BLACK-BOX MODELING OF DISTORTION CIRCUITS WITH BLOCK-ORIENTED MODELS

Felix Eichas, Udo Zölzer

Department of Signal Processing and Communications,
Helmut Schmidt University
Hamburg, Germany
`felix.eichas@hsu-hh.de`

ABSTRACT

This paper describes black-box modeling of distortion circuits. The analyzed distortion circuits all originate from guitar effect pedals, which are widely used to enrich the sound of an electric guitar with harmonics. The proposed method employs a block-oriented model which consists of a linear block (filter) and a non-linear block. In this study the nonlinear block is represented by an extended parametric input/output mapping function. Three distortion circuits with different nonlinear elements are analyzed and modeled. The linear and nonlinear parts of the circuit are analyzed and modeled separately. The Levenberg–Marquardt algorithm is used for iterative optimization of the nonlinear parts of the circuits. Some circuits could not be modeled with high accuracy, but the proposed model has shown to be a versatile and flexible tool when modeling distortion circuits.

1. INTRODUCTION

Virtual analog modeling has widely been done before. The focus in virtual analog modeling of electric guitar equipment lies on recreating an analog reference device as close as possible. In [1–7] this has been done with great success by analyzing the analog reference circuit and transferring the circuit into a mathematical model, which is able to recreate the original's characteristics. This circuit-based approach achieves very convincing results and the digital model is mostly indistinguishable from the analog reference device for the human ear. But this precise approach also has drawbacks. To create the digital model the circuit diagram has to be known, as well as the characteristic curves of every nonlinear circuit element, e.g. diodes, transistors, transformers or vacuum tubes. If no circuit diagram is obtainable, time-consuming reverse engineering of the circuit has to be performed, as described in e.g. [8].

Another drawback of this method is the computational effort which arises due to nonlinear circuit elements. For every nonlinear circuit element at least one nonlinear equation has to be solved per time step. Depending on the nonlinear solver and the initial parameter set, this can drastically influence the computational load of the digital model. Although Holmes et al. described a method for improving the nonlinear solver in [9], the computational effort is still high, especially for complex circuits with multiple nonlinearities.

Alternative approaches for modeling of distortion circuits are found in [10] and [11]. Block-oriented models are used to represent the distortion circuit. A block-oriented model generally consists of linear blocks and static nonlinear blocks. Some common topologies have conventional names, like Hammerstein model (static nonlinearity followed by a filter in series), Wiener model (linear filter followed by a static nonlinearity in series) or Wiener–Hammerstein model (filter followed by static nonlinearity followed by filter). In [10] a parallel, generalized polynomial Hammerstein model is used. Each parallel branch represents the different harmonic components of the reference system. In [11] principal component analysis is used to reduce the complexity of the model presented in [10].

In [12] a completely parametric Wiener–Hammerstein model is used to automatically identify distortion guitar effect pedals. The structure of the used Wiener–Hammerstein model is the series connection of a parametric filter followed by a nonlinear block, which is also used in this work, followed by another parametric filter. The filters are identified by iterative parameter optimization, using low-level noise as input signal. To find the initial parameter set for the nonlinear part of the model, a time-consuming grid search is performed.

As in [12] the basic idea behind this work is to analyze and model nonlinear distortion circuits, without knowledge of the circuit itself. Only input/output measurements are performed to adjust the parameters of a block-oriented, nonlinear model to recreate the characteristics of the analog reference device. In this work, only parts of distortion circuits should be modeled, which use different electronic components to create distortion. In this work the Wiener–Hammerstein model from [12] could be reduced to an extended Wiener model (filter followed by nonlinear block), because all chosen circuit parts did not have any filter at the output. The aim of this work is to analyze how well such a simple model can recreate the behavior of the circuits with an automated optimization procedure.

Section 2 describes the analyzed distortion circuits and Sec. 3 explains the topology of the extended Wiener model. Sections 4 and 5 describe the modeling process. Section 6 compares the identified models to the reference circuits and Sec. 7 concludes the paper.

2. HARDWARE

Three different distortion circuits were analyzed in this work. The first one was a simple diode clipper with pre-amplification of the input signal, the second one a BJT distortion stage of an Electro Harmonix - Big Muff Pi[TM] and the operational amplifier based distortion stage of an Ibanez - Tube Screamer[TM]. All circuits were simulated with a spice circuit simulator, `LTSpice`[13].

