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 July/August).
DOI: https://doi.org/10.17743/jaes.2021.0029

Neural Modeling of Phaser and Flanging Effects

ALEC WRIGHT, AND VESA VÄLIMÄKI, *AES Fellow*
(alec.wright@aalto.fi) (vesa.valimaki@aalto.fi)

Acoustics Lab, Dept. of Signal Processing and Acoustics, Aalto University, FI-02150 Espoo, Finland

This article further explores a previously proposed gray-box neural network approach to modeling LFO (low-frequency oscillator) modulated time-varying audio effects. The network inputs are both the unprocessed audio and LFO signal. This allows the LFO to be freely controlled after model training. This paper introduces an improved process for accurately measuring the frequency response of a time-varying system over time, which is used to annotate the neural network training data with the LFO of the effect being modeled. Accuracy is improved by using a frequency domain synthesized chirp signal and using shorter and more closely spaced chirps. A digital flanger effect is used to test the accuracy of the method and neural network models of two guitar effects pedals, a phaser and flanger, were created. The improvement in the system measurement method is reflected in the accuracy of the resulting models, which significantly outperform previously reported results. When modeling a phaser and flanger pedal, error-to-signal ratios of 0.2% and 0.3% were achieved, respectively. Previous work suggests errors of this size are often inaudible. The model architecture can run in real time on a modern computer while using relatively little processing power.

0 INTRODUCTION

The use of neural networks and deep learning has recently become popular in audio processing research [1–5]. The first neural models for audio effects processing imitated time-invariant linear and nonlinear filtering [6–8]. Deep neural models are the newest phase in the black-box modeling of audio devices, which had previously relied on nonlinear system identification methods, such as Volterra series [9–12] or the Wiener-Hammerstein model [13]. The present paper focuses on neural modeling of time-variant effects, which requires a different approach and poses a challenge during training, as the target behavior varies over time.

A system is time variant if its output depends on the time at the moment of input, as well as the input signal itself. For time-varying audio effects this is often achieved by modulating the parameters of the effect over time. The modulation signal varies for different effects, with some effects, such as chorus, often utilizing low-passed noise as a modulation signal, and with others, such as phasers and flangers, using a periodic modulation signal, such as a rectified sine or triangle wave [14]. A periodic modulation signal is commonly referred to as a Low-Frequency Oscillator (LFO).

This paper presents a method for modeling LFO-modulated audio effects. Models of two LFO-modulated analog effects pedals, a *phaser* and *flanger*, are created to demonstrate our approach. This work refines our previous results published at the DAFx-20 conference [15] by introducing an improved method for measuring the behavior of a time-varying system over time, which we show improves the reliability and accuracy of measuring the LFO signals of audio effects. Improvements to our method include using a frequency domain synthesized chirp signal, a new method for tracking spectral notches/peaks over time and improving the way the measured LFO data is extrapolated when creating the training dataset. To verify the accuracy of the LFO measurement method it is applied to a digital flanger algorithm, so the measured LFO signal can be easily compared to the true LFO signal. For training the deep neural network, the unprocessed audio and estimated LFO signal are given as inputs and the processed audio is used as a target.

This paper is organized as follows. Sec. 1 reviews the principles of phasing and flanging. Sec. 2 describes the

neural network used in this study. Sec. 3 details the method used to predict time-varying effects LFO signals. Sec. 4 reports experiments conducted to validate the LFO prediction method. Sec. 5 describes the creation of phaser and flanger pedal datasets for use in training neural network models. Sec. 6 presents the results, and Sec. 7 concludes.

1 TIME-VARYING AUDIO EFFECTS

This section briefly describes two time-varying audio effects, phasing and flanging, and how they are achieved.

1.1 Phasing

A phasing effect is characterized by a series of moving notches in the system's frequency response [16, 17]. The frequency of some or all of the nulls is varied over time by the LFO signal. The nulls can be introduced using notch filters but are more traditionally created using a series of cascaded low-order allpass filters [18–20]. Allpass filters have a flat magnitude response but introduce a nonlinear phase delay. For a first-order analog allpass filter, for example, the phase response decreases monotonically from 0° toward -180° as frequency is increased.

Cascading a series of N first-order allpass filters results in their phase responses being summed, producing an overall phase response that decreases monotonically from 0° toward $-N \times 180^\circ$ [18, 14]. The phasing effect is created by summing the original signal with the output of the cascaded first-order allpass filters. Frequencies where the phase response of the cascaded allpass filters is an odd-integer multiple of -180° will cancel out when summed with the original signal. The number of notches introduced depends on the number and order of allpass filters, such that for example two first-order allpasses produce one notch. The frequencies of the nulls can be varied by modulating the allpass filter parameters.

1.2 Flanging

Flanging is traditionally achieved by mixing a signal with a delayed version of itself [21, 16, 22, 17]. This is equivalent to applying a feedforward comb filter to the signal $x(t)$ [18]:

$$y(t) = b_0 x(t) + x(t - \tau), \tag{1}$$

where $y(t)$ is the output signal, b_0 is a linear gain factor that determines how pronounced the notches in the filter's frequency response are, and τ is the time delay. The transfer function is given by [23]:

$$H_{\text{ff}}(s) = b_0 + e^{-s\tau}. \tag{2}$$

where the latter term is the Laplace transform of time delay. In practical analog flangers the time delay is realized with a non-ideal delay implementation, such as using a bucket-brigade circuit [19, 24], but here it is assumed that an idealized continuous-time delay is used.

The resulting magnitude response can be seen in Fig. 1 for varying values of b_0 and τ. The flanging effect is created by using the LFO signal to modulate the length of the delay, which results in the notch frequencies moving up and down the spectrum. The notch locations are harmonically related, with the difference in frequency between consecutive notches being equal. A negative value of b_0 results in the first notch being located over the DC frequency, which causes the bass frequencies to be excessively filtered, as such a positive value of b_0 is generally used.

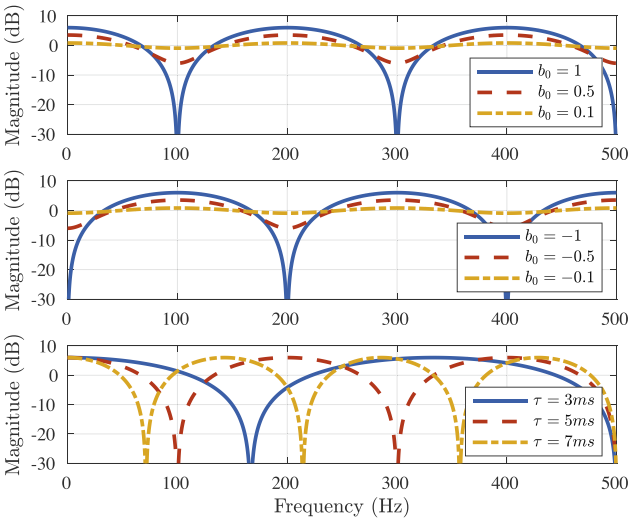

Fig. 1. Frequency response of feedforward comb filter for (top) positive b_0, (middle) negative b_0, and (bottom) different values of τ.

Feedback comb filters can also be used to create a flanging effect, with the following difference equation and transfer function [23]:

$$y(t) = x(t) + a y(t - \tau), \tag{3}$$

$$H_{\text{fb}}(s) = \frac{1}{1 - a e^{-s\tau}}, \tag{4}$$

where a is a gain factor, limited to $|a| < 1$ for filter stability, which determines the strength of the comb filtering effect. The magnitude response of the feedback comb filter can be seen in Fig. 2 for varying values of a and τ.

Combining the two filters results in a combined feedforward and feedback comb filter, with the time-domain description and transfer function, respectively:

$$y(t) = b_0 x(t) + x(t - \tau) + a y(t - \tau) \tag{5}$$

and

$$H_{\text{c}}(s) = \frac{b_0 + e^{-s\tau}}{1 - a e^{-s\tau}}. \tag{6}$$

If parameters are selected such that $b_0 = a$, then the filter becomes an allpass filter, but for flanging a positive b_0 and negative a are often chosen, so that the peaks of the feedback comb filter coincide with the notches of the feedforward comb filter. By adjusting the magnitudes of b_0 and a the sound of the comb filter can be altered to make the peaks and notches more pronounced without actually changing the locations of the peak and notches. This can be seen in the magnitude response shown in Fig. 3.

