GUITAR TONE STACK MODELING WITH A NEURAL STATE-SPACE FILTER

Tantep Sinjanakhom

Neural DSP Technologies
Helsinki, Finland
tantep@neuraldsp.com

Eero-Pekka Damskägg

Complex Root Audio
Helsinki, Finland
eero-pekka@complexroot.com

Stylianos I. Mimilakis, Athanasios Gotsopoulos

Neural DSP Technologies
Helsinki, Finland
{stelios, tan}@neuraldsp.com

Vesa Välimäki

Acoustics Lab, Dept. of Information and Communications Engr.
Aalto University
Espoo, Finland
vesa.valimaki@aalto.fi

ABSTRACT

In this work, we present a data-driven approach to modeling tone stack circuits in guitar amplifiers and distortion pedals. To this aim, the proposed modeling approach uses a feedforward fully connected neural network to predict the parameters of a coupled-form state-space filter, ensuring the numerical stability of the resulting time-varying system. The neural network is conditioned on the tone controls of the target tone stack and is optimized jointly with the coupled-form state-space filter to match the target frequency response. To assess the proposed approach, we model three popular tone stack schematics with both matched-order and over-parameterized filters and conduct an objective comparison with well-established approaches that use cascaded biquad filters. Results from the conducted experiments demonstrate improved accuracy of the proposed modeling approach, especially in the case of over-parameterized state-space filters while guaranteeing numerical stability. Our method can be deployed, after training, in real-time audio processors.

1. INTRODUCTION

Virtual analog (VA) modeling refers to digitally emulating analog audio processing devices [1, 2]. In the last few years, machine learning techniques have drawn a lot of attention in the context of VA modeling tasks including distortion effects [3], guitar amplifiers [4, 5], phasing and flanging [6, 7], and dynamic range compressor [8–10] modeling. An appealing aspect of such techniques relies on their ability to model devices with intricate combinations of linear and non-linear characteristics without any prior knowledge of the device's circuitry. Such approaches, referred to as black-box modeling [11], are fully data-driven as they only require pairs of input and output signals. In contrast, traditional audio signal processing techniques, referred to as white-box modeling [12], require circuit analysis at a component level, a process that is repetitive and prone to human errors. The architectures of modern VA black-box models are commonly based on convolutional neural networks (CNNs) [3,4] and recurrent neural networks (RNNs) [5].

More recent studies have adopted the concept of differentiable digital signal processing (DDSP) [13] for audio effect modeling [14–16]. In the DDSP paradigm, audio signal processing stages are attached to machine learning models and optimized in an end-to-end setup. Kuznetsov et al. [17] demonstrated that digital finite impulse response (FIR) and infinite impulse response (IIR) filters can be made differentiable and trained to emulate characteristics of target filters. Related to filter learning, several methods have been proposed for modeling equalization (EQ) matching and learning EQ shaping devices.

Two closely related works from Nercessian [18] and Pepe et al. [19] proposed deep feedforward neural networks to estimate the parameters of biquad filters. Colonel et al. [20] suggested the use of a neural network to predict poles and zeros of cascaded IIR filters. In [18, 19] and [20], the networks were conditioned on the target magnitude response. In a similar manner, Bhattacharya et al. [21] proposed a method to predict the parameters of peak and shelving filters.

The previously mentioned approaches address static settings, that is, the outcome of the neural network cannot be adjusted by a set of controls that would allow for richer tone-shaping capabilities. To overcome this limitation, the work presented in [15] allows the use of controls by combining neural networks and the nodal discrete-Kirchoff (DK) method [12] in order to predict values of individual circuit components. Nonetheless, the nodal DK method requires an a priori knowledge of the circuit, responsible for the tone-shaping control, which in many cases is unknown. Another work that proposed to emulate analog audio effects with non-static settings is presented in [16] which is an extension from the previous study in [18]. This model utilizes a set of differentiable cascaded biquad filters. To ensure that the filter is stable in a time-varying system, the filter has to be implemented in a stable form, such as the state-variable filter or state-space structure [22], which is suitable for cascaded second-order recursive filters.

