NEURAL NET TUBE MODELS FOR WAVE DIGITAL FILTERS

Champ C. Darabundit [*], *Dirk Roosenburg* [†] *and Julius O. Smith III*

Center for Computer Research in Music and Acoustics
Stanford University
Stanford, CA
champ@ccrma.stanford.edu | dirk@ccrma.stanford.edu | jos@ccrma.stanford.edu

ABSTRACT

Herein, we demonstrate the use of neural nets towards simulating multiport nonlinearities inside a wave digital filter. We introduce a resolved wave definition which allows us to extract features from a Kirchhoff domain dataset and train our neural networks directly in the wave domain. A hyperparameter search is performed to minimize error and runtime complexity. To illustrate the method, we model a tube amplifier circuit inspired by the preamplifier stage of the Fender Pro-Junior guitar amplifier. We analyze the performance of our neural nets models by comparing their distortion characteristics and transconductances. Our results suggest that activation function selection has a significant effect on the distortion characteristic created by the neural net.

1. INTRODUCTION

Vintage audio gear based on analog tube amplifier circuits is responsible for many of the distinctive sounds that characterize music today. For example, genres such as blues, rock, and metal continue to rely heavily on guitar tube amplifiers. Tube-based vintage recording studio gear, such as as the LA-2A leveling amplifier and the Pultec EQP-1A program equalizer, continues to be widely used. Tube-based analog audio gear is expensive to reproduce, making it inaccessible to many, and, even worse, many popular pieces of audio gear derive their sound from electrical components that are no longer manufactured. As ever dwindling supply drives prices to unreasonable levels, the demand for effective digital emulation increases.

Virtual analog modeling seeks to create digital algorithms that accurately replicate the behavior of analog audio effects. Models can be broadly divided into two categories: "black-box" models that only emulate the effect of a specific piece of gear by analyzing and replicating only its input-output behavior, and "white-box" models that emulate the effect by analyzing internal behavior and replicating its physical operation.

Wave-digital filters (WDFs) have been thoroughly investigated as a white-box method for modeling many audio circuits [1–8]. Tubes have been modeled with WDFs in [9–11] using a simplified model with ad hoc delays and in [12] using a blockwise K-method.

[*] Authors listed alphabetically and should be considered co-authors.

[†] Correspondence should be addressed to Dirk Roosenburg (dirk@ccrma.stanford.edu)

WDFs are especially effective for simulating the linear parts of circuits, providing a high degree of modularity and desirable properties [13]. Circuits containing multiple or multiport nonlinearities have been specifically addressed in WDF models by [4, 14–16]. Most nonlinear circuits do not have closed form WDF solutions, requiring resolution of delay-free loops via table-lookup, functional approximation, or iterative techniques (typically using Newton's method or its variants) [1,3,15–18]. As an alternative to these resolution methods, we propose to use of machine learning to capture the behavior of the nonlinear ports in the wave domain.

Machine learning has been usefully applied to virtual analog modeling in the Kirchhoff domain using black-box models for tubes circuits in [19], generally in [19–23], and in grey-box models which utilize knowledge of the circuit to train neural networks within a white-box structure [24]. Machine learning has more recently been applied to grey-box modeling in the WDFs for single-port nonlinearities [25].

In this article, we propose a wave domain definition based on the K-method [17, 26] that explicitly resolves the delay-free loop for multiport nonlinearities. We use this definition to extract the feedforward wave domain behavior from a dataset describing a nonlinearity in the Kirchhoff domain. This dataset is then used to train a neural net to functionally approximate the resolved wave behavior and replace the multiport adaptor in a WDF model of a tube amplifier circuit. The combined neural net and wave digital filter model can run in real-time.

Section 2 presents theoretical background on WDFs and derives the resolved wave transformation. Section 3 provides some background on machine learning and the design of the neural net models used in our case study, the first stage of the Fender Pro-Junior preamplifier. Case study results appear in Section 4. Section 5 summarizes and concludes the paper.

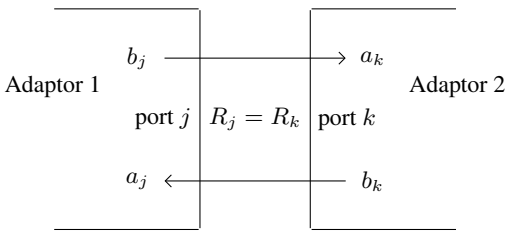
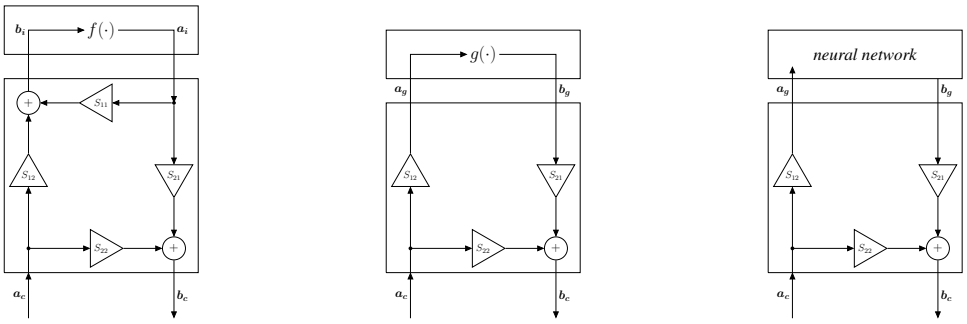


Figure 1: *Two adaptors connected by a single port. Connecting them enforces the equality of wave variables* $a_k = b_j$, $a_j = b_k$, *and port resistance* $R_j = R_k$.



