POLITECNICO MILANO 1863

SCUOLA DI INGEGNERIA INDUSTRIALE E DELL'INFORMAZIONE

OCD Emulator: Combining Black-Box and White-Box Models to Efficiently Emulate Audio Analog Circuitry in Real Time

M.Sc. Music & Acoustic Engineering

Selected Topics in Music & Acoustic Engineering Course

Stefano Ravasi, 224801

Professor:
M. J. Yee-King

Academic year:
2023/2024

Abstract: Nowadays, deep learning algorithms are widely used to solve many kinds of problem since they have proven to be very helpful in modelling complex behaviours which are difficult to define in terms of mathematical equations. Recurrent Neural Networks (RNNs) made possible processing time series input data, such as audio and video files. In particular, Long Short-Term Memory (LSTM) units turned out to be one of the best choices to emulate highly non-linear analog circuits. Though they also demonstrated to be very efficient at doing so, their computational cost becomes higher as their hidden size increases. Since a lot of audio circuitry contains linear stages, some already known white-box approaches, such as Wave Digital Filters (WDFs), can be employed to efficiently process audio signals in real-time applications. A mixed black/white-box approach could be beneficial to reduce computational costs while maintaining high fidelity in analog circuits virtual modelling. In addition, recording tons of processed audio data from physical instrumentation, varying each time their parameters, often results in a great effort: resorting to numerical simulated dataset could be the best way to save time and money in building advanced plugins that carefully reproduce "old school" analog sounds.

Key-words: C++, CMake, JUCE, RNN, LSTM, WDF

1. Introduction

This paper will describe the VST plugin emulation of a very popular distortion pedal released by Fulltone in 2004: the OCD (Obsessive Compulsive Drive). The processing stages and the user interface are entirely coded in a JUCE project (C++ programming language) built with CMake. The dataset employed during the training of the RNN model is obtained through LTspice simulations performed on a detailed reproduction of the real circuit (fig.A).

The power supply circuit has been significantly simplified since all the capacitors and diodes present have no use in the digital domain: the voltage supplied by the battery B_1 will always be constant and not subject to any fluctuation or disturbance. As a result, only the voltage divider made up of resistors R_{13} and R_{14} will be considered for feeding power to operational amplifiers and transistors.

Figure 1: Plugin GUI

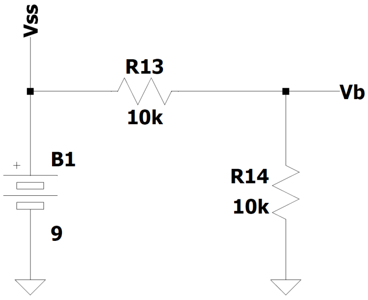


Figure 2: Simplified power supply circuit

The preamplification and clipping stage of the circuit instead, are fully modelled by means of a conditioned recurrent neural network based on long short-term memory units and a final dense layer to obtain the single channel signal output. Two inputs are passed to the RNN: the drive gain, which determines the resistance value of the logarithmic potentiometer in the preamplification stage, and the clean audio sample. An input limiter is also employed to avoid inputs that exceed the training range [-1,1] typical of wave files.

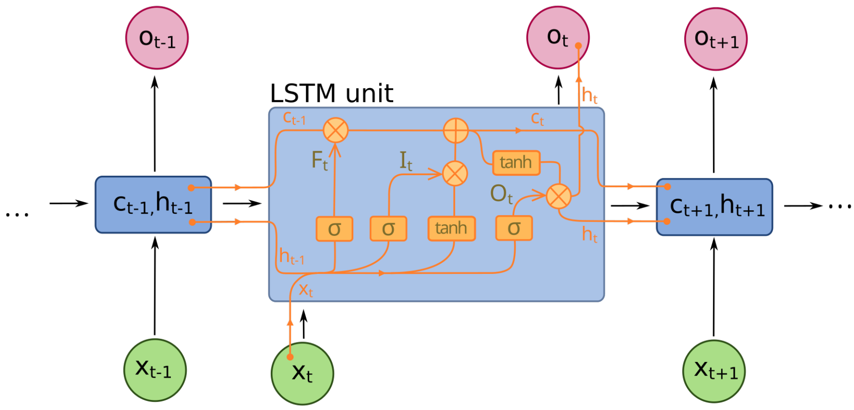


Figure 3: Recurrent neural network scheme

The tone control circuit will be modelled using the well-known time domain technique of wave digital filters for virtual analog emulation. The almost perfect linearity of this circuit makes this method really suitable even in real-time applications.