In this work, we propose a differentiable digital filter to model tone stacks. The structure of the proposed differentiable digital filter is based on a coupled-form state-space filter to promote numerical stability and to learn a single filter of arbitrary order. The entries of the filter are predicted by a neural network. The neural network is conditioned on tone controls, such as bass, middle, and

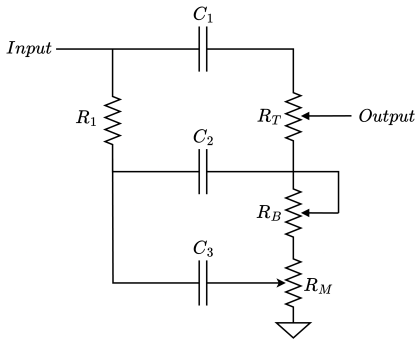

Figure 1: *The Fender '59 Bassman (FMV) tone stack schematic.*

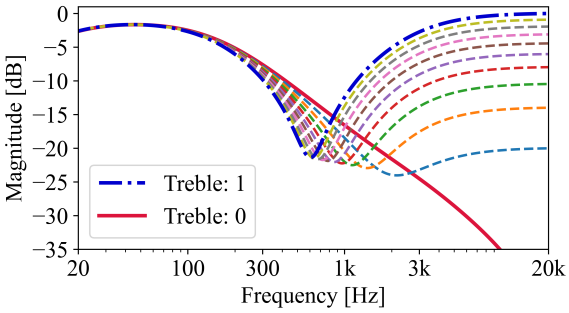

Figure 2: *Magnitude responses of the Fender '59 Bassman tone stack circuit, Bass = 0.5, Mid = 0, Treble = [0, 0.1, 0.2, ..., 1].*

treble. This leads to a trained neural network that can accurately replicate the sound of a given device in any control setting. Furthermore, the filter adopts a state-space representation where the system matrix is given in a coupled form. As such, the predicted filter is ensured to be stable in a time-varying system [23], unlike the direct-form biquad structures that have been previously proposed [18, 20]. This work focuses only on black-box methods due to the appealing feature of requiring no a priori knowledge of the audio circuit. As such, a comparison against white-box methods is beyond the scope of this work.

The remainder of this paper is structured as follows: Section 2 reviews the background on tone stack circuits. Section 3 presents the proposed method. Section 4 describes the experiment details. Section 5 shows the results. Finally, Section 6 concludes.

2. BACKGROUND

Tone stacks refer to the parametric EQ circuits designed for guitar effects and amplifiers [24]. They are typically passive filters consisting of capacitors and resistors [12, 25]. These tone circuits are generally positioned between the pre-amplifier and the power amplifier, serving as the main spectral shaping circuit in many amplifiers [12, 15]. A tone stack usually features three controls: *treble*, *middle*, and *bass*. Each adjusts the frequency response of its respective band. However, some tone circuits found in distortion pedals may have fewer controls [25, 26]. One of the most common tone stack topologies is used in the Fender, Marshall, and Vox amplifiers, often referred to as the FMV tone stack [24]. While tone stacks offer a broad range of EQ flexibility in an amplifier, their original purpose was to compensate for the attenuated frequencies in the output of guitar pickups [25, 27]. Figure 1 shows the schematic of the FMV from a Fender '59 Bassman amplifier.

Due to the passive nature of the circuits, interactions between controls can occur, where changing one control affects the behavior of the other frequency bands. An example of the interaction is shown in Fig. 2. Here we can observe that by decreasing the treble, the attenuation point in the middle frequency is affected and shifted upward. Accurately modeling these characteristics can be challenging, especially with a high-order tone stack.

The FMV tone stack has been modeled using various techniques in prior studies, including conventional white-box modeling [28] as well as differentiable models [14, 15]. These methods require a priori knowledge of the circuitry. In contrast, we propose a black-box model capable of generally emulating tone stacks without requiring detailed circuit comprehension, while also adjusting filter characteristics based on parametric inputs.

3. PROPOSED METHOD

The proposed method for modeling tone stacks accepts as input a vector containing control values in the interval $[0, 1]$, and produces as output the parameters of a coupled-form state-space filter. The mapping from the input controls to the state-space filter parameters is done by a feedforward fully connected neural network (FNN). During training, the error between the state-space filter frequency response and that of the circuit is minimized. An illustration of the proposed method is given in Fig. 3.

In more detail, the FNN input $\boldsymbol{x} \in \mathbb{R}^G$, where G is the number of circuit controls, consists of the values of the tone control positions, e.g., treble, middle, and bass. The FNN consists of N layers of affine transformations followed by leaky rectified linear unit (leakyReLU) activation functions [29]. The output of the FNN passes through a final linear affine transformation, yielding the output vector $\boldsymbol{y} \in \mathbb{R}^F$, which is transformed into the matrices of a discrete-time state-space filter. The output size F depends on the chosen order of the state-space filter.

