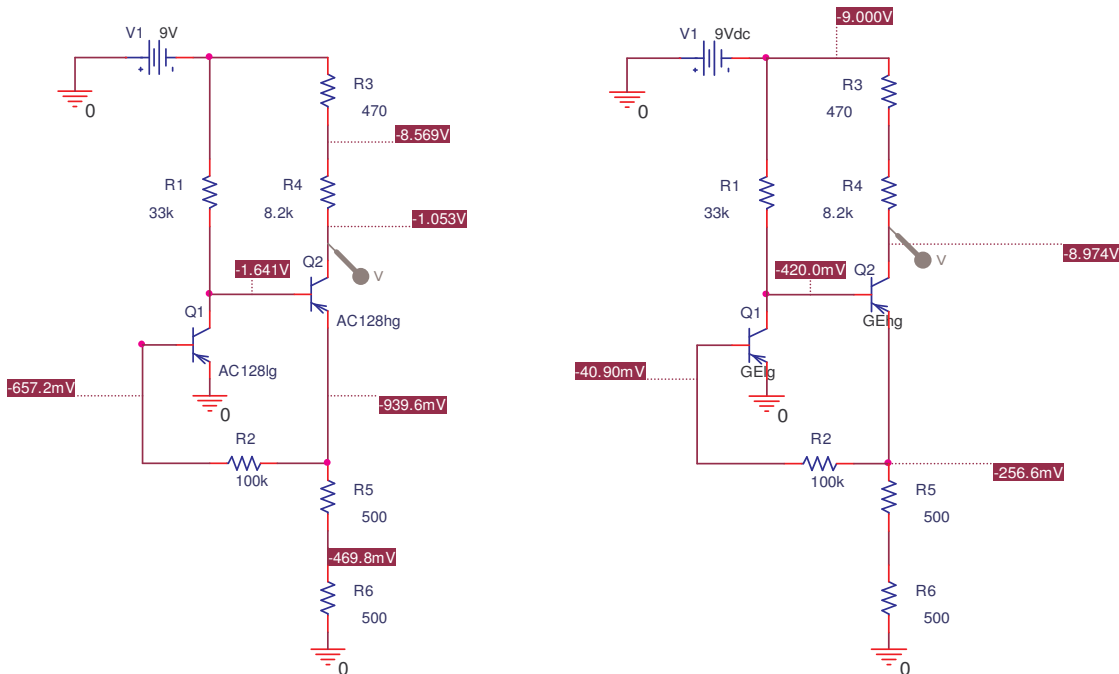


Figure 4.4 – DC bias voltages, before and after parameter modification

The bias voltage at the base of Q1 was -40 mV, which is similar to what was measured in lab. However, the rest of the simulated bias voltages were different than what was measured. This could be due to leakage effects not accounted for in the model.

A new DC analysis was attempted after updating the SPICE transistor parameters; however, the output became pegged to the negative supply rail. We are uncertain why using non-ideal saturation leakage currents caused PSPICE to do this. The inability to plot the output with these updated transistor models made it impossible to get a truly accurate comparison between transfer characteristics obtained with PSPICE and MATLAB.

#### **4.2 Analog Circuit Parameters**

The parameters extracted from a real Fuzz Face circuit led to some important facts about the Fuzz Face. The properties of the transistors did not have a significant effect on the DC bias levels of the circuit. As described in Chapter 3.2, two pairs, AC128-3/AC128-4 and AC128-6/AC125, were tested. Q1 and Q2 were swapped in the first pair for a third test. Although these transistors had different properties, the resulting node voltages were nearly identical in both the four-node and five-node circuits. The maximum percent error with respect to the mean values is shown in the table below.

*Table 4.2 – Maximum percent error compared to mean*

	Node 1	Node 2	Node 3	Node 4	Node 5
4 Node Circuit	3.7%	0%	6.4%	9.3%	----
5 Node Circuit	4.3%	0%	6.7%	7.2%	4%
4 and 5 Node Circuits	5.8%	0%	10.3%	----	----

As shown in this table, values never deviated more than 10% from the mean value. All of these values were in millivolts, except for Node 2 which was the power supply, and its deviation was never more than 10 mV. Therefore, it appears that the transistor parameters have little effect on the DC bias levels of the circuit. In addition, the last line of the table shows the maximum percent error across values from both the four-node and five-node circuits where equal comparisons were possible. The errors are higher, but only slightly so. Therefore, adding in the fuzz pot (Node 5) also had minimal effect on the DC bias points of the circuit. Thus, the DC operation point of this circuit is fairly stable under a variety of transistor gains, leakage currents, and resistance settings. However, as shown in PSPICE, significant changes in the Ebers-Moll parameters of the transistors do have a significant effect on the bias point; therefore, these transistors have similar characteristics relative to the amount of change needed to affect the bias point.

Another important observation was that germanium transistors are extremely sensitive to temperature changes. After placing these transistors in the circuit, they needed up to five minutes to stabilize. Even heat from proximity could cause the readings to change rapidly. Furthermore, this implies that the readings are only valid at the temperature of the room at that time. While it is uncertain how much the temperature must change for the ear to be able to perceive a difference, it is conceivable that the same circuit would sound different in different locations. This extreme sensitivity to temperature underscores the benefit of having a digital model of this circuit.

The last analog parameters extracted were those associated with the Ebers-Moll equation. There was one anomaly concerning the saturation leakage currents. From (2.2), we can infer that  $I_{EO}$  will be less than  $I_{CO}$  if  $\alpha_I$  is less than  $\alpha_N$ . This holds true for AC128-3. However, our measured  $I_{EO}$  is greater than our measured  $I_{CO}$  for AC128-4, which contradicts the equations. There is no obvious explanation for this discrepancy. We may be observing highly nonideal effects that are not predicted by the Ebers-Moll model; there may also be limitations to our measurement procedures. However, since the goal is to model the behavior of the real transistors, the values that were obtained in lab were used when running the algorithm, with the hope that these unusual parameters give the “best fit” to the true transistor characteristics. As noted in Section 4.1 when these values were added to the SPICE transistor models, the circuit appeared to bias up correctly.

### **4.3 Algorithm Design**

The goal of the MATLAB algorithm was to accept a clean (i.e., not distorted) guitar signal as a WAV file, process it as a vector using the equations derived in Chapter 3.3.2, and output this distorted signal. As shown previously, the code must solve a nonlinear equation for  $V_{B2}$  given an input  $V_{B1}$ , after which  $V_{B1}$  and  $V_{B2}$  can be plugged in to another equation to find the output voltage  $V_{C2}$ . MATLAB’s ‘fzero’ function was used to solve the nonlinear equation. The left-hand side of (3.21) was programmed as a MATLAB function that is called by ‘fzero.’

A system modeling program was written to compute and display the transfer function of the model. It first declares all constants, particularly the resistances and transistor

parameters. This allows for easy modifications of the digital “components.” The transfer function is then computed over a pre-specified range by first solving for  $V_{B2}$  using the current value of  $V_{B1}$  via MATLAB’s ‘fzero’ function. The  $V_{B1}$  and  $V_{B2}$  values are then plugged into (3.18) and (3.19), and the resulting intermediate results are then plugged into (3.22) to find the output voltage  $V_{C2}$ . It is not necessary to go through the intermediate variables in MATLAB, but keeping them made the programming more manageable. The value of  $V_{C2}$  and the input for which it was calculated were appended to vectors. This portion looped until the desired range was covered. A range of -0.1 V to 0.1 V was selected. We originally stepped the input by 0.1 mV increments, but we later found that stepping by 1 mV intervals yielded virtually the same result with faster processing time. Using linear interpolation, the norms of the resulting outputs were only off by 0.5%. Increasing the interval size by an additional factor of ten caused the norms to be off by 50%, so 1 mV intervals were selected for the final program. The transfer function is stored in a .mat file.

A processing algorithm was written to actually process the guitar signals. It read in a guitar signal and its sampling frequency using MATLAB’s ‘wavread’ function. This vector, originally in stereo, is converted to mono. The output voltage vector is then computed for each index of the input voltage using MATLAB’s ‘interp1’ function to interpolate the transfer function previously computed by the system modeling program. This allows the output to be computed quickly, compared to solving the nonlinear equation for each index. The mean of the output vector is then subtracted from the output vector to ensure there is no DC offset. Finally, the output is scaled by its maximum value



to so that it does not exceed the range of  $[-1 \ 1]$ . Vectors outside that range would be unnaturally clipped by nature of MATLAB's 'sound' function. This ensures that any distortion in the audio signal only comes from the Fuzz Face model.

#### **4.4 MATLAB Algorithm Performance**

A clean (undistorted) guitar signal was recorded using a BOSS 1600-CD digital audio recorder and transferred to the computer via USB. This WAV file was read into MATLAB for processing. The algorithm did produce a distorted guitar output, but it did not perform perfectly compared with what we had hoped for. The transfer function resembled what was obtained in PSPICE, but it was not an exact match. (This is to be expected, since PSPICE models behavior that is not present in the Ebers-Moll model.) It also clipped to ground quite quickly, which may have resulted in clicking sounds being present in the output. Also, 'fzero' stopped calculating values just before -0.15 V, although the range over which the transfer function was obtained was sufficient for getting an output when the signal was biased correctly.