2.1. Diode Clipper

The circuit of the diode clipper can be seen in Fig. 1. It is an extension of the diode clipper circuit from [14] with an additional non-inverting amplifier.
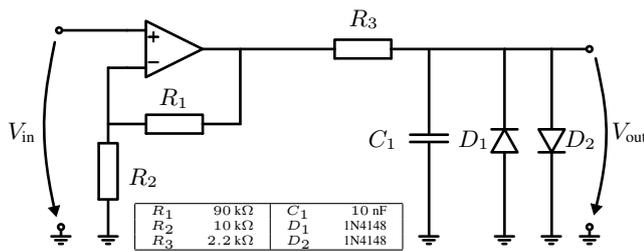
Figure 1: *Circuit of the diode clipper with pre-amplification.*

The input signal is amplified by the non inverting amplifier with a factor of $1 + R_1/R_2 = 10$ and the resulting amplified signal passes through a low-pass RC circuit with cut-off frequency $f_c = 1/(2\pi R_3 C_1) \approx 7.23$ kHz and finally through the anti-parallel 1N4148 diodes. The supply voltage of the operational amplifier was set to a relatively high value of ± 30 V to avoid additional clipping. The simulated operational amplifier was a TL072.

2.2. Big Muff Distortion Stage

The distortion stage of the Big Muff can be seen as an extended BJT transimpedance gain stage as described in [15]. It was also modeled in [7], using wave digital filters.
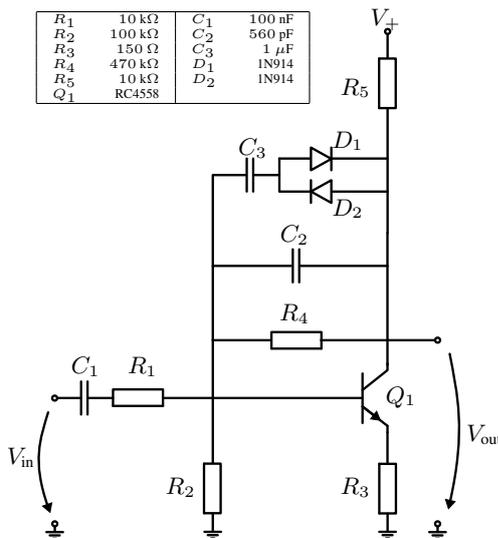

Figure 2: *Circuit of a Big Muff transimpedance gain stage.*

Fig. 2 shows the circuit around the NPN transistor. In addition to the feedback from collector to base via R_4 and C_2, there are two anti-parallel diodes D_1, D_2 as well as the capacitor C_3. These diodes introduce further clipping in addition to the clipping of the BJT circuit itself. The supply voltage was set to $V_+ = 9$ V.

2.3. Tube Screamer Distortion Stage

The Tube Screamer is based on an operational amplifier gain stage, also described in [15–18]. The input signal is amplified and additionally distorted by two anti-parallel diodes in the feedback path from output to negative input of the op-amp.
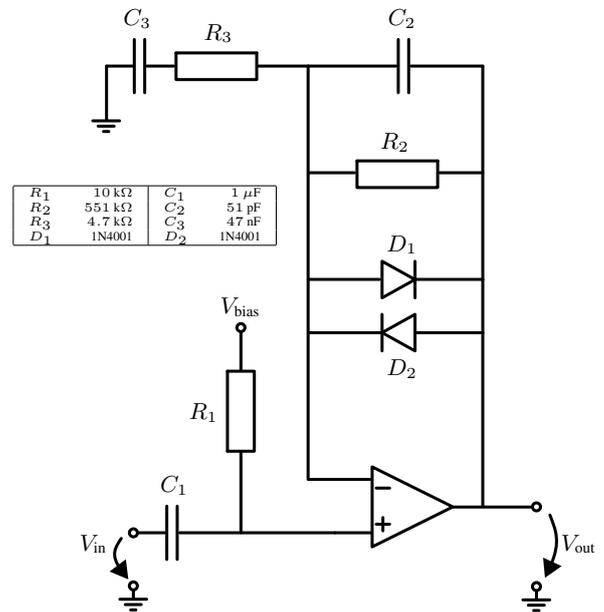

Figure 3: *Circuit of a Tube Screamer operational amplifier gain stage.*

Fig. 3 shows the circuit with resistor R_2 which is a potentiometer in the Tube Screamer circuit. It takes values from $R_{2,\text{min}} \approx 51$ kΩ to $R_{2,\text{max}} = 551$ kΩ. In this work the potentiometer was always set to $R_{2,\text{max}}$ to maximize the amplification of the operational amplifier and thus the distortion of the output signal. The bias voltage was set to $V_{\text{bias}} = 4.5$ V, which is half the supply voltage of the operational amplifier. The used op-amp was a general-purpose amplifier RC4558 by Texas Instruments.

