Simulating a Triode Tube Guitar Amplifier Using a Non-Linear State-Space Representation

Arthur Befumo\(^*\), Advisor Scott Petersen\(^\dagger\)

Yale CPSC 490

1 Introduction

The triode vacuum tube, invented by Lee De Forest in 1906, is a simple yet powerful device which can be utilized to amplify audio signals. Triode tubes remained widely in use until the 1970s, at which point transistors became favored for their small size and relative power efficiency. Nonetheless in certain applications, particularly electric guitar amplifiers, the “signature” sound which triode tubes create is still valued. Although difficult to describe, triode tubes create a pleasant distortion when the input signal rises above a certain threshold. Everything from the screaming guitar solos of classic rock to the warm and fuzzy tone of older jazz guitar recordings owe their distinctive sound to the effects of triode tube distortion.

There are several different approaches for simulating a triode tube amplifier. Many commercially available software treats the tube amplifier as a “black box,” employing a variety of digital signal processing techniques which are unrelated to the physics governing a triode tube\(^{[2]}\). This project takes a more physics-based approach, wherein a triode tube circuit is simulated (with some necessary approximations) to process a digital signal.

\(^*\)arthur.befumo@yale.edu
\(^\dagger\)scott.petersen@yale.edu
2 The triode amplifier circuit

A basic vacuum tube (or diode tube) consists of two terminals enclosed inside a vacuum. One of the terminals, called the cathode, is heated which causes electrons to pool around that electrode through a process called thermionic emission. Those electrons will flow to the other terminal, called the plate (or anode), at a rate dependant on the voltage applied to the cathode. A triode tube adds a third terminal in between the anode and cathode, called the grid. The voltage between the grid and the plate acts as a barrier for electrons flowing towards the cathode, thus enabling the an input signal connected to the grid to modulate the plate-to-cathode current[2]. The circuit being emulated is shown in Figure 1.

![Figure 1: A simple triode amplifier circuit](image)

The plate-to-cathode current \( I_p \) and grid-to-cathode current \( I_g \) are non-linear functions of both the grid-to-cathode and plate-to-cathode voltages. Norman Koren gives a closed form equation for which closely matches real world measurements[4]:

\[
I_p = \frac{E_1^E}{K_g} \left(1 + \text{sign}(E_1) \right)
\]

\[
E_1 = \frac{V_{gk}}{K_p} \ln \left[ 1 + \exp \left( K_p \left( \frac{1}{\mu} + \frac{V_{gk}}{\sqrt{K_{vb} + V_pk^2}} \right) \right) \right]
\]
The values for $\mu, K_p, K_{vb}, K_g$, and $E_x$ are dependant on the triode tube model. Below are the values for the 12AX7 triode tube.

<table>
<thead>
<tr>
<th>$\mu$</th>
<th>$K_p$</th>
<th>$K_{vb}$</th>
<th>$K_g$</th>
<th>$E_x$</th>
</tr>
</thead>
<tbody>
<tr>
<td>100</td>
<td>600</td>
<td>300</td>
<td>1060</td>
<td>1.4</td>
</tr>
</tbody>
</table>

Since $I_g$ is normally very small and has only a slight effect on the output voltage, its value is ignored in this simulation.

### 3 Extended state-space representations

A popular approach for simulating triode tube circuits is through an extended nonlinear state-space representation[1]. In the linear case, a state-space representation take a dynamic vector $X$ which represents the state of the system, an input vector $U$, and an output vector $Y$. $A, B, C, D$ are matrices which capture the linear behavior of the system.

\[
dX/dt = AX + BU
\]

\[
Y = CX + DU
\]

Since the triode tube circuit is nonlinear, this representation is extended by adding a static nonlinear state vector $W$ to account for implicit equations, and expressing $dX/dt$ and $Y$ has nonlinear functions of $X, W$, and $U$.

\[
dX/dt = f(X, W, U)
\]

\[0 = g(X, W, U)\]

\[Y = h(X, W, U)\]

Previous work by Cohen and Hélie[1] give the functions $f, g, h$ as well as the choice of state variables in $X$ and $W$.

\[U = V_{in}\]

\[X = [V_k \ V_{out} - V_p]\]

\[W = [V_p \ V_g]\]

\[Y = V_{out}\]
Note that the following equations are slightly modified from those by Cohen and Hélie, since here the grid-to-cathode current $I_g$ is ignored.

$$f(X, W, U) = \left[ \frac{-X_1}{R_k C_k} + \frac{I_p}{C_k} - \frac{X_2 + W_1}{R_o C_o} \right] = \frac{dX}{dt}$$

$$g(X, W, U) = \left[ W_1 + R_p \left( I_p + \frac{X_2 + W_1}{R_o} \right) - V_{bias} \right] - W_2 - U = 0$$

$$h(X, W, U) = X_2 - W_1 = V_{out}$$

The various resistances and capacitances for the circuit are given below.