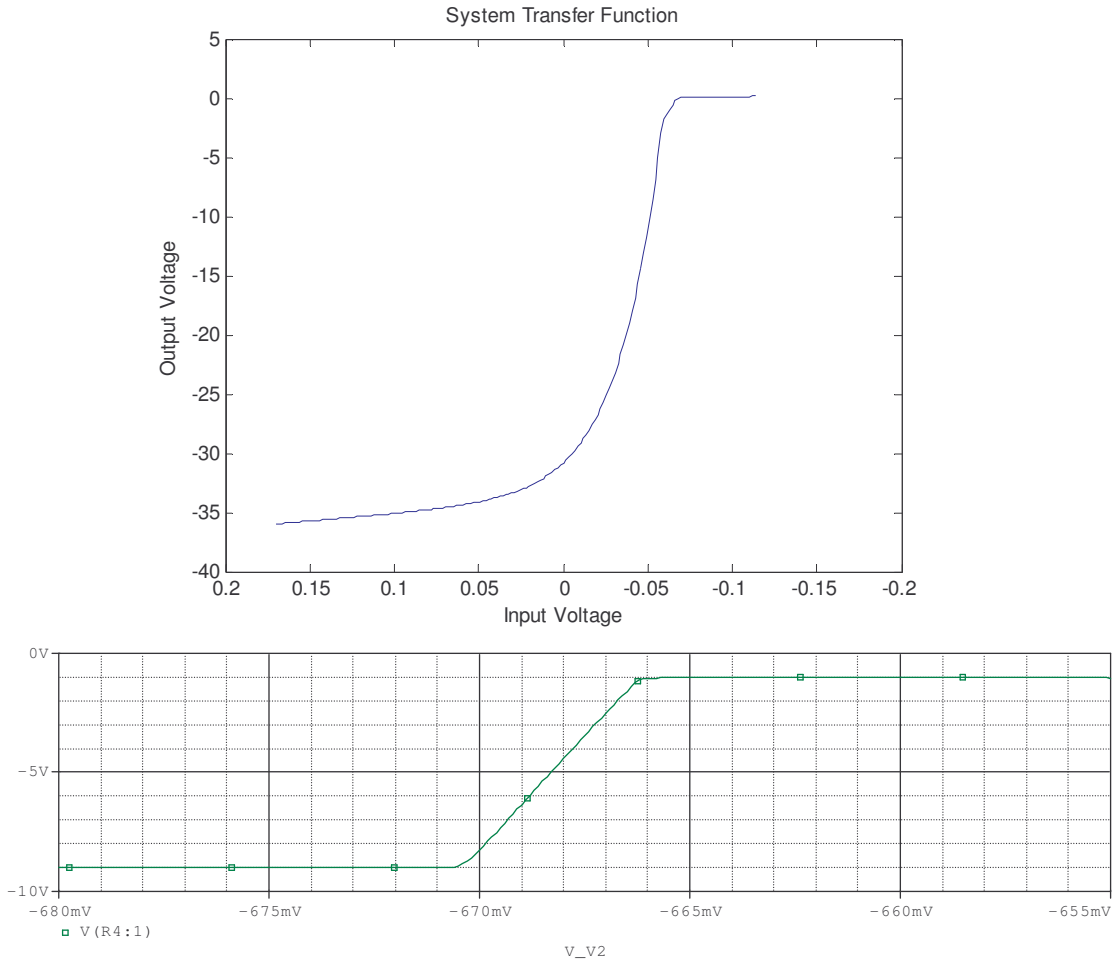
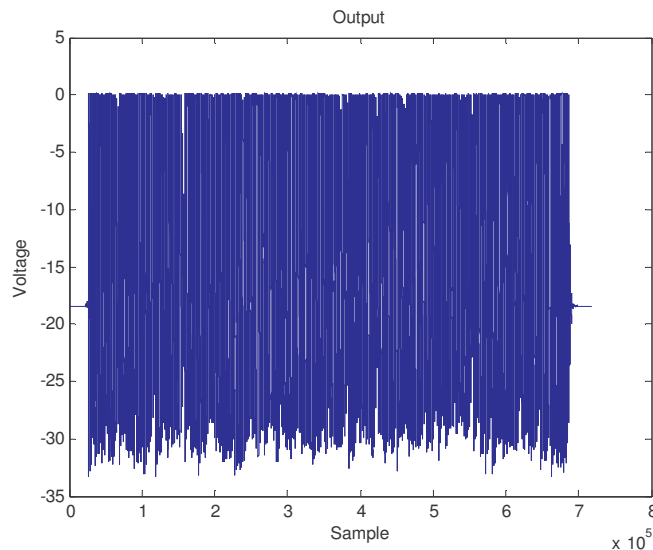


Figure 4.5 – Comparison of MATLAB and PSPICE transfer functions

The chosen bias points of the circuit were determined by inspection of Figures 4.5. The MATLAB model was “biased” at around -0.04 V, whereas the PSPICE model was biased at -0.66 V. The chosen MATLAB value closely corresponds to the -0.047 V expected circuit analysis. Since the incoming audio signal was centered around zero, the bias voltage was added to the input vector so as to place it in a reasonable operating region. This bias voltage had to be added manually, because the MATLAB algorithm currently has no way to calculate the correct bias point. A voltage of -0.04 V was added since that

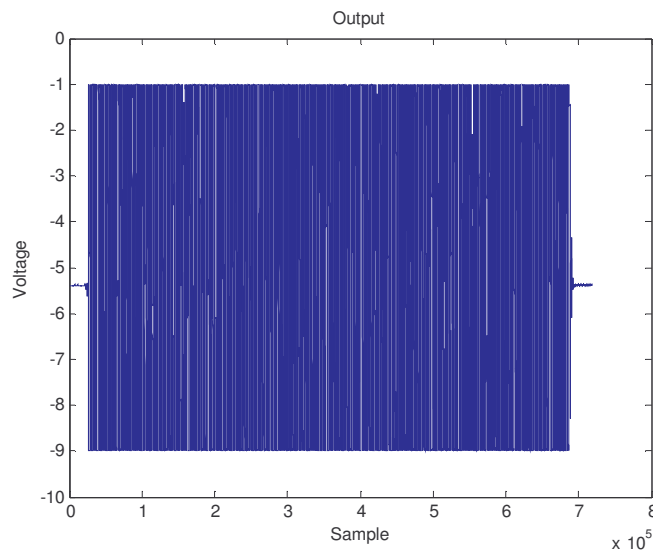
was the bias voltage calculated by PSPICE for the measured circuit parameters. We use PSPICE here, since we currently do not have an Ebers-Moll based mathematical analysis of the biasing behavior. Such an analysis, which would essentially involve removing the input voltage source in Section 3.3.2 and allowing  $V_{B1}$  to float, remains an avenue for future work. A sample output signal created using MATLAB with the manual biasing method is shown below.



*Figure 4.6 – Processed guitar output, Ebers-Moll modeling (overton\_william\_e\_200605\_mast\_fig46\_em-modeling.wav, 1920K)*

To determine how close the MATLAB result was to SPICE in terms of the effect on an audio file, the SPICE transfer function in Figure 4.5 was exported into Excel and read into MATLAB. The input bias calculated by PSPICE was -0.66 V. However, this placed the bias point on the flat portion of the transfer function, allowing for no voltage swing in one direction. This is not reasonable, so manual biasing was again used. The input bias was changed to -0.6685 V to place it at a reasonable operating point. This signal did not exhibit the clicks resulting from the Ebers-Moll modeling, perhaps because the transfer

function did not clip as hard to ground as seen in the MATLAB output in Figure 4.5. However, the input bias of -0.6685 was considerably different than the measured value; this was expected, since the parameters used by the model different than those in the actual circuit. The overall quality of the distortion sounded better than that achieved using the Ebers-Moll model, which exhibited unpleasant clicks. On the other hand, the clipping from the PSPICE model was unexpectedly and undesirably symmetric.



*Figure 4.7 – Processed guitar output, PSPICE modeling  
(overton\_william\_e\_200605\_mast\_fig47\_pspice-modeling.wav, 1920K)*

The PSPICE model was altered to include the transistor parameters measured in lab to try to correct this problem. After updating the model, the circuit did in fact bias at -0.04 according to PSPICE, as expected from our lab experiments. However, PSPICE would no longer perform a reasonable DC or transient analysis with these parameter values. It simply pegged the output at the collector of the second transistor to the negative supply voltage. We currently do not have explanation for this extremely odd behavior.

An algorithm was also written to process the guitar signal using the simplified Ebers-Moll equations. The output actually sounded better than the full Ebers-Moll model, although it also required a different manual bias; this model had a reasonable bias point at around -0.09 V.