3. MODEL TOPOLOGY

The digital model which was chosen to represent these distortion circuits is an extended Wiener model. It consists of a linear time invariant block followed by a nonlinear block. Fig. 4 shows the block diagram of the extended Wiener model. In this work the LTI
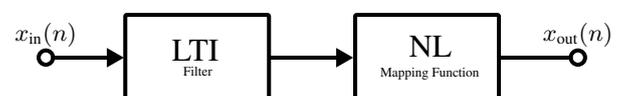

Figure 4: *Block diagram of a Wiener model.*

block is represented by a FIR filter, but it could be easily modified to include any other LTI system e.g. state-space systems or IIR filters. The nonlinear block consists mainly of a mapping function, mapping input amplitude to output amplitude. Figure 5 illustrates the function principle of a static, memory-less mapping function. Each input sample is processed by the nonlinear equation, creating
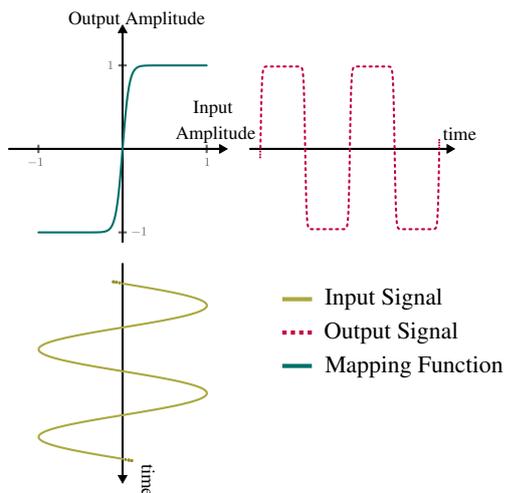
Figure 5: *Function principle of a static mapping curve.*

a distorted output signal. Some extensions have been made to the mapping function (see Sec. 5.2). Especially the side-chain with low-pass filter leads to a nonlinear block which is not memory-less anymore. So the model is not a Wiener model in the classical sense, it is an extended Wiener model with distortion-circuit specific refinements.

4. MODELING: LINEAR PART

The linear part of all circuits was measured with exponential sine sweeps, as described in [19]. The sweep is described in terms of start frequency $\omega_1 = 2\pi f_{start}/f_s$, stop frequency $\omega_2 = 2\pi f_{stop}/f_s$ and amplitude A,

$$x_{sw}(n) = A \cdot \sin\left(\frac{\omega_1 \cdot (L-1)}{\log\left(\omega_2/\omega_1\right)} \cdot \left(e^{\left(\frac{n}{L-1}\log(\omega_2/\omega_1)\right)} - 1\right)\right),$$
(1)

where L is the total length of the sweep in samples. An inverse-filter signal to the sweep can be created which fulfills the condition

$$x_{sw}(n) * x_{inv}(n) \approx c \cdot \delta(n - n_0).$$
(2)

This means that the sweep convolved with the inverse filter yields a Dirac delta function which is only shifted in time and scaled by some factor c. Due to the assumption that very low signal levels will pass through the linear region of the distortion circuits, the maximum amplitude of the sweep was set to $A = 0.01$ V, to ensure that no nonlinear distortion occurs while measuring the output signal. The output sweep $y_{sw}(n)$ is recorded and convolved with the inverse filter

$$x_{inv}(n) = x_{sw}(L - 1 - n) \cdot (\omega_2/\omega_1)^{\frac{-n}{L-1}}$$
(3)

to get the impulse response of the system

$$h(n) = \frac{1}{c} \cdot x_{inv}(n) * y_{sw}(n).$$
(4)

When using this technique $h(n)$ does not only contain the linear response of the system. It also contains the impulse responses for

higher order harmonics, as described in [10]. Therefore the impulse response has to be segmented in time-domain and is normalized to a maximum magnitude of 0 dB in frequency-domain. Afterwards it is saved and directly used as FIR filter coefficients in the digital model.