<table>
<thead>
<tr>
<th>$R_p$</th>
<th>$R_g$</th>
<th>$R_o$</th>
<th>$R_k$</th>
<th>$V_{bias}$</th>
<th>$C_o$</th>
<th>$C_k$</th>
</tr>
</thead>
<tbody>
<tr>
<td>100 KΩ</td>
<td>220 KΩ</td>
<td>22 KΩ</td>
<td>2.7 KΩ</td>
<td>300 V</td>
<td>20 nF</td>
<td>10 µF</td>
</tr>
</tbody>
</table>

4 Design and implementation

The circuit simulation was initially implemented in MATLAB, which enabled quicker iteration and bug-fixing along with the ability to easily display the output. Once the MATLAB code was tested on audio with satisfactory results, it was rewritten in C for efficiency. The input to the circuit simulation is a digital audio stream (read from a file) with values in the range $[-1, 1]$.

The heart of the code is a state machine which computes the state vectors in the state-space representation of the circuit. State vectors $X$ and $W$ are initially set to 0. Let $X_n, W_n$ be the state vectors after receiving $n$ inputs. Given $U_{n+1}$, $X$ is updated using the Runge-Kutta second order method.

$$k_1 = \Delta t \cdot f(X_n, W_n, U_{n+1})$$

$$X_{n+1} = X_n + \Delta t \cdot f(X_n + \frac{k_1}{2}, W_n, U_{n+1})$$

Here $\Delta t = 1$/samplerate, which for typical audio files is between 44.1 and 96 KHz.

The equation $g(X_{n+1}, W_n, U_{n+1}) = 0$ is then solved using Newton-Raphson. Cohen and Hélie found that 4 iterations for the Newton-Raphson are sufficient to solve for $W_{n+1}$ with relatively high accuracy[1].
The process of transforming the output voltage to a digital sound sample is not entirely straightforward, and there is little published work on methods for doing so. Before processing audio, the state machine running the circuit simulation on an empty input to calculate a baseline output voltage ($\approx 43.4875$ for the 12AX7 tube). That number is then subtracted to center the output voltage around 0. The input file is processed in blocks, and each block keeps track of its maximum absolute value. The absolute maximum voltage over all blocks is then used to create a scaling factor which is applied before writing the data to an output file. This ensures that the final audio file will not undergo clipping due to values outside of the range $[-1, 1]$.

### 4.1 Live audio processing

Some work was done to implement a VST plugin in C++ using JUCE library and the C implementation of circuit simulation. Ultimately, the lack of a proper realtime method for converting the output voltage from the simulation to a digital audio signal led to a significant amount of clipping in the VST implementation, and quite a bit of noise occurred even at low input volume levels. However, the computational cost of the circuit simulation itself is not prohibitive to live audio, as demonstrated by other implementations of VSTs using similar methods[1][3].

### 5 Examples

All of the following audio samples were processed using the final C implementation of the circuit simulator and then graphed in MATLAB. Figure 2 shows a 440 Hz sine wave at several stages of processing. In order from left to right, top to bottom, are the signal after passing through zero, one, two, or three simulated triode tube circuits. The triode tubes "saturate" the raw signal, resulting in an output more closely resembling a square wave.\(^1\)

---

\(^1\)Note that each circuit flips the sign of the input. This is because a higher input voltage restricts the plate-to-cathode current, resulting in a lower output voltage. For this reason guitar preamplifiers often contain an even number of triode tubes chained together.
Figures 3 and 4 both show the same note played at different volumes on a guitar. On the left is the raw signal and on the right is the same signal after passing through two simulated triode tube circuits. When played quietly, the note is amplified in a relatively linear fashion, resulting in a waveshape very similar to the raw signal but with slightly greater amplitude. At a higher volume, the amplitude increases non-linearly, resulting in a distortion of the original waveshape.
Figure 3: A quiet D4 ($\approx 293.67$ Hz)

Figure 4: A loud D4 ($\approx 293.67$ Hz)
References

[1] Ivan Cohen, Thomas Hélie. Simulation of a guitar amplifier stage for several triode models: examination of some relevant phenomena and choice of adapted numerical schemes. 127th Convention of Audio Engineering Society, Oct 2009, New York, United States. <hal-00631757>