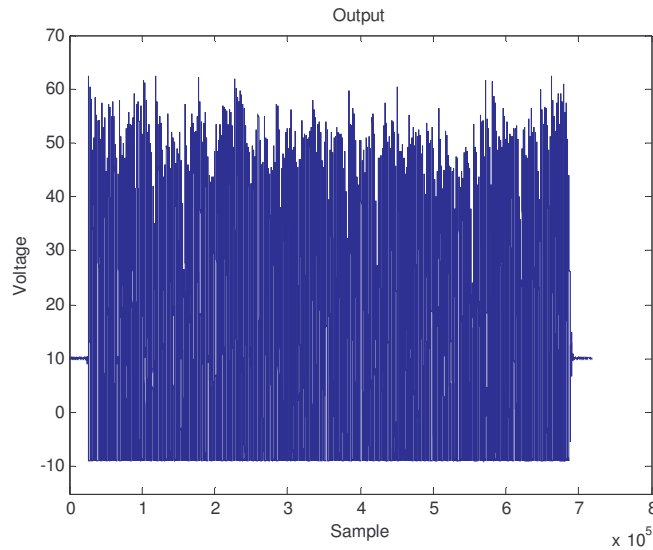


Figure 4.8 – Processed guitar output, Simplified Ebers-Moll model  
(*overtone\_william\_e\_200605\_mast\_fig47\_simplified-modeling.wav, 1920K*)

The clipping appears mostly asymmetrical for each model, although as noted from the transfer function, the PSPICE model seems to exhibit symmetrical clipping. Graphs of the input signal and the outputs for each distortion method are shown below. The original Ebers-Moll model inverted the voltage signal while the other two models did not<sup>3</sup>. Therefore, that plot was inverted to allow for a fair comparison.

---

<sup>3</sup> This suggests that one of the presented calculations contains an error, either in the original analytic work or in the MATLAB coding. We have presented these results as-is in the hopes of inspiring work by other researchers.

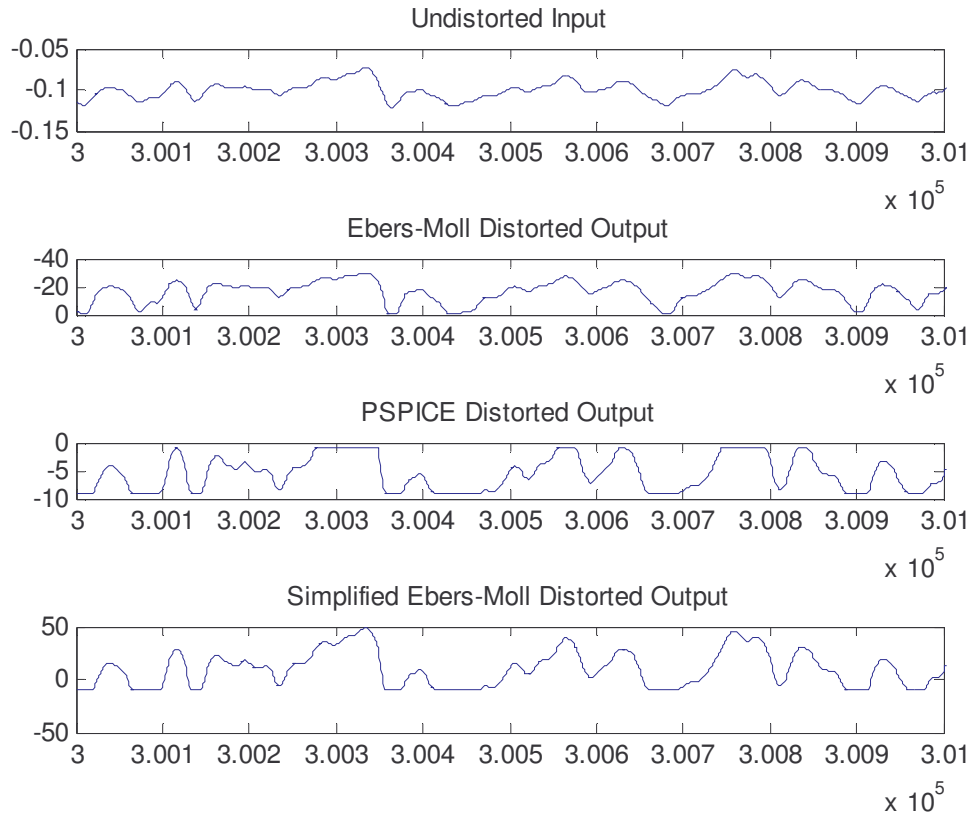


Figure 4.9 – Clean and distorted guitar signals.  
*x-axis – samples. y-axis – voltage*

The source code for the m-files, along with definitions of the key functions used, can be found in Appendix C.

## CHAPTER 5: CONCLUSIONS

The Ebers-Moll-based models derived for the Fuzz Face circuit were not perfect, but they were a step in the right direction. Even at the PSPICE stage, it was apparent that small changes in transistor parameters had profound effects on the operation of the circuit.

Thus, musicians' obsessions with finding perfectly matched transistor pairs appear to be valid. MATLAB also exhibited sensitivity to these parameters, with 'fzero' sometimes being unable to find solutions to the nonlinear equation depending on what parameters were entered. The bias point also had to be selected manually, because those selected by PSPICE did not match the values obtained in the lab or give reasonable operating points.

Also, the bias values arising from PSPICE modeling were different than the bias values measured in lab. This bias is crucial because if it lands in the wrong place, the signal gets clipped immediately in one (or both) swing directions. Reasonable bias values were obtained by inspection.

PSPICE simply would not perform a reasonable DC or transient analysis when using non-ideal collector and emitter saturation leakage currents. It would simply peg the entire signal to the negative power supply. Furthermore, the acquired DC transfer function when using ideal saturation leakage currents appeared to be almost perfectly symmetric. However, the transient analysis showed the expected asymmetrical clipping.



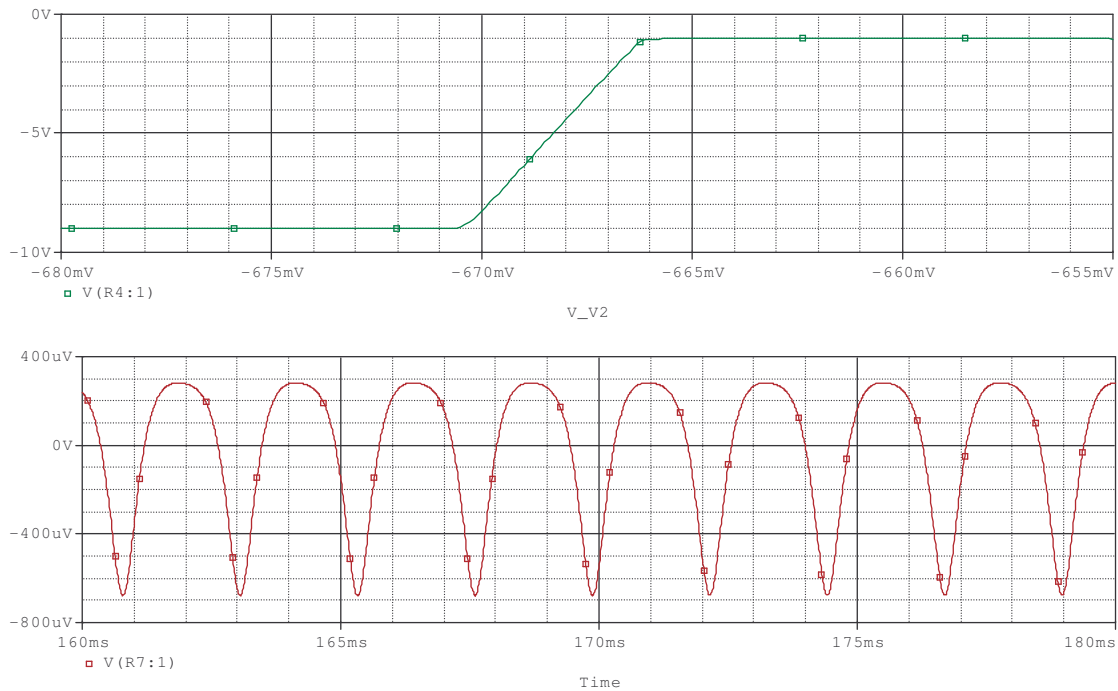


Figure 5.1 – PSPICE DC and transient analysis

This clearly does not seem to add up, and there is no apparent explanation for this discrepancy. The capacitors may have had a greater effect on the operation of the circuit than anticipated. Acquiring a PSPICE transfer function for a transistor with the desired non-ideal parameters and matching the DC and transient analyses are outstanding issues.

By using the Ebers-Moll modeling method, it is easy to modify circuit parameters. The transistor parameters and circuit resistances can easily be set prior to running the algorithm. This allows any transistor to be modeled based solely on the four Ebers-Moll parameters: forward-active current gain, inverse-active current gain, and the collector and emitter saturation leakage currents. It should work with PNP or NPN transistors as long as the parameters are obtained with the correct current reference. In addition, circuit modifications based on changing resistance values, such as the Roger Mayer Mod, are

easy to implement. However, new components cannot be added without reworking the model equations. Therefore, mods such as the Fuller Mod or Vox Tone Bender cannot be implemented under the current set of equations. These components would need to be figured into the model, and then could be set to zero for standard operation.

The Fuzz Face appears to be a simple circuit, but performing circuit-level emulation proved to be extraordinarily complex. Although the sound of the Fuzz Face was not perfectly modeled, equations that provided a reasonable approximation of the operation of the analog circuit were obtained.

