Embedded Virtual Analog Modelling with Recurrent Neural Networks

Jatin Chowdhury

Center for Computer Research in Music and Acoustics (CCRMA)

Virtual Analog Modelling

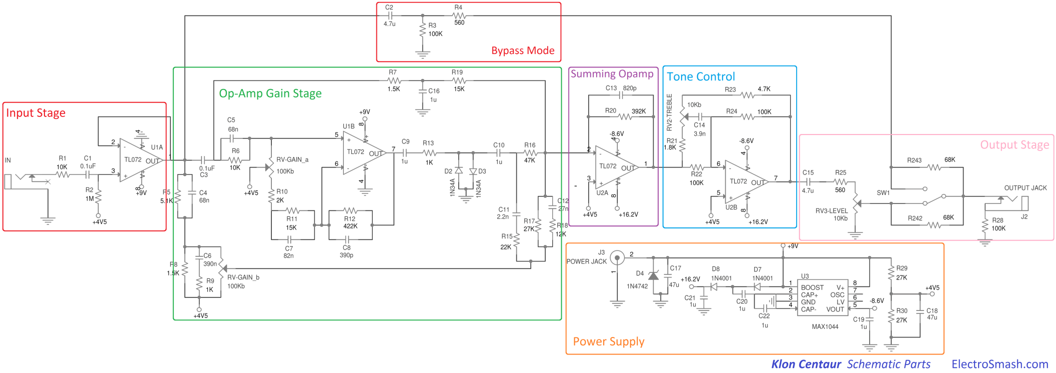
Creating a digital emulation of a classic analog audio effects.

- Provide access to effects that are old or rare.
- Lower cost.
- Convenience.
- Improved understanding.

Klon Centaur

Guitar pedal made by Bill Finnegan (MIT) from 1994-2000

Klon Centaur Circuit Schematic



Traditional Circuit Modelling

Nodal Analysis[1]

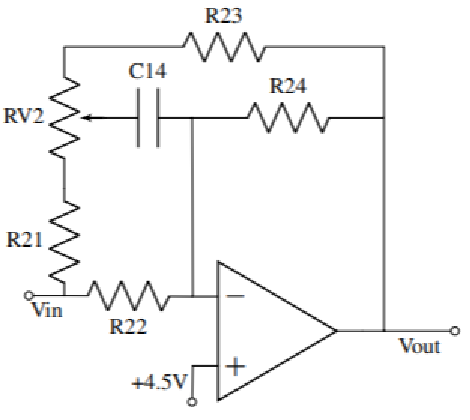


Figure 1: *Klon Centaur Tone Control Circuit*

[1] Smith, *Physical Audio Signal Processing*; Maby, *Solid State Electronics*.

Nodal Analysis

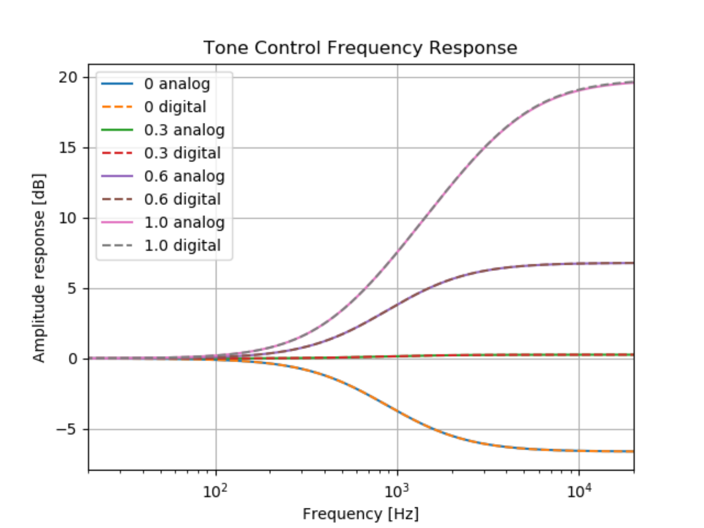
Laplace domain transfer function

$$\frac{V_{out}(s)}{V_{in}(s)} = \frac{C_{14}\left(\frac{1}{R22}+\frac{1}{R21+R_{v2b}}\right)s+\frac{1}{R22}\left(\frac{1}{R21+R_{v2b}}+\frac{1}{R23+R_{v2a}}\right)}{C_{14}\left(\frac{1}{R23+R_{v2a}}+\frac{1}{R24}\right)s+\frac{-1}{R24}\left(\frac{1}{R21+R_{v2b}}+\frac{1}{R23+R_{v2a}}\right)} \tag{1}$$