(a) *unresolved adaptor* (b) *resolved adaptor* (c) *neural net adaptor*

Figure 2: *The proposed loop resolution process. The resolved wave definition — equation (7b) — is used to transform the nonlinearity in (a) to the form* $\mathbf{g}(\cdot)$ *in (b) which is then approximated using a neural net as in (c).*

2. WAVE DIGITAL FILTER THEORY

2.1. Background

Wave-digital filters represent electrical circuits in the *wave domain* — i.e. in terms of *traveling-wave components* — in place of the usual Kirchhoff-domain variables: voltage and current. A vector voltages, $\mathbf{v}$, and currents, $\mathbf{i}$, across and through a port are related to their traveling-wave components $\mathbf{a}$ and $\mathbf{b}$ by the parametric wave definition:

$$\begin{aligned}\mathbf{a} &= \mathbf{R}^{\rho-1}\mathbf{v} + \mathbf{R}^{\rho}\mathbf{i} \\ \mathbf{b} &= \mathbf{R}^{\rho-1}\mathbf{v} - \mathbf{R}^{\rho}\mathbf{i} \end{aligned} \tag{1a}$$

$$\begin{aligned}\mathbf{v} &= \frac{1}{2}\mathbf{R}^{1-\rho}\mathbf{a} + \frac{1}{2}\mathbf{R}^{1-\rho}\mathbf{b} \\ \mathbf{i} &= \frac{1}{2}\mathbf{R}^{-\rho}\mathbf{a} - \frac{1}{2}\mathbf{R}^{-\rho}\mathbf{b}, \end{aligned} \tag{1b}$$

where ρ is a constant that changes the physical units of the wave variables in the adaptor [4, 13]. For voltage waves, $\rho = 1$. Applying the parametric wave definition results in a change of basis between the Kirchhoff domain (i, v) and the wave domain (a, b), and the circuit can now be viewed as an electric transmission line with *port impedance* $\mathbf{R}$.

WDFs are composed of a network of adaptors connected to ports such as a resistor, capacitor, or an inductor. The port impedances of a resistor R, capacitor C, and inductor L, are R, $T/(2C)$, and $2L/T$ respectively, where T denotes the sampling interval in seconds. These values are obtained by discretizing the continuous-time reflectances using the bilinear transform and choosing the port impedance that suppressing the instantaneous reflected wave. This step is known as *adapting* the model which additionally enforces causality in the WDF.

The WDF is preferably arranged in a *binary connection tree* [27], which requires only three-port adaptors. When two ports are connected, as shown in Figure 1, the incident wave of one is equal to the reflected wave of the other ($a_k = b_j$ and $a_j = b_k$). Connecting two different wave impedances results in a *scattering junction* called an *adaptor*.The free port impedance of an adaptor (the port facing the root of the tree and connecting to another adaptor) is chosen to eliminate the instantaneous reflection from that adaptor as seen from the root of the tree. This practice eliminates delay-free loops in the model's overall structure [28]. Specifically, a series adaptor sets its root-facing port to the sum of the two port impedances it connects, while a parallel adaptor presents the parallel combination of those impedances.

The binary connection tree is only sufficient for series and parallel connections. In more general circuits, the adaptors may be systematically defined based on an SPQR tree decomposition [14] that allows any given circuit to be represented by Series, Parallel, and Rigidly connected WDF elements. The Rigid node is not decomposable into series and or parallel subgraphs and an $\mathcal{R}$-type adaptor is used to represent the complex topology. In [8], Werner et al. demonstrated how modified nodal analysis (MNA) may be used to create custom $\mathcal{R}$-type adaptors for any rigid topological connection between circuit elements. Several adaptors for nonlinear circuit elements exist, such as the Chua diode [29] and the Schottky diode [2], but their use is usually limited to a single nonlinear element or element combination at the root of the binary connection tree. This is because only the root of the tree may have an instantaneous reflection without introducing a delay-free loop.

2.2. Delay-Free Loop Resolution

Before [1], a general procedure for incorporating multiple port nonlinearities, such as tubes and transistors, into WDF models, was unknown. When the parametric wave definition is applied to an existing Kirchhoff domain nonlinearity function of the form $\mathbf{i} = k(\mathbf{v})$, it is rarely possible to express the resulting wave-domain reflectance $\mathbf{b} = f(\mathbf{a})$ in closed form. Furthermore, the resulting nonlinear adaptor must be attached via multiple ports to an $\mathcal{R}$-type adaptor, forming delay-free loops between them. In [1,30], Werner et al. proposed applying the K-method [17, 26] to resolve the delay free loop. Here, we propose a *resolved wave definition*, which uses generalized wave variables [1, 31] to resolve the delay-free loop between the $\mathcal{R}$-type adaptor and wave domain nonlinearity.

Consider an $\mathcal{R}$-type adaptor connected by multiple ports to some nonlinearity described, in the Kirchhoff domain, by $\mathbf{i}_f = k(\mathbf{v}_f)$ and $f(\mathbf{a}_f) = \mathbf{b}_f$ in the wave domain. The $\mathcal{R}$-type adaptor is generally described by a linear scattering matrix,

$$\begin{bmatrix}\mathbf{b}_i \\ \mathbf{b}_c\end{bmatrix} = \begin{bmatrix}\mathbf{S}_{11} & \mathbf{S}_{12} \\ \mathbf{S}_{21} & \mathbf{S}_{22}\end{bmatrix}\begin{bmatrix}\mathbf{a}_i \\ \mathbf{a}_c\end{bmatrix} \tag{2}$$

Here, each wave vector has been partitioned into wave vectors of the ports attached to the the wave-domain nonlinearity $(\mathbf{a}_i, \mathbf{b}_i)$, and the wave vectors of the ports attached to adapted ports of other

linear adaptors ($\mathbf{a}_c, \mathbf{b}_c$). The nonlinearity is attached such that $\mathbf{a}_f = \mathbf{b}_i$ and $\mathbf{a}_i = \mathbf{b}_f$, like in Figure 1.

At each sample, this portion of the WDF model must compute $\mathbf{b}_c$, the reflected wave-vector from the root node. The vector $\mathbf{a}_c$ is known at all times from the tree below. To compute $\mathbf{b}_i$ or $\mathbf{b}_c$, $\mathbf{a}_i = \mathbf{b}_f$ must first be found, which in turn depends instantaneously on $\mathbf{a}_f = \mathbf{b}_i$ resulting in a delay-free loop with the nonlinearity.

The nonlinear delay-free loop has previously been resolved by minimizing

$$\|\hat{\mathbf{a}}_i - \mathbf{f}(\underbrace{\mathbf{S}_{11}\hat{\mathbf{a}}_i + \mathbf{S}_{12}\mathbf{a}_c}_{\hat{\mathbf{a}}_f})\| \tag{3}$$

over $\hat{\mathbf{a}}_i$ using Newton-Raphson iterations [12, 15]. This method can be costly, since iteration must be carried out for each sample.

