

The TubeScreamer TS808: A Digital Model

JACEY SCHELL,*  AES Member
(jxs7684@miami.edu)

Department of Music Engineering Technology, Frost School of Music, University of Miami, Coral Gables, Florida, USA

This paper presents a digital model of the Ibanez Tube Screamer TS808 overdrive pedal based on a subsystem decomposition of its analog circuitry. The input buffer, clipping-stage filters, tone and volume network, and output buffer are represented using continuous-time transfer functions and converted to discrete-time filters using the bilinear transform. Nonlinear diode clipping is approximated using a drive-dependent soft-clipping function. The resulting MATLAB implementation is evaluated using frequency-response, impulse-response, and pole-zero analyses under multiple drive, tone, and volume configurations. The model reproduces the principal filtering and nonlinear processing behavior of the TS808 while providing a computationally simple structure suitable for real-time implementation.

Keywords: Tube Screamer TS808, Virtual Analog Modeling, Digital Signal Processing, Nonlinear Audio Processing, Guitar Effects, Bilinear Transform, Soft Clipping

1 INTRODUCTION

The TS808 was the first of the TubeScreamer guitar pedal series. Through the boosting of mid-frequencies and warm harmonics, the TS808 provides a rich overdrive while preserving the input signal's characteristics. This paper explores how the linear and nonlinear components of the TS808 can be digitally modeled in MATLAB.

2 CIRCUIT DECOMPOSITION: SIGNAL FLOW AND SUBSYSTEMS

To analyze the TS808 circuitry and signal flow, I will reference the schematic breakdown from ElectroSmash [1] as presented in Figure 1. There are six major subsystems in the schematic, but for the purpose of digital modeling I incorporated only the four that affect the digital signal: Input Buffer, Clipping Amp, Tone/Volume Stage, and Output Buffer.

2.1 Input Buffer

The first block that the signal encounters is the input buffer, which consists of a first-order RC high-pass filter

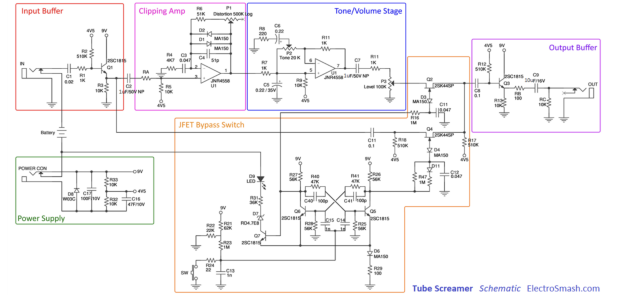


Fig. 1. TS808 circuit decomposed into labeled subsystems. Reproduced from ElectroSmash [1].

followed by a unity-gain emitter follower. This stage acts as a DC block, filtering any low humming that might be present in the signal and lowering the output impedance for the upcoming blocks in the circuit. The input buffer subsystem is shown in Figure 2.

Using an estimated input impedance, I determined the R_{Total} value to be 338 k Ω . This resulted in the high-pass filter holding a cutoff frequency of approximately 23.5 Hz. After filtering, the signal runs through a transistor with a gain scaling value of 0.993 – acting as a unity gain factor. The purpose of this buffer is to lower the output impedance while keeping the high input impedance, ultimately avoiding frequency response alterations when flowing from a high-impedance source to a low-impedance load. The transfer function from the input buffer stage, $H_1(s)$, is as follows:

* To whom correspondence should be addressed, email: jxs7684@miami.edu.

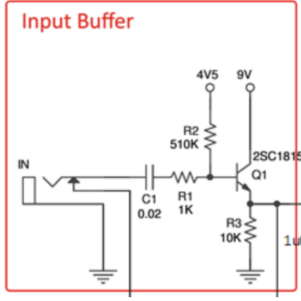


Fig. 2. Input Buffer subsection of the TS808 circuit.

$$H_1(s) = A_v \frac{RCs}{RCs + 1},$$

where $R = 338 \text{ k}\Omega$ is the resistance, $C = 20 \text{ nF}$ is the capacitance, and $A_v = 0.993$ is the voltage gain of the input buffer.