To describe the transformation from the output $\boldsymbol{y}$ to the state-space filter parameters, let us consider a second-order filter as an example. In this case, $\boldsymbol{y} \in \mathbb{R}^7$ is transformed into the following state-space matrices:

$$\boldsymbol{A} = \begin{bmatrix} \mathcal{R}(p_0) & -\mathcal{I}(p_0) \\ \mathcal{I}(p_0) & \mathcal{R}(p_0) \end{bmatrix}, \boldsymbol{B} = \begin{bmatrix} y_2 \\ y_3 \end{bmatrix}, \boldsymbol{C}^{\mathrm{T}} = \begin{bmatrix} y_4 \\ y_5 \end{bmatrix}, \boldsymbol{D} = \begin{bmatrix} y_6 \end{bmatrix},$$

where $\mathcal{R}(\cdot)$ and $\mathcal{I}(\cdot)$ are operators that retain the real and imaginary parts of a complex number, respectively, and p_0 and p_0^* are the complex-conjugate poles of the state-space filter computed using vector elements of $\boldsymbol{y}$ as:

$$p_0 = f(y_0 + jy_1),$$

where $j = \sqrt{-1}$, the numerical subscripts denoting the indices of the vector elements, and

$$f(z) = \tanh(|z|)\frac{z}{\max(|z|,\ \epsilon)}, \tag{1}$$

where z is a complex number and $\epsilon = 10^{-7}$ is used to prevent division by zero. The transformation $f(z)$ ensures that the eigenvalues of the system matrix $\boldsymbol{A}$ lie within the unit circle, guaranteeing the stability of the system. This transformation is also used in [16, 20].

The mapping in Eq. (1) compresses the magnitudes of the complex-valued poles to be less than 1 while preserving the original angles. An illustration of this operation is shown in Fig. 4.

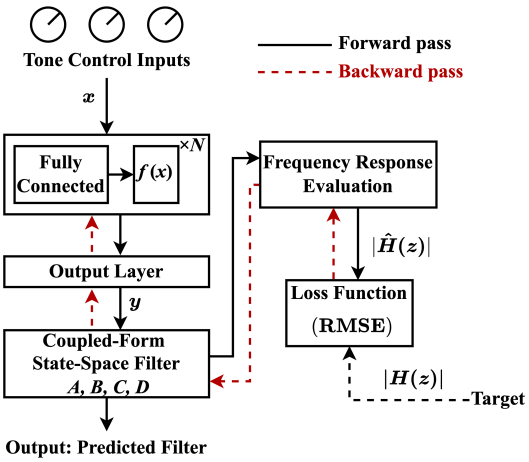

Figure 3: *Illustration of the proposed method. The left side of the figure illustrates the proposed method during inference. Computational blocks depicted on the right side are used during training.*

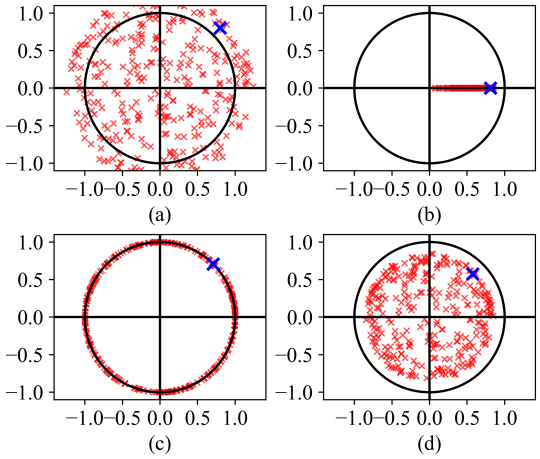

Figure 4: *Parameterization of predicted poles (Eq.* (1)*). (a) Unstable poles, (b) compressed magnitudes* $(\tanh(|p|))$*, (c) original angles* $\left(\frac{p}{\max(|p|,\,\epsilon)}\right)$ *and, (d) stabilized poles.*

Note the positions of the highlighted pole (blue cross). Additionally, this form of $\boldsymbol{A}$ is known as the *coupled form*. It has been shown to be asymptotically stable with time-varying coefficients [23], given that all the poles are inside the unit circle. In contrast, direct-form IIR filters are not guaranteed bounded-input bounded-output (BIBO) stable in this setting [23]. Moreover, the coupled form is less sensitive to errors from quantization [30, 31], which is preferable for real-time discrete systems. For higher-order filters, $\boldsymbol{A}$ is structured as a coupled-form block diagonal matrix.