The K method [17, 26] defines a state-space representation of the system plus an auxiliary nonlinear function that is resolved as such:

$$\begin{aligned}
\mathbf{x}_n &= \mathbf{A}\mathbf{x}_{n-1} + \mathbf{B}\mathbf{u}_n + \mathbf{C}\mathbf{i}_n \\
\mathbf{i} &= \mathbf{k}(\mathbf{v}) \\
\mathbf{v} &= \mathbf{D}\mathbf{x} + \mathbf{E}\mathbf{u} + \mathbf{F}\mathbf{i} \\
\mathbf{v}_n &= \mathbf{\Gamma}(\mathbf{p}_n) \\
\mathbf{i}_n &= \mathbf{f}(\mathbf{\Gamma}(\mathbf{p}_n)) \\
\text{where} & \\
\mathbf{p}_n &= \mathbf{D}\mathbf{x}_{n-1} + \mathbf{E}\mathbf{u}_n \\
\text{so that} & \\
0 &= \mathbf{p}_n + \mathbf{F}\mathbf{f}(\mathbf{v}_n) - \mathbf{v}_n
\end{aligned}$$

In the Nodal K -Method described in [26], the nonlinear resolving function $\mathbf{\Gamma}$ is precomputed using Newton homotopy and stored in a lookup table for real-time use through multidimensional linear interpolation based on `interpn` from the Octave distribution. All variables are in the Kirchhoff domain.

In [1], the nonlinear delay-free loop resolution of the K-method is applied to the WDF multiport nonlinearity problem. To allow for easier tabulation of the lookup, the $\mathcal{R}$-type adaptor is first transformed from the wave domain to the Kirchoff domain using a w-K converter. Then, the K-method is applied to the transformed $\mathcal{R}$-type adaptor.

Instead, consider that both the K-method and the parametric wave definition constitute a linear transformation or change of variables. Furthermore, the parametric wave definition is chosen as such because it is useful for preventing delay free loops within a WDF model's structure. Thus, if we can preserve model behavior, it is reasonable to choose an alternative linear transformation of Kirchhoff variables that prevents delay free loops.

We desire transformed wave variables $(\mathbf{a}_g, \mathbf{b}_g)$ for which the transformed nonlinear function $\mathbf{g}$ has no delay-free loop,

$$\mathbf{g}(\mathbf{a}_g) = \mathbf{b}_g. \tag{4}$$

as shown in (3). To break the dependency of $\mathbf{b}_i$ and $\mathbf{a}_i$, we must effectively set $\mathbf{S}_{11} = 0$. This corresponds to matching the port impedance of the nonlinearity with the instantaneous impedance, dependent on $\mathbf{a}_c$, of the nonlinearity itself.

These transformed wave-variables are related to the normal wave variables by the K-method:

$$\begin{bmatrix}\mathbf{a}_g \\ \mathbf{b}_g\end{bmatrix} = \begin{bmatrix}\mathbf{I} & \mathbf{K} \\ \mathbf{0} & \mathbf{I}\end{bmatrix}\begin{bmatrix}\mathbf{a}_f \\ \mathbf{b}_f\end{bmatrix}, \tag{5}$$

where $\mathbf{I}$ is the identity matrix with the same dimensions as $\mathbf{S}_{11}$ and $\mathbf{K} = \mathbf{S}_{11}$ for a system with no dynamics, as is the case here. Substituting in (1a) gives the relationship to the Kirchhoff domain.

$$\begin{bmatrix}\mathbf{a}_g \\ \mathbf{b}_g\end{bmatrix} = \begin{bmatrix}\mathbf{I} & \mathbf{S}_{11} \\ \mathbf{0} & \mathbf{I}\end{bmatrix}\begin{bmatrix}\mathbf{R}^{\rho-1} & \mathbf{R}^{\rho} \\ \mathbf{R}^{\rho-1} & -\mathbf{R}^{\rho}\end{bmatrix}\begin{bmatrix}\mathbf{v}_f \\ \mathbf{i}_f\end{bmatrix}, \tag{6}$$

This combined relationship defines the *resolved wave definition*:

$$\begin{aligned}
\mathbf{a}_g &= \mathbf{R}^{\rho-1}\left(\mathbf{I} - \mathbf{S}_{11}\right)\mathbf{v}_f + \mathbf{R}^{\rho}\left(\mathbf{I} + \mathbf{S}_{11}\right)\mathbf{i}_f \\
\mathbf{b}_g &= \mathbf{R}^{\rho-1}\mathbf{v}_f - \mathbf{R}^{\rho}\mathbf{i}_f \qquad (7a) \\
\mathbf{v}_f &= \frac{1}{2}\mathbf{R}^{1-\rho}\left(\left(\mathbf{S}_{11} + \mathbf{I}\right)\mathbf{b}_g + \mathbf{a}_g\right) \\
\mathbf{i}_f &= \frac{1}{2}\mathbf{R}^{-\rho}\left(\left(\mathbf{S}_{11} - \mathbf{I}\right)\mathbf{b}_g + \mathbf{a}_g\right). \qquad (7b)
\end{aligned}$$

While this definition is not convenient for the tabulation of a lookup table, it is useful for data transformation. A Kirchhoff domain nonlinearity described by $\mathbf{i}_f = k(\mathbf{v}_f)$ and transformed using the resolved wave definition can be inserted directly into the WDF model, as shown in Figure 2b. This WDF model is equivalent to the one described by Figure 2a, but contains no delay free loop, by definition of the transformed wave variables.