Bilinear transform (or other conformal map)

$$s \leftarrow \frac{2}{T}\frac{1-z^{-1}}{1+z^{-1}} \tag{2}$$

Nodal Analysis



Wave Digital Filters (WDFs)[2]

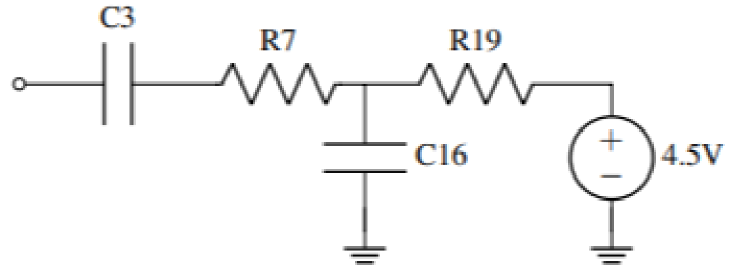


Figure 3: *Klon Centaur Feed-Forward Network 1 Circuit*

[2] Fettweis, "Wave digital filters: Theory and practice"; Werner, "Virtual Analog Modeling of Audio Circuitry Using Wave Digital Filters".

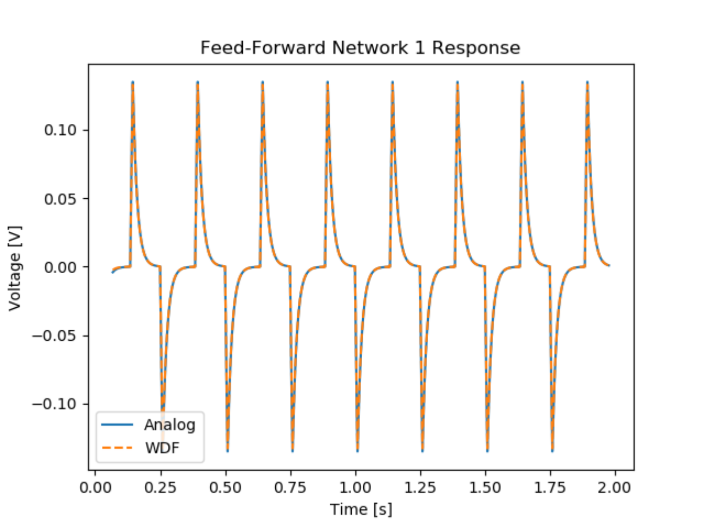
Wave Digital Filters (WDFs)

- Use wave variables instead of voltage/current.
- Port resistance free parameter.
- Discretize each circuit element separately.

$$
\begin{tikzcd}
 & & & R_7 & & R_{19} \\
 & & \mathcal{S}_2 \arrow[ur] \arrow[dr] & & \mathcal{S}_3 \arrow[ur] \arrow[dr] & \\
V_{in} \arrow[r] & \mathcal{S}_1 \arrow[ur] \arrow[dr] & & \mathcal{P}_1 \arrow[ur] \arrow[dr] & & V_{4.5} \\
 & & C_3 & & C_{16} &
\end{tikzcd}
$$

WDF tree for the Klon Centaur Feed-Forward Network 1 Circuit. $\mathcal{S}$ and $\mathcal{P}$ nodes refer to series and parallel adaptors respectively.