After the transformation of $\boldsymbol{y}$, the discrete-time state-space filter is realized as:

$$\boldsymbol{s}[n+1] = \boldsymbol{A}\boldsymbol{s}[n] + \boldsymbol{B}\boldsymbol{u}[n], \tag{2}$$

$$\boldsymbol{v}[n] = \boldsymbol{C}\boldsymbol{s}[n] + \boldsymbol{D}\boldsymbol{u}[n], \tag{3}$$

where $\boldsymbol{s}[n]$, $\boldsymbol{u}[n]$, and $\boldsymbol{v}[n]$ denote the state, input, and output vectors, respectively, at time-step n. In practice, the process of filtering using Eq. (2) is not easily parallelizable due to its recursive nature. When training on GPUs, it is therefore preferable to evaluate the frequency response of the state-space filter directly. The frequency response is given by [32]:

$$\hat{H}(z)\Big|_{z=e^{j\omega_k}} = \boldsymbol{D} + \boldsymbol{C}(e^{j\omega_k}\boldsymbol{I} - \boldsymbol{A})^{-1}\boldsymbol{B}, \tag{4}$$

where ω_k corresponds to the frequency bin $k \in 0, 1, ..., N-1$ in the evaluation, and N is the total frequency bins. Evaluating the frequency response allows us to minimize the error between the magnitude responses $|\hat{H}(z)|$ of the model and the target $|H(z)|$.

4. EXPERIMENTS

To assess the performance of the proposed tone stack method, denoted as DiffSF, we compare it to two baseline models.

4.1. Baseline Models

For comparison, we reproduced two baseline models, namely the IIRNet developed by Colonel et al. [20] and a differentiable cascaded biquad filter model proposed by Nercessian et al. [16, 18], which we decide to call DiffBQ. Both models are based on an FNN predicting parameters for a set of differentiable cascade biquad filters. Originally, they were designed for EQ matching tasks where the input layer takes in the desired magnitude response and the output layer gives the poles, zeros, and gains.

In this study, the input layer now receives the parametric tone controls instead. The predicted poles of both baseline models are parameterized with Eq. (1) to stabilize the filter. In the IIRNet model, this parameterization is also applied to the zeros to obtain a minimum-phase filter. The architecture of the IIRNet model is an FNN with two hidden layers of 512 units and the activation function leakyReLU with $\alpha = 0.2$. The architecture of the second baseline model, the DiffBQ, is slightly different from the IIRNet. It has three hidden layers of 256 units, and the activation function is ReLU. The resulting poles, zeros, and gains from the neural network are mapped to the biquad filter coefficients and computed for the frequency response. If the filter is of odd order, the last filter section will have a real pole and a real zero.

4.2. Dataset Generation

In this work, three tone stack circuits found in the Electro-Harmonix Big Muff Pi [26], the Fender '59 Bassman [28], and the Marshall JCM 800 [12] are modeled. The Marshall JCM 800 and Fender '59 Bassman tone stacks are both third-order filters, whereas the Electro-Harmonix Big Muff Pi tone circuit is a second-order filter. The datasets are the collections of generated paired data obtained from LTSpice[1]. The circuits are simulated to obtain the frequency responses of various tone control settings. The simulation uses a unit impulse as an input to the tone circuit at each control setting. The output impulse response is then transformed into the frequency response by evaluating the frequency response at 1/12-octave band frequencies from 0.1 Hz to 24 kHz. Using fractional octave bands requires fewer sampling points for adequate resolution at low frequencies compared to linearly spaced frequency vectors.

The tone control combinations are sampled based on a parameter grid. In the training set, the position of each control ranges

[1] https://www.analog.com/en/resources/design-tools-and-calculators/ltspice-simulator.html

Table 1: *Validation results where the smallest errors on each row are highlighted with bold font. The table shows that the proposed methods DiffSF-2 and DiffSF-3 are outperforming in all cases*

Tone stack	Order	RMSE				MAE (dB)			
		IIRNet	**DiffSF-2**	DiffBQ	**DiffSF-3**	IIRNet	**DiffSF-2**	DiffBQ	**DiffSF-3**
BigMuff	Matched	0.011	0.009	0.011	0.009	0.32	0.22	0.32	0.22
	Over-param	0.002	**0.001**	0.002	**0.001**	0.04	**0.01**	0.05	**0.01**
Fender Bassman	Matched	0.050	0.050	0.050	0.050	0.91	0.90	0.91	0.90
	Over-param	0.004	**0.001**	0.004	**0.001**	0.09	**0.01**	0.08	0.02
Marshall JCM 800	Matched	0.199	0.264	0.030	0.030	2.71	2.69	0.38	0.36
	Over-param	0.002	**0.001**	0.002	**0.001**	0.04	**0.01**	0.04	**0.01**