This methods performs better than the iterative parameter optimization approach based on white noise, described in [12]. The small-signal impulse response is directly measured and used in the model, instead of iteratively adapting several filters to approximate the frequency response of the circuit.

5. MODELING: NONLINEAR PART

Modeling of the nonlinear part is done by creating a reference signal, in this case the output voltage of each circuit to a specific (known) input signal. Afterwards it is compared to the output of the digital model to compute the error between both signals according to a cost function. This cost function has to be minimized to find the optimal set of parameters for the given reference signal.

5.1. Input Signal

When designing the input signal it is important to consider the influence of the parameters on the output. The nonlinear block in the extended Wiener model is frequency independent, which means that it is not necessary for the input signal to excite more than one frequency. But it is most important to excite all possible amplitudes of the input signal, so their modification by the reference system can be observed. A single frequency sine wave with logarithmically rising amplitude was used as the input signal,

$$x_{nl}(n) = a(n) \cdot \sin\left(\frac{2\pi f_0 n}{f_s}\right).$$
(5)

The fundamental frequency was set to $f_0 = 1000$ Hz. The amplitude scaling function $a(n)$ is logarithmically increasing from a start value of $a(1) = 1 \cdot 10^{-5}$ to the largest value $a(N) = 1$, with N as the total amount of samples.

5.2. Parametric Nonlinear Block

The parametric nonlinear block is based on a mapping function, described in Sec. 3. It was already used in [12] to model distortion pedals. Figure 6 shows the nonlinear block. Its main component is
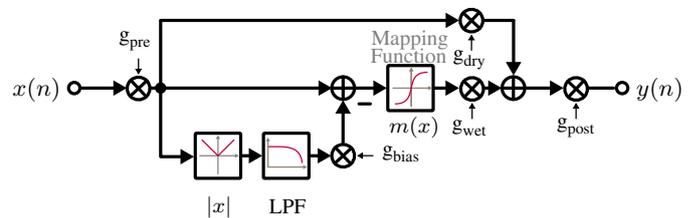

Figure 6: *Nonlinear mapping function with extensions.*

the mapping function, which is a combination of three hyperbolic tangent functions. The combination of the three functions, with the amplitude of the signal x as input, is shown by (6).

$$m(x) = \begin{cases} \tanh(k_p) - \left[\frac{\tanh(k_p)^2 - 1}{g_p} \tanh(g_p x - k_p) \right] & \text{if } x > k_p \\ \tanh(x) & \text{if } -k_n \le x \le k_p \\ -\tanh(k_n) - \left[\frac{\tanh(k_n)^2 - 1}{g_n} \tanh(g_n x + k_n) \right] & \text{if } x < -k_n \end{cases}$$
(6)

The additional terms for $x > k_p$ and $x < -k_n$ are needed to ensure that $m(x)$ has a continuous derivative at the connection points k_n and k_p. Besides the mapping function, the nonlinear block has a pre-gain g_{pre} to scale the input signal and a post-gain g_{post} to scale the output. The side-chain envelope detector, consisting of absolute value calculation and low-pass filtering with a cut-off frequency of $f_{\text{c,LP}} = 5\,\text{Hz}$, and a dry/wet mixing stage. Note that the dry gain parameter is calculated automatically by $g_{\text{dry}} = 1 - g_{\text{wet}}$. The envelope subtraction from the direct signal is used to emulate the signal-dependent bias-point shift which occurs for vacuum tubes or transistors. It is an adapted version of the so called 'tube stage' from [20], but to avoid feedback, it is constructed in a feed-forward loop. Please note that this extension prevents the nonlinear block from being memoryless, because the output is dependent on the previous values of the input signal. This leads to a total amount of eight parameters, four gains and four parameters for the mapping function, which are combined in the parameter vector

$$\mathbf{p} = \left(g_{\text{pre}} \quad g_{\text{bias}} \quad k_p \quad k_n \quad g_p \quad g_n \quad g_{\text{wet}} \quad g_{\text{post}} \right)^T.$$
(7)

With $g_{\text{wet}} = 1 - g_{\text{dry}}$.

5.3. Parameter Optimization

The nonlinear block is initialized with a parameter set that only introduces a slight distortion for high signal levels. To improve the robustness of convergence during optimization, the parameters are optimized in three different steps. The used algorithm is the gradient-based Levenberg–Marquardt optimization procedure [21, 22].