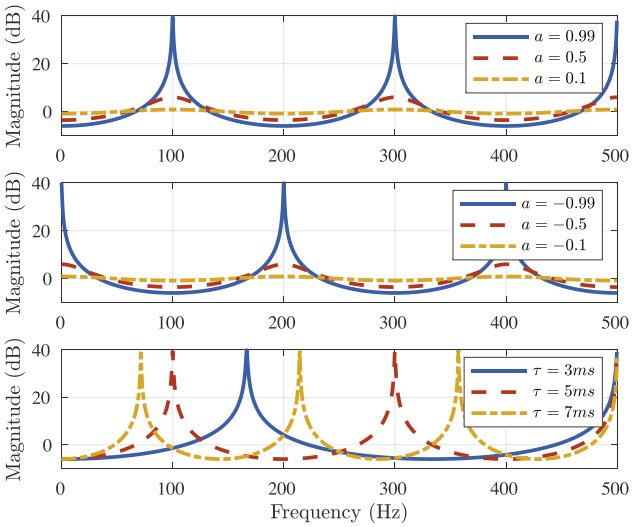

Fig. 2. Frequency response of feedback comb filter for (top) positive a, (middle) negative a, and (bottom) different values of τ.

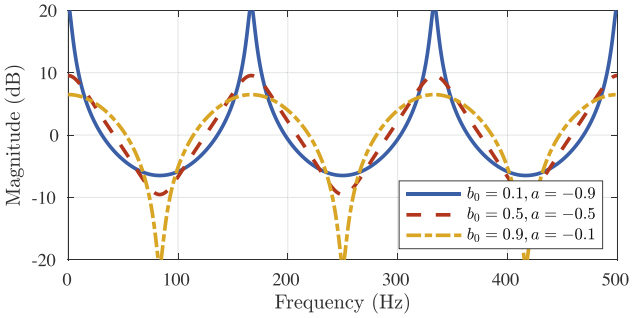

Fig. 3. Frequency response of combined comb filter with $\tau = 5$ ms.

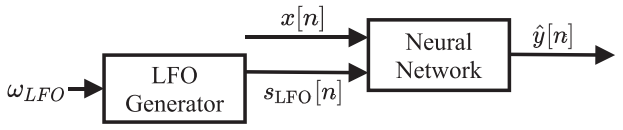

Fig. 4. Model structure, where x[n] is the input audio signal, $s_{\text{LFO}}[n]$ is the LFO signal, f_0 is the desired frequency of the LFO, and $\hat{y}[n]$ is the network's predicted output sample.

2 NEURAL NETWORK AUDIO EFFECT MODELING

The proposed model processes the input audio on a sample-by-sample basis, with the time-varying aspects of the effect modeled using an LFO signal as an additional input to the model. The general form of the trained model is depicted in Fig. 4. This is identical to the model used in our previous work [15].

By including the LFO signal as an input to the network, the required model complexity is reduced, in comparison to a model that would have to learn the shape and frequency of the LFO from the training data. Additionally the shape and frequency of the LFO can be controlled freely after the model is trained. One issue with this approach is that

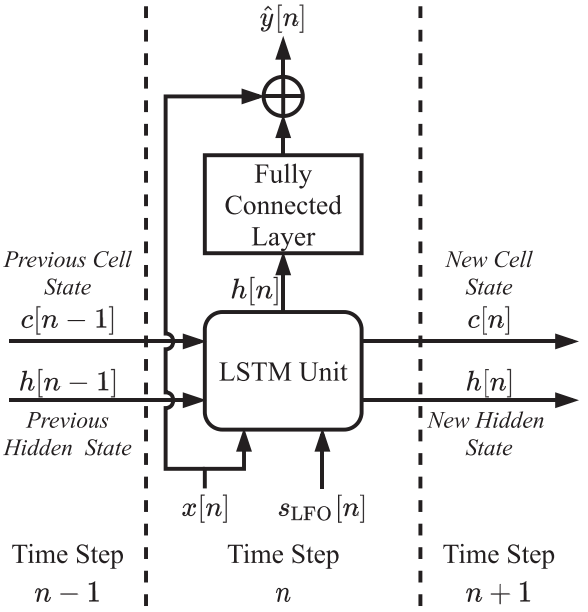

Fig. 5. RNN structure, where x[n] is the input audio signal, $S_{\text{LFO}}[n]$ is the input LFO signal, h and c are the LSTM Hidden and Cell states, respectively, and $\hat{y}[n]$ is the network's predicted output sample.

training the neural network requires input/output data from the target device, annotated with the LFO signal. Predicting the LFO signal accurately is not trivial and this is discussed further in Sec. 3.

2.1 Neural Network Model

In this study validation of this approach was carried out using the Recurrent Neural Network (RNN) model identical to that used in our previous paper [15], depicted in Fig. 5, consisting of a single Long Short-Term Memory (LSTM) [25] recurrent unit followed by a fully connected layer. The input to the model is a two-channel signal at audio rate,

$$u[n] = \left[x[n]\ ,\ s_{LFO}[n]\right], \tag{7}$$

where x is the input audio signal and s_{LFO} is the LFO signal.

The *Hidden State*, $h[n]$, and *Cell State*, $c[n]$, of the LSTM are updated at each time step as follows:

$$i[n] = \sigma(W_{ii}u[n] + b_{ii} + W_{hi}h[n-1] + b_{hi}), \tag{8}$$

$$f[n] = \sigma(W_{if}u[n] + b_{if} + W_{hf}h[n-1] + b_{hf}), \tag{9}$$

$$\tilde{c}[n] = \tanh(W_{ic}u[n] + b_{ic} + W_{hc}h[n-1] + b_{hc}), \tag{10}$$

$$o[n] = \sigma(W_{io}u[n] + b_{io} + W_{ho}h[n-1] + b_{ho}), \tag{11}$$

$$c[n] = f[n]c[n-1] + i[n]\tilde{c}[n], \tag{12}$$

$$h[n] = o[n]\tanh(c[n]), \tag{13}$$

where $i[n]$ is the input gate, $f[n]$ is the forget gate, $\tilde{c}[n]$ is the candidate cell state, $o[n]$ is the output gate, tanh(.) is the hyperbolic tangent function, and σ(.) is the logistic sigmoid function.

The fully connected layer performs an affine transformation of the LSTM hidden state and sums it with the input value to produce the predicted output for that timestep:

$$\hat{y}[n] = x[n] + W_{fc}h[n] + b_{fc}. \tag{14}$$

During training the network learns the weight matrices and bias vectors, denoted by W and b in the above equations. The LSTM *hidden size*, h, is a user-defined parameter that determines the size of the cell and hidden state vectors and thus the required size of the weight matrices and bias vectors. The model was implemented using PyTorch [26]. While this study only shows results achieved using the RNN model, the same method should work for other audio processing neural networks, such as the WaveNet-style model proposed in [27].

2.2 Training

The training process used was similar to that described in [15]. The Training Dataset was split into 1-s segments and processed in mini-batches. During the processing of each mini-batch 1,000 samples were initially processed to allow the LSTM states to initialize, then parameter updates were carried out every 1,024 samples. The networks were trained to minimize the Error-to-Signal Ratio (ESR) with first-order highpass pre-emphasis filtering and DC loss term:

$$\mathcal{E}_{\text{ESR}} = \frac{\sum_{n=0}^{N-1} |y_p[n] - \hat{y}_p[n]|^2}{\sum_{n=0}^{N-1} |y_p[n]|^2}, \tag{15}$$

$$\mathcal{E}_{\text{DC}} = \frac{|\frac{1}{N}\sum_{n=0}^{N-1}(y[n] - \hat{y}[n])|^2}{\frac{1}{N}\sum_{n=0}^{N-1} |y[n]|^2}, \tag{16}$$

$$\mathcal{E} = \mathcal{E}_{\text{ESR}} + \mathcal{E}_{\text{DC}}, \tag{17}$$

where y_p is the pre-emphasized target signal and $\hat{y}_p$ is the pre-emphasized neural network output. The first order pre-emphasis filter coefficient used was 0.85. The training was carried out using the Adam optimizer [28]. The segments were shuffled at the end of each epoch.

3 ACCURATE LOW-FREQUENCY OSCILLATOR MEASUREMENT

For the proposed neural network model to work an accurate measurement of the LFO signal that is modulating the time-varying effect needs to be obtained. While in the case of an analog effects pedal it is possible to directly measure the LFO signal from the circuit, the method used in this article infers the LFO purely based on measurements from the output of the pedal. This increases the complexity of determining the LFO signal considerably; however it allows the method to be applied to other phasing or flanging effects pedals with minimal effort and additionally creates the possibility that non-technical users could create datasets for modeling any pedals that they own.