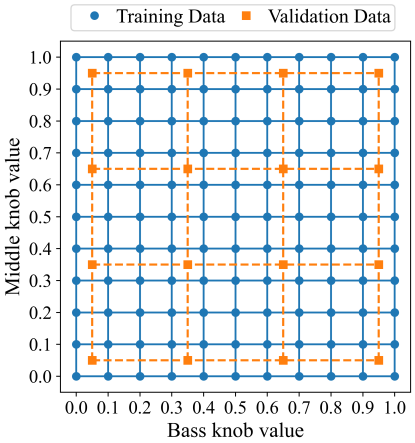

Figure 5: *Example sampling point combinations for the bass and middle controls.*

from 0 to 1, with a 0.1 increment. In the validation set, data is gathered at points between those of the training set, specifically [0.05, 0.35, 0.65, 0.95]. Figure 5 shows how the training and validation data are sampled in a two-knob case. The training sets are designed to cover a wide range of combinations within the control space, while the validation set is structured to include intermediate points. This ensures that the model can be effectively validated for its ability to interpolate across settings. From observations, the training loss and validation loss demonstrate similar convergence patterns, although the validation loss consistently remains higher.

4.3. Training

We compare a differentiable coupled-form state-space filter to cascaded direct-form biquad filters. We adhere to the neural network architectures used in the IIRNet and DiffBQ for a fair comparison. The proposed model to be compared with the IIRNet is named the DiffSF-2, indicating the utilization of an FNN with two hidden layers. Another model is the DiffSF-3, denoting three hidden layers, and it is compared with the DiffBQ. Since architectures are the same as those used in the baseline methods, the only differences are the filter structure and the size of the output layer.

Every model undergoes 50,000 iterations of training throughout the whole dataset using Adam optimizer [33]. Validation is conducted every 1,000 iterations to monitor the training process and mitigate the risk of overfitting in the model. The neural network is trained to minimize the magnitude-response error. The learning rate is scheduled to be reduced over time. The maximum learning rate is $3 \cdot 10^{-4}$ and the minimum is 10^{-5}. The training set is divided into mini-batches, each comprising half of the dataset. Each tone stack is trained using two models of different sizes, namely, the matched-order model and the over-parameterized model. The matched-order model aligns with the filter order of the original circuit. The over-parameterized model is designed with a higher order of $M_{\text{OP}} = 2M - 1$, where M is the filter order of the modeled circuit. This is expected to help increase flexibility and enhance the learnability of the neural network. The loss function used during the training is the root mean squared error (RMSE), which can be written as:

$$L_{\text{RMSE}} = \sqrt{\frac{1}{N}\sum_{k=0}^{N-1}\left(|H(e^{j\omega_k})| - |\hat{H}(e^{j\omega_k})|\right)^2}, \quad (5)$$

where $|H(e^{j\omega_k})|$ and $|\hat{H}(e^{j\omega_k})|$ are the target and predicted magnitude responses, respectively, N is the total number of frequency points, and $\omega_k = \pi f_k / f_s$ is the angular frequency. To further measure the error produced by each model in a more perceptually meaningful manner, an additional validation metric is the mean absolute error (MAE) in decibels (dB), formulated as

$$E_{\text{MAE}} = \frac{1}{N}\sum_{k=0}^{N-1} 20\left|\log(|H(e^{j\omega_k})|) - \log(|\hat{H}(e^{j\omega_k})|)\right|. \quad (6)$$

The error function E_{MAE} is used only to evaluate the results, and not during training. We prioritize optimizing the magnitude response over the phase response, as differences in magnitude are more perceptually significant. For the sake of compactness, results for phase matching are not discussed in this paper.

5. RESULTS

Table 1 shows the validation results in terms of RMSE loss and MAE in dB. For the Big Muff Pi tone stack, the validation losses of the baseline models and the proposed models are both relatively low. However, in the case that the tone control is set close to 0, the baseline models exhibit higher errors. Figure 6 illustrates the magnitude responses and the errors in dB of all Big Muff Pi models given input tone control at 0.05. The errors of the proposed methods (DiffSF-2, DiffSF-3) are small at all frequencies.