Wave Digital Filters (WDFs)



Neural Network Circuit Modelling

Recurrent Neural Network[3]

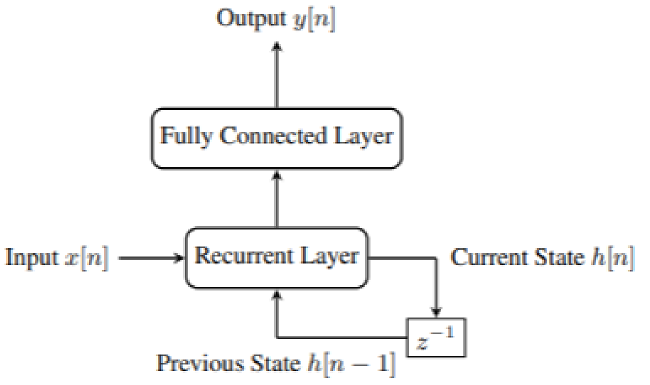


Figure 6: *RNN Architecture.*

[3] Wright, Damskagg, and Valimaki, “Real-Time Black-Box Modelling with Recurrent Neural Networks”.

Recurrent Neural Network

Recurrent layer: Gated Recurrent Unit

$$z[n] = \sigma(W_z x[n] + U_z h[n-1] + b_z) \tag{3}$$

$$r[n] = \sigma(W_r x[n] + U_r h[n-1] + b_r) \tag{4}$$

$$c[n] = \tanh(W_c x[n] + r[n] \circ U_c h[n-1] + b_c) \tag{5}$$

$$h[n] = z[n] \circ h[n-1] + (1 - z[n]) \circ c[n] \tag{6}$$

Recurrent Neural Network

Training Data:

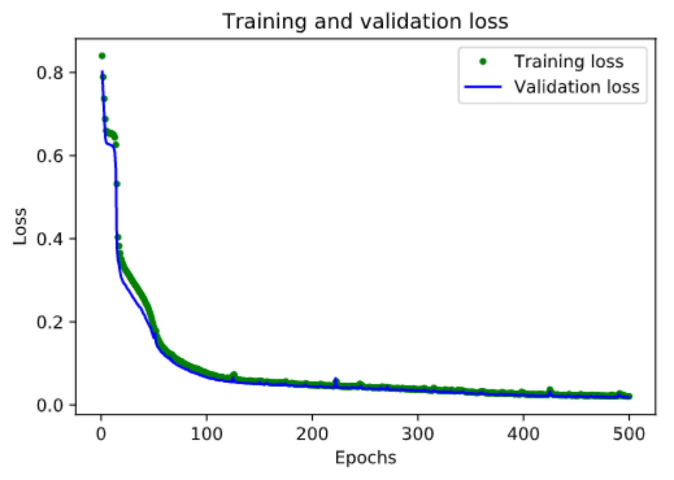
- ~ 4 minutes of guitar recordings (direct)
- Split into 0.5 second segments
- 400 training samples, 25 validation samples

Loss Function: Error-to-Signal Ratio

$$\mathcal{E}_{ESR} = \frac{\sum_{n=0}^{N-1} |y[n] - \hat{y}[n]|^2}{\sum_{n=0}^{N-1} |y[n]|^2} \tag{7}$$