5.3.1. Levenberg–Marquardt

Consider a reference system which produces output $y_{\text{sys}}(n)$ (after A/D conversion) and the corresponding digital model which produces output $y_{\text{mod}}(n, \mathbf{p})$, depending not only on the input signal $x(n)$, but also on the parameter vector $\mathbf{p}$. The residual

$$\text{res}(n, \mathbf{p}) = y_{\text{sys}}(n) - y_{\text{mod}}(n, \mathbf{p})$$
(8)

describes the difference of reference system and digital model.

The Levenberg–Marquardt algorithm is a combination of the *gradient-descent* and the *Gauss–Newton* methods. The parameter vector $\mathbf{p}$ is updated by,

$$\mathbf{\Delta p} = \left(\mathbf{J}^T \mathbf{J} + \lambda \cdot \text{diag}(\mathbf{J}^T \mathbf{J}) \right)^{-1} \cdot \text{grad}(\mathbf{p}).$$
(9)

$\mathbf{J}$ is the Jacobi matrix, where each column represents the derivative of the residual with respect to the parameter vector $\mathbf{p}$,

$$\mathbf{J} = \begin{bmatrix} \frac{\partial \text{res}(1,\mathbf{p})}{\partial p_1} & \cdots & \frac{\partial \text{res}(1,\mathbf{p})}{\partial p_M} \\ \frac{\partial \text{res}(2,\mathbf{p})}{\partial p_1} & \cdots & \frac{\partial \text{res}(2,\mathbf{p})}{\partial p_M} \\ \vdots & \ddots & \vdots \\ \frac{\partial \text{res}(N,\mathbf{p})}{\partial p_1} & \cdots & \frac{\partial \text{res}(N,\mathbf{p})}{\partial p_M} \end{bmatrix}$$
(10)

with M as the total number of parameters and N as the total length of the residual. The gradient

$$\text{grad}(\mathbf{p}) = \mathbf{J}^T \cdot \left[\text{res}(1, \mathbf{p}) \quad \cdots \quad \text{res}(N, \mathbf{p}) \right]^T$$
(11)

describes in which direction of each entry in the parameter vector we have to descend to minimize the error between reference system and digital model.

For large values of λ the algorithm behaves more like *gradient-descent*, while for small values of λ the algorithm behaves more like *Gauss–Newton* [23]. Although *Gauss-Newton* is an efficient method there exist cases where the algorithm needs a long time to converge into the minimum, or does not converge at all. If λ is initially set to a relatively small value and the cost function does not decrease, e.g.

$$C(\mathbf{p}) < C(\mathbf{p} + \mathbf{\Delta p}),$$

λ is increased to get quicker convergence with the *gradient-descent* method.

If the current step was successful, e.g.

$$C(\mathbf{p}) > C(\mathbf{p} + \mathbf{\Delta p}),$$

the parameter vector for each iteration k is updated,

$$\mathbf{p}_{k+1} = \mathbf{p}_k + \mathbf{\Delta p}_k,$$

and λ is decreased to make use of the advantageous properties of the *Gauss–Newton* algorithm near the solution.

5.3.2. Cost Function

The choice of the cost function is crucial for the robustness of the optimization process. The straight-forward approach would be to simply calculate the difference of digital model output and analog reference output in time-domain, as shown by (8). But if the phase characteristic of reference and model is not matched perfectly, the time-domain error is quite high, which does not necessarily represent the human perception of the difference between the two signals. To neglect any phase shift between reference system and digital model the cost function is designed to match the envelopes of both systems. The envelopes are calculated for positive and negative signal amplitudes separately, because the nonlinear mapping function, described in Sec. 5.2 is able to shape positive and negative amplitudes independently. To calculate the envelope the signals are half-wave rectified and low-pass filtered by a second order IIR low-pass with a cut-off frequency of $f_c = 5\,\text{Hz}$. To calculate the envelope for negative amplitudes the signals are multiplied with -1 before half-wave rectification.

Since the Levenberg Marquardt algorithm is gradient-based, convergence into a local minimum is possible when the initial set of parameters is too far from the global minimum of the cost function. This is why the main optimization procedure is divided into three parts.