2.2 Clipping Amp

The second stage of the TS808 circuit is the clipping amp, as shown in Figure 3. The combination of diodes, capacitors, and resistors in the feedback loop of the op-amp produces a filtered overdrive that falls dependent on a potentiometer. The clipping amp block can be broken down further into four subsections: Input Biasing, High-Pass Filter, Clipping Section, and Low-Pass Filter.

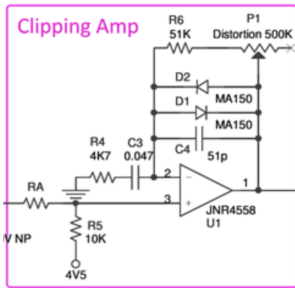


Fig. 3. Clipping amp subsection of the TS808 circuit.

2.2.1 Input Biasing and High-Pass Filter

As the circuit runs off of a 9V power supply, the signal must first flow through a biasing section in the clipping amp block resulting in a centered virtual ground around 4.5V rather than 0V. This is the case so that the positive half of the waveform can reach up to 9V and the negative half can reach down to 0V. The next substage is built off of R_4 and C_3 which together form a high-pass filter with a frequency cutoff of approximately 720 Hz. This filter prevents any muddiness that might come from distorting the bass frequencies of the signal. The corresponding transfer function, $H_2(s)$, can be expressed as:

$$H_2(s) = \frac{RCs}{RCs + 1},$$

where $R = 4.7 \text{ k}\Omega$ is the resistance and $C = 47 \text{ nF}$ is the capacitance of the high-pass stage.

2.2.2 Clipping Stage and Low-Pass Filter

After the high-pass filter, the signal reaches the nonlinear clipping stage which includes an inverting amplifier followed by parallel diodes with an estimated threshold of 0.5-0.7V. A potentiometer P_1 controls the amplification amount applied to the signal in series with R_6 , with higher values of P_1 equating to more amplification, and vice versa. When the signal from the amplifier exceeds the diode threshold, a soft clipping forms from increased feedback. The last portion of the clipping amp block occurs in the feedback loop with components R_6 and C_4 which create an RC low-pass filter with a cutoff frequency of approximately 60 kHz. Having such a high cutoff frequency due to the 51 pF capacitor maintains the quality of the frequency range while also improving stability in the feedback loop and ensuring control over harmonics. The transfer function describing this final low-pass stage, $H_3(s)$, is:

$$H_3(s) = \frac{1}{RCs + 1},$$

where $R = 51 \text{ k}\Omega$ is the resistance and $C = 51 \text{ pF}$ is the capacitance of the low-pass stage.

2.3 Tone and Volume

Following the clipping amp block is the tone and volume stage of the TS808. This is the section where low and high frequencies are attenuated or boosted depending on the value of a potentiometer – shaping the frequency content of the signal. The volume potentiometer, as shown in Figure 4, is also located within this stage, just before the signal reaches the final output buffer.

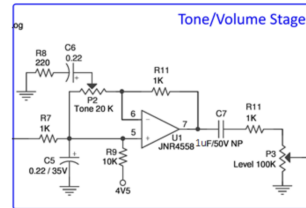


Fig. 4. Tone and volume subsection of the TS808 circuit.

Within the tone and volume network, there are three primary phases in the signal path. The first is a passive low-pass filter using components R_7 and C_5 . This filter, with a cutoff frequency of approximately 723 Hz, attenuates the harsh harmonics generated from the previous clipping amp subsystem. The second phase of the tone and volume block is the active mid-frequency filtering circuit which differs depending on the value of the associated potentiometer, P_2 . When P_2 holds a value of 0Ω , R_8 and C_6 hold a stronger

connection to the feedback network and act as a second low-pass filter due to increased high-frequency feedback. This results in an added pole to the transfer function meaning a second-order low-pass filter in relation to the low-pass filter from the first phase. Contrastingly, when $P_2 = 20 \text{ k}\Omega$, the high-frequency feedback is reduced which introduces a zero to the transfer function. This added zero manifests as a band-pass filter in the frequency response with mid-frequency boost. The last phase of the tone and volume network is the volume control, which simply acts as a gain factor after the signal has gone through phases one and two. The tone and volume block can be mathematically represented by this transfer function, $H_4(s)$:

$$H_4(s) = KV \frac{s + \omega_z}{(s + \omega_{p1})(s + \omega_{p2})},$$

where K is the stage gain, V is the normalized volume control, ω_z is the zero frequency, and ω_{p1} and ω_{p2} are the pole frequencies, defined as

$$K = \frac{R_{\text{load}} + R_f}{R_{\text{load}}},$$

$$V = \frac{\text{Volume}}{100},$$

$$\omega_{p1} = \frac{1}{C_z(R_z + R_f)},$$

$$\omega_{p2} = \frac{1}{C_z R_{\text{load}}},$$

$$\omega_z = \frac{1}{C_z(R_z + R_l)}.$$

$$R_l = \begin{cases} TR_{\text{pot}}, & TR_{\text{pot}} \geq 1, \\ 1, & TR_{\text{pot}} < 1, \end{cases} \quad T = \frac{\text{Tone}}{100}.$$

where $C_z = 220 \text{ nF}$, $R_z = 220 \Omega$, $R_f = 1 \text{ k}\Omega$, $R_{\text{load}} = 10 \text{ k}\Omega$, and $R_{\text{pot}} = 20 \text{ k}\Omega$.

2.4 Output Buffer

The final block of the TS808 circuit breakdown is the output buffer represented in Figure 5. This stage is almost identical to the input buffer with a DC bias at 4.5V, an emitter follower to lower output impedance, and a high-pass filter to rid any low-frequency humming noise.

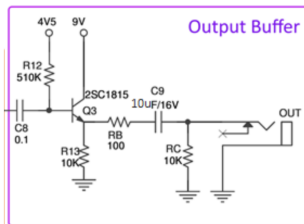


Fig. 5. Output buffer subsection of the TS808 circuit.

Similar to the input buffer, the output buffer works at a unity-gain factor of approximately 1. As the signal flows

through the transistor, the output impedance is lowered allowing for drive capabilities and preparing for the next stages of an amplifier. The component R_B provides stability in the circuit by slightly reducing the gain. Once the signal passes through C_9 , DC bias is effectively removed and the AC signal is centered around 0V. C_9 also works in series with RC to form a high-pass filter with a cutoff frequency of around 1.59 Hz. The transfer function for the final output buffer stage, $H_5(s)$, can be represented by:

$$H_5(s) = \frac{RCs}{RCs + 1},$$

where $R = 10 \text{ k}\Omega$ is the resistance and $C = 10 \mu\text{F}$ is the capacitance of the buffer stage.

3 ANALOG-TO-DIGITAL CONVERSION

After acquiring the five transfer functions, I then continued my modeling process by converting the subsystem representations to digital models using the bilinear transform, or *bilinear()* command in MATLAB. From there, I extracted the difference equation for each digital model.

3.1 $H_1(s)$

Transfer function:

$$H_1(s) = A_v \frac{RCs}{RCs + 1}$$

Z-transform:

$$H_1(z) = \frac{0.9915 - 0.9915z^{-1}}{1 - 0.9969z^{-1}}$$

Difference equation:

$$y_1[n] = 0.9915x[n] - 0.9915x[n-1] + 0.9969y[n-1]$$

3.2 $H_2(s)$

Transfer function:

$$H_2(s) = \frac{RCs}{RCs + 1}$$

Z-transform:

$$H_2(z) = \frac{0.9550 - 0.9550z^{-1}}{1 - 0.9099z^{-1}}$$

Difference equation:

$$y_2[n] = 0.9550x[n] - 0.9550x[n-1] + 0.9099y[n-1]$$

3.3 $H_3(s)$

Transfer function:

$$H_3(s) = \frac{1}{RCs + 1}$$

Z-transform:

$$H_3(z) = \frac{0.8002 + 0.8002z^{-1}}{1 + 0.6004z^{-1}}$$

Difference equation:

$$y_3[n] = 0.8002x[n] + 0.8002x[n-1] - 0.6004y[n-1]$$

3.4 $H_4(s)$

Transfer function:

$$H_4(s) = KV \frac{s + \omega_z}{(s + \omega_{p1})(s + \omega_{p2})}.$$

Z-transform:

$$H_4(z) = \frac{0.0001(0.1054 + 0.0018z^{-1} - 0.1036z^{-2})}{1 - 1.8220z^{-1} + 0.8253z^{-2}}$$