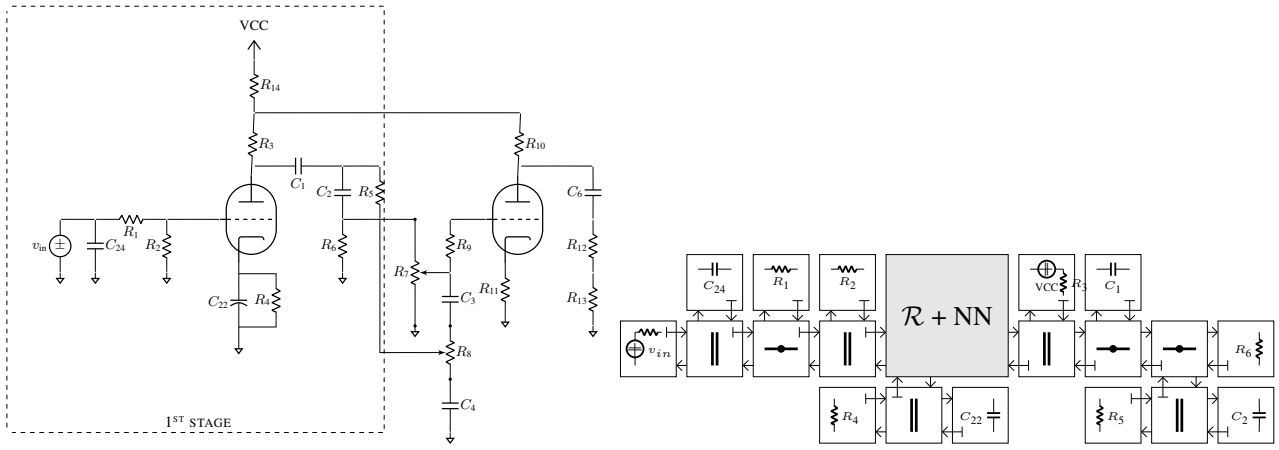
3. MACHINE LEARNING FOR TRIODES

Sets of voltages, $\mathbf{v}_f$, and currents, $\mathbf{i}_f$, at each port of a nonlinearity can be obtained through either direct measurement or circuit modeling, forming a Kirchhoff domain dataset for that nonlinearity. The resolved wave definition — derived in the previous section — allows us to transform said set of $\mathbf{v}_f$'s and $\mathbf{i}_f$'s creating a set of known solutions to the nonlinear function (4). As in the K-method [17], $\mathbf{g}(\cdot)$ represents the explicit solution resolving the delay-free loop.

$\mathbf{g}(\cdot)$ can be implemented using iteration representation in a table lookup that is optionally compressed by functional approximation (e.g., using linear interpolation or splines) [26, 32]. Both methods result in an approximation of the desired resolved nonlinearity. Function approximation of this kind is also a task well-suited for neural networks, which have been shown to be universal approximators, capable of replicating any continuous nonlinear function with arbitrary precision [33–35]. Due to the enormity of the lookup tables required for multiple nonseparable nonlinearities, neural net functional approximations present an interesting alternative implementation.

Training our neural network as a universal approximator of $\mathbf{g}(\cdot)$ is a regression task, where the $\mathbf{a}_g$ and $\mathbf{b}_g$ are the respective features and outcomes. Thus, the resolved wave definition becomes a feature extraction method applied to the Kirchhoff domain dataset. For a memoryless nonlinearity such as a triode or transistor, the function we need to approximate is necessarily memoryless; this suggests we should use a memoryless feedforward neural network [36]. Within feedforward networks, if our nonlinearity is represented by an amplitude mapping, this further suggests the use of a multilayer perceptron (MLP) network [24] rather than a convolution neural network (CNN), which is akin to FIR filtering [37].

Consider that encapsulating our neural network inside a WDF structure greatly reduces the complexity of the function the net must learn to approximate. When a circuits contains stateful elements, such as capacitors and inductors, modeling the system with



(a) *schematic* (b) *1st stage WDF model.*

Figure 3: *Preamplifier stage of the Fender Pro-Junior guitar amplifier. The modeling method was applied to the 1st stage of the circuit.*

a neural net would similarly require the use of stateful recurrent neural networks (RNN) [36], as in [21]. Instead, in our model, we have separated the linear portion in to WDF model that also maintains all the stateful portions of the circuit model. Our nonlinear behavior is independent from these states, reducing the complexity of the machine learning task.This allows our model to use small neural nets that can easily run in real time.

We modeled the first stage of a two-stage tube preamplifier from a Fender Pro-Junior guitar amplifier. A schematic of the circuit and its first stage is shown in Figure 3a and the accompanying WDF and neural net model is shown in Figure 3b . The preamplifier utilizes a 12AX7 triode vacuum tube.

3.1. Dataset Generation

To generate a Kirchhoff domain dataset for the 12AX7 triode, the circuit was modelled using LTspice. The SPICE model proposed by Dempwolf et al. [38] was used to simulate each 12AX7 triode.

To generate a dataset that accurately captured the behavior of the nonlinearity, we devised a 10-second test signal. The signal was composed of dry guitar samples and logarithmic sine sweeps with additive white noise. Each sine sweep is 3.0 seconds long, and has varying digital signal levels, from ± 0.25 to ± 1.0. The level of noise was equal to the level of the swept sine tones to ensure a wide range of frequencies were captured by the neural nets during training. A sampling rate of 96 kHz was used for both measurement and training.

In the LTspice model, full scale digital audio was scaled to 10 V_{pp}, which exceeds reasonable real-world voltage levels expected on the input of the amplifier. This was chosen to ensure that both the nonlinearity completely saturates and the Kirchhoff domain dataset of the tube extends beyond reasonable usage behavior in the circuit model. Voltage and current data were collected at the grid, anode and cathode. Finally, the dataset was transformed into the resolved wave domain using equation (7b).

3.2. Training

To train our model, we initially used a combination of error-to-signal ratio (ESR) loss and dc offset loss functions as suggested by Wright and Välimäki in [39]. Loss was computed individually along each dimension of the output vector, with the total loss being the sum of the losses in each dimension. Our networks had difficulty training due to a large variance in the loss, and we found it helpful to sample normalized each loss function. The normalized ESR loss used is given by

$$\mathcal{E}_{\text{ESR}} = \left(\frac{1}{N} \frac{\sum_{n=0}^{N-1} |y_n - \hat{y}_n|^2}{\sum_{n=0}^{N-1} |y_n|^2} \right)^{1/2}, \tag{8}$$

where N is the number of signal samples, y_n is the expected output value, and $\hat{y}_n$ is the output value predicted by the neural net. The normalized dc offset loss used is given by

$$\mathcal{E}_{\text{dc}} = \left(\frac{1}{N} \frac{|\frac{1}{N} \sum_{n=0}^{N-1} (y_n - \hat{y}_n)|^2}{\frac{1}{N} \sum_{n=0}^{N-1} |y_n|^2} \right)^{1/2}. \tag{9}$$

Combining normalized ESR and dc loss provided suitable initial results.