In terms of RMSE, the baseline models provide slightly smaller losses than the proposed models in the Marshall JCM 800 match-order cases, whereas in the Fender '59 Bassman matched-order case, all models perform equally well. Nevertheless, when the model is over-parameterized, the DiffSF-2 performed the best both in terms of RMSE and MAE dB. Examples of magnitude responses and errors for the Fender '59 Bassman models are shown in Fig. 7. The trained models can be observed to adequately interpolate the filter parameters for unseen tone control values. An edge case can be found with the bass control is rolled off. Figure 8 shows the prediction of the DiffSF-2 model for the Marshall

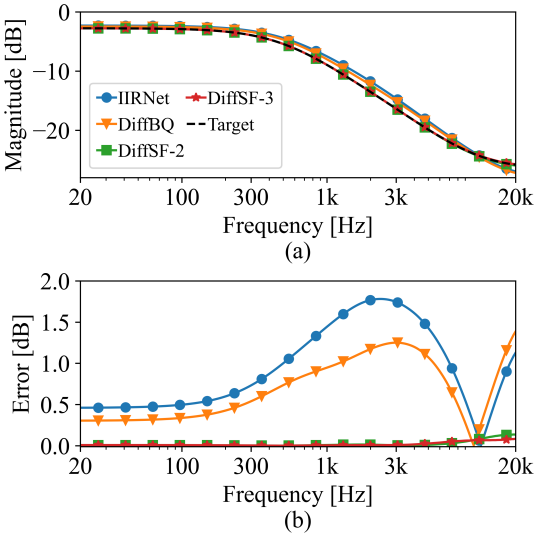

Figure 6: *(a) Magnitude responses and (b) differences in magnitude responses of over-parameterized ($M_{OP} = 3$) models for the Electro-Harmonix Big Muff Pi (Tone = 0.05).*

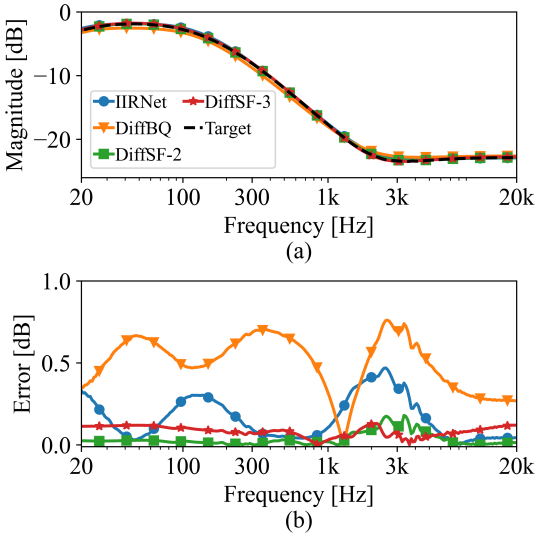

Figure 7: *(a) Magnitude responses and (b) differences in magnitude responses of over-parameterized ($M_{OP} = 5$) models for the Fender '59 Bassman. Bass = 0.95, Mid = 0.05, and Treble = 0.05.*

JCM 800 when each control is swept. In most cases, the errors are below 0.1 dB, except when the bass control is at 0.05, where errors noticeably increase. This suggests that the positions of zeros may be less accurate. Additionally, the energy at low-frequency bands is low in that setting, resulting in a small impact on the training loss. Empirically, over-parameterized models outperform matched-order counterparts across different filter structures due to increased flexibility, leading to higher accuracy. Thus, it can be safer to use the over-parameterized model when the exact order of the target tone circuit is unknown.

Many existing tone stack circuits can be realized as passive filters with only real-valued poles. While direct-form biquad and coupled-form state-space filters induce complex conjugate poles, if we want the poles to be real-valued, then the predicted filter parameters must have repeated poles, i.e., a pair of poles with 0 as their imaginary parts, rather than a single real pole. However, the state-space filter is not restricted to having complex conjugate zeros, meaning that the model can predict real-valued zeros anywhere. Therefore, the real-valued zero can be placed over the repeated poles to cancel the effect of the excess pole(s).

6. CONCLUSIONS

To summarize, a differentiable state-space filter is proposed for tone stack modeling. The model incorporates a feedforward fully connected neural network for mapping from parametric tone controls to the entries of the state-space matrices. The system matrix of the filter is parameterized to be stable in a linear time-varying system using the coupled form. The experiments focus on three different tone stacks used in popular guitar distortion pedals and amplifiers. The model directly minimizes the magnitude response error in the frequency domain. The presented filter structure is compared against direct-form cascaded biquad filters. Two variants of the proposed model outperform the baseline models in terms of all evaluation metrics. In addition to tone stack modeling, the differentiable state-space filter can be adapted for various audio applications, such as audio equalizer design.