Difference equation:

$$\begin{aligned} y_4[n] = & 1.054 \times 10^{-5}x[n] + 1.8 \times 10^{-7}x[n-1] \\ & - 1.036 \times 10^{-5}x[n-2] + 1.8220y[n-1] \\ & - 0.8253y[n-2]. \end{aligned}$$

3.5 $H_5(s)$

Transfer function:

$$H_5(s) = \frac{RCs}{RCs + 1}$$

Z-transform:

$$H_5(z) = \frac{0.9999 - 0.9999z^{-1}}{1 - 0.9998z^{-1}}$$

Difference equation:

$$y_5[n] = 0.9999x[n] - 0.9999x[n-1] + 0.9998y[n-1]$$

4 MATLAB IMPLEMENTATION AND RESULTS

The derived analog transfer functions were implemented in MATLAB and converted to discrete-time filters using the bilinear transform. The linear stages were evaluated using their frequency responses, impulse responses, and pole-zero locations. Three combinations of the user-controlled parameters were then examined to demonstrate the effect of the drive, tone, and volume controls on the modeled system.

Because the clipping stage is nonlinear, it cannot be fully represented by a conventional linear transfer function. The clipping subsystem was therefore implemented separately in the time domain and inserted between the linear high-pass and low-pass stages.

4.1 Linear System Response

The cascaded linear model was evaluated through its frequency response, impulse response, and pole-zero representation. The frequency response demonstrates the combined effect of the input buffer, clipping-stage filtering, tone network, and output buffer. The low-frequency roll-off is primarily introduced by the high-pass stages, while the tone network provides the principal frequency-dependent variation controlled by the user.

The impulse response provides a time-domain representation of the combined linear filtering, while the pole-zero plot provides an indication of the stability of the resulting discrete-time system. For the parameter settings tested, the poles remain within the unit circle, indicating a stable digital filter implementation.

4.2 Parameter Comparison

Three control configurations were examined: (a) Drive = 0, Tone = 0, Volume = 100; (b) Drive = 0, Tone = 100, Volume = 100; and (c) Drive = 80, Tone = 60, Volume = 70. Configurations (a) and (b) isolate the effect of the tone control, while configuration (c) combines an intermediate tone setting with reduced output volume.

The frequency responses, impulse responses, and pole-zero plots of the cascaded linear filters are shown in Figures 6, 7, and 8, respectively. These analyses characterize the linear filtering stages; the nonlinear clipping operation is evaluated separately.

Figure 6 shows that all three configurations attenuate low and high frequencies while retaining a broad response in the intervening frequency range. Comparing panels (a) and (b) shows that changing the tone setting alters the magnitude and shape of the linear frequency response. Panel (c) reflects the combined effect of a different tone setting and reduced volume.

In Figure 7, each response exhibits an initial transient followed by a decaying tail that approaches zero, with differences in transient shape and amplitude across configurations.

The pole-zero plots in Figure 8 show poles inside the unit circle, including poles close to its boundary, consistent with stable filters and slowly decaying response components. These comparisons describe the linear filtering stages and do not measure nonlinear distortion introduced by the drive control.

4.3 Nonlinear Clipping Implementation

The nonlinear clipping behavior was approximated using a hyperbolic tangent function. A normalized drive parameter,

$$D = \frac{\text{Drive}}{100},$$

was used to determine the amount of pre-clipping gain,

$$G = 1 + 9D^2.$$

The amplified signal was then computed as

$$x_g[n] = Gx[n],$$

and soft clipping was applied according to

$$x_{nl}[n] = V_t \tanh\left(\frac{x_g[n]}{V_t}\right),$$

where $V_t = 0.3$ controls the transition into saturation. The final clipping-stage output was obtained by crossfading between the unclipped and nonlinear signals,

$$x_{clip}[n] = (1 - D)x[n] + Dx_{nl}[n].$$

This implementation allows the drive control to continuously vary the model from approximately linear operation at