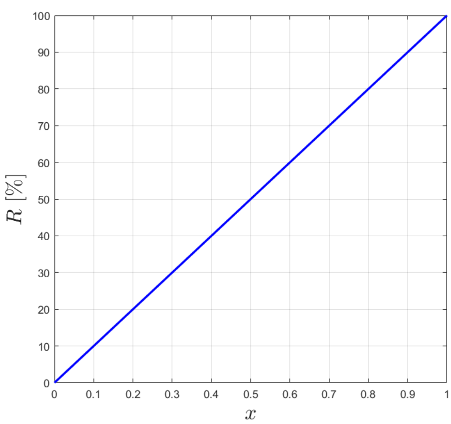


Figure 4: Linear potentiometer curve

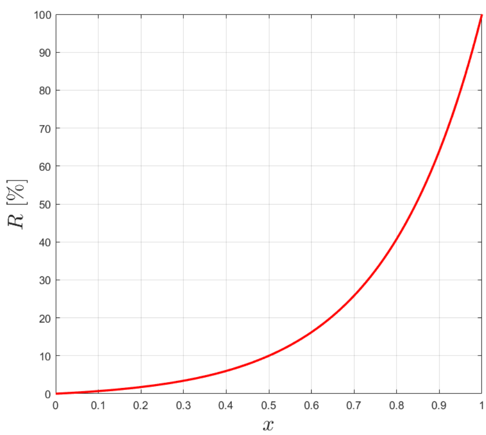


Figure 5: Logarithmic potentiometer curve

2. Distortion Circuit RNN Modelling

2.1. Creating the dataset

The dataset for training the conditioned neural network model were made using a clean electric guitar wave file (48 KHz - 16 bit PCM) of 7:06 minutes. Such input file was recorded to accurately represent the full dynamic range and capabilities of the instrument: chords, single notes, song sections and string noises were included to provide the widest variety of sounds possible.

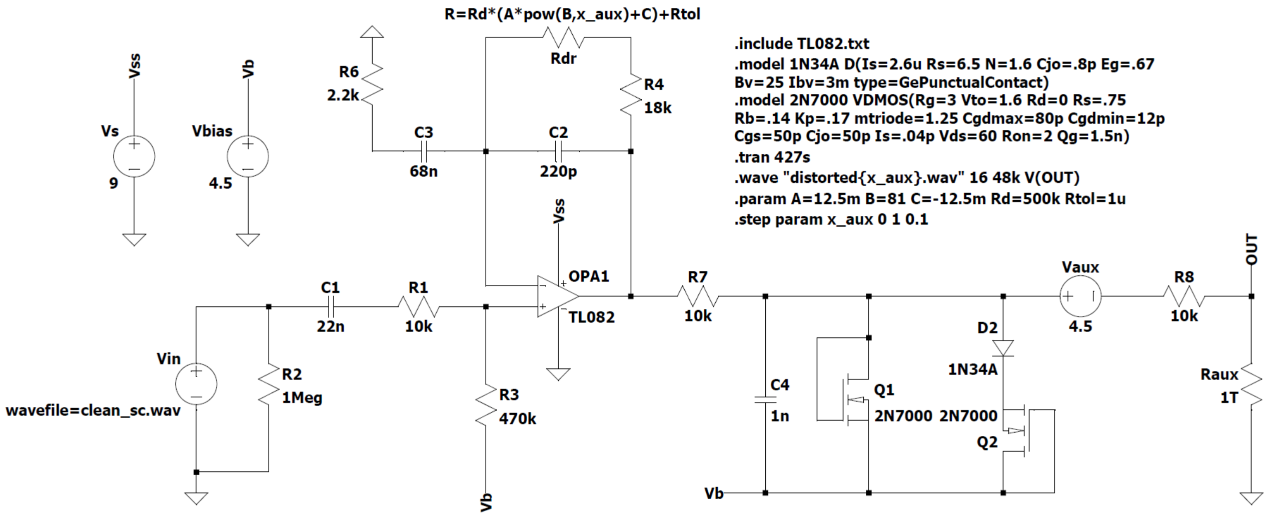


Figure 6: Auxiliary distortion circuit

The circuit is disconnected from the last part (before the second opamp) and two auxiliary elements are added as shown in fig.6:

- A DC voltage generator V_{aux} in series to the resistor R_8 in order to cancel the offset imposed by the bias voltage imposed before R_3 (needed to translate the input signal into the operating range of the opamp OPA_1).
- A resistor R_{aux} of 1 TΩ to simulate the huge input impedance of the opamp OPA_2.

The output files are obtained by processing iteratively the entire wave file by LTspice 11 times by varying the parameter x_{aux} of 0.1 in the range $[0, 1]$. The parameter modifies at each iteration the resistance value of the logarithmic potentiometer R_{dr}, increasing each time the preamplifier stage gain that drives the clipping circuit (realized by means of two 2N7000 N-channel mosfets and a 1N34A germanium diode). Each output signal is saved in a wave file with the same characteristics of the input one.

Then, two passages are performed on the output signal in order to optimize the training results:

1. All the files are normalized at -14 dB LUFS-M max using the software Reaper by Cockos.
2. Each output is normalized according to the maximum amplitude absolute value found among all the 11 files.

This procedure aims at standardize the amplitude level out of the distortion stage, reducing the perception of a rise in volume as distortion increases.

2.2. Training the model

The conditioned LSTM model was trained adaptating slightly the open source code by Alec Wright forked by Keith Bloemer on GuitarML GitHub page.

The initial step consists in the configuration files setup:

1. In the first one, the number of parameters must be specified along with their values and the relative input/output files location.
2. In the second one, the hidden size, the unit type, the loss function and the pre-filtering options have to be set.