7. ACKNOWLEDGMENTS

The authors would like to thank Dr. Thomas Sherson and Aleksi Peussa for their helpful advice and insightful discussions.

8. REFERENCES

[1] J. Pakarinen, V. Välimäki, F. Fontana, V. Lazzarini, and J. S Abel, "Recent advances in real-time musical effects, synthesis, and virtual analog models," *EURASIP J. Advances in Signal Process.*, vol. 2011, pp. 1–15, 2011.

[2] V. Välimäki, S. Bilbao, J. O. Smith, J. S. Abel, J. Pakarinen, and D. Berners, "Virtual analog effects," in *DAFX: Digital Audio Effects*, Udo Zölzer, Ed., pp. 473–522. Wiley, Chichester, UK, 2nd edition, 2011.

[3] M. A. M. Ramírez and J. D. Reiss, "Modeling nonlinear audio effects with end-to-end deep neural networks," in *Proc. IEEE Int. Conf. Acoustics, Speech and Signal Process.*, 2019, pp. 171–175.

[4] E.-P. Damskägg, L. Juvela, E. Thuillier, and V. Välimäki, "Deep learning for tube amplifier emulation," in *Proc. IEEE Int. Conf. Acoustics, Speech and Signal Process.*, 2019, pp. 471–475.

[5] A. Wright, E.-P. Damskägg, L. Juvela, and V. Välimäki, "Real-time guitar amplifier emulation with deep learning," *Applied Sciences*, vol. 10, no. 3, pp. 766, 2020.

[6] A. Wright and V. Välimäki, "Neural modeling of phaser and flanging effects," *J. Audio Eng. Soc.*, vol. 69, no. 7/8, pp. 517–529, 2021.

[7] A. Carson, C. Valentini-Botinhao, S. King, and S. Bilbao, "Differentiable grey-box modelling of phaser effects using frame-based spectral processing," in *Proc. Int. Conf. Digital Audio Effects*, 2023, pp. 219–226.

[8] C. J. Steinmetz and J. D. Reiss, "Efficient neural networks for real-time modeling of analog dynamic range compression," in *Proc. Audio Eng. Soc. Convention*, 2022.

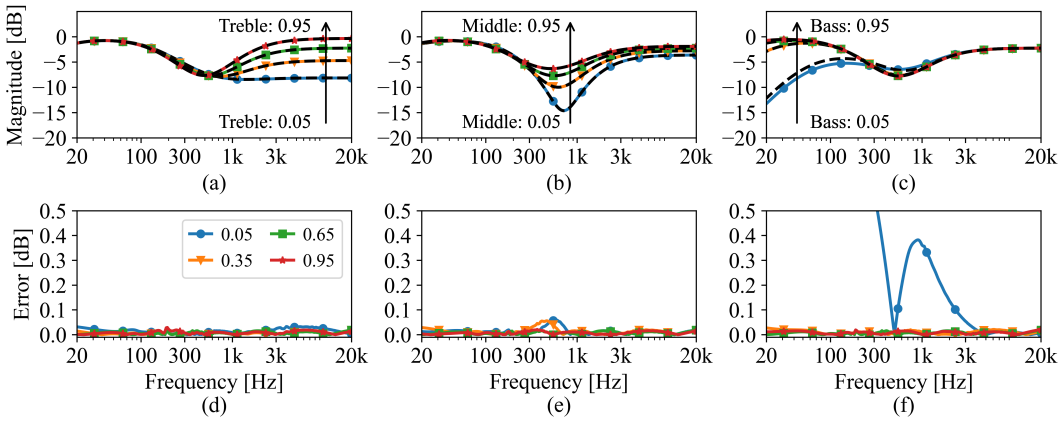

Figure 8: *Magnitude responses of the over-parameterized ($M_{OP} = 5$) DiffSF-2 for the Marshall JCM 800, illustrating (a) fixed Bass and Mid, (b) fixed Bass and Treble, and (c) fixed Mid and Treble settings. Corresponding difference plots are shown in (d), (e), and (f), respectively. Dashed black lines indicate the target magnitude responses.*

[9] R. Simionato and S. Fasciani, "Fully conditioned and low-latency black-box modeling of analog compression," in *Proc. Int. Conf. Digital Audio Effects*, 2023, pp. 288–295.

[10] S. H. Hawley, B. Colburn, and S. I. Mimilakis, "Profiling audio compressors with deep neural networks," in *Proc. Audio Eng. Soc. Convention*, 2019.