1. Optimize parameters for positive amplitudes (while ignoring parameters which change negative amplitude)

2. Optimize ignored parameters from step 1. for negative amplitudes

3. Refine all parameters for positive and negative amplitudes

Another benefit of matching the signals' envelopes is a robust identification. The output of the digital model depends on fewer parameters in step one and two than in step three. In step one, for example, the parameters k_n and g_n can be ignored, because they only influence negative amplitudes. If the initial parameter set leads to a slightly nonlinear mapping function, the Levenberg–Marquardt algorithm always matches the envelopes satisfactorily. If this is done for positive and negative amplitudes, the parameter vector will be close to the optimal solution when starting the refinement in step three.

6. RESULTS

In this section the results of the modeling process are presented and each digital model is compared to the corresponding reference system.

6.1. Metrics Definition

To rate the result of the optimization, a recorded guitar track was played back through both systems and the percentage of error energy or 'error to signal ratio' (ESR) was calculated. It is defined as the ratio of error energy to the energy of the reference output,

$$\text{ESR} = \frac{E_{\text{res}}}{E_{\text{sys}}} = \frac{\sum_{n=-\infty}^{\infty} |y_{\text{sys}}(n) - y_{\text{mod}}(n, \mathbf{p})|^2}{\sum_{n=-\infty}^{\infty} |y_{\text{sys}}(n)|^2}. \quad (12)$$

Another way to calculate the difference is via the correlation coefficient, which describes the linear dependence of two random variables. The computation of the correlation coefficient is shown by

$$\rho(\text{A,B}) = \frac{\text{cov(A,B)}}{\sigma_A \sigma_B}, \quad (13)$$

where $A = y_{\text{sys}}(n)$ and $B = y_{\text{mod}}(n, \mathbf{p})$ are the random variables (in our case reference and model output), cov(A,B) is the covariance of A and B and $\sigma_{\text{A,B}}$ is the standard deviation of the random variables.

6.2. Modeling Results

The results of the modeling process are shown in Table 1. The diode clipper obtained the best results with an error to signal ratio of only ESR = 5.78% and a correlation coefficient of $\rho = 0.9983$. The informal listening test proved that there is no noticeable difference between the output of the circuit and the digital model. The

| Circuit	ESR	$\rho(\mathbf{A,B})$
Diode Clipper	0.0578	0.9983
Big Muff	0.0901	0.9578
Tube Screamer	0.1832	0.9062

Table 1: *Results of the modeling process.*

results for the other circuits are not as good. The Big Muff BJT gain stage has an ESR = 9.01% and a correlation coefficient of $\rho = 0.9578$, which already leads to a slightly perceivable difference between the signals. This can be explained by the feedback

path from the collector of the NPN transistor to its base (see Fig. 2). This feedback path is not modeled in the extended Wiener model, so the result of the modeling process is only an approximation of the real circuit.

The Tube Screamer has an ESR= 18.32% and a correlation coefficient of $\rho = 0.9062$, which also leads to a small noticeable difference between digital model and circuit output. This difference can also be explained by the simplicity of the extended Wiener model. In the original circuit, there is a feedback path from operational amplifier output to its negative input. This could be modeled by a parallel path to the mapping function with a filter, whose frequency response is unknown, because only the global frequency response can be measured, without detailed measurements of the analog circuit.
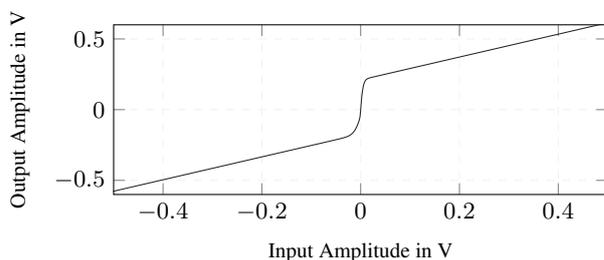
Figure 7 shows the mapping function of the *digital model* after the optimization procedure is finished. It can be seen that the diode clipper circuit has no dry signal at the output of the nonlinear block, because the slope of the mapping function at $x(n) = -1$ and $x(n) = 1$ is zero.
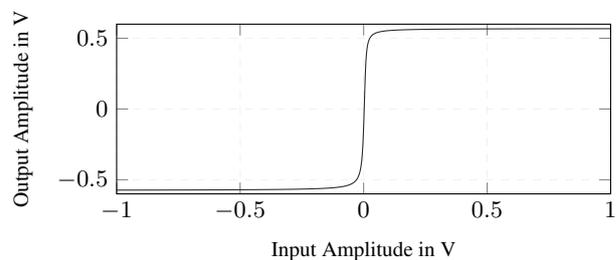
(a) Diode Clipper
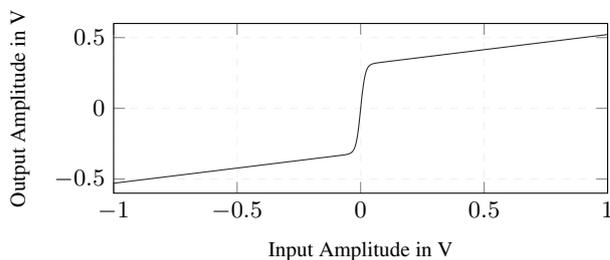
(b) Big Muff