This article proposes a number of improvements for the LFO signal measurement technique proposed in [15] that are described in this section. An overview of the complete process for measuring the LFO of a device is shown in Fig. 6.

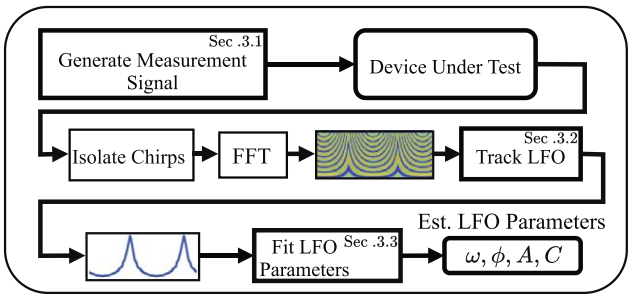

Fig. 6. Process for estimating LFO frequency, ω, phase, ϕ, amplitude, A, and offset, C. Numbers in the top right hand corner of boxes refer to the relevant subsections.

3.1 Measurement Signal

The frequency response of a Linear Time-Invariant (LTI) system is usually found by analyzing its response to a measurement signal, such as a sine-sweep or Maximum Length Sequence [29]. Generally a longer measurement signal is used to improve the Signal-to-Noise Ratio (SNR) achieved during the measurement; however when measuring a time-variant system a shorter measurement signal is advantageous, as it allows for more measurements to be taken over a short period of time [14].

In our previous work [15] we used the impulse response of cascaded first-order allpass filters as a measurement signal, as proposed in [14]. The transfer function of the digital first-order allpass filter is

$$A_{ap}(z) = \frac{a_1 + z^{-1}}{1 + a_1 z^{-1}}. \tag{18}$$

By definition it has unity gain for all frequencies, a phase response of

$$\phi_{ap}(\omega) = -\omega + 2\arctan\left(\frac{a_1 \sin\omega}{1 + a_1\cos\omega}\right), \tag{19}$$

and a group delay of

$$\tau_{ap}(\omega) = \frac{a_1^2 - 1}{a_1^2 + 2a_1\cos(\omega) + 1}. \tag{20}$$

In our previous work the chirp signal was generated using $N_{ap} = 64$ cascaded allpass filters, with $a_1 = -0.9$, resulting in an approximately 27-ms long measurement signal. It should be noted that for a negative a_1 the group delay τ_{ap} decreases with frequency, so the chirp frequency in this case will start at the Nyquist frequency and decrease with time. To generate a chirp signal that takes T_c s to traverse from w_{st} to w_{en}, the required number of allpass filters, N_{ap}, can be calculated as follows:

$$N_{ap} = \frac{T_c}{\tau_{ap}(\omega_{en}) - \tau_{ap}(\omega_{st})}. \tag{21}$$

An alternate method for generating chirp signals is to start by defining the group delay of the desired chirp [30], $\tau(\omega, \eta)$, where η is a factor controlling the rate of change in chirp frequency. The chirp can be synthesized in the

frequency domain, as a complex signal with unit magnitude, and phase angle calculated based on the group delay:

$$\phi(\omega) = -\int_0^{\omega} \tau(\omega, \eta) d\omega. \tag{22}$$

The time domain chirp signal can then be created by taking the Inverse Discrete Fourier Transform (IDFT).

This method was used to generate two chirp signals, one with a linear group delay function and one with an exponential group delay function. The linear group delay function is defined as

$$\tau_{lin}(\omega) = \eta\omega, \tag{23}$$

and has the corresponding phase response of

$$\phi_{lin}(\omega) = -\frac{\eta\omega^2}{2}. \tag{24}$$

The exponential group delay function is defined as

$$\tau_{exp}(\omega) = \eta \ln(\omega), \tag{25}$$

and has the corresponding phase response of

$$\phi_{exp}(\omega) = -\eta\omega(\ln(\omega) - 1). \tag{26}$$

To create a chirp of length T_c s that starts at a frequency of ω_{st} and finishes at a frequency ω_{en} the parameter η can be selected such that

$$\tau(\omega_{en}) - \tau(\omega_{st}) = T_c, \tag{27}$$

for whichever group delay function, $\tau_{lin}(\omega)$ or $\tau_{exp}(\omega)$, is being used to synthesize the chirp signal.

3.2 LFO Tracking

The processed measurement signal is analyzed by isolating the individual chirps and then taking the Discrete Fourier Transform (DFT), which is usually implemented using the Fast Fourier Transform (FFT) algorithm, as summarized in Fig. 6. This creates a spectrogram-like image that shows the magnitude response of the system over the chirp frequencies at the time of each chirp. An example of this can be seen in Fig. 8. The LFO behavior can be inferred by tracking the movement of a frequency notch over time.

To initiate the LFO tracking, the location of a notch at one of the chirp times is selected manually. The notch at the adjacent chirp times can then be tracked using the initial LFO frequency and notch frequency range estimates to provide a range of possible values the notch could appear at:

$$\omega_{n+1} = \omega_n \pm p\ F_c\ \alpha_i\ \omega_{LFOi}, \tag{28}$$

where p is a factor chosen by the user (set to $p = 2$ during this work), ω is the frequency of the notch, F_c is the chirp rate, α_i is the difference between the maximum and minimum frequencies of the notch being tracked, and ω_{LFOi} is the initial estimate of the LFO frequency. The notch frequency at each subsequent chirp is simply selected as the frequency bin with the lowest magnitude within the range of frequencies provided by the above equation.

The frequency of the notch over the duration of each chirp is now known, as well as the start time of each chirp in the measurement signal. As each chirp is spread over time, the group delay of the chirp signal can be used to more precisely determine the time at the which the measured notch frequency was put into the system being measured. As the notch frequency varies over time, this means the LFO is being sampled at a non-uniform sampling rate.

The method of notch tracking described in this section can also be easily modified to track a peak instead, by simply searching the frequency spectrum for a maximum instead of a minimum. This is useful in cases where the peaks are more pronounced than the notches.

3.3 LFO Fitting

Once the notch position over time has been measured an LFO signal can be fitted to the data, as indicated in Fig. 6. To fit the LFO function, as proposed in our previous work, we use a nonlinear least-squares solver. For example, where the LFO is a rectified sine-wave, the solver minimizes the following equation:

$$f(\omega, \phi, A, C) = A|\sin[t(\omega/2) + (\phi/2)]| + C - s_{LFO}(t), \tag{29}$$

where ω and ϕ are the predicted LFO frequency and phase, A and C are the LFO magnitude and offset, s_{LFO} is the measured LFO signal, and t is the sampling time of the LFO. The nonlinear least-squares solver requires initial estimates for the parameters. The initial frequency estimate can be chosen by taking the DFT of s_{LFO} and finding the frequency bin with the highest energy. Initial estimates for A and C are chosen by looking at the maximum and minimum values observed in the measured LFO signal, and finally the initial phase estimate is set to 0, as it was found that this does not affect the outcome of the LFO fitting.

Once an estimate for the LFO parameters is obtained the LFO can then be extrapolated beyond the region where the LFO was measured. In our previous work we fit the LFO to a single segment of the measurement signal and then extrapolated out to the surrounding data [15]. In this article we found that a more accurate estimation of the LFO parameters could be achieved by fitting the LFO to two segments of the LFO measurement signal that are separated by the guitar audio that will be used to train the neural network. As the LFO is effectively being fit over the duration of both measurement signals and the gap in between them, small frequencies error will result in a significant phase misalignment and as such this results in a more accurate frequency measurement. Fig. 7 shows the previous method of fitting an LFO to each segment of the measurement signal in comparison to the proposed method of fitting the LFO jointly to two consecutive segments of the measurement signal.