## CHAPTER 6: DIRECTION FOR FUTURE STUDY

There are several areas where further study and additional techniques could increase the accuracy of the model and the effectiveness of the algorithm. The circuit used as the basis for the Ebers-Moll modeling was simplified. Simplifications included lumping some resistances, leaving out the potentiometers, and treating the capacitors as short circuits. The lumping of resistances caused the output to be recorded at a different point in the model than in the analog circuit. Although we suspect this results in a simple shifting and scaling relative to the actual output, the lack of loading by the volume control introduced error into the model. Furthermore, potentiometers were treated as simple resistors with values representing how much of the pot was “in use.” These resistances also influence the transfer function of the system. Though the capacitors were shown to have minimal effect on guitar input signals, the second-order effects of the capacitances could have significant effects on the transfer function of the circuit. Incorporating all these elements would allow the circuit to be more completely and accurately modeled. The second-order effects of the capacitors, and even parasitic capacitance in the transistors, could possibly have a significant effect on the transfer function of the circuit, and thus should be studied further.

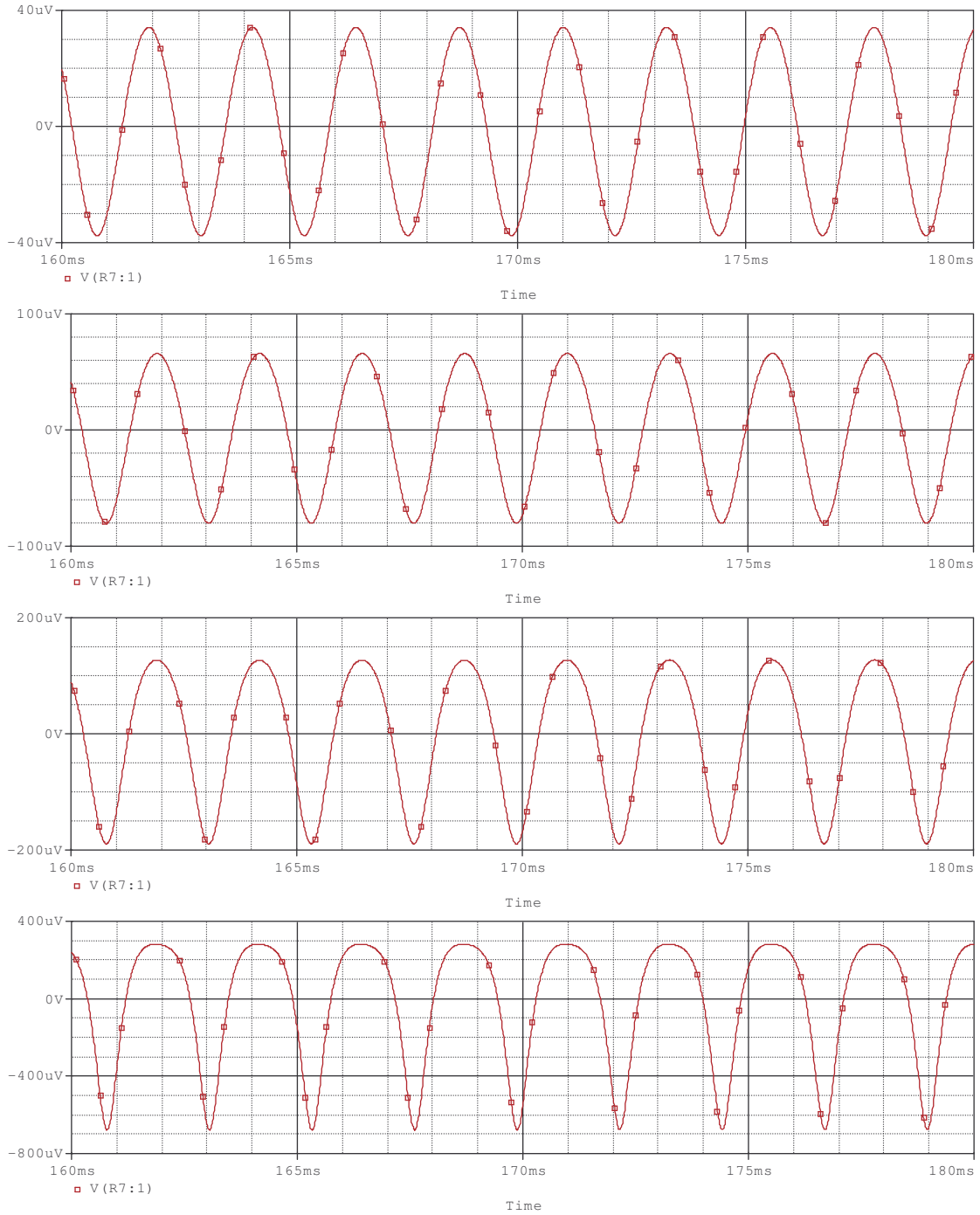
There were also disagreements between the results obtained through SPICE, in the lab, and using MATLAB. It would be instructive to determine where the discrepancies came from, and how to account for them in both the simulations and algorithms. It is difficult to assess the effectiveness of the algorithm without a clear basis for comparison. It should

be determined how to get PSPICE to analyze the circuit when non-ideal transistor parameters are used.

Additionally, more study is needed on how the different constants, including resistances and transistor parameters, affect the transfer function of the model. The best way to explore this would be to have sliders that control the various parameters, and run the transfer function generating algorithm continuously to see the change in the transfer function caused by modifying the parameters. However any discrepancies in the transfer function calculation should be resolved before exploring these aspects of the model.