(c) Tube Screamer

Figure 7: *Nonlinear mapping functions $m(x)$ after optimization.*

The Big Muff circuit has a little dry signal mixed together with the wet signal, which can be explained as a 'compensation' of the missing feedback path in the model. The Tube Screamer has a lot

of dry signal mixed together with the distorted wet signal, which confirms the assumption that the circuit can be modeled with a parallel dry signal, which can be mixed with the distorted signal.

All mapping functions look symmetrical, which is due to using the same model for the two diodes in each `Spice` simulation, leading to the same shape for positive and negative amplitudes. Only in the Tube Screamer circuit there is a slight difference between the shape of positive amplitudes and negative amplitudes, which is also visible in the mapping function, Fig. 7 (c), because the transition from steep middle part of the mapping function to higher amplitudes is a little softer for negative amplitudes.

The time-domain signals for each reference circuit and their comparison to the corresponding model are shown in Fig. 8. The input signal was a self-recorded riff played on a stratocaster-type electric guitar using the humbucker bridge-pickup. The guitar was directly connected to an RME - Fireface 800 audio interface. For the diode clipper, the digital model waveform is very close to the waveform of the reference signal, which leads to no perceivable difference between the two signals. With a rising ESR value for Big Muff and Tube Screamer the waveform of the digital model differs more and more from the reference output. Generally, it can be observed that the difference in the waveforms is proportional to the input amplitude, because for the first 500 samples of the test signal, all models are close to the reference signal, while for higher input amplitudes (sample 600 to 2000) the model is not accurate enough to recreate the more complex reference circuits.

In addition to these scores, an informal listening test was conducted. The participants of the test were five experienced researchers in virtual analog modeling, who should test if they are able to hear a difference between digital model and reference signal. In case of the diode clipper none of the participants was able to hear a difference between simulation and reference. For Big Muff and Tube Screamer, the results were not as convincing, since every test subject was able to hear a difference. Nevertheless all of the participants confirmed that the overall characteristic of the reference device could be captured by the corresponding digital version.

6.3. Listening Examples

To give the reader an impression of the model and compare it to the reference system some listening examples were created. The input signal consists of single notes, played by an electric guitar and decaying guitar chords as well as decaying single notes, played by an electric bass-guitar. No effect has been used to alter the signals before or after processing them with the digital model or the reference circuit simulation. The listening examples can be found on-line at [24].