4 LFO MEASUREMENT TESTING

As it is not straightforward to verify the accuracy of an LFO measured from an analog device, a digital implementation of the flanger audio effect described in Sec. 1.2

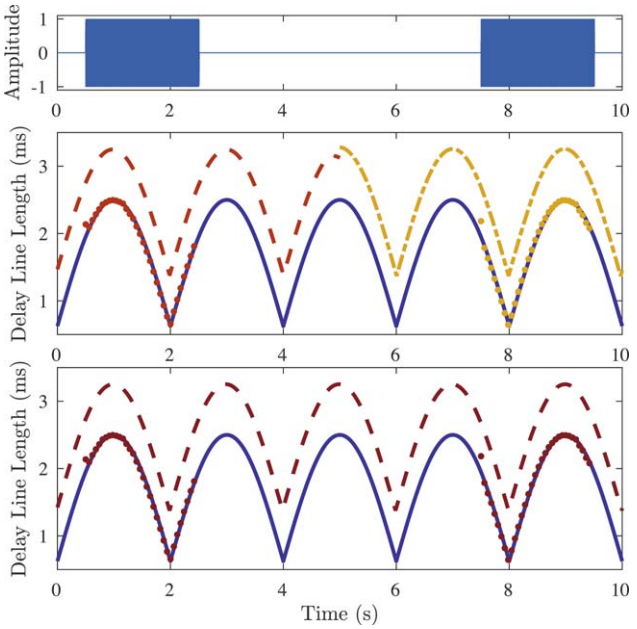

Fig. 7. Measurement signal (top) and actual LFO signal (solid line), measured LFO position (circles), and extrapolated LFO signal (offset for clarity), where one LFO signal is fit to each segment of measured LFO (middle), or one LFO signal is fit to both segments of measured LFO concurrently (bottom). Phase misalignment in middle plot is exaggerated for illustrative purposes.

was used to test the LFO measurement method, with the following transfer function:

$$H(z) = \frac{b_0 + z^{-M}}{1 - az^{-M}}, \tag{30}$$

where M is the length of the delay line in samples. As the digital flanger requires a continuously varying delay-line length, a fractional delay is required [31], which was implemented using cubic interpolation for this article. The LFO signal in this case controls the length of the delay line, with the peaks and troughs of the system moving with the delay-line length. A chirp signal can be used to determine the frequency location of a peak or trough at a certain time and this can be used to infer what the delay-line length was at that time.

For the feedforward comb filter with a positive value of b_0, the notch number n_{notch} will appear at frequency:

$$f_{\text{notch}} = \frac{2n_{\text{notch}} - 1}{2\tau}, \tag{31}$$

where τ is the delay-line length in seconds, and for the feedback comb filter with a negative value of a, the peak number n_{peak} will appear at frequency:

$$f_{\text{peak}} = \frac{n_{\text{peak}}}{\tau}, \tag{32}$$

where the peak corresponding to $n_{\text{peak}} = 1$ is the first peak appearing above the DC frequency. If the frequency of a notch or peak is measured the above equations can easily be rearranged to determine the corresponding delay-line length, τ. This can then be compared directly to the real LFO value at that time.

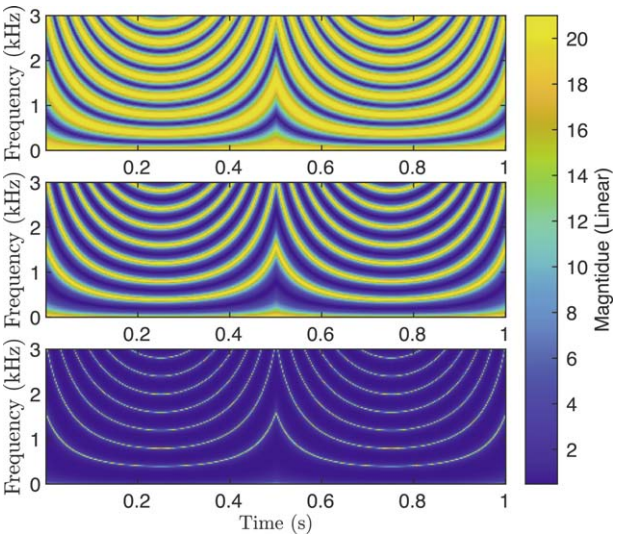

Fig. 8. Frequency response over time for digital flanger effect, with (top) $b_0 = 0.95$ and $a = 0.05$, (middle) $b_0 = 0.5$ and $a = 0.5$, and (bottom) $b_0 = 0.05$ and $a = 0.95$.

The LFO measurement algorithm was tested on two settings of the combined comb filter, with the parameters for the feedforward and feedback gain, b_0 and a being varied. In all cases the delay-line length was varied by a rectified sine wave, from 0.6250 ms to 2.5 ms. Additionally white noise at −60 dBFS was added to the measurement signal to try and emulate the measurement conditions of an analog device. Fig. 8 shows a spectrogram-like image of the magnitude response of the flanger effect over time. It can be seen that when a is close to 1, the notches are wide, and the peaks are very sharp. As such in this configuration it should be easier to track the LFO via a peak as opposed to a notch, as the center of the notches are less pronounced than the center of the peaks. When b_0 is close to 1 the opposite is true, so in this case the notch should be tracked instead.

For each stage of the LFO measurement and prediction process this article has proposed an improvement over the existing method [15], and these two methods are directly compared in this section.

4.1 Measurement Signal

Three different chirp signals were tested: the previously proposed impulse response of cascaded first-order allpass filters, s_{ap}, the linearly swept chirp, s_{lin}, and the exponentially swept chirp, s_{exp}. Additionally the effect of chirp rate on the measurement accuracy was also tested. The chirp spacing refers to the time elapsed between the start of successive chirps. The chirp rate is thus the reciprocal of the chirp spacing. Chirp spacings of 10, 20, and 30 ms were trialed, corresponding to chirp rates of 100, 50, and 33.3 chirps per second, respectively. The chirp signal lengths were adjusted such that successive chirps do not overlap and the overall measurement signal used was 5.0 s long.

Table 1 shows the results, with all cases showing that the cascaded allpass chirp performs worst in terms of mean error when compared to the linearly and exponentially swept

Table 1. Accuracy of the LFO measurement of the digital flanger effect for the three tested chirp signals; left column indicates flanger parameters and which notch or peak was tracked to achieve the reported result.

	Chirp Rate (/s)	Mean Error (%) s_{ap}	s_{exp}	s_{lin}
$b_0 = 0.95$	100	1.50	1.03	**0.51**
$a = 0.05$	50	1.41	**0.19**	0.25
Notch 1	33.3	2.04	**0.15**	0.26
$b_0 = 0.05$	100	4.99	5.64	**1.85**
$a = 0.95$	50	3.02	2.62	**1.28**
Peak 1	33.3	2.63	1.97	**1.52**

Table 2. Accuracy of the LFO measurement of the digital flanger effect for the previously proposed peak picking algorithm and the newly proposed candidate frequency range method; left column indicates flanger parameters and which notch or peak was tracked to achieve the reported result.

	Chirp Rate (/s)	Mean Error (%) Method of [15]	Proposed
$b_0 = 0.95$	100	0.51	0.51
$a = 0.05$	50	0.26	0.26
Notch 1	33.3	0.26	0.26
$b_0 = 0.05$	100	78.3	**1.79**
$a = 0.95$	50	12.0	**1.32**
Peak 1	33.3	25.9	**1.66**

chirps. The linearly swept chirp seems to perform best out of the tested signals. The results also show that even with 10 ms–long chirp signals, an accurate measurement of the LFO signal can be achieved.

4.2 LFO Tracking

Two different methods for tracking the notch location over successive chirps were tested: the new method described in Sec. 3.2 and the previously proposed method [15], which used a peak picking algorithm. The two methods were trialed on the measurement signal consisting of linearly swept chirps, with a chirp rate of 100, 50, or 33.3 chirps per second. The results are shown in Table 2, where it can be seen that in the case with a small feedback gain, a, the tracking algorithms achieve the same error, indicating that they have both tracked the same peak or notch. However when the feedback gain is larger, the proposed tracking algorithm performs considerably better.

4.3 LFO Fitting

The LFO function is fitted to the measured LFO using a nonlinear least-squares solver. The existing method of fitting the LFO to one segment of a measurement signal and the proposed method of fitting an LFO jointly to two separated measurement signals were compared. In both cases a measurement signal totaling 20 s in length was used, but in the proposed method 60 s of audio was inserted in the middle of the measurement signal. The test was conducted with a chirp rate of 100, 50, or 33.3 chirps per second. The results in Table 3 show that while both methods achieve a very small error the proposed method performs better in all the tests conducted. It is noted that while the error is very small in all cases shown here, we found that when measuring analog devices the errors were generally much higher and as such the improvement in frequency estimation should be more significant when applied to analog effects pedals.

Table 3. Performance of the existing and proposed LFO extrapolation method, in terms of error in the predicted LFO frequency.