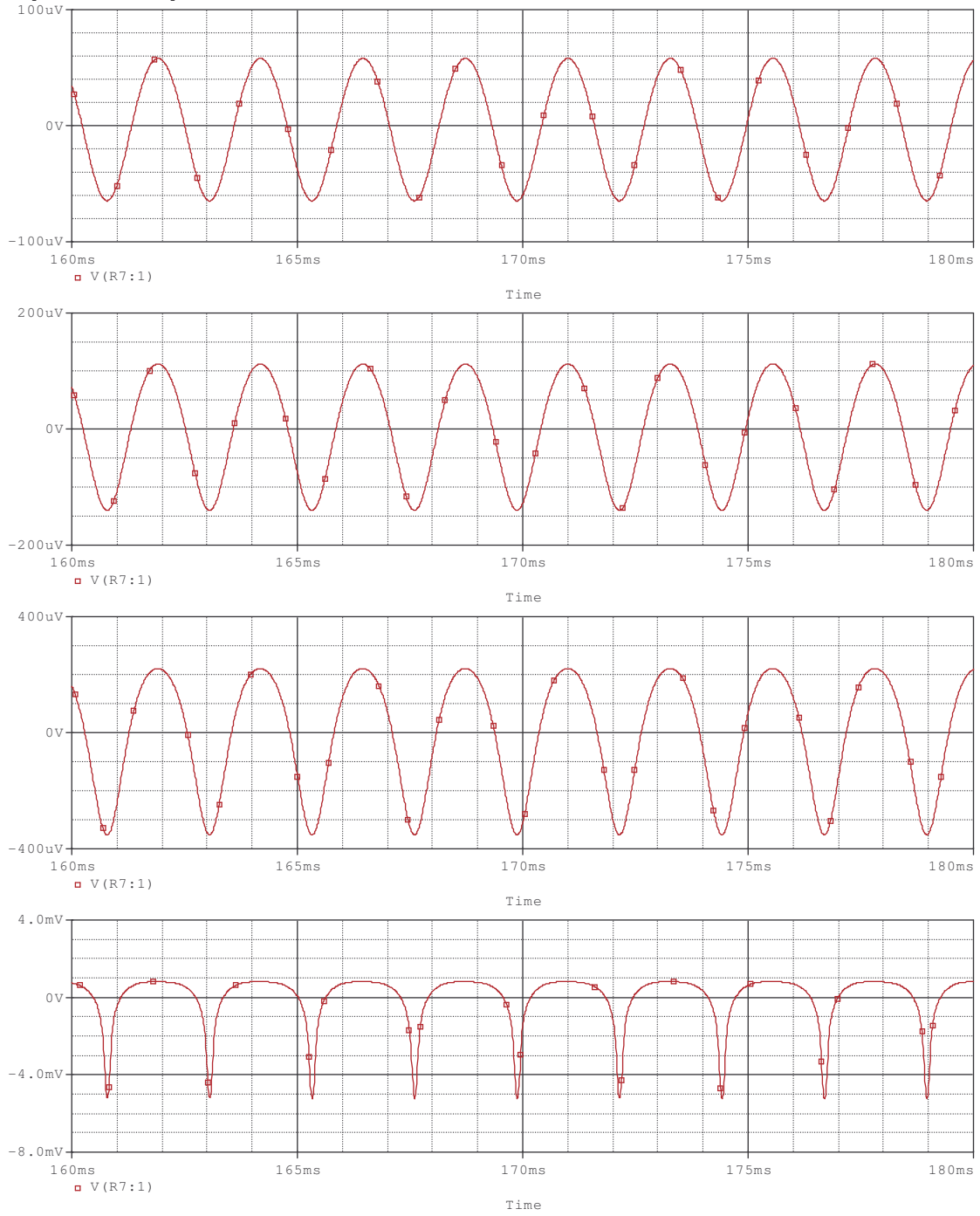
**APPENDIX A**  
**SIMULATION DATA**

**Q1:  $\beta = 70$ , Q2:  $\beta = 120$**



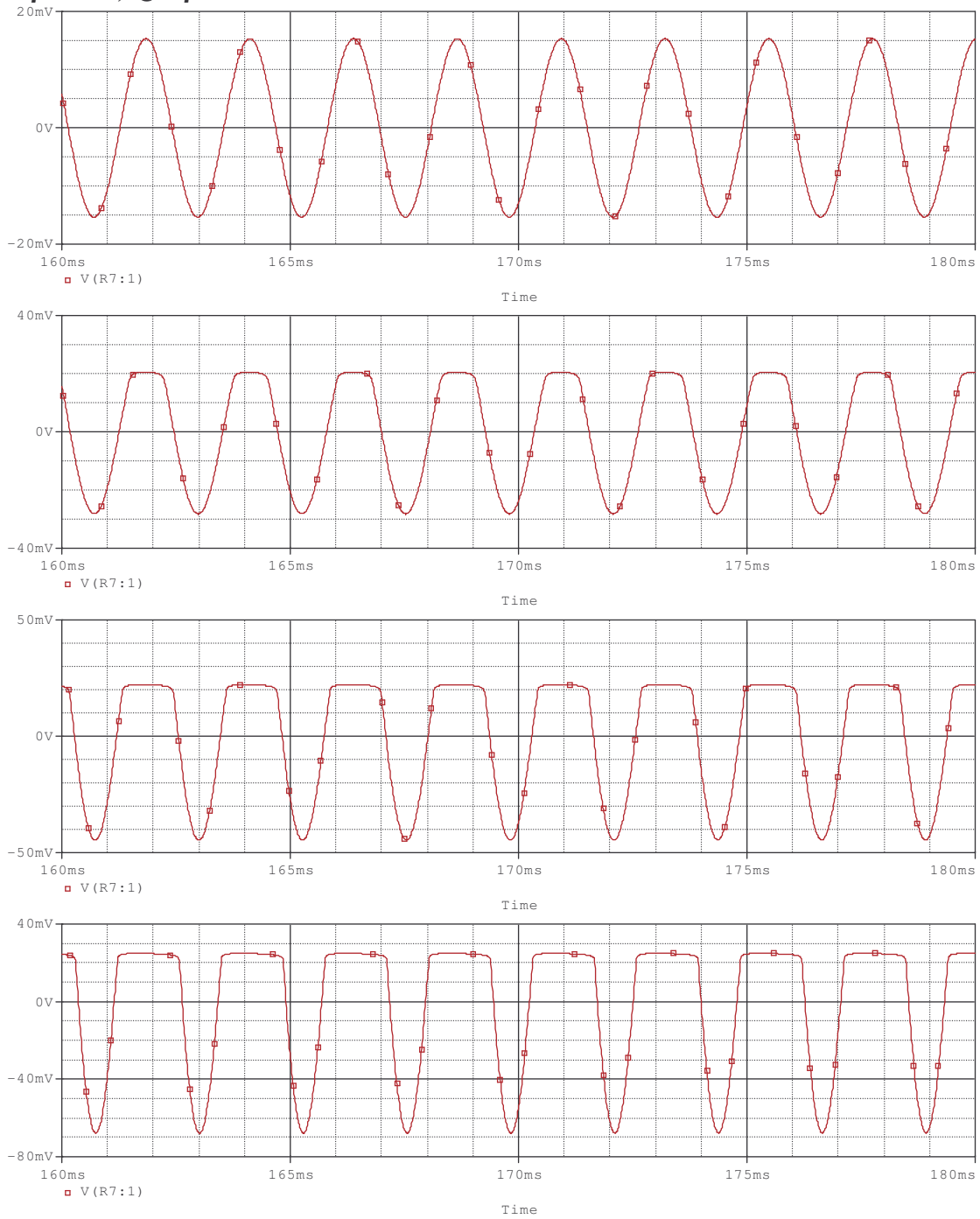
*Figure A.1 – Simulated output for  $\beta_1=70$ ,  $\beta_2=120$ ; Inputs of 0.5, 1, 1.5, and 2 mV*

**Q1:  $\beta = 70$ , Q2:  $\beta = 70$**



*Figure A.2 – Simulated output for  $\beta_1=70$ ,  $\beta_2=70$ ; Inputs of 0.5, 1, 1.5, and 2 mV*

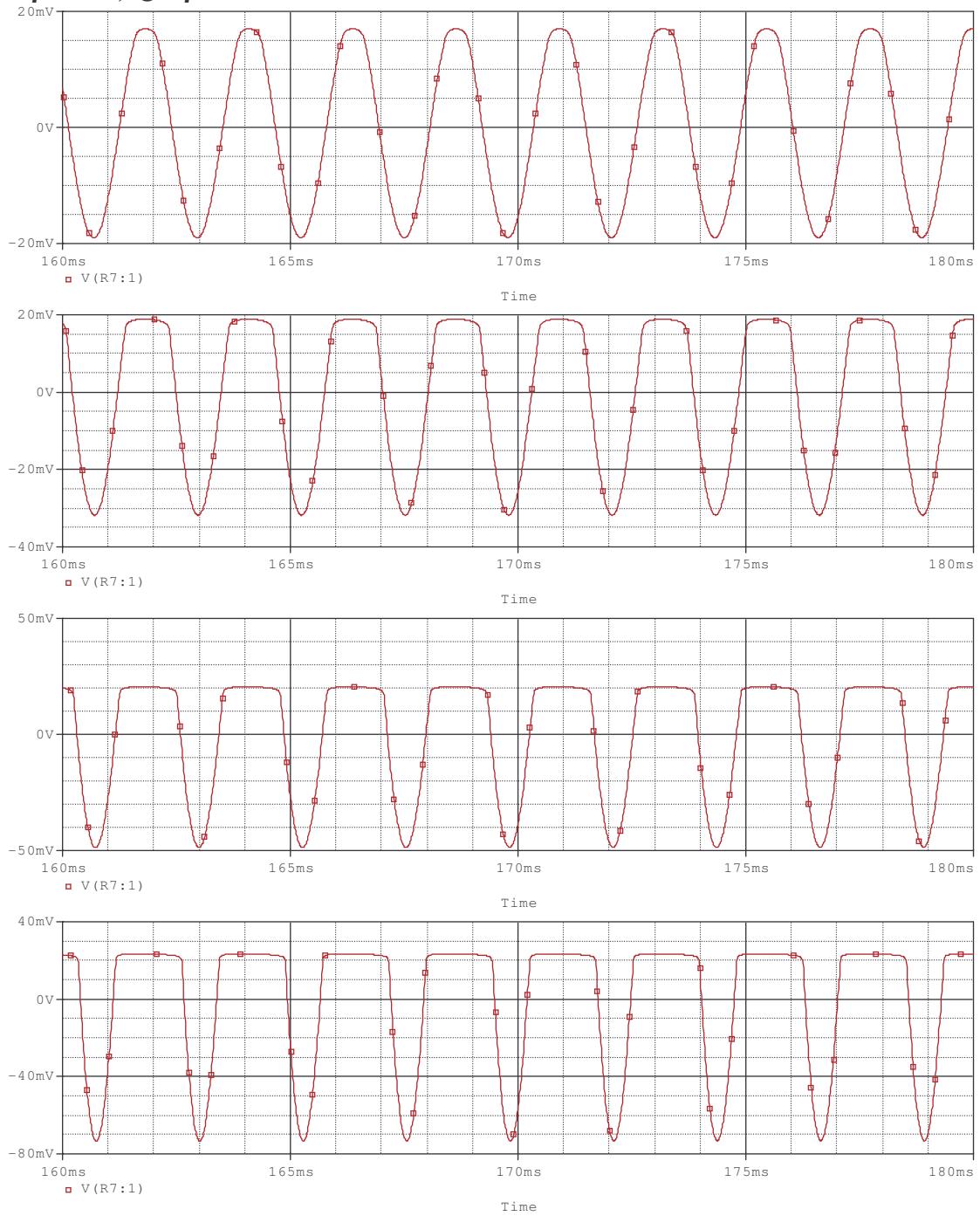
**Q1:  $\beta = 120$ , Q2:  $\beta = 70$**



*Figure A.3 – Simulated output for  $\beta_1=120$ ,  $\beta_2=70$ ; Inputs of 0.5, 1, 1.5, and 2 mV*