Then, the main script can be directly executed after choosing the number of training epochs, the number of inputs and the initial seed. The code automatically produces the trained model in .json format using PyTorch backend.

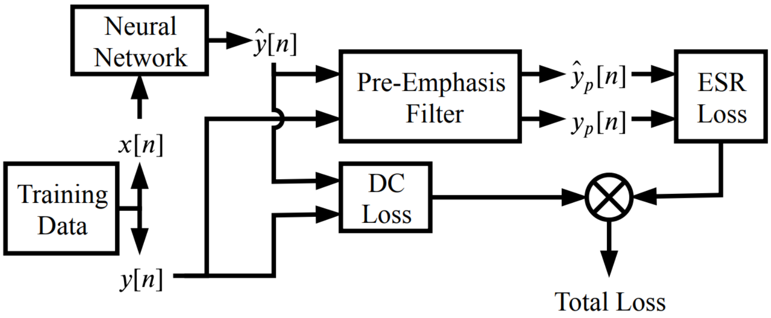


Figure 7: Training scheme adopted

For the specific model analysed in this report, the following specifications have been chosen:

- Number of parameters: 1 (preamplification stage gain)
- Number of inputs: 2 (1 sample plus 1 parameter)
- Hidden size: 32
- Unit type: LSTM
- Loss function: Linear combination of ESR (Error-to-signal ratio) and DC (Direct current offset)

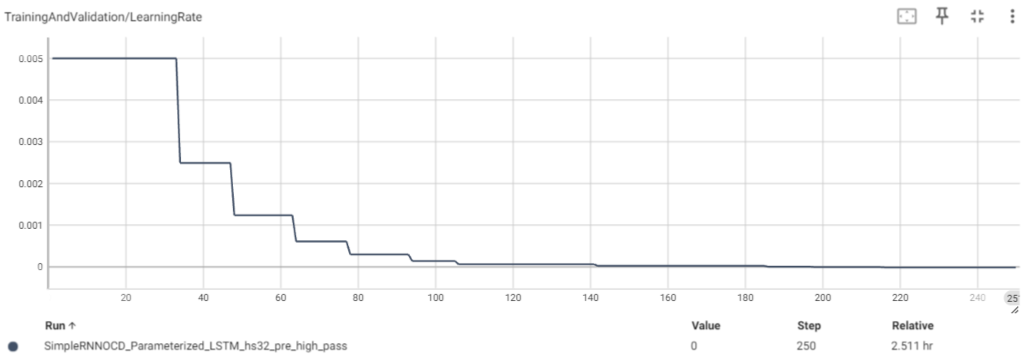
$$\mathcal{E}_{loss} = 0.75\mathcal{E}_{ESR} + 0.25\mathcal{E}_{DC} = \frac{1}{4}\left\{\frac{3\sum_{n=0}^{N-1}|y_p[n]-\hat{y}_p[n]|}{\sum_{n=0}^{N-1}|y_p[n]|^2} + \frac{|\sum_{n=0}^{N-1}(y[n]-\hat{y}[n])|^2}{N\sum_{n=0}^{N-1}|y[n]|^2}\right\}$$

- Pre-filtering stage: high-pass filter

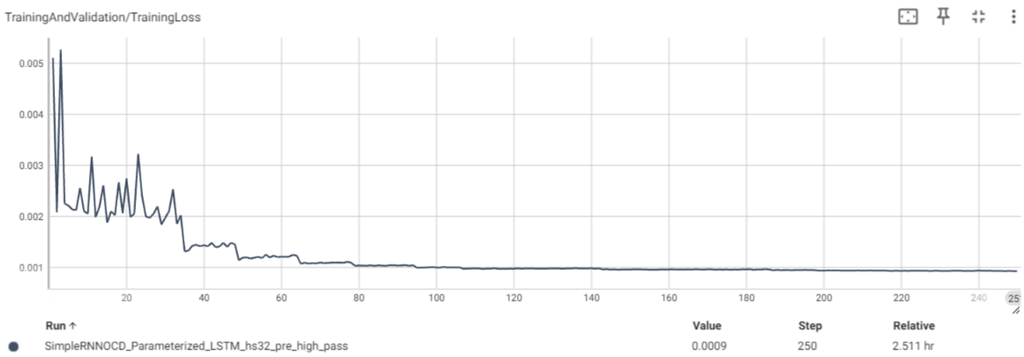
$$H_{HP}(z) = 1 - 0.85z^{-1}$$

- Number of epochs: 250
- Initial seed: 39

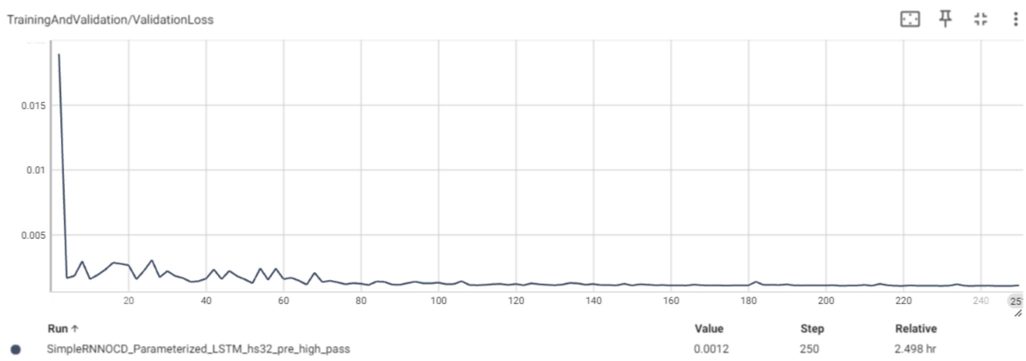
A Tensorboard session documented the entire training process:



(a) Learning rate curve



(b) Training loss curve



(c) Validation loss curve

Figure 8: Learning rate and losses trend during the training

2.3. Loading the model

The weights and the biases of the neural network (for both the LSTM units and the final dense layer) are then loaded with the RTNeural inferencing engine in a separate class. Such class also defines the main signal processing function. The model is embedded in the project as binary data: in this way the plugin will not search at each startup for any external file.

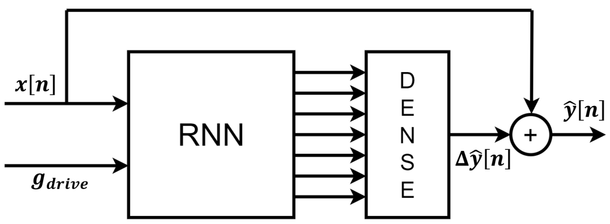


Figure 9: Neural network processing logic

Notice that the Python code is set to learn the difference between the input and the target output.

3. Tone Control Circuit WDF Modelling

3.1. Analysing the circuit topology

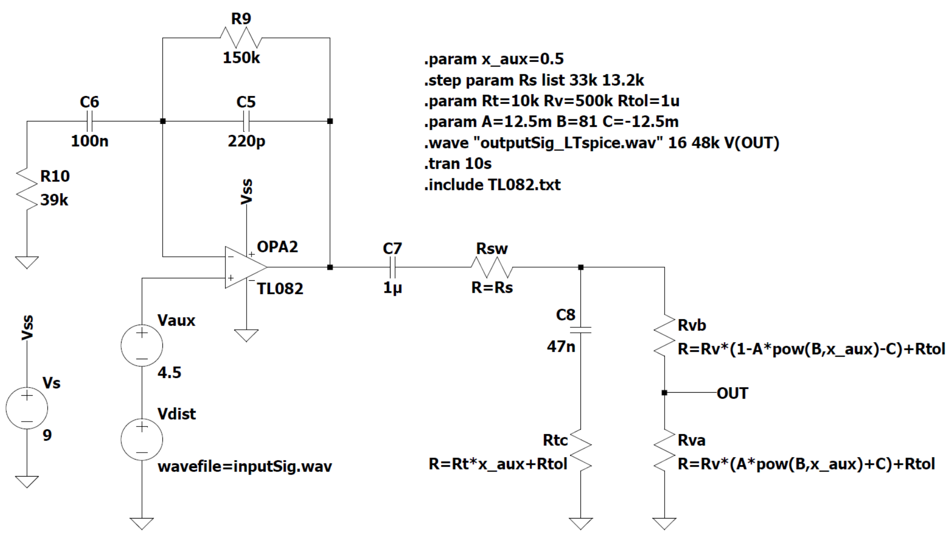


Figure 10: Tone control circuit (V_{aux} is accounted only for LTspice simulation)

As stated before, the last part of the circuit is linear: this assumption holds due to the fact that the opamp OPA_2 has a very wide linear operating range and a huge GBWP (its open-loop gain starts to decay around 20 KHz). These characteristics allows to treat it as an ideal operational amplifier: the integrated circuit can be easily modelled as a nullor with a grounded pin. The nullor is a virtual electrical device composed by a nullator and a norator: the first behaves like a short-circuit when the circuit is solved for voltages and as an open-circuit when solved for currents while the latter behaves exactly the opposite.

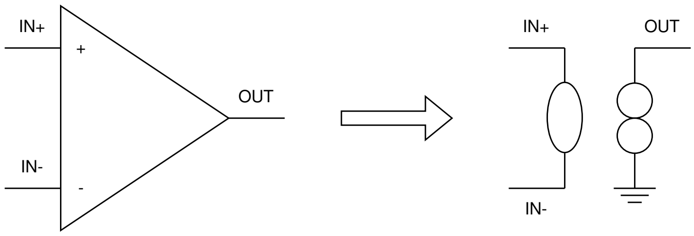
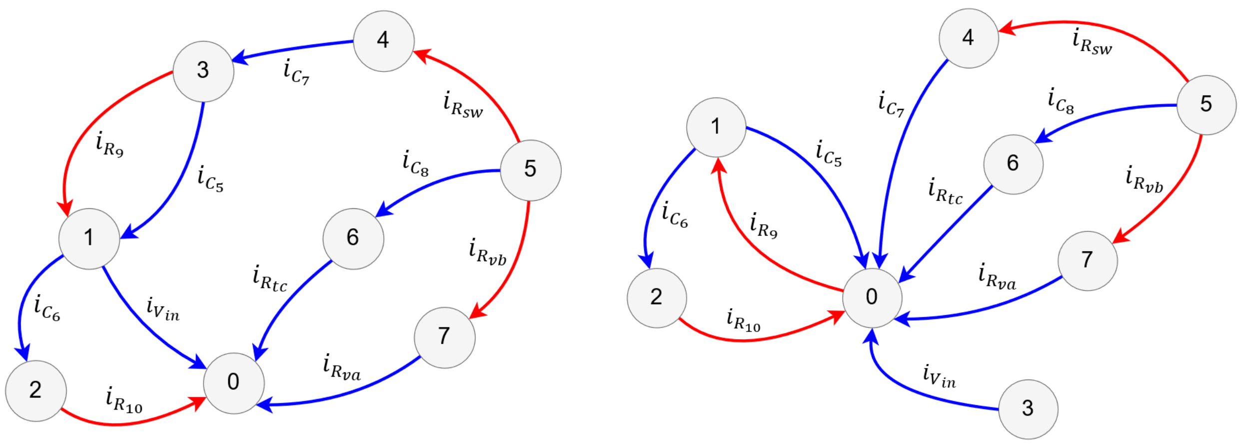