	Chirp Rate (/s)	Freq Error ($\times 10^{-6}$) Hz Method of [15]	Proposed
$b_0 = 0.95$	100	1.6	**0.5**
$a = 0.05$	50	1.6	**0.4**
Notch 1	33.3	0.3	**0.3**
$b_0 = 0.05$	100	12	**4.9**
$a = 0.95$	50	7.9	**0.7**
Peak 1	33.3	25	**0.1**

5 ANALOG EFFECTS MODELING

Neural network models of two effects pedals were created, the Behringer VP-1 Vintage Phaser and Donner Jet Convolution Flanger. The phaser pedal has a "Rate" control that sets the LFO frequency and a "Tone" switch. The flanger pedal has a "Rate," "Color," and "Range" control and a mode selector switch, which determines whether the LFO is on ("Normal" mode) or off ("Filter" mode). The effects of each of these controls are explored further in Secs. 5.2 and 5.3.

5.1 Dataset Creation

The accuracy and reliability of the LFO measurement technique was tested by creating input-output datasets of both of the analog pedals and annotating them with the predicted LFO signal. By using the LFO-annotated datasets to train neural network models of the effects, it should be clear if the measured LFO signal is accurate, as a poorly fit LFO signal will result in the neural network failing to train well. Based on experiments conducted in our previous work [15], we consider a training dataset of 60 s in length to be sufficient to train a neural network model.

To create the datasets, the LFO measurement and prediction method described in Sec. 3 was used, with the measurement signal being inserted periodically throughout the training dataset. A measurement signal of 20 s in length was inserted at the start of the dataset, then every subsequent 80 s, and finally again at the end of the dataset. This means that each 80-s segment of guitar audio has two 20-s measurement signals adjacent to it, one preceding and one following it. For each of the 80-s segments, the LFO signal is tracked over the duration of the adjacent measurement signals, and an LFO is fitted simultaneously to the measured LFO from both measurement signals. This improves the accuracy of the predicted LFO frequency, as a small error in frequency

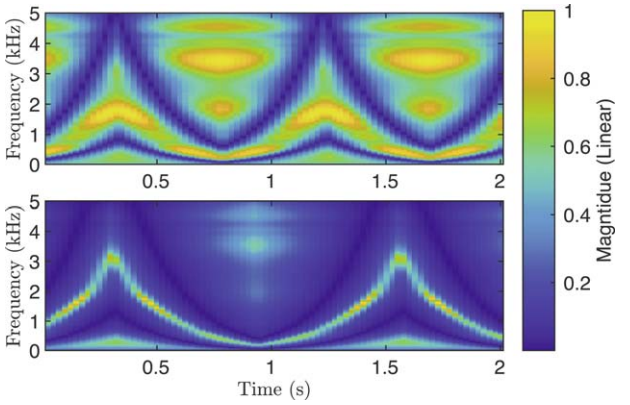

Fig. 9. Measurements of the Behringer Vintage Phaser Pedal's frequency response over time with the "tone" switch off (top) and on (bottom).

will result in the predicted LFO gradually going more out of sync at each subsequent LFO period.

The audio used in the dataset consisted of 16 guitar clips of 15 s each, 12 of which were taken from the *IDMT-SMT-Guitar* database [32] for guitar transcription and 4 of which were recorded by the authors. Eight of the clips were of country or classical style and processed through a guitar amplifier emulator on a clean setting. The other eight clips were of the metal or rock style and processed through a guitar amplifier on an overdriven setting. Twelve of the clips were selected for use in the training set, two were used in the validation set, and the remaining two made up the test set. Each 15-s clip was split into three 5-s segments, and three sub-datsets of 80 s in length were formed by concatenating a 5-s segment from each of the 16 clips.

This was done to ensure that each of the three sub-datasets contained one 5-s segment from each of the 16 original 15-s guitar clips. For each 80-s sub-dataset, the first 60 s makes up the training dataset, the following 10 s makes up the validation dataset, and the final 10 s makes up the test dataset. The clips were ordered such that the training, validation, and test datasets were always drawn from the same original guitar clips. Each sub-dataset can then be used to train a different neural network emulation of the effect, and the loss achieved by the network on the test dataset can be used to infer whether the LFO measurement was sufficiently accurate.

5.2 Phaser Pedal

The Behringer VP-1 Vintage Phaser pedal has a "Rate" control that controls the LFO rate as well as a "Tone" switch. Using the chirp train method described in Sec. 3, a spectrogram-like image was created, shown in Fig. 9, for both settings of the "Tone" switch, with the "Rate" set to 5. It is clear from the image that the LFO rate is decreased when the tone switch is in the on position. Furthermore the peaks in between the notches of the phaser effect become much more pronounced, and the frequency range of the two notches shown becomes greater.

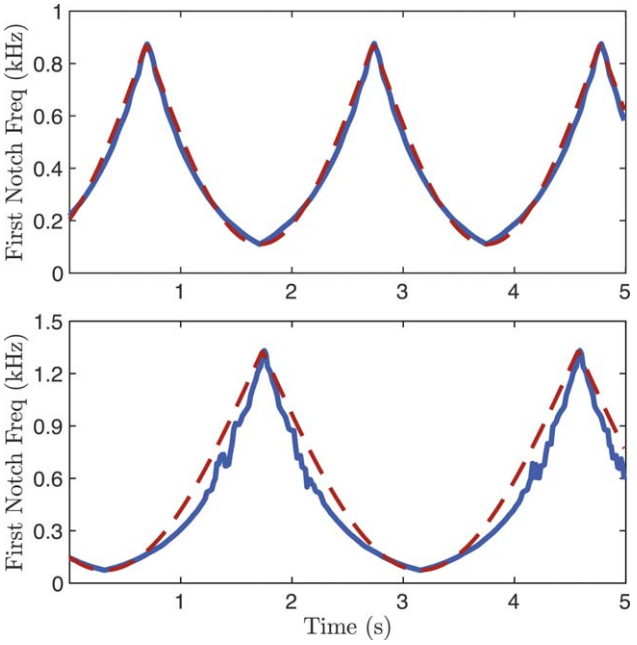

Fig. 10. Measured LFO (solid line) and fitted LFO (dashed line) for the phaser pedal with the "Tone" switch set to either off (top) or on (bottom).

Two datasets were created for the phaser pedal, one with the "Tone" switch in the on position and one with the "Tone" switch in the off position. The "Rate" control was set to three o'clock. Both datasets consist of three sub-datasets, created as described in Sec. 5.1. Examples of the measured LFO signal and resulting fitted LFO signal for the phaser pedal with the "Tone" switch set to either off or on are shown in Fig. 10 (top) and Fig. 10 (bottom), respectively. When the "Tone" switch is set to off, the measured notch frequency is very clean. The fitted LFO also very closely matches the measured LFO. When the "Tone" switch is set to on the measurement is noisier and the fitted LFO does not fit as well. It also appears that the measured LFO signal does not exactly match the shape of a rectified sine-wave anymore.

Each of the six sub-datasets were used to train two neural network models of the phaser pedal, with the neural networks having a hidden size of either 16 or 32. Training was run for 500 epochs and took approximately 2.5 hours.

5.3 Flanger Pedal

The Donner flanger pedal has a "Rate" control that controls the LFO rate, as well as a "Color" and "Range" control. Using the chirp train method described in Sec. 3, spectrogram-like images were created with the "Color" control, Fig. 11, or the "Range" control, Fig. 12, being set to either the 9 (top), 12 (middle), or 3 (bottom) o'clock positions.

Increasing the "Color" control results in the peaks becoming more pronounced and the notches becoming wider, while decreasing the "Color" control produces the opposite effect. This behavior is similar to that observed in the combined feedforward-feedback comb filter frequency re-

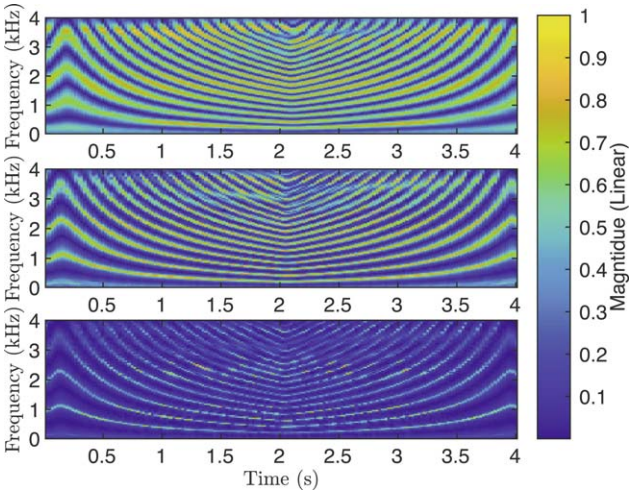

Fig. 11. Measurements of the Donner Flanger Pedal's frequency response over time with the "Color" knob set to (top) nine, (middle) twelve, and (bottom) three o'clock positions.