We found that including an additional normalized mean squared error (MSE) loss given by

$$\mathcal{E}_{\text{MSE}} = \frac{1}{N} \sum_{n=0}^{N-1} \frac{(y_n - \hat{y}_n)^2}{y_n^2} \tag{10}$$

in the loss function helped to further match the desired transconductance behavior of the neural network. See Section 4.2 for our transconductance analysis.

During training we utilized the Adam optimizer with values suggested by an initial Optuna study [40] for the learning rate α and decay rates β_1 and β_2. Our batch size was fixed at 256 samples. All training and verification of our neural networks was performed using Keras [41] and TensorFlow [42].

3.3. Hyperparameter Search

Two hyperparameter searches were performed using Optuna [40]. We first conducted an initial hyperparameter search to roughly determine adequate hyperparater values. Then a final search was done with a reduced set hyperparameters based on the results from

Hyperparameter	Searched Range	Spacing
# Hidden Layers	$1-8$	linear
Hidden Layer Size	$8-128$	log
α*	$10^{-6}-10^{-2}$	log
β_1*	$0.8-1.0$	log
β_2*	$0.9-1.0$	log
Activation	ReLU, tanh*, ELU,	N/A

Table 1: *The maximum and minimum bounds of each hyperparameter included in the Optuna search. Activation type was also included as a hyperparameter. * indicates the following parameter was not utilized in the final hyperparameter search*

the initial study. The hyperparameters included in our search and their respective minimum and maximum possible ranges are given in Table 1. Included in our initial search were three activation functions: rectified linear units (ReLU), tanh, and exponential linear units (ELU) [43]. Optuna worked to minimize the validation loss of 100 different neural network architectures over 250 epochs. The final validation loss was then stored for comparison.

To better understand the effects of layer width and depth, following our initial study, an approximate median of best performing neural nets' learning rate and decay rates were used in lieu of including these hyperparameters in our search. This corresponded to $\alpha = 10^{-4}$, $\beta_1 = 0.9$ and $\beta_2 = 0.999$. The tanh activation performed worse compared to the the ReLU and ELU activations, and it was removed from our final hyperparameter search.

As the goal of our model is real-time implementation, we compared each neural network's validation loss to an estimation of their runtime using single instruction, multiple data (SIMD) optimization. This estimation was derived by computing the number of multiplies and additions each neural network requires, and the computation time of the different activation functions relative to the ReLU activation.

A final Optuna study of 100 neural network architectures was carried out and in Figure 4 the final validation loss was compared to a runtime metric proportional to the size of the network. Of interest is that there exists a Pareto front, suggesting that there is a region where the loss is close to optimal for a given runtime.

Following this final hyperparameter search, three neural network architectures — given in Table 2 — were chosen from the left most points on the Pareto front that best minimized validation loss and maximized runtime efficiency.

Included in this table is the real-time factor (RTF) at a 96 kHz sampling rate for each neural net. The RTF is found by ratio of input duration and runtime-duration. For our analysis, a 10-second long input at 96 kHz was processed by each neural net on 2.4 GHz 8-Core Intel Core i9 CPU and the runtime of the net was used in the denominator of the RTF ratio. RTF analysis and audio validation of the neural nets was implemented in C++ using the RTNeural[1] library.

3.4. WDF Implementation

To test our method, we implemented a WDF model of the by the first stage of the Fender Pro-Junior Guitar amplifier. The WDF model was derived using the methods shown in [4]. The model diagram is shown in Figure 3b. The wave digital filter models were

[1] `https://github.com/jatinchowdhury18/RTNeural`

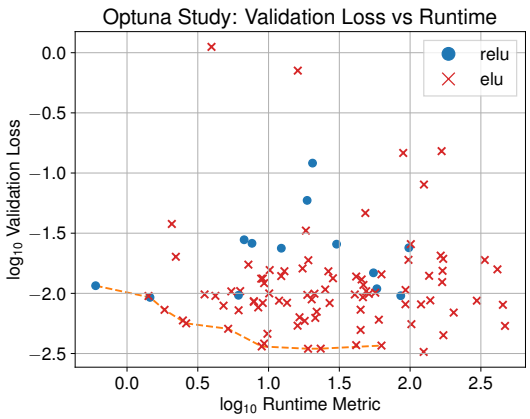


Figure 4: *Comparison of validation loss and runtime as generated by the Optuna study. The dashed line indicates the Pareto front.*

implemented in Python using the Differentiable Wave-Digital Filters library[2]. A custom combined $\mathcal{R}$-type [3] and the neural net adaptor was implemented as shown in Figure 2c.

The voltage across R_6 as was used as the output of the model. The circuit modeled only included the elements denoted in Figure 3a; any parasitic capacitances assiociated with the 12AX7 triode were left out for sake of simplicity. If desired, these could be added to the WDF model by reforming the $\mathcal{R}$-type and connecting capacitance adaptors of the relevant values.

4. MODEL ANALYSIS

Figure 5 compares the behavior of the three trained neural nets to the LTspice model which generated the training data. The top two plots of Figure 5 display the grid and plate reflected wave outputs as predicted by the neural net for a subset of our validation dataset. The results are compared to the values transformed from the LTspice generated Kirchhoff domain data. The signal shown is a dry guitar sample sampled at 96 kHz on the input of the circuit. These graphs demonstrate the neural nets suitably learned the input-output relationship of our dataset in the resolved wave domain.