Figure 11: Ideal operational amplifier nullor model

This simplification enables to identify two distinct networks (fig.12) from the tone control circuit, one for voltages and another one for currents, that can be represented in thanks to graph theory.
It is worth noticing that both networks must share the same tree (highlighted in blue) and cotree (highlighted in red) decomposition to correctly perform the analysis: basically, the edges of the voltage network tree must be the same of the current network one (regardless of the adjacent nodes). In this way, these subcircuits share the same independent set of currents (the ones that flow through the cotree edges) and their fundamental loop matrices can be employed simultaneously to solve the system.
The fundamental loop matrix basically relates the independent set of currents to all the currents that flow in the circuit. It can be obtained by writing the Kirchhoff current laws at each node r for both networks.



(a) Voltage network graph (b) Current network graph

Figure 12: Tone control circuit representations according to graph theory

$$\sum_{n=1}^{N} i_n^{(r)} = 0 \qquad \Longrightarrow \qquad \begin{cases} \mathbf{i}_\mathrm{V} = \mathbf{B}_\mathrm{V}^\mathrm{T} \mathbf{i}_l \\ \mathbf{i}_\mathrm{I} = \mathbf{B}_\mathrm{I}^\mathrm{T} \mathbf{i}_l \end{cases}$$

$$\mathbf{i}_{\mathrm{V/I}} = \begin{bmatrix} i_{C_5} & i_{C_7} & i_{C_8} & i_{V_{in}} & i_{C_6} & i_{R_{tc}} & i_{R_{va}} & i_{R_9} & i_{R_{10}} & i_{R_{sw}} & i_{R_{vb}} \end{bmatrix}^T$$

$$\mathbf{i}_l = \begin{bmatrix} i_{R_9} & i_{R_{10}} & i_{R_{sw}} & i_{R_{vb}} \end{bmatrix}^T$$

$$\mathbf{B}_\mathrm{V} = \begin{bmatrix} -1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 1 & 1 & -1 & 1 & 0 & -1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & -1 & 0 & 0 & -1 & 1 & 0 & 0 & 0 & 1 \end{bmatrix} \quad , \quad \mathbf{B}_\mathrm{I} = \begin{bmatrix} -1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & -1 & 0 & 0 & -1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & -1 & 0 & 0 & -1 & 1 & 0 & 0 & 0 & 1 \end{bmatrix}$$

3.2. Building the WDF

Wave Digital Filters are based on a simple change of variables and the introduction of a scalar degree of freedom: the voltage across a bipolar element can be written as half the sum of an incident (to the element) wave b and a reflected (from the element) wave a while the current through it is expressed as half their difference scaled by the newly introduced reference port resistance Z (to be considered constant in time for the sake of simplicity).

$$v[k] = \frac{a[k]+b[k]}{2} \quad , \quad i[k] = \frac{a[k]-b[k]}{2Z} \tag{1}$$

For each linear element, it is always possible to find in advance a particular value of Z that cancels the instantaneous dependence of the reflected wave from the incident one. In other words, the reflected wave at the k-th sample $b[k]$ will only depend from a constant or previous values of the waves $a[n-m]$ and $b[n-m]$, with $m \in \mathbb{N}^+$.
The tone control circuit only contains a voltage generator (theoretically ideal but modelled as a resistive one imposing a tiny series resistance of 1 nΩ to make it easily adaptable), resistors and capacitors. The way to derive the right reference port resistances and the relative scattering relations is to substitute (1) in the constitutive equations of such elements, apply an appropriate discretization method for derivatives when needed (in this case the Tustin's method is employed) and then put to zero the coefficient of the instantaneous reflection $a[k]$.