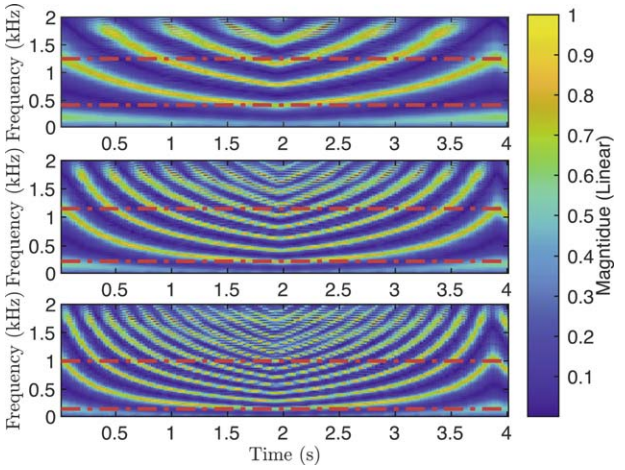

Fig. 12. Measurements of the Donner Flanger Pedal's frequency response over time with the "Range" knob set to (top) nine, (middle) twelve, and (bottom) three o'clock positions, With the dotted lines indicating the approximate minimum and maximum frequencies of the first peak.

sponse shown in Fig. 3, where increasing the gain of the feedback results in the peaks becoming more pronounced. As such this suggests that the "Color" control increases the feedback gain of the flanger pedal.

Increasing the "Range" control results in a reduction of the minimum and maximum frequencies of the notches and peaks. This means that there are more peaks/notches visible in the spectrum, as less of them exceed the Nyquist frequency. As can be seen in Figs. 1 and 2, increasing the length of the delay line results in more peaks/notches and also reduces the frequency at which they occur. This suggests that the "Range" control adjusts the LFO range, increasing both the maximum and minimum delay-line length used in the flanger.

Six datasets were created for the flanger pedal: three with the "Color" switch set to the 9 o'clock position and the "Range" switch set to the 9, 12, or 3 o'clock position and three with the "Color" switch set to the 12 o'clock position and the "Range" switch again being set to the 9, 12, or 3 o'clock position. Again, each of these six datasets consist of three sub-datasets, created as described in Sec. 5.1. This results in a total of 18 sub-datasets.

Examples of the measured LFO signal and resulting fitted LFO signal for the flanger pedal with "Color" set to 9 are shown in Fig. 13 and with "Color" set to the 12 o'clock position 12 in Fig. 14. When "Color" is set to the 9 o'clock position the tracked notch frequency reading is also fairly smooth. The shape of the measured LFO again deviates from the rectified sine-wave shape, although not significantly. When "Color" is set to the 12 o'clock position the measurements become noisier, with the noisiness increasing as the "Range" control is increased. In the last case when "Range" is set to the 3 o'clock position, the LFO fitting algorithm seems to perform quite poorly, with the fitted LFO being clearly out of sync with the measured LFO. The poorer performance when the "Color" is on a higher setting is similar to the findings from Sec. 4, where increasing the

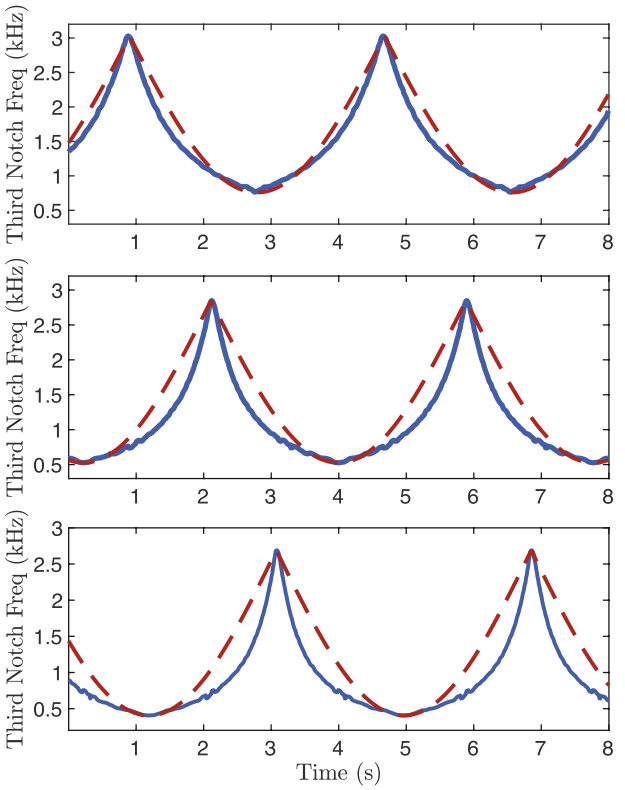

Fig. 13. Measured LFO (solid line) and fitted LFO (dashed line) for the flanger pedal with the "Color" set to the nine o'clock position and the "Range" set to the (top) nine, (middle) twelve, or (top) three o'clock position.

feedback gain in the digital flanger effect also resulted in less accurate LFO measurement.

Each of the 18 sub-datasets were used to train 2 neural network models. For the flanger pedal, neural network hidden sizes of either 32 or 48 were used. Training was again run for 500 epochs and took approximately 2.5 hours.

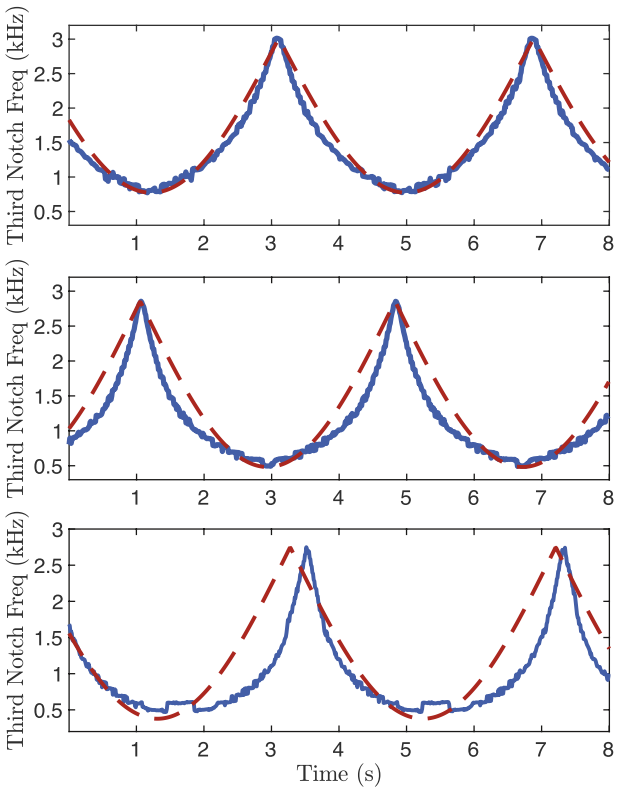

Fig. 14. Measured (solid line) and fitted LFO frequency (dashed line) for the flanger pedal with the "Color" set to the twelve o'clock position and the "Range" set to either the (top) nine, (middle) twelve, or (bottom) three o'clock position.

Table 4. Prediction of LFO frequency for each of the three sub-datasets created using the Behringer Vintage Phaser Pedal with the tone switch off or on, as well as the ESR loss achieved on the test dataset for a neural network model with hidden size of 16 or 32.

Dataset #	Pred. LFO Freq (Hz)	ESR (%)	
	Tone Switch: Off	$h = 16$	$h = 32$
1	0.4904	0.47	**0.19**
2	0.4904	0.31	**0.29**
3	0.4905	0.47	**0.29**
	Tone Switch: On		
1	0.3524	1.05	**0.24**
2	0.3524	2.4	**0.67**
3	0.3524	0.71	**0.48**

6 RESULTS

For the phaser pedal Table 4 shows the predicted LFO frequency and ESR for the neural networks of hidden size 16 and 32. For both datasets the predicted LFO frequencies for each of the three sub-datasets are identical up to three significant figures. This consistency between predictions provides confidence that the predicted LFO frequency is accurate. While it was observed that the LFO shape deviated from the rectified sine wave when the "Tone" switch was activated, the ESR loss achieved on the test dataset was still very low for all models trained, with most of them achieving a loss of less than 1%. However the model generally performed better when the "Tone" switch was set to off. It is not clear from these results whether the difference in model performance is due to the fitting of the LFO or the behavior of the effect itself.