The bottom plot of Figure 5 compares the output behavior of the neural net WDF model to the LTspice model for a 3 V_{pp} 220 Hz sine tone signal. Our models show minimal deviation from the LTspice model. Thus, the learned relationship translates well to the final output when the neural net is inserted into the wave digital

[2] `github.com/jatinchowdhury18/differentiable-wdfs`
[3] derived using `https://chowdsp.com/rsolver`

# Hidden Layers	Hidden Layer Size	Activation	Final $\mathcal{E}$	RTF @ 96 kHz
4	8	ELU	5.07e-4	39.2457x
2	32	ELU	1.77e-3	25.8181x
2	32	ReLU	5.48e-3	39.573x

Table 2: *Final neural nets used in analysis, their final loss and their real-time factor at 96 kHz.*

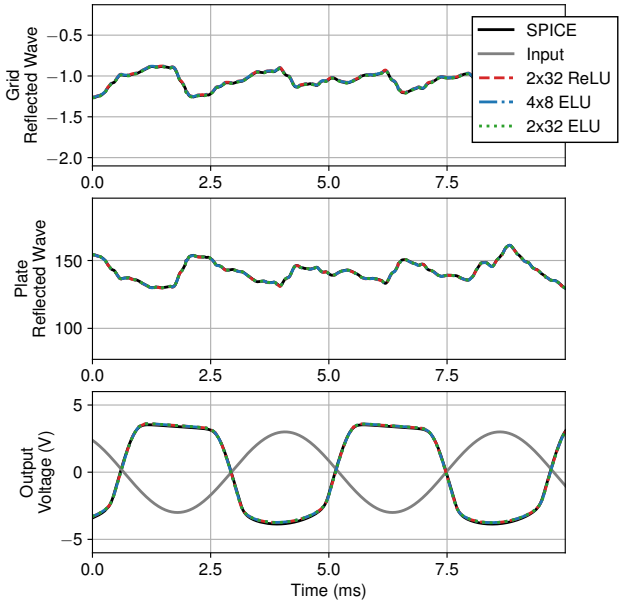


Figure 5: *The top two plots compare the components of the predicted output vector, $\mathbf{b}_f$ to the desired output as generated by LTspice for a guitar input signal. The bottom plot compares the output behavior of the 1st stage of the preamplifier circuit to the LTspice model output.*

filter. These plots experimentally confirm our proposed delay-free loop resolution method and supports using the resolved wave definition as a feature extraction tool when training a wave domain nonlinearity from a Kirchhoff domain dataset.

4.1. Activation Function Comparison

Figure 6 shows a detailed comparison of the different neural network architectures in response to the aforementioned 220 Hz sine tone signal. Note how different activation types exhibit different clipping profiles; the model with ReLU activations exhibits harsh transition points compared to the ELU networks that are relatively smooth. This behavior can be ascribed to piecewise linear approximators used in ReLU networks [35]. The ELU networks, on the other hand, do not exhibit these sharp edges as the "knee" of the activation function has been softened with an exponential. These time-domain artifacts are reflected in the frequency-domain where the ReLU network produces more high-frequency noise compared to the ELU networks. Frequency-domain analysis further suggests that the flatter ELU network (2x32) performs better for this particular function approximation.

4.2. Transconductance Analysis

To better understand the behavior of our trained networks, we experimentally examined their transconductance behavior in the Kirchhoff domain and compared to the LTspice model equations from Dempwolf et al. [38]. Through the resolved wave definition (7b), the behavior of our neural nets can be directly related to the Kirchhoff domain allowing us a better understanding of the neural nets behavior.

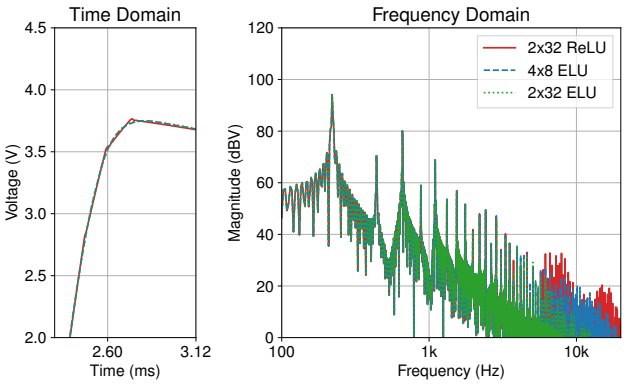


Figure 6: *Detailed comparison of the different neural net architectures in response to a 3 V_{pp} 220 Hz sine tone. Note how different architectures create different harmonic profiles in the high frequencies.*

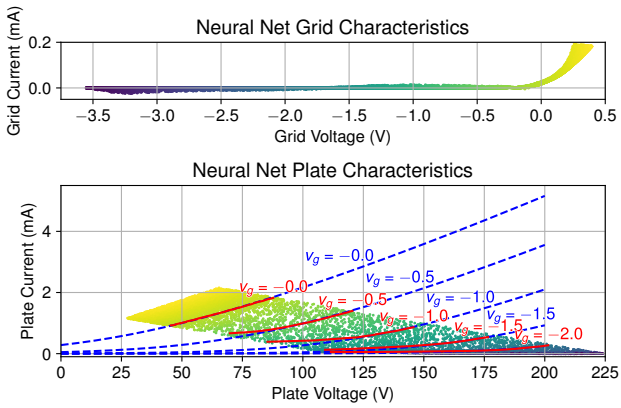


Figure 7: *Transconductance plot of the 4x8 ELU neural network model. Extracted curves of the neural net (solid red) closely adhere to the Dempwolf model (dashed blue)*

To extract the transconductance of our neural nets, the neural net's output was captured for a set of random input vectors with a Gaussian distribution over the expected range of the input. This range was bounded by the minimum and maximum found for each input-vector element in the training set. (7b) was then used to transform the input and output vectors into a Kirchhoff domain dataset, yielding a set of input voltages and output currents that characterize the neural net's transconductance.

The transconductance behavior for the 4x8 ELU neural net is shown in Figure 7. The experimentally extracted grid transconductance curves of the neural net (solid red), closely match the transconductance curves of the Dempwolf model (dashed blue). There is some deviation from the expected curves for lower voltages, possibly due to a lack of training data in this region. Overall, this demonstrates that our model has learned the Kirchhoff domain behavior of the 12AX7 triode through our feature extraction.

5. CONCLUSIONS

In this paper, we demonstrated the use of neural nets for simulating multiport nonlinearities in wave digital filter models. Previously, nonlinear elements have been modeled using iterative methods, K-method lookup tables, or functional approximations to lookup tables.