- Resistive voltage source:

$$v[k] = V_g[k] + R_g i[k] \quad \Longrightarrow \quad b[k] = \frac{R_g - Z}{R_g + Z}a[k] + \frac{2Z}{R_g+Z}V_g \quad \Longrightarrow$$

$$\begin{cases} Z_g = R_g \\ b_g[k] = V_g[k] \end{cases} \tag{2}$$

- Ideal resistor:

$$v[k] = Ri[k] \quad \Longrightarrow \quad b[k] = \frac{R-Z}{R+Z}a[k] \quad \Longrightarrow$$

$$\begin{cases} Z_R = R \\ b_R[k] = 0 \end{cases} \tag{3}$$

- Ideal capacitor:

$$i(t) = C\frac{dv(t)}{dt} \quad \Longrightarrow \quad I(s) = sCV(s) \quad \Longrightarrow$$

$$I(z) = 2f_s\frac{1-z^{-1}}{1+z^{-1}}CV(z) \quad \Longrightarrow \quad i[k] + i[k-1] = 2f_sC(v[k]-v[k-1]) \quad \Longrightarrow$$

$$b[k] = \frac{1-2f_sCZ}{1+2f_sCZ}a[k] + \frac{1+2f_sCZ}{1+2f_sCZ}a[k-1] - \frac{1-2f_sCZ}{1+2f_sCZ}b[k-1] \quad \Longrightarrow$$

$$\begin{cases} Z_C[k] = \dfrac{1}{2f_sC} \\ b_C[k] = a_C[k-1] \end{cases} \tag{4}$$

After retrieving the expressions (2), (3), (4), it is possible to compute the numerical value for each element of the circuit and finally write the reference port resistances diagonal matrix $\mathbf{Z}$:

$$\mathbf{Z} = \mathrm{diag}\left(Z_{C_5}, Z_{C_7}, Z_{C_8}, Z_g, Z_{C_6}, Z_{R_{tc}}, Z_{R_{va}}, Z_{R_9}, Z_{R_{10}}, Z_{R_{sw}}, Z_{R_{vb}}\right)$$

Introducing an 11x11 identity matrix $\mathbf{I}$, the scattering matrix $\mathbf{S}$ can now be computed as:

$$\mathbf{S} = \mathbf{I} - 2\mathbf{Z}\mathbf{B}_{\mathrm{I}}^{\mathrm{T}}\left(\mathbf{B}_{\mathrm{V}}\mathbf{Z}\mathbf{B}_{\mathrm{I}}^{\mathrm{T}}\right)^{-1}\mathbf{B}_{\mathrm{V}}$$

For each input sample $V_g[k]$, the incident waves vector $\mathbf{b}[k]$ is computed according to the scattering relation of each element. Then the reflected waves vector $\mathbf{a}[k]$ is obtained evaluating the following matrix-vector product:

$$\mathbf{a}[k] = \mathbf{S}\mathbf{b}[k]$$

$$\mathbf{a}[k] = \begin{bmatrix} a_{C_5} & a_{C_7} & a_{C_8} & a_{V_{in}} & a_{C_6} & a_{R_{tc}} & a_{R_{va}} & a_{R_9} & a_{R_{10}} & a_{R_{sw}} & a_{R_{vb}} \end{bmatrix}^T$$

$$\mathbf{b}[k] = \begin{bmatrix} b_{C_5} & b_{C_7} & b_{C_8} & b_{V_{in}} & b_{C_6} & b_{R_{tc}} & b_{R_{va}} & b_{R_9} & b_{R_{10}} & b_{R_{sw}} & b_{R_{vb}} \end{bmatrix}^T$$

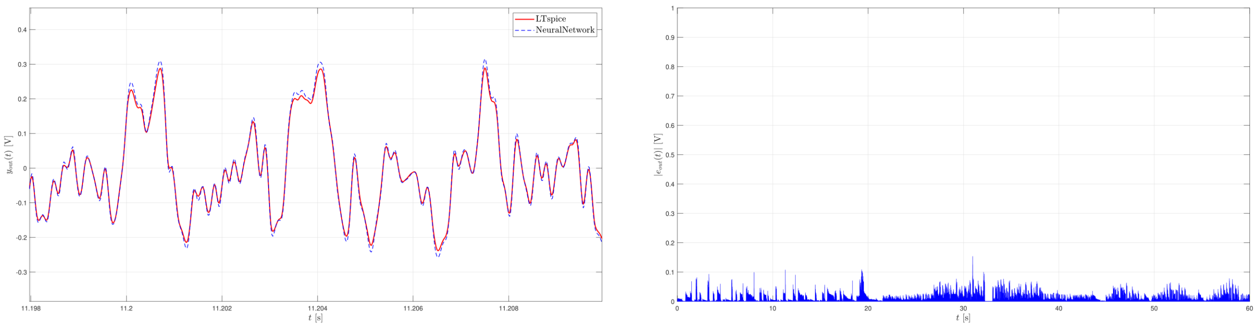
Recalling (1), the actual output sample (the voltage across the volume potentiometer grounded side R_{va}) is finally obtained:

$$v_{out}[k] = v_{R_{va}}[k] = \frac{a_{R_{va}}[k] + b_{R_{vb}}[k]}{2} = \frac{a_{R_{va}}[k]}{2}$$

4. Technical Evaluation

4.1. RNN implementation

The RNN implementation of the distortion circuit (preamplifier plus clipping stage) has been tested excluding momentarily the tone control stage from the processing chain of the plugin and feeding both the VST and LTspice with a 60 seconds long recording of a clean guitar and comparing the final results with a MATLAB script. A short window and the absolute value over the entire simulation time have been chosen to show the accuracy of the neural network modelling:



(a) Observation window

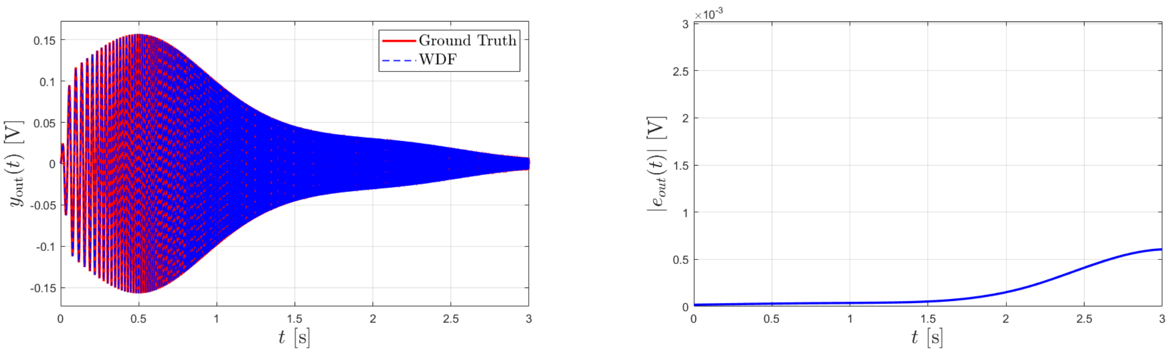
(b) Absolute error graph

Figure 13: Comparison between the Neural Network implementation and the LTspice simulation

The absolute error graph present a maximum value of 0.15 V, the mean absolute error value is $4.6 \cdot 10^{-3}$ V and the MSE is $5.6 \cdot 10^{-5}$ V: these values prove that the RNN is really good at simulating the real circuit modelled on LTspice.
Although the results are more than satisfactory, it should be pointed out that in some particular cases few outlier peaks become quite evident. This happens mainly when the incoming audio signal has really sharp transients and varies its dynamic range very fast (in less than 5 samples at 48kHz). The maximum value registered in the absolute error is around 0.42 V, while the average error and the MSE maintain approximately the same order of magnitude previously reported.

4.2. WDF implementation

The WDF implementation of the tone control circuit has been tested with a MATLAB script against the equivalent circuit simulated in LTspice by feeding both a sinusoidal exponential sweep ranging from 20 Hz to 20kHz in 3 s.



(a) Output comparison

(b) Absolute error envelope

Figure 14: Comparison between the MATLAB WDF implementation and the LTspice simulation

It can be easily noticed that the error introduced by the simplifying assumptions made to model the circuit with a wave digital structure has almost no effects on the output signal. The maximum absolute error value is $6 \cdot 10^{-4}$ V and it is located where the sweep reaches the maximum frequency: the derivative discretization

method inevitably introduce a small error that increases as the sampling frequency does. In other words, the introduced warping is mainly due to the bilinear transform: the ideal opamp assumption holds and approximate almost perfectly the original circuit.
In addition, the RTR measured repeating 1000 times the same processing cycles on 10 seconds of audio at 48 KHz has a mean value of $9.5 \cdot 10^{-3}$: the WDF realisation of the tone control is blazingly fast and its computational cost is very low.

5. Conclusions

The plugin gave satisfactory results both from a technical and a musical standpoint:

- The software is completely usable in real-time applications, even on non-high-end platforms.
- Only few lines of source code must be edited in order to recompile the software on/for a different platform.
- According to Cockos Reaper measurements (done on the following system: Intel i7-11800H, 16 GB DDR4 RAM, NVIDIA GeForce RTX 3060 Laptop, 1 TB SSD NVMe), the plugin consumes only the 0.8% of the available resources (0.7% dedicated to the neural network and 0.1% used by the limiter and the wave digital filter).
- The user interface resembles the original pedal, allowing musicians to easily recognise the physical controls.
- The user controllable parameters can be easily mapped to MIDI controllers in a DAW to tweak knobs and switches remotely.
- The distortion effect can be accurately set and does not assume extreme values instantaneously (due to logarithmic mapping).

Creating the dataset for training the neural network with computer aided simulations has a huge advantage: it enables users that are not practical with analog devices (or simply does not own them) to model circuitry with a high perceptual quality of the processed sound. Moreover, it allows to model with separate approaches different sections of the circuitry: in fact, the example proposed in this paper has been trained by running only 11 simulations on LTspice ($\tilde{5}$ h) and very few "short" training sessions ($\tilde{2}.5$ h).
If the dataset had been obtained from a real pedal, even increasing the sampling factor to 0.2 for each parameter, it would have taken at least 6x6x2=72 .wav files (not considering the volume control which could easily be implemented digitally) to model drive, tone and the filter switch. This approach would have drastically increased the training time of the network and could have required an increase in the hidden size of the recurrent layer (without considering all the possible difficulties that could be encountered in recording the output through an audio interface).
Even though this approach probably can lead to some lack of warmth in the processed sound, it has demonstrated to be a more efficient and feasible alternative for any user who does not have a powerful computer or a lot of time to create a working plugin.