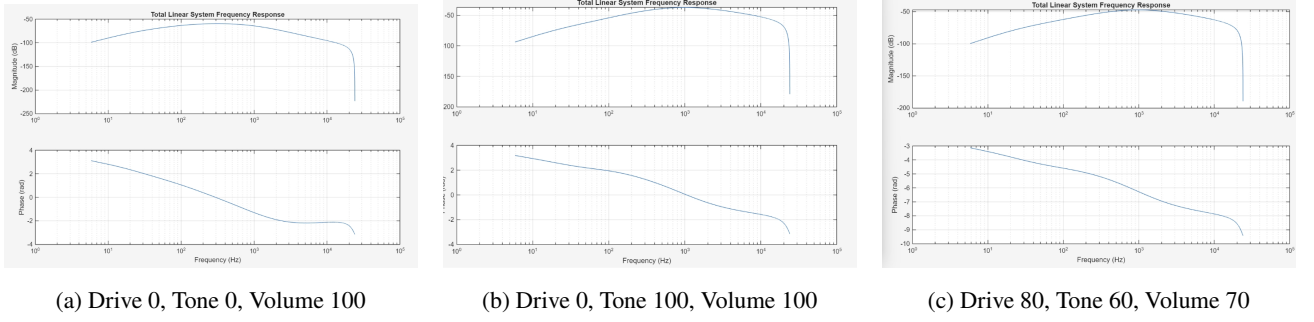


Fig. 6. Frequency responses of the cascaded linear TS808 model for three control configurations.

$D = 0$ to increasingly strong soft clipping as D approaches 1.

5 DISCUSSION

The resulting model demonstrates that the primary behavior of the TS808 can be represented as a cascade of linear filtering stages surrounding a nonlinear clipping operation. The high-pass filtering in the input and clipping stages reduces low-frequency content before distortion, while the tone network provides a user-controlled spectral shaping stage after the nonlinearity. The output buffer then provides a final high-pass response before the signal leaves the modeled pedal.

The discrete-time implementation also demonstrates the usefulness of subsystem-based virtual analog modeling. Rather than attempting to model the entire circuit as a single transfer function, each approximately linear section can be represented independently while nonlinear components are implemented directly in the time domain. This structure also allows individual stages to be modified or replaced without redesigning the complete system.

5.1 Model Limitations

Several simplifications limit the accuracy of the current model. First, the transistor input and output buffers are represented primarily by their frequency response and approximate voltage gain rather than by detailed nonlinear semiconductor models. Second, the diode clipping behavior is approximated using a hyperbolic tangent function rather than solving the nonlinear diode equations directly. As a result, the model reproduces the general behavior of soft clipping but does not explicitly model the voltage-current characteristics of the physical diodes.

The model also assumes nominal component values and does not account for component tolerances, temperature variation, transistor variation, or loading between all adjacent circuit stages. In addition, the bilinear transform introduces frequency warping when mapping the analog filters to the digital domain. Although these effects are relatively small for several of the modeled stages, they may contribute to differences between the simulated response and that of a physical TS808.

Finally, the present evaluation is based primarily on analytical transfer functions and MATLAB simulations. A more rigorous validation would compare the digital model against measurements obtained directly from a physical TS808 using frequency-response measurements, swept-sine analysis, and time-domain comparison under multiple drive and tone settings.

6 CONCLUSION

This work presented a digital model of the Ibanez Tube Screamer TS808 based on a subsystem decomposition of the analog circuit. The input buffer, clipping-stage filters, tone and volume network, and output buffer were represented using analog transfer functions and converted to discrete-time filters using the bilinear transform. The nonlinear clipping behavior was implemented separately using a drive-dependent soft-clipping function.

MATLAB simulations were used to examine the frequency response, impulse response, and pole-zero behavior of the resulting system under several control configurations. The model reproduces the principal signal-processing structure of the TS808, including low-frequency attenuation before clipping, nonlinear saturation, tone-dependent spectral shaping, and output-level control.

Although the implementation uses several circuit simplifications, it provides a practical framework for virtual analog modeling of the TS808 and serves as a foundation for a real-time audio plugin implementation. Future work could improve the model through circuit measurements, more accurate diode and transistor models, and direct comparison against a physical pedal.