[11] F. Eichas and U. Zölzer, "Black-box modeling of distortion circuits with block-oriented models," in *Proc. Int. Conf. Digital Audio Effects*, 2016, pp. 5–9.

[12] D. T. Yeh, *Digital Implementation of Musical Distortion Circuits by Analysis and Simulation*, Ph.D. thesis, Stanford University, CA, 2009.

[13] J. Engel, L. Hantrakul, C. Gu, and A. Roberts, "DDSP: Differentiable digital signal processing," in *Proc. Int. Conf. Learning Representations*, 2019.

[14] F. Esqueda, B. Kuznetsov, and J. D. Parker, "Differentiable white-box virtual analog modeling," in *Proc. Int. Conf. Digital Audio Effects*, 2021, pp. 41–48.

[15] S. Miklánek, A. Wright, V. Välimäki, and J. Schimmel, "Neural grey-box guitar amplifier modelling with limited data," in *Proc. Int. Conf. Digital Audio Effects*, 2023, pp. 151–158.

[16] S. Nercessian, A. Sarroff, and K. J. Werner, "Lightweight and interpretable neural modeling of an audio distortion effect using hyperconditioned differentiable biquads," in *Proc. IEEE Int. Conf. Acoustics, Speech and Signal Process.*, 2021, pp. 890–894.

[17] B. Kuznetsov, J. D. Parker, and F. Esqueda, "Differentiable IIR filters for machine learning applications," in *Proc. Int. Conf. Digital Audio Effects*, 2020, pp. 297–303.

[18] S. Nercessian, "Neural parametric equalizer matching using differentiable biquads," in *Proc. Int. Conf. Digital Audio Effects*, 2020, pp. 265–272.

[19] G. Pepe, L. Gabrielli, S. Squartini, C. Tripodi, and N. Strozzi, "Deep optimization of parametric IIR filters for audio equalization," *IEEE/ACM Trans. Audio, Speech, and Language Process.*, vol. 30, pp. 1136–1149, 2022.

[20] J. T. Colonel, C. J. Steinmetz, M. Michelen, and J. D. Reiss, "Direct design of biquad filter cascades with deep learning by sampling random polynomials," in *Proc. IEEE Int. Conf. Acoustics, Speech and Signal Process.*, 2022, pp. 3104–3108.

[21] P. Bhattacharya, P. Nowak, and U. Zölzer, "Optimization of cascaded parametric peak and shelving filters with backpropagation algorithm," in *Proc. Int. Conf. Digital Audio Effects*, 2020, pp. 101–108.

[22] A. Wishnick, "Time-varying filters for musical applications," in *Proc. Int. Conf. Digital Audio Effects*, 2014, pp. 69–76.

[23] J. Laroche, "On the stability of time-varying recursive filters," *J. Audio Eng. Soc.*, vol. 55, no. 6, pp. 460–471, 2007.

[24] M. Blencowe, *Designing Valve Preamps for Guitar and Bass*, 2012.

[25] J. Lähdevaara, *The Science of Electric Guitars and Guitar Electronics*, 2012.

[26] B. Holmes and M. Van Walstijn, "Potentiometer law modelling and identification for application in physics-based virtual analogue circuits," in *Proc. Int. Conf. Digital Audio Effects*, 2019, pp. 332–339.

[27] R. C. D. de Paiva, J. Pakarinen, and V. Välimäki, "Acoustics and modeling of pickups," *J. Audio Eng. Soc.*, vol. 60, no. 10, pp. 768–782, 2012.

[28] D. T. Yeh and J. O. Smith, "Discretization of the '59 Fender Bassman tone stack," in *Proc. Int. Conf. Digital Audio Effects*, 2006, pp. 1–6.

[29] A. L. Maas, A. Y. Hannun, A. Y. Ng, et al., "Rectifier nonlinearities improve neural network acoustic models," in *Proc. Int. Conf. Machine Learning*, vol. 30, p. 3.

[30] B. Gold and C. M. Rader, "Effects of quantization noise in digital filters," in *Proc. Spring Joint Computer Conference*, 1966, pp. 213–219.

[31] C. Barnes, "Roundoff noise and overflow in normal digital filters," *IEEE Trans. Circuits Syst.*, vol. 26, no. 3, pp. 154–159, 1979.

[32] J. O. Smith, *Introduction to Digital Filters with Audio Applications*, W3K Publishing, http://www.w3k.org/books/, 2007.

[33] D. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proc. Int. Conf. Learning Representations*, San Diego, CA, USA, 2015.