References

[1] K. J. Werner - Virtual Analog Modeling of Audio Circuitry Using Wave Digital Filters - Ph.D. Dissertation - 2016.

[2] A. Wright, E. P. Damskägg, V. Välimäki - "Real-Time Black-Box Modelling with Recurrent Neural Networks" - Proc. of the 22nd Int. Conf. on DAFx - 2019.

[3] A. Wright, V. Välimäki - "Perceptual Loss Function for Neural Modelling of Audio Systems" - IEEE Inter. Conf. on Ac., Sp. and Sig. Proc. - 2019.

[4] R. Giampiccolo, M.G. De Bari, A. Bernardini, A. Sarti - "Wave Digital Modelling and Implementation of Nonlinear Audio Circuits With Nullors" - IEEE/ACM Trans. on Aud., Sp., and Lang. Proc. - 2021.

[5] GeneralGuitarGadgets website - "OC Dist schematics" - Accessible at: https://generalguitargadgets.com - Accessed: 08-07-2024

[6] STMicroelectronics website - “TL082 operational amplifier datasheet", "2N7000 n-channel mosfet datasheet" - Accessible at: https://www.st.com - Accessed: 17/07/2024.

[7] DatasheetCatalog website - "1N34A point contact germanium diode datasheet" - Accessible at: https://pdf.datasheetcatalog.com - Accessed: 17/07/2024.

[8] GuitarML GitHub page - "Automated-GuitarAmpModelling repository" - Accessible at: https://github.com/GuitarML/Automated-GuitarAmpModelling/tree/main - Accessed: 28/07/2024.

A. Full Schematics

Here the Fulltone OCD full schematics is represented:

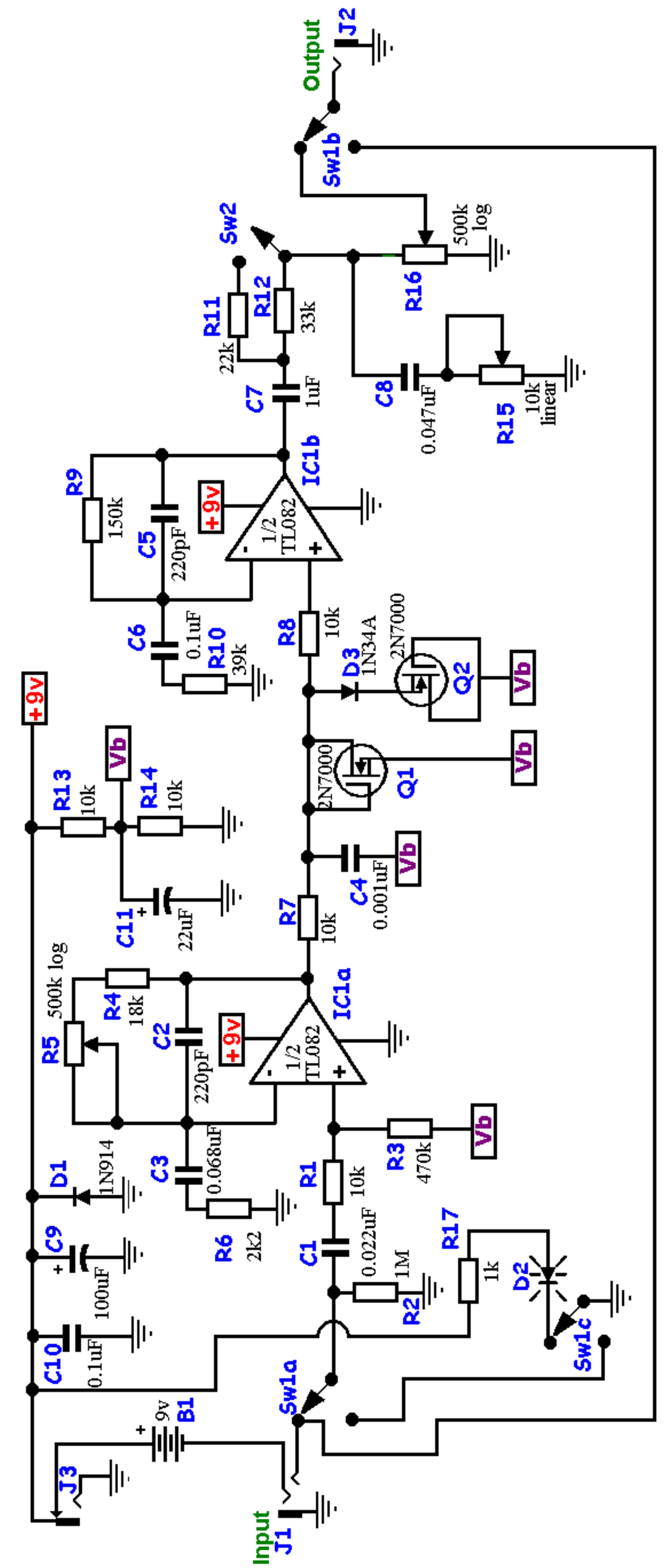


Figure 15: Full schematics of the pedal