We introduced the resolved wave definition — a wave transformation that directly resolves the delay free loop created by a multiport nonlinearity — and used it to perform feature extraction from a Kirchhoff-domain dataset of a 12AX7 triode simulated in LTspice. Our results demonstrate that small neural nets are capable of learning the behavior of circuit nonlinearities in the resolved-wave domain and, due to complexity reduction in our grey-box modeling method, are easily capable of running in real time when implemented in C++.

Transconductance analysis of our neural net domain showed our neural nets closely replicated the tubes Kirchhoff domain behavior over an expected range of input. We found that hyperparameter selection in the network changed the distortion characteristic of the model independent of measured loss. For small dimension neural nets, we found the choice of activation function had a noticeable effect on aliasing artifacts, suggesting smooth activation functions are desirable. More detailed research is needed to understand how hyperparameters, such as the activation function, can effectively be chosen to minimize undesirable effects such as aliasing.

Because a nonlinearity's behavior is circuit invariant in the Kirchhoff domain, with a sufficient Kirchhoff domain dataset a nonlinearity could be appropriately transformed and placed into any WDF model. Even if the mathematical model of a nonlinear device is not well understood, it could still be trivially treated by the proposed method and used in a WDF model. In the future, the authors hope to collect real-world data on various nonlinearities and explore how this method can simulate them inside a WDF model.

The authors also hope to investigate how to include $\mathbf{S}_{11}$ as a learnable parameter, such that parameter control of the model is not limited by a static resolved wave definition. It is reasonable to expect that a sufficiently large neural net could extract the behavior of $\mathbf{S}_{11}$ in addition to modeling the wave domain behavior, given sufficient data.

6. ACKNOWLEDGMENTS

Many thanks to Jatin Chowdhury and Christopher Johann Clark for their fruitful discussions on wave digital filters and machine learning. Additional thanks to Kurt Werner for helping with the wave digital filter figures. Thanks to the anonymous reviewers for their helpful comments.

7. REFERENCES

[1] Kurt Werner, Vaibhav Nangia, Julius Smith, and Jonathan Abel, "Resolving wave digital filters with multiple/multiport nonlinearities," in *Proc. 18th Int. Conf. Digital Audio Effects (DAFx-15), Trondheim, Norway*, November 30 - December 3 2015.

[2] Kurt James Werner, Vaibhav Nangia, Alberto Bernardini, Julius O. Smith III, and Augusto Sarti, "An improved and generalized diode clipper model for wave digital filters," in *139th Convention of the Audio Engineering Society*, October 2015.

[3] T. Schwerdtfeger and A. Kummert, "Newton's method for modularity-preserving multidimensional wave digital filters," in *2015 IEEE 9th Int. Workshop on Multidimensional (nD) Systems (nDS)*, Sept 2015, pp. 1–6.

[4] Kurt James Werner, *Virtual Analog Modeling of Audio Circuitry Using Wave Digital Filters*, Ph.D. thesis, Stanford University, 2016.

[5] Kurt James Werner, W. Ross Dunkel, Maximilian Rest, Michael Jørgen Olsen, and Julius O. Smith III, "Wave digital filter modeling of circuits with operational amplifiers," in *24th European Signal Processing Conf. (EUSIPCO)*, 2016.

[6] Kurt James Werner, W. Ross Dunkel, and François G Germain, "A computational model of the hammond organ vibrato/chorus using wave digital filters," in *Proc. 19th Int. Conf. on Digital Audio Effects (DAFx-16), Brno, Czech*, 2016, pp. 5–9.

[7] W. Ross Dunkel, Maximilian Rest, Kurt James Werner, Michael Jørgen Olsen, and Julius O. Smith III, "The Fender Bassman 5F6-A family of preamplifier circuits—a wave digital filter case study," in *Proc. 19th Int. Conf. on Digital Audio Effects (DAFx-16), Brno, CZ*, Sept. 2016.

[8] K. J. Werner, A. Bernardini, J. O. Smith, and A. Sarti, "Modeling circuits with arbitrary topologies and active linear multiports using wave digital filters," *IEEE Trans. on Circuits and Systems I: Regular Papers*, vol. 65, no. 12, pp. 4233–4246, Dec. 2018.

[9] Matti Karjalainen and Jyri Pakarinen, "Wave digital simulation of a vacuum-tube amplifier," in *Proc. Int. Conf. on Acoust. Speech and Signal Processing (ICASSP)*, 2006.

[10] Jyri Pakarinen, Miika Tikander, and Matti Karjalainen, "Wave digital modeling of the output chain of a vacuum-tube amplifier," in *Proc. of the 12th Int. Conf. on Digital Audio Effects (DAFx-09)*, Como, Italy, September 2009.

[11] Jyri Pakarinen and Matti Karjalainen, "Enhanced wave digital triode model for real-time tube amplifier emulation," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. 18, no. 4, pp. 738–746, May 2010.

[12] Jingjie Zhang and Julius O. Smith III, "Real-time wave digital simulation of cascaded vacuum tube amplifiers using modified blockwise method," in *Proc. of the 21st Int. Conf. on Digital Audio Effects (DAFx-18)*, September 2018.

[13] Klaus Meerkötter, "On the passivity of wave digital networks," *IEEE Circuits and Systems Magazine*, 2018.

[14] Kurt James Werner, Julius O. Smith, and Jonathan S. Abel, "Wave digital filter adaptors for arbitrary topologies and multiport linear elements," in *Proc. of the 18th Int. Conf. on Digital Audio Effects (DAFx-15)*, Trondheim, Norway, November 30 - December 2015 2015.

[15] Michael Jørgen Olsen, Kurt James Werner, and Julius O. Smith III, "Resolving grouped nonlinearities in wave digital filters using iterative techniques," in *Proc. of the 19th Int. Conf. on Digital Audio Effects (DAFx-16)*, September 2016.

[16] Alberto Bernardini, Alessio E. Vergani, and Augusto Sarti, "Wave digital modeling of nonlinear 3-terminal devices for virtual analog applications," *Springer Science+Business Media*, January 2020.

[17] Gianpaolo Borin, Giovanni De Poli, and Davide Rocchesso, "Elimination of delay-free loops in discrete-time models of nonlinear acoustic systems," *IEEE Trans. On Speech And Audio Processing*, vol. 8, no. 5, 2000.