7 SUPPLEMENTAL MATERIAL

The digital model can be downloaded at <https://github.com/jacesch/TubeScreamerDigitalModel>. A further-developed plugin based off of the TubeScreamer digital model is available at <https://github.com/jacesch/TubeScreamer-Emulation-Plugin>.

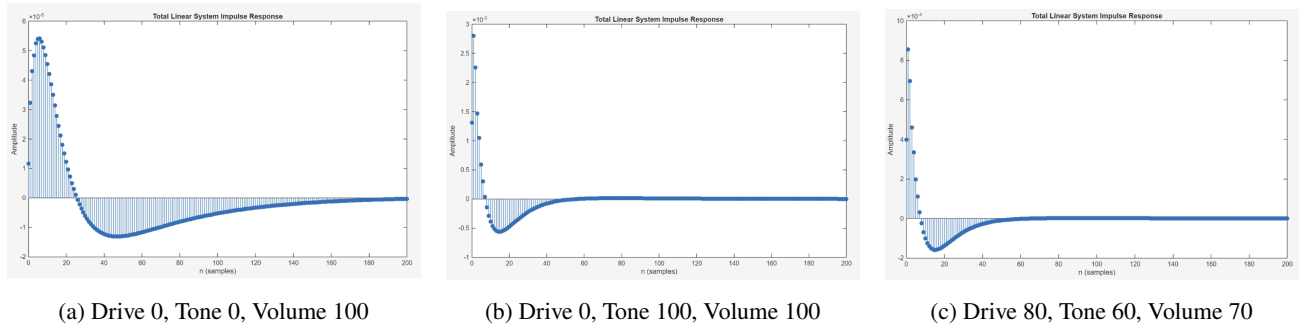


Fig. 7. Impulse responses of the cascaded linear TS808 model for three control configurations.

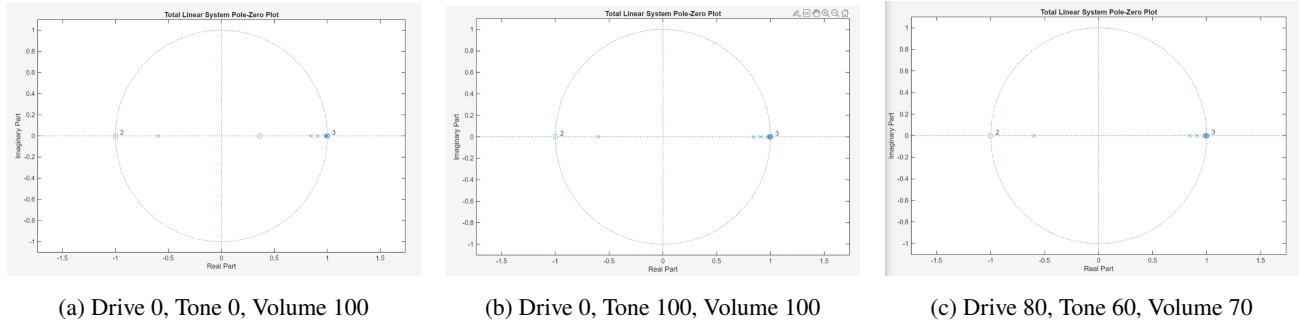


Fig. 8. Pole-Zero Plots of the cascaded linear TS808 model for three control configurations.

8 ACKNOWLEDGMENT

This work has been completed as a requirement of a course that is part of the Music Engineering Technology Department curriculum at the Frost School of Music, University of Miami: Audio Software Development I- Dr. Chris Bennett.

9 REFERENCES

[1] ElectroSmash, “Tube Screamer Analysis,” <https://electrosmash.mas-effects.com/tube-screamer-analysis.html>, electroSmash Archive. Accessed October 1, 2026.

THE AUTHOR



Jacey Schell

Jacey Schell is an audio engineer and DSP-focused developer pursuing a B.S. in Music Engineering Technology with a minor in Computer Engineering along with an M.S.

in Music Engineering Technology. Her work spans real-time audio processing, ML-driven systems, creative DSP, and immersive sound.