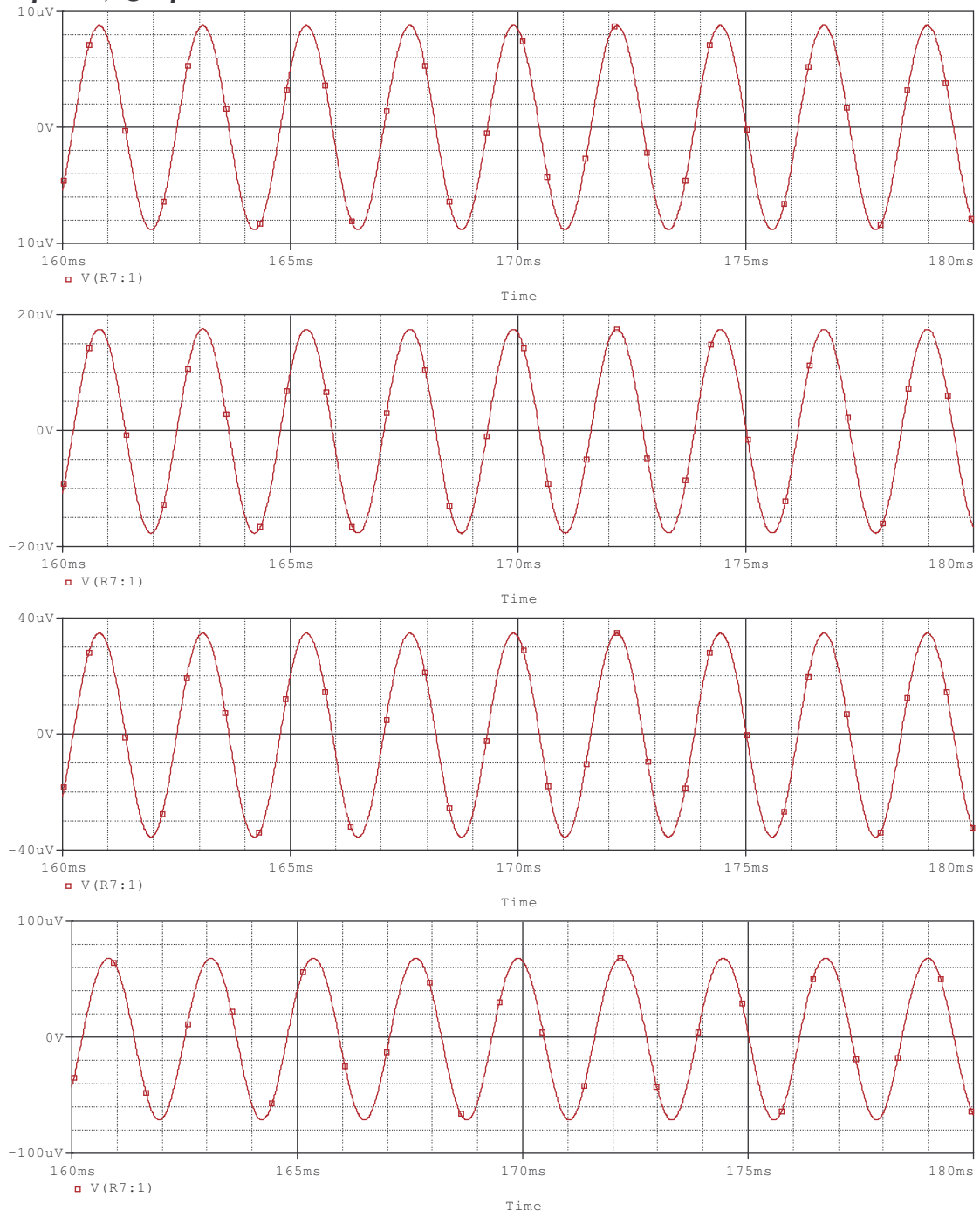


**Q1:  $\beta = 120$ , Q2:  $\beta = 120$**



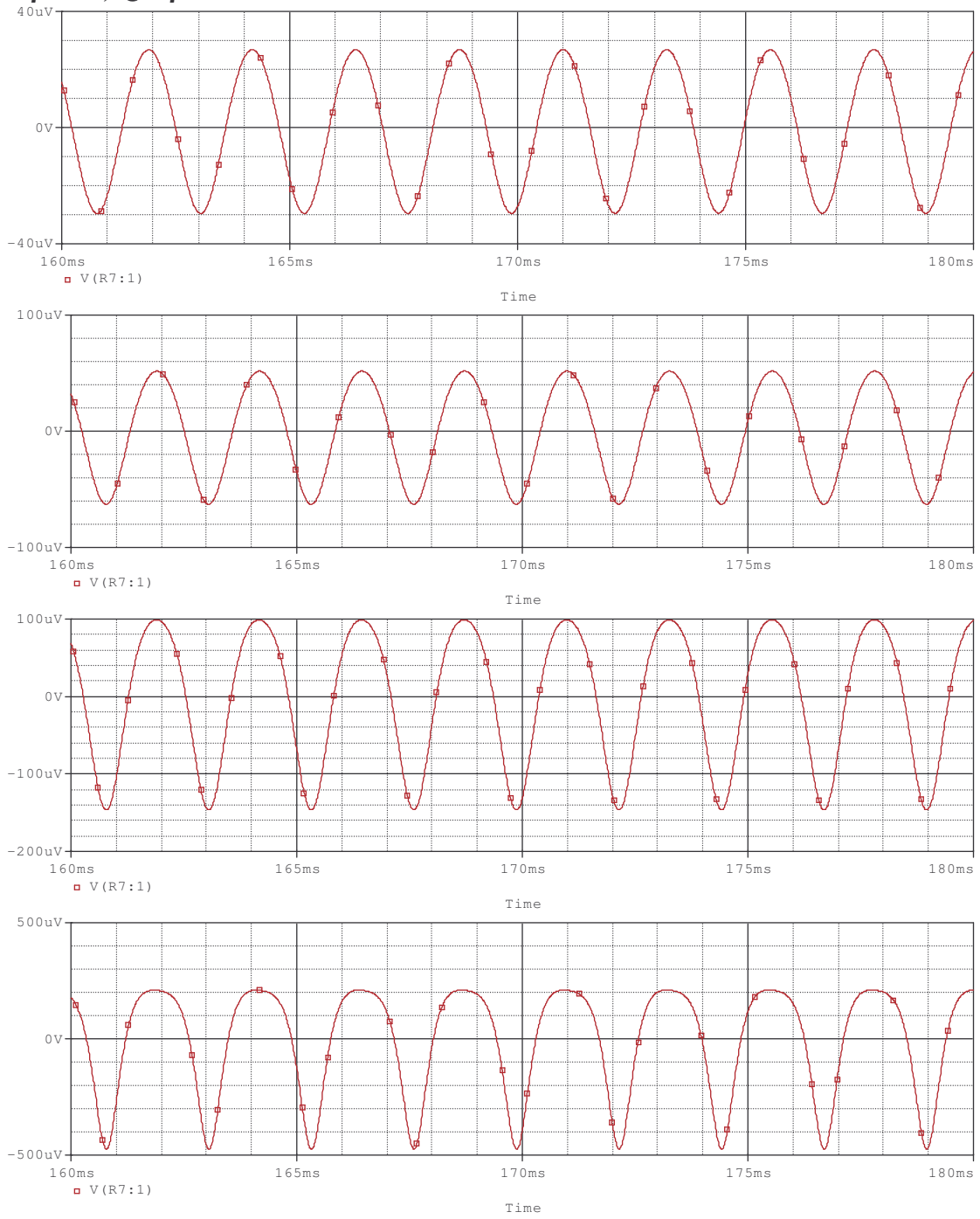
*Figure A.4 – Simulated output for  $\beta_1=120$ ,  $\beta_2=120$ ; Inputs of 0.5, 1, 1.5, and 2 mV*

**Q1:  $\beta = 40$ , Q2:  $\beta = 120$**



*Figure A.5 – Simulated output for  $\beta_1=40$ ,  $\beta_2=120$ ; Inputs of 0.5, 1, 1.5, and 2 mV*

**Q1:  $\beta = 70$ , Q2:  $\beta = 180$**



*Figure A.6 – Simulated output for  $\beta_1=70$ ,  $\beta_2=180$ ; Inputs of 0.5, 1, 1.5, and 2 mV*

**APPENDIX B**  
**PSPICE TRANSISTOR MODEL PARAMETERS**

Table B.1 – Bipolar transistor parameters [15]

Name	Description	Default
<b>DC Large Signal Forward Bias</b>		
BF	Ideal Maximum Forward Beta. Basic parameter for Ebers-Moll and Gummel-Poon models.	100
IKF	Knee Current for Forward Beta High Current Roll-off. Models variation in forward Beta at high collector currents. Use if device is to be used with high collector currents.	∞ Amp
IS	Transport Saturation Current. Basic parameter for Ebers-Moll and Gummel-Poon models.	$1 \times 10^{-16}$ Amp
ISE	Base Emitter Leakage Saturation Current. Models variation in forward Beta at low base currents. Use if device is to be used with low base emitter voltage.	0 Amp
NE	Base Emitter Leakage Emission Coefficient. Models variation in forward Beta at low base currents. Use if device is to be used with low base emitter voltage.	1.5
NF	Forward Current Emission Coefficient. Used to model deviation of emitter base diode from ideal (usually approximately 1).	1.0
VAF	Forward Early Voltage. Models base collector bias effects. Used to model base collector bias on forward Beta and IS.	∞ volt
<b>DC Large Signal Reverse Bias</b>		
BR	Ideal Maximum Reverse Beta. The basic parameter for both Ebers-Moll and Gummel-Poon models. Use when the transistor is saturated or operating in reverse mode.	1.0
IKR	Knee Current for Reverse Beta High Current Roll-off. Specifies variation in reverse Beta at high emitter currents. Needed only if transistor is operated in reverse mode.	∞ Amp
ISC	Base Collector Leakage Saturation Current. Specifies variation in reverse Beta at low base currents. Models base current at low base collector voltage. Use only if transistor is operated in reverse mode.	0 Amp
NC	Base Collector Leakage Emission Coefficient. Specifies variation in reverse Beta at low currents. Models base current at low base collector voltage. Use only if transistor is operated in reverse mode.	2.0
NR	Reverse Current Emission Coefficient. Used to model deviation of base collector diode from the ideal (usually about 1).	1.0
VAR	Reverse Early Voltage. Models emitter base bias effects. Use to model emitter base bias on reverse Beta and IS.	∞ Volt