[18] David T. Yeh, *Digital Implementation of Musical Distortion Circuits by Analysis and Simulation*, Ph.D. thesis, CCRMA, Electrical Engineering Dept., Stanford University (CCRMA), June 2009, https://ccrma.stanford.edu/~dtyeh/.

[19] E.-P. Damskägg, Lauri Juvela, Etienne Thuillier, and Vesa Välimäki, "Deep learning for tube amplifier emulation," in *Proc. of the IEEE Int. Conf. on Acoust., Speech and Signal Processing (ICASSP)*, May 2019.

[20] Udo Zölzer Felix Eichas, Stephan Möller, "Block-oriented gray box modeling of guitar amplifiers," in *Proc. of the 20th Int. Conf. on Digital Audio Effects (DAFx-17),*, September 2017.

[21] Alec Wright, E.-P. Damskägg, and Vesa Välimäki, "Real-time black-box modelling with recurrent neural networks," in *Proc. of the 22nd Int. Conf. on Digital Audio Effects (DAFx-19)*, September 2019.

[22] Jatin Chowdhury, "A comparison of virtual analog modelling techniques for desktop and embedded implementations," arXiv, September 2020.

[23] Yuto Matsunaga, Naofumi Aoki, Yoshinori Dobash, and Tsuyoshi Yamamoto, "A digital modeling technique for distortion effect based on a machine learning approach," in *2018 Asia-Pacific Signal and Information Processing Association Annual Summit and Conf. (APSIPA ASC)*, 2018, pp. 1888–1892.

[24] Julian D. Parker, Fabián Esqueda, and André Bergner, "Modelling of nonlinear state-space systems using a deep neural network," in *Proc. of the 22nd Int. Conf. on Digital Audio Effects (DAFx-19)*, September 2019.

[25] Jatin Chowdhury and Christopher Johann Clarke, "Emulating diode circuits with differentiable wave digital filters," in *Proc. Sound and Music Computing Conf. (SMC)*, June 2022.

[26] David T. Yeh, "Automated physical modeling of nonlinear audio circuits for real-time audio effects—part ii: Bjt and vacuum tube examples," *IEEE Trans. on Audio, Speech, and Language Processing*, vol. 20, no. 4, pp. 1207–1216, 2012.

[27] G. De Sanctis, A. Sarti, and S. Tubaro, "Automatic methods for the physical modeling of sounds in the wave digital domain," in *Proc. Conf. Digital Audio Effects (DAFx-03), Queen Mary, University of London*, Sept. 2003.

[28] A. Fettweis and K. Meerkötter, "On adaptors for wave digital filters," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. 23, no. 6, 1975.

[29] Klaus Meerkötter and R. Scholz, "Digital simulation of nonlinear circuits by wave digital filter principals," in *Proc. of 1989 IEEE Int. Symp. on Circuits and Systems*, 1989.

[30] Kurt James Werner, Vaibhav Nangia, Julius O. Smith III, and Jonathan S. Abel, "A general and explicit formulation for wave digital filters with multiple/multiport nonlinearities and complicated topologies," in *Proc. IEEE Workshop on Applications of Signal Processing to Audio and Acoust., New Paltz, NY*, New York, October 2015, IEEE Press.

[31] S. S. Lawson, "On a generalization of the wave digital filter concept," *Int. J. Circuit Theory Appl.*, vol. 6, no. 2, pp. 107–120, 1978.

[32] David T. Yeh, Jonathan S. Abel, and Julius O. Smith III, "Automated physical modeling of nonlinear audio circuits for real-time audio effects—part i: Theoretical development," *IEEE Trans. on Audio, Speech, and Language Processing*, vol. 18, no. 4, pp. 728–737, 2010.

[33] G. Cybenko, "Approximation by superpositions of a sigmoidal function," *Mathematics of Control Signals, and Systems*, vol. 2, pp. 303–314, 1989.

[34] Kurt Hornik, "Approximation capabilities of multilayer feedforward networks," *Neural Networks*, vol. 4, no. 2, pp. 251–257, 1991.

[35] Changcun Huang, "Relu networks are universal approximators via piecewise linear or constant functions," *Neural Computation*, vol. 32, no. 11, pp. 2249–2278, 2020.

[36] Ian J. Goodfellow, Yoshua Bengio, and Aaron Courville, *Deep Learning*, MIT Press, Cambridge, MA, USA, 2016.

[37] Jesse Engel, Lamtharn Hantrakul, Chenjie Gu, and Adam Roberts, "Ddsp: Differential digital signal processing," in *Proc. of the Int. Conf. of Learning Representations (ICLR)*, Addis Ababa, Ethiopia, 2020.

[38] Kristjan Dempwolf, Martin Holters, and Udo Zolzer, "A triode model for guitar amplifier simulation with individual parameter fitting," in *131 Convention of the Audio Engineering Society*, October 2011.

[39] Alec Wright and Vesa Välimäki, "Perceptual loss function for neural modelling of audio systems," arXiv, 2019.

[40] Takuya Akiba, Shotaro Sano, Toshihiko Yanase, Takeru Ohta, and Masanori Koyama, "Optuna: A next-generation hyperparameter optimization framework," in *Proc. of the 25rd ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining*, 2019.

[41] François Chollet et al., "Keras," `https://keras.io`, 2015.

[42] Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Yangqing Jia, Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dandelion Mané, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng, "TensorFlow: Large-scale machine learning on heterogeneous systems," 2015, Software available from tensorflow.org.

[43] Djork-Arné Clevert, Thomas Unterthiner, and Sepp Hochreiter, "Fast and accurate deep network learning by exponential linear units (elus)," in *Proc. of the 2016 Int. Conf. on Learning Representations (ICLR)*, 2016.

[44] S. D'Angelo, J. Pakarinen, and V. Välimäki, "New family of wave-digital triode models," *Audio, Speech, and Language Processing, IEEE Trans. on*, vol. 21, no. 2, pp. 313–321, Feb 2013.

[45] Stefano D'Angelo, Leonardo Gabrielli, and Luca Turchet, "Fast approximation of the lambert w function for virtual analog modelling," in *Proc. of the 22nd Int. Conf. on Digital Audio Effects (DAFx-19)*, September 2019.