Table 5. Prediction of LFO frequency for each of the three sub-datasets created using the Donner Flanger with various "Color" and "Range" settings, as well as the ESR loss achieved on the test dataset for a neural network model with hidden size of 32 or 48.

Dataset #	Pred. LFO Freq (Hz)	ESR (%)	
	Color: 9, Range: 9	$h = 32$	$h = 48$
1	0.2649	1.2	**1.1**
2	0.2648	0.9	**0.3**
3	0.2648	0.9	**0.6**
	Color: 9, Range: 12		
1	0.2649	3.2	**1.4**
2	0.2648	1.7	**1.1**
3	0.2649	3.6	**1.4**
	Color: 9, Range: 3		
1	0.2650	3.4	**2.7**
2	0.2650	5.7	**2.9**
3	0.2649	**2.4**	2.5
	Color: 12, Range: 9		
1	0.2650	**1.9**	**1.9**
2	0.2650	2.9	**2.3**
3	0.2649	**2.4**	2.5
	Color: 12, Range: 12		
1	0.2649	9.3	**5.7**
2	0.2649	**2.4**	4.6
3	0.2649	9.3	**8.7**
	Color: 12, Range: 3		
1	0.2543	**53**	54
2	0.2636	**16**	19
3	0.2545	**54**	57

For the flanger pedal Table 5 shows the predicted LFO frequency and ESR achieved on the test dataset for each sub-dataset for the neural networks of hidden size 32 and 48. Generally the LFO predictions between the different datasets are again very consistent.

The exception to this is when the "Color" and "Range" controls are set to the 12 and 3 o'clock positions, respectively. In this case the predicted LFO frequencies vary considerably for each sub-dataset. This indicates that the LFO tracking has performed poorly in this case, most probably due to noisy measurements, such as that shown in Fig. 14 (bottom). As might be expected this resulted in a poor performance in terms of the test loss achieved by the neural network models. It should be noted that for sub-dataset 2, when the predicted frequency was reasonably close to the majority of the other predicted LFO frequencies, the test

Table 6. Number of parameters and floating point operations per sample for neural network model with various LSTM hidden sizes.

Hidden Size	Parameters	Ops/sample	GFLOPS
8	393	1,849	0.08
16	1,297	4,721	0.21
32	4,641	13,537	0.60
48	10,033	26,449	1.17

loss is much lower than for sub-datasets 1 and 3. This indicates that the model may still be capable of modeling the flanger effect on this setting, if it were provided with more accurate LFO data.

The general pattern that can be observed is that accuracy achieved by the models decreases as the "Color" and "Range" settings increase. It is not clear whether this is due to lower LFO tracking accuracy, limitations of the neural network model, or a combination of both. It is possible that the model struggles to deal with the longer delay-line length or increased number of notches/peaks required when the "Range" parameter is increased. Likewise the increase in the feedback gain introduced by increasing the "Color" setting may also be hard for the network to emulate.

It can also be noted that for both the phaser and flanger pedals some parameter settings cause the LFO signal to deviate from the rectified sine-wave shape. In these cases the LFO fitting algorithm is often still able to generate a reasonable frequency estimate, and in some cases the neural network is still able to accurately model the effect. However further work could extend the method such that it could fit alternate LFO shapes more accurately.

For most of the settings trialed, the network was able to achieve a loss of less than 3%, which previous listening tests, conducted for guitar amplifiers models [3], suggest correspond to very minor differences being perceived between audio produced by the model and the target device. Informal listening also indicates that the models sound very similar to the target devices, although further listening tests should be conducted to confirm the accuracy of the models. Audio examples are available at the accompanying web page [33]. The Matlab code used to create the datasets and the Python code used to create and train the neural network models are also available at https://github.com/Alec-Wright/NeuralTimeVaryFx.

6.1 Computational Complexity

In Table 6 we report the number of learnable parameters in the model, as well as the number of floating point operations per sample and number of floating point operations per second (FLOPS), assuming the model is run at a sample rate of 44.1 kHz. The number of floating point operations were estimated by counting the number of multiplications, additions, and activation function evaluations, assuming 30 operations for both tanh(.) and sigma(.). The estimated FLOPS are well within the capabilities of real-time performance for a modern computer.

While the required processing power for running the neural network model was not measured in this study, the model is very similar to a previously tested model [34], which, when run on an Apple iMac with 2.8 GHz Intel Core i5 processor, could process 1 second of audio in 0.12 s when an LSTM hidden size of 32 was used, or 0.24 s for a hidden size of 64.

7 CONCLUSION

This paper has shown how a neural network can be used to model LFO-modulated effects, such as phaser and flanger pedals. During training, the target is audio processed by the effect, and the neural network inputs are the unprocessed audio and an estimated LFO signal. An improved method for estimating the LFO frequency and phase was introduced that is more accurate than a previous method. The LFO estimation method was tested on a digital flanger algorithm, in which case the measured LFO signal could be compared to the real LFO signal. This showed that a sufficiently accurate measurement of the effect behavior could be taken, even when taking measurements as frequently as every 10 ms. Furthermore the error in the predicted LFO frequency was generally very small and often less than 0.01%.

Real analog phaser and flanger pedals were then modeled by first estimating the frequency and phase of their LFO signal. This turned out to be more reliable in the case of the phaser pedal than the flanger. The data obtained from the LFO estimation was used for training neural network models, with hidden sizes between 16 and 48. Using an LSTM with a hidden size of 32 for the phaser pedal, an ESR of 0.2% was obtained. An LSTM with a hidden size of 48 achieved an ESR of 0.3% for the flanger pedal. According to our previous studies these values correspond to hardly audible modeling errors. A formal listening test would still be needed to verify the good perceptual quality obtained with the best neural models of this work. This is left for future work.

The results of this paper show that time-varying effects pedals can be successfully emulated with a deep neural network when the LFO signal is modeled and shown to the neural network separately. The neural networks used here are sufficiently compact to be included in real-time applications.

8 ACKNOWLEDGMENT

This research is part of the activities of the "Nordic Sound and Music Computing Network" (NordicSMC), NordForsk project no. 86892.

9 REFERENCES

[1] E.-P. Damskägg, L. Juvela, and V. Välimäki, "Real-Time Modeling of Audio Distortion Circuits With Deep Learning," in *Proceedings of the 16th Sound & Music Com-*

puting Conference (SMC), pp. 332–339 (Malaga, Spain) (2019 May).

[2] J. D. Parker, F. Esqueda, and A. Bergner, "Modelling of Nonlinear State-Space Systems Using a Deep Neural Network," in *Proceedings of the 22nd International Conference on Digital Audio Effects (DAFx-19)* (Birmingham, UK) (2019 Sep.).

[3] A. Wright, E.-P. Damskägg, L. Juvela, and V. Välimäki, "Real-Time Guitar Amplifier Emulation With Deep Learning," *Appl. Sci.*, vol. 10, no. 3 (2020 Jan.), paper 766. https://doi.org/10.3390/app10030766.

[4] M. A. Martínez Ramírez, E. Benetos, and J. D. Reiss, "Deep Learning for Black-Box Modeling of Audio Effects," *Appl. Sci.*, vol. 10, no. 2, p. 638 (2020 Jan.). https://doi.org/10.3390/app10020638.

[5] F. Rumsey, "Modeling Audio Effects," *J. Audio Eng. Soc.*, vol. 68, no. 1/2, pp. 100–104 (2020 Jan.).

[6] J. Covert and D. L. Livingston, "A Vacuum-Tube Guitar Amplifier Model Using a Recurrent Neural Network," in *Proceedings of the IEEE Southeastcon*, pp. 1–5 (Jacksonville, FL) (2013 Apr.). https://doi.org/10.1109/secon.2013.6567472.

[7] Z. Zhang, E. Olbrych, J. Bruchalski, T. J. McCormick, and D. L. Livingston, "A Vacuum-Tube Guitar Amplifier Model Using Long/Short-Term Memory Networks," in *Proceedings of the SoutheastCon*, pp. 1–5 (St. Petersburg, FL) (2018 Apr.). https://doi.org/10.1109/secon.2018.8479039.

[8] M. A. Martínez Ramírez and J. D. Reiss, "End-to-End Equalization With Convolutional Neural Networks," in *Proceedings of the 21st International Conference on Digital Audio Effects (DAFx-18)*, pp. 296–303 (Aveiro, Portugal) (2018 Sep.).