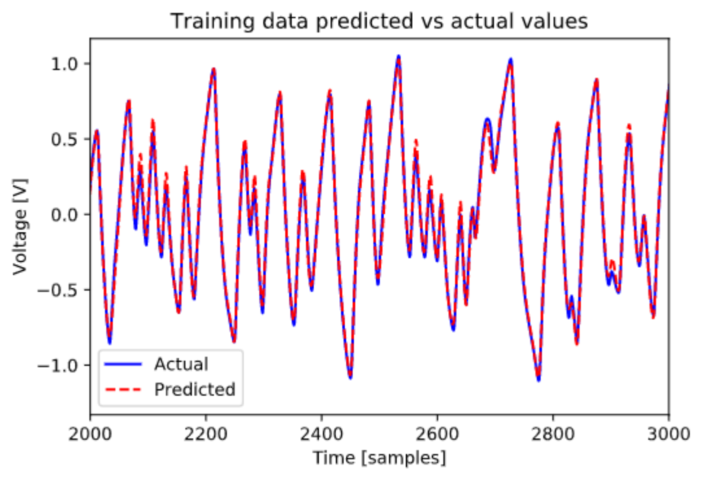
Recurrent Neural Network

Training: 500 epochs, ~ 8 hours



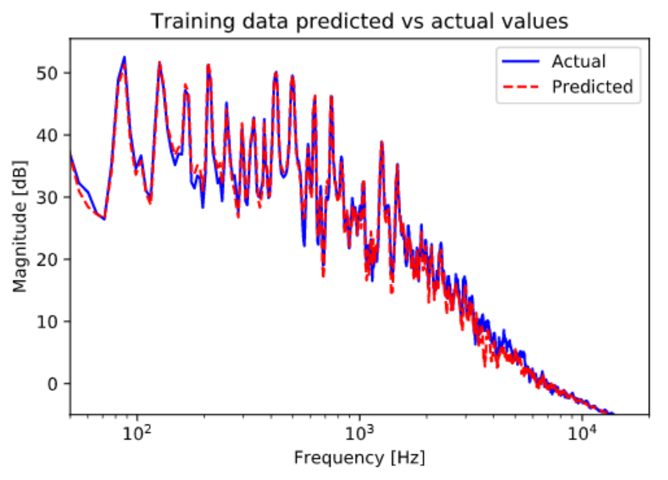
Recurrent Neural Network

Training results (time domain)



Recurrent Neural Network

Training results (frequency domain)



Real-Time Implementation

Implementation

Non-ML Implementation

- Use a combination nodal analysis, WDFs
- Control parameters for Treble, Gain, Level

ML Implementation

- RNN model for Gain Stage, nodal analysis elsewhere
- Fade between models for variable Gain control
- Custom GRU and Dense layer implementations in C++

Implementation

Inference Engine

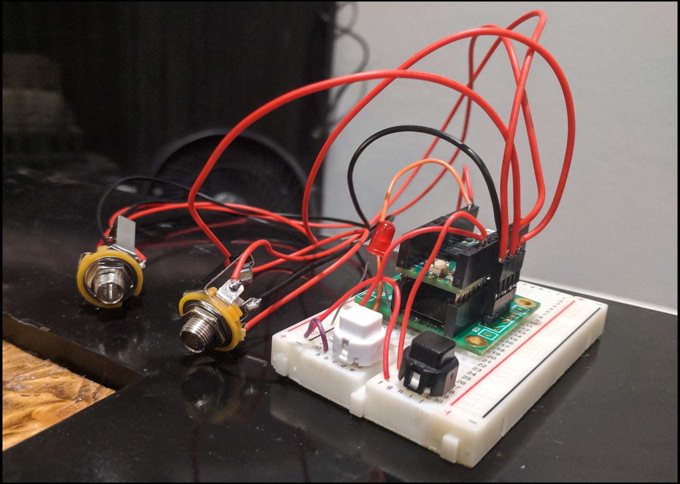
- Why use a custom implentation?
  - GRU support in TFLite is still experimental
  - Real-time audio thread: no locking, no memory allocation
- Currently using STL functions (`std::inner_product`)
- Eventually upgrade to Eigen or TFLite

Implementation

Desktop Audio Plugin (JUCE/C++)

Implementation

Teensy 4.0, Teensy Audio Shield, Teensy Audio Library



Results: Performance

Compute time per second of audio.

Block Size	NonML Speed	ML Speed
8	0.0723437	0.0528792
16	0.0703079	0.0510437
32	0.0652856	0.0511147
64	0.0662835	0.0502434
128	0.0666593	0.0495194
256	0.0696844	0.0480298
512	0.0669037	0.0477946
1024	0.060816	0.0488841
2048	0.0695175	0.0488309
4096	0.0623839	0.0472191

Results: Summary

- Subjectively, non-ML and ML models sound very similar.
- ML model has slightly damped high frequency response, (not a big deal on guitar input; more noticeable on other audio).
- ML model is more efficient!

Live Performance!