Table B.1 – Continued [15]

<b>Series Resistance</b>		
IRB	Base Resistance Roll-off Current. Models the base current at which the base resistance is halfway between minimum and maximum.	∞ Amp
RB	Zero Bias Base Resistance. Maximum value of parasitic resistance in base.	0 Ohm
RBM	Minimum Base Resistance. The minimum value of base resistance at high current levels. Models the way base resistance varies as base current varies.	RB Ohm
RC	Collector Resistance. Parasitic resistance in the collector. Important in high current and high frequency applications.	0 Ohm
RE	Emitter Resistance. Parasitic resistance in the emitter. Important in small signal applications.	0 Ohm
<b>Capacitance</b>		
CJC	Base Collector Zero Bias Capacitance. Helps model switching time and high frequency effects.	0 Farad
CJE	Base Emitter Zero Bias Capacitance. Helps model switching time and high frequency effects.	0 Farad
CJS	Zero Bias Substrate Capacitance. Helps model switching time and high frequency effects.	0 Farad
MJC	Base Collector Junction Grading Coefficient. Models the way junction capacitance varies with bias.	0.33
MJE	Base Emitter Junction Grading Coefficient. Models the way junction capacitance varies with bias.	0.33
MJS	Substrate Junction Grading Coefficient. Models the way junction capacitance varies with bias.	0.33
VJC	Base Collector Built-in Potential. Models the way junction capacitance varies with bias.	0.75 Volt
VJE	Base Emitter Built-in Potential. Models the way junction capacitance varies with bias.	0.75 Volt
VJS	Substrate Junction Built-in Potential. Models the way junction capacitance varies with bias.	0.75 Volt
XCJC	Fraction of Base Collector. Capacitance that connects to the internal base node. Important in high frequency applications.	1.0
FC	Coefficient for Forward Bias Capacitance Formula. Provides continuity between capacitance equations for forward and reverse bias.	0.5
<b>AC Small-Signal</b>		
ITF	High Current Parameter for Effect on TF. Models decline of TF with high collector current.	∞ Amp

Table B.1 – Continued [15]

PTF	Excess Phase at FT. Models excess phase at FT.	0 Degree
TF	Ideal Forward Transit Time. Models finite bandwidth of device in forward mode.	0 Sec
TR	Ideal Reverse Transit Time. Models finite bandwidth of device in reverse mode.	0 Sec
VTF	Voltage Describing TF Dependence on Base-Collector Voltage. Models base-collector voltage bias effects on TF.	0 Volt
XTF	Coefficient for Bias Dependence on TF. Models minimum value of TF at low collector-emitter voltage and high collector current.	0
<b>Temperature Effects</b>		
EG	Energy Gap for Modeling Temperature. Effect on IS, ISE, and ISC. Used to calculate the temperature variation of saturation currents in the collector, and base-emitter and collector base diodes.	1.11EV
XTB	Forward and Reverse Beta Temperature Exponent. Models the way Beta varies with temperature.	0
XTI	Temperature Exponent for Modeling. Temperature Variation of IS. Models the way saturation current varies with temperature.	3.0
TNOM	This global variable can be assigned temperature values in degrees C, for use by extractions and simulations.	27°C

**APPENDIX C**  
**MATLAB CODE AND DATA FILES**



```

% Bill Overton - funct_VB2
% Transistor Fuzz Modeling
%
% This function is a nonlinear expression relating the Fuzz Face
% voltages V_B1 (given) and V_B2

function f = funct_VB2(V_B2,V_B1,R_C1,R_C2,R_fb,R_fuzz,alpha_N1,...
    alpha_I1,I_CO1,I_EO1,alpha_N2,alpha_I2,I_CO2,V_T,VDC)

% Function in terms of voltages V_B1 and V_B2 only
f = (((((I_CO1*(exp((V_B2-V_B1)/V_T)-1)+V_B1/R_fb)/(1-alpha_N1)-(V_B2-
VDC)/R_C1+VDC/R_C2)-((I_EO1*(exp((-V_B1/V_T))-1)-alpha_I1*V_B1/R_fb-(1-
alpha_I1))*((V_B2-VDC)/R_C1-VDC/R_C2)))/((1-alpha_I1)/R_fuzz-
alpha_I1/R_fb))* (1/R_fuzz+1/(R_fb*(1-alpha_N1)))))/(1/R_C2-(1-
alpha_I1)/(R_C2*((1-alpha_I1)/R_fuzz-
alpha_I1/R_fb))* (1/R_fuzz+1/(R_fb*(1-alpha_N1)))))-VDC)/R_C2-
alpha_N2*(V_B1/R_fb-((I_EO1*(exp((-V_B1/V_T))-1)-alpha_I1*V_B1/R_fb-
(1-alpha_I1))*((V_B2-VDC)/R_C1-VDC/R_C2)))/((1-alpha_I1)/R_fuzz-
alpha_I1/R_fb))-((1-alpha_I1)*(((I_CO1*(exp((V_B2-V_B1)/V_T)-
1)+V_B1/R_fb)/(1-alpha_N1)-(V_B2-VDC)/R_C1+VDC/R_C2)-((I_EO1*(exp((-
V_B1/V_T))-1)-alpha_I1*V_B1/R_fb-(1-alpha_I1))*((V_B2-VDC)/R_C1-
VDC/R_C2)))/((1-alpha_I1)/R_fuzz-alpha_I1/R_fb))* (1/R_fuzz+1/(R_fb*(1-
alpha_N1)))))/(1/R_C2-(1-alpha_I1)/(R_C2*((1-alpha_I1)/R_fuzz-
alpha_I1/R_fb))* (1/R_fuzz+1/(R_fb*(1-alpha_N1)))))/R_C2)/((1-
alpha_I1)/R_fuzz-alpha_I1/R_fb))* (1/R_fb+1/R_fuzz))-
I_CO2*(exp((((I_CO1*(exp((V_B2-V_B1)/V_T)-1)+V_B1/R_fb)/(1-alpha_N1)-
(V_B2-VDC)/R_C1+VDC/R_C2)-((I_EO1*(exp((-V_B1/V_T))-1)-
alpha_I1*V_B1/R_fb-(1-alpha_I1))*((V_B2-VDC)/R_C1-VDC/R_C2)))/((1-
alpha_I1)/R_fuzz-alpha_I1/R_fb))* (1/R_fuzz+1/(R_fb*(1-
alpha_N1)))))/(1/R_C2-(1-alpha_I1)/(R_C2*((1-alpha_I1)/R_fuzz-
alpha_I1/R_fb))* (1/R_fuzz+1/(R_fb*(1-alpha_N1)))))-V_B2)/V_T)-1);

```

```

% Bill Overton – Fuzz6
% Transistor Fuzz Modeling
% Ebers-Moll Method
%
% This mfile models the distortion of the Fuzz Face using the derived
% Ebers-Moll equations of the circuit.

% Initialize Program
clear;

% Circuit Resistances and Power Supply
R_C1 = 33e3;
R_C2 = 8.67e3;
R_fb = 100e3;
R_fuzz = 1000;
VDC = -9;

% Thermal Voltage
V_T = 0.025695;

% Q1 Transistor Parameters
alpha_N1 = 0.99;
alpha_I1 = 0.914;
I_CO1 = 4.2e-6;
I_EO1 = 3.4e-6;

% Q2 Transistor Parameters
alpha_N2 = 0.992;
alpha_I2 = 0.918;
I_CO2 = 3e-6;
I_EO2 = 3.9e-6;

% FIND THE TRANSFER FUNCTION
%
% Calculate V_C2 for a range of inputs by running fzero iteratively for
% different values of V_B1.

% Initialize loop
V_B1 = 0.17;
index=[];
transfer=[];
points = 200;
inc = abs(V_B1)*2/points;

for n=1:points

```

```

% Solve using fzero
V_B2=fzero(@(V_B2) funct_VB2(V_B2,V_B1,R_C1,R_C2,R_fb,R_fuzz,alpha_N1,
    alpha_I1,I_CO1,I_EO1,alpha_N2,alpha_I2,I_CO2,V_T,VDC),1e-3);

% Solve for V_C2
K1=((I_EO1*(exp((-V_B1/V_T))-1)-alpha_I1*V_B1/R_fb-(1-alpha_I1)...
    *((V_B2-VDC)/R_C1-VDC/R_C2))/((1-alpha_I1)/R_fuzz-alpha_I1/R_fb));
K2=((I_CO1*(exp((V_B2-V_B1)/V_T)-1)+V_B1/R_fb)/(1-alpha_N1)-(V_B2-VDC)...
    /R_C1+VDC/R_C2);
V_C2=(K2-K1*(1/R_fuzz+1/(R_fb*(1-alpha_N1))))/(1/R_C2-(1-alpha_I1)...
    /(R_C2*((1-alpha_I1)/R_fuzz-alpha_I1/R_fb))*(1/R_fuzz+1/(R_fb*(1-...
    alpha_N1))));

% Set new values
transfer=[transfer V_C2];
index=[index V_B1];
V_B1=V_B1-inc;
end;

% Plot the transfer function
%hold on;
plot(index,transfer,'b');
title('System Transfer Function');
xlabel('Input Voltage');
ylabel('Output Voltage');

% Now read in input
[V_B1in, fs] = wavread('riff3.wav');

% Account for DC bias and scale
scale = 0.1;
input_bias = -0.04;
V_B1=(V_B1in(:,1)*scale+input_bias);

% Apply the distortion
% Calculate V_C2 using interp1
V_C2 = interp1(index, transfer, V_B1);

%plot(V_C2);

% Normalize to [-1 1]
x = V_C2;
xmin = min(x);
xmax = max(x);
slim = [xmin xmax];
dx=diff(slim);