[9] T. Hélie, "Volterra Series and State Transformation for Real-Time Simulations of Audio Circuits Including Saturations: Application to the Moog Ladder Filter," *IEEE Trans. Audio Speech Lang. Process.*, vol. 18, no. 4, pp. 747–759 (2010 May). https://doi.org/10.1109/TASL.2009.2035211.

[10] L. Tronchin, "The Emulation of Nonlinear Time-Invariant Audio Systems With Memory by Means of Volterra Series," *J. Audio Eng. Soc.*, vol. 60, no. 12, pp. 984–996 (2012 Dec.).

[11] T. Schmitz and J.-J. Embrechts, "Hammerstein Kernels Identification by Means of a Sine Sweep Technique Applied to Nonlinear Audio Devices Emulation," *J. Audio Eng. Soc.*, vol. 65, no. 9, pp. 696–710 (2017 Sep.).

[12] S. Orcioni, A. Terenzi, S. Cecchi, F. Piazza, and A. Carini, "Identification of Volterra Models of Tube Audio Devices Using Multiple-Variance Method," *J. Audio Eng. Soc.*, vol. 66, no. 10, pp. 823–838 (2018 Oct.). https://doi.org/10.17743/jaes.2018.0046.

[13] F. Eichas and U. Zölzer, "Gray-Box Modeling of Guitar Amplifiers," *J. Audio Eng. Soc.*, vol. 66, no. 12, pp. 1006–1015 (2018 Dec.). https://doi.org/10.17743/jaes.2018.0052.

[14] R. Kiiski, F. Esqueda, and V. Välimäki, "Time-Varying Gray-Box Modeling of a Phaser Pedal," in *Proceedings of the 19th International Conference on Digital Audio Effects (DAFx-16)*, pp. 31–38 (Brno, Czech Republic) (2016 Sep.).

[15] A. Wright and V. Välimäki, "Neural Modelling of Periodically Modulated Time-Varying Effects," in *Proceedings of the 23rd International Conference on Digital Audio Effects (DAFx2020)*, pp. 281–288 (Vienna, Austria) (2020 Sep.).

[16] W. M. Hartmann, "Flanging and Phasers," *J. Audio Eng. Soc.*, vol. 26, no. 6, pp. 439–443 (1978 Jun.).

[17] H. Bode, "History of Electronic Sound Modification," *J. Audio Eng. Soc.*, vol. 32, no. 10, pp. 730–739 (1984 Oct.).

[18] J. O. Smith, "An Allpass Approach to Digital Phasing and Flanging," in *Proceedings of the International Computer Music Conference (ICMC)*, pp. 103–109 (Paris, France) (1984 Oct.). https://ccrma.stanford.edu/files/papers/stanm21.pdf.

[19] A. Huovilainen, "Enhanced Digital Models for Analog Modulation Effects," in *Proceedings of the 8th International Conference on Digital Audio Effects (DAFx-05)*, pp. 155–160 (Madrid, Spain) (2005 Sep.).

[20] V. Välimäki, S. Bilbao, J. O. Smith, J. S. Abel, J. Pakarinen, and D. Berners, "Virtual Analog Effects," in U. Zölzer (Ed.), *DAFX: Digital Audio Effects*, 2nd ed., pp. 473–522 (Wiley, Hoboken, NJ, 2011).

[21] B. Bartlett, "A Scientific Explanation of Phasing (Flanging)," *J. Audio Eng. Soc.*, vol. 18, no. 6, pp. 674–675 (1970 Dec.).

[22] M. L. Beigel, "A Digital "Phase Shifter" for Musical Applications, Using the Bell Labs (Alles-Fischer) Digital Filter Module," *J. Audio Eng. Soc.*, vol. 27, no. 9, pp. 673–676 (1979 Sep.).

[23] P. Dutilleux, M. Holters, S. Disch, and U. Zölzer, "Filters and Delays," in U. Zölzer (Ed.), *DAFX: Digital Audio Effects*, 2nd ed., pp. 70–76 (Wiley, Hoboken, NJ, 2011).

[24] M. Holters and J. D. Parker, "A Combined Model for a Bucket Brigade Device and Its Input and Output Filters," in *Proceedings of the 21st International Conference on Digital Audio Effects (DAFx-18)*, pp. 11–18 (Aveiro, Portugal) (2018 Sep.).

[25] S. Hochreiter and J. Schmidhuber, "Long Short-Term Memory," *Neural Comp.*, vol. 9, no. 8, pp. 1735–1780 (1997 Dec.).

[26] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, et al.,"PyTorch: An Imperative Style, High-Performance Deep Learning Library," in *Proceedings of the 33rd Conference on Neural Information Processing Systems (NeurIPS 2019)*, pp. 8024–8035 (Vancouver, Canada) (2019 Dec.).

[27] E.-P. Damskägg, L. Juvela, E. Thuillier, and V. Välimäki, "Deep Learning for Tube Amplifier Emulation," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 471–475 (Brighton, UK) (2019 May). https://doi.org/10.1109/ICASSP.2019.8682805.

[28] D. P. Kingma and J. L. Ba, "Adam: A Method for Stochastic Optimization," in *Proceedings of the Interna-*

tional Conference on Learning Representations (ICLR-15) (San Diego, CA) (2015 May).

[29] M. Holters, T. Corbach, and U. Zölzer, "Impulse Response Measurement Techniques and Their Applicability in the Real World," in *Proceedings of the 12th International Conference on Digital Audio Effects (DAFx-09)*, pp. 108–112 (Como, Italy) (2009 Sep.).

[30] E. K. Canfield-Dafilou and J. S. Abel, "An All-pass Chirp for Constant Signal-to-Noise Ratio Impulse Response Measurement," presented at the *144th Convention of the Audio Engineering Society* (2018 May), paper 10014.

[31] T. I. Laakso, V. Välimäki, M. Karjalainen, and U. K. Laine, "Splitting the Unit Delay: Tools for Fractional Delay Filter Design," *IEEE Signal Process. Mag.*, vol. 13, no. 1, pp. 30–60 (1996 Jan.).

[32] C. Kehling, J. Abeßer, C. Dittmar, and G. Schuller, "Automatic Tablature Transcription of Electric Guitar Recordings by Estimation of Score- and Instrument-Related Parameters," in *Proceedings of the 17th International Conference on Digital Audio Effects (DAFx-14)*, pp. 219–226 (Erlangen, Germany) (2014 Sep.).

[33] A. Wright and V. Välimäki, "Neural Modelling of Phaser and Flanging Effects," http://research.spa.aalto.fi/publications/papers/jaes-tvfx/ (accessed Apr. 30, 2021).

[34] A. Wright, E.-P. Damskägg, and V. Välimäki, "Real-Time Black-Box Modelling With Recurrent Neural Networks," in *Proceedings of the 22nd International Conference on Digital Audio Effects (DAFx-19)* (Birmingham, UK) (2019 Sep.).

THE AUTHORS

Alec Wright

Vesa Välimäki

Alec Wright received his M.Eng. degree in Mechanical Engineering from The University of Manchester, UK, in 2014 and his M.Sc. degree in Acoustics and Music Technology from the University of Edinburgh in 2018. He is currently a doctoral candidate in the Acoustics Lab of Aalto University in Espoo, Finland. His research interests include virtual analog modeling of guitar amplifiers and other audio effects and machine learning for audio applications.

•

Vesa Välimäki is Professor of Audio Signal Processing at Aalto University, Espoo, Finland. He is the Vice Dean for Research in the Aalto University School of Electrical Engineering. He received his M.Sc. and D.Sc. degrees from the Helsinki University of Technology (TKK) in 1992 and 1995, respectively. In 1996 he was a Postdoctoral Research Fellow at the University of Westminster, London, UK. In 2001 to 2002 he was Professor of Signal Processing at the Pori School of Technology and Economics, Tampere University of Technology, Pori, Finland. In 2006 to 2007 he was the head of the TKK Laboratory of Acoustics and Audio Signal Processing. During the academic year 2008 to 2009 he was a visiting scholar at the Center for Computer Research in Music and Acoustics (CCRMA), Stanford University, Stanford, CA, USA. His research interests include audio effects processing, digital filter design, loudspeaker and headphone signal processing, and sound synthesis. Prof. Välimäki is a Fellow of the AES, a Fellow of the IEEE, and a Life Member of the Acoustical Society of Finland. He was the General Chair of the 11th International Conference on Digital Audio Effects DAFx-08 in 2008 and the General Chair of the 14th International Sound and Music Computing Conference SMC-17 in 2017. Currently Prof. Välimäki is the Editor-in-Chief of the *Journal of the Audio Engineering Society*.