```

```
if dx==0,  
    % Protect against divide by zero  
    V_0 = zeros(size(V_C2));  
else  
    V_O = (x-slim(1))/dx*2-1;  
end  
  
%plot(V_O);  
%sound(V_O,fs);
```

```

% Bill Overton - Fuzz7
% Transistor Fuzz Modeling
% PSPICE Method
%
% This mfile models the distortion of the Fuzz Face using a transfer
% function imported from PSPICE

% Initialize program
clear

% Read in transfer function and input
data = xlsread('pspicedata');
index = data(:,1);
transfer = data(:,2);
plot(index, transfer);
[V_Blin, fs] = wavread('riff3.wav');

% Account for input bias and scale
scale = 0.02;
input_bias = -0.6685;
V_B1=V_Blin(:,1)*scale+input_bias;

% Apply the distortion
% Calculate V_C2 using interp1
V_C2 = interp1(index, transfer, V_B1);

%plot(V_C2)

% Normalize to [-1 1]
x = V_C2;
xmin = min(x);
xmax = max(x);
slim = [xmin xmax];
dx=diff(slim);
if dx==0,
    % Protect against divide by zero
    V_0 = zeros(size(V_C2));
else
    V_0 = (x-slim(1))/dx*2-1;
end

%plot(V_O);
%sound(V_O,fs);

```

```

function f = funct_VE2(V_E2,V_B1,R_C1,R_fb,R_fuzz,I_S1,I_S2,V_T,VDC)

% Bill Overton
% Transistor Fuzz Modeling
%
% This function is a nonlinear expression relating the Fuzz Face voltages
% V_B1 (given) and V_E2

% Function in terms of voltages V_B1 and V_B2 only
f = V_E2-(1/(1/R_fb+1/R_fuzz))*(V_B1/R_fb-I_S2*(exp(-(R_C1*I_S1*(exp(-V_B1...
/V_T)-1)+VDC)/V_T)*exp(V_E2/V_T)-1));

```

```

% Bill Overton - Fuzz8
% Transistor Fuzz Modeling
% Simplified Ebers-Moll Method
%
% This mfile models the distortion of the Fuzz Face using the simplified
% Ebers-Moll equations of the circuit.

% Initialize Program
clear;

% Circuit Resistances and Power Supply
R_C1 = 33e3;
R_C2 = 8.67e3;
R_fb = 100e3;
R_fuzz = 1000;
VDC = -9;

% Thermal Voltage
V_T = 0.025695;

% Q1 Transistor Parameters
I_S1 = 4.2e-6;

% Q2 Transistor Parameters
I_S2 = 3.9e-6;

% FIND THE TRANSFER FUNCTION
%
% Calculate V_C2 for a range of inputs by running fzero iteratively for
% different values of V_B1.

% Initialize loop
V_B1 = 0.2;
index=[];
transfer=[];
points = 200;
inc = abs(V_B1)*2/points;

for n=1:points

    % Initial Guess
    lambda1 = 1/(1/R_fb+1/R_fuzz);
    lambda2 = exp(-(R_C1*I_S1*(exp(-V_B1/V_T)-1)+VDC)/V_T);
    guess = lambda1*(V_B1/R_fb-I_S2*(lambda2-1))/(1+I_S2*lambda1*lambda2/V_T);

    % Solve using fzero

```

```

V_E2=fzero(@(V_E2)
funct_VE2(V_E2,V_B1,R_C1,R_fb,R_fuzz,I_S1,I_S2,V_T,VDC),guess);

% Solve for V_C2
V_B2 = R_C1*I_S1*exp(-V_B1/V_T-1)+VDC;
V_C2 = R_C2*(V_B1/R_fb-V_E2*(1/R_fb+1/R_fuzz))+VDC;

% Set new values
transfer=[transfer V_C2];
index=[index V_B1];
V_B1=V_B1-inc;
end;

% Plot the transfer function
%hold on;
plot(index,transfer,'b');
title('System Transfer Function');
xlabel('Input Voltage');
ylabel('Output Voltage');

% Now read in input
[V_B1in, fs] = wavread('riff3.wav');

% Account for DC bias and scale
scale = 0.08;
input_bias = -0.1;
V_B1=(V_B1in(:,1)*scale+input_bias);

% Apply the distortion
% Calculate V_C2 using interp1
V_C2 = interp1(index, transfer, V_B1);

%plot(V_C2);

% Normalize to [-1 1]
x = V_C2;
xmin = min(x);
xmax = max(x);
slim = [xmin xmax];
dx=diff(slim);
if dx==0,
    % Protect against divide by zero
    V_0 = zeros(size(V_C2));
else
    V_O = (x-slim(1))/dx*2-1;
end

```



## REFERENCES

- [1] Wikipedia, "Audio Distortion" online article available at [http://en.wikipedia.org/wiki/Distortion#Audio\\_distortion](http://en.wikipedia.org/wiki/Distortion#Audio_distortion).  
Date accessed: 1 April 2006
- [2] Based off a drawing by R. G. Keen. "Basic Fuzz Face." online image available at [http://www.geofex.com/Article\\_Folders/fuzzface/fftech1.gif](http://www.geofex.com/Article_Folders/fuzzface/fftech1.gif).  
Date accessed: 21 Feb 2005
- [3] Keen, R. G. "The Technology of the Fuzz Face." online article available at [http://www.geofex.com/Artical\\_Folders/fuzzface/fftech.htm](http://www.geofex.com/Artical_Folders/fuzzface/fftech.htm).  
Date accessed: 21 Feb 2005
- [4] Castledine, S. "The Fuzz Face." online article available at <http://homepage.ntlworld.com/s.castledine/greenfuzz/fface.html>.  
Date accessed: 25 Jan 2006
- [5] "Analog Man Sun Face (Internal)." online image available at <http://www.analogman.com/graphics/sunfsmli.jpg>. Date accessed: 21 Feb 2005
- [6] "Analog Man Sun Face (External)." online image available at <http://www.analogman.com/graphics/sunfacenkt.jpg>. Date accessed: 21 Feb 2005
- [7] JD-F2 Fuzz Face. online audio clip available at [http://www.jimdunlop.com/products/electronics/fuzzface/audio/fuzz\\_face.wav](http://www.jimdunlop.com/products/electronics/fuzzface/audio/fuzz_face.wav).  
Date accessed: 4 April 2006
- [8] Based off a drawing by R. G. Keen. "Fuzz Face With "Roger Mayer" or "Jimi Hendrix" mods. online image available at [http://www.geofex.com/Article\\_Folders/fuzzface/fftech2.gif](http://www.geofex.com/Article_Folders/fuzzface/fftech2.gif).  
Date accessed: 21 Feb 2005
- [9] Based off a drawing by R. G. Keen. "Fuzz Face With Fuller Mods." online image available at [http://www.geofex.com/Article\\_Folders/fuzzface/fftech3.gif](http://www.geofex.com/Article_Folders/fuzzface/fftech3.gif).  
Date accessed: 21 Feb 2005
- [10] Based off a drawing by R. G. Keen. "Vox Tone Bender 5/67." online image available at [http://www.geofex.com/Article\\_Folders/fuzzface/fftech4.gif](http://www.geofex.com/Article_Folders/fuzzface/fftech4.gif).  
Date accessed: 21 Feb 2005
- [11] "SPICE History." online article available at <http://www.ecircuitcenter.com/SpiceTopics/History.htm>.  
Date accessed: 12 Oct 2005

- [12] Perlman, David. "The Ebers-Moll Transistor Model." online article available at <http://www.rit.edu/~dnpeee/EE481/Suppl11-Ebers-Moll.pdf>.  
Date accessed: 22 Nov 2005
  
- [13] Keen, R. G. "A Musical Distortion Primer." online article available at <http://www.geofex.com/effxfaq/distn101.htm>. Date accessed: 21 Feb 2005
  
- [14] Keen, R. G. "Asymmetrical Hard Clipping." online image available at [http://www.geofex.com/effxfaq/d101\\_10.gif](http://www.geofex.com/effxfaq/d101_10.gif). Date accessed: 21 Feb 2005
  
- [15] "Model Parameters" online article available at [http://eesof.tm.agilent.com/docs/iccap2002/ic\\_mdl/icim014.html](http://eesof.tm.agilent.com/docs/iccap2002/ic_mdl/icim014.html).  
Date accessed: 12 Oct 2005