7. CONCLUSION

Three distortion circuits have been modeled by an extended Wiener model, consisting only of one linear time-invariant block (filter) and an extended nonlinear mapping block, which maps input amplitudes to output amplitudes. The modeling was very successful for the diode clipper circuit, because it matches the model topology. Although the model is not able to emulate more complex circuits perfectly, it is able to recreate the reference circuit to a satisfying degree, given its simplicity. To really capture all the characteristics of a distortion circuit, especially for bigger, more complex circuits, the model needs to be expanded. A serial approach, concatenating several simple models, to refine the results
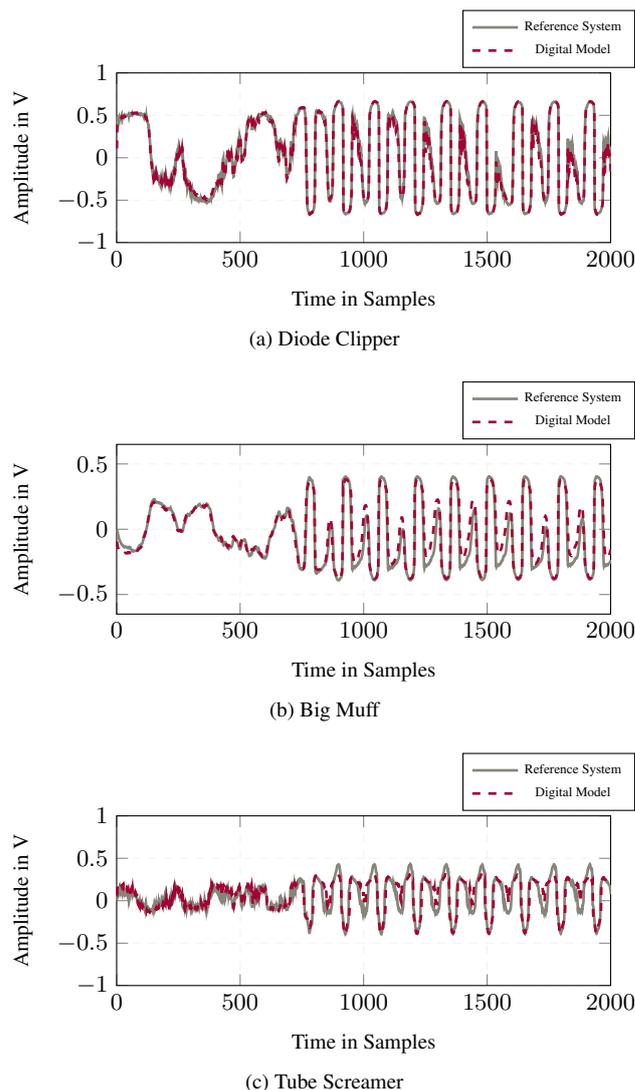

(a) Diode Clipper

(b) Big Muff

(c) Tube Screamer

Figure 8: *Time-domain response to a recorded guitar input for all three circuits.*

would be conceivable. Another expansion could be a feedback path with a unit delay, which allows more possibilities of shaping the waveform.

8. REFERENCES

[1] D.T. Yeh and J.O. Smith, "Simulating guitar distortion circuits using wave digital and nonlinear state-space formulations," in *Proc. Digital Audio Effects (DAFx-08)*, Espoo, Finland, Sept. 1–4, 2008, pp. 19–26.

[2] D.T. Yeh, J.S. Abel, and J.O. Smith, "Automated physical modeling of nonlinear audio circuits for real-time audio effects: part 1—theoretical development," in *IEEE Trans. Audio, Speech, and Language Process.*, May 2010, vol. 18, pp. 203–206.

[3] J. Mačák, *Real-time Digital Simulation of Guitar Amplifiers*

as Audio Effects, Ph.D. thesis, Brno University of Technology, 2011.

[4] K. Dempwolf, *Modellierung analoger Gitarrenverstärker mit digitaler Signalverarbeitung*, Ph.D. thesis, Helmut-Schmidt-Universität, 2012.

[5] M. Holters and U. Zölzer, "Physical modelling of a wah-wah effect pedal as a case study for application of the nodal dk method to circuits with variable parts," in *Proc. Digital Audio Effects (DAFx-11)*, Paris, France, Sept. 19–23, 2011, pp. 31–35.

[6] K.J. Werner, J.O. Smith, and J.S. Abel, "Wave digital filter adaptors for arbitrary topologies and multiport linear elements," in *Proc. Digital Audio Effects (DAFx-15)*, Trondheim, Norway, Nov. 30 – Dec. 3 2015.

[7] K.J. Werner, V. Nangia, J.O. Smith, and J.S. Abel, "Resolving wave digital filters with multiple/multiport nonlinearities," in *Proc. Digital Audio Effects (DAFx-15)*, Trondheim, Norway, Nov. 30 – Dec. 3 2015.

[8] C. Gnegy and K.J. Werner, "Digitizing the Ibanez Weeping Demon wah pedal," in *Proc. Digital Audio Effects (DAFx-15)*, Trondheim, Norway, Nov. 30 – Dec. 3 2015.

[9] B. Holmes and M. van Walstijn, "Improving the robustness of the iterative solver in state-space modelling of guitar distortion circuitry," in *Proc. Digital Audio Effects (DAFx-15)*, Trondheim, Norway, Nov. 30 – Dec. 3 2015.

[10] A. Novak, L. Simon, P. Lotton, and J. Gilbert, "Chebyshev model and synchronized swept sine method in nonlinear audio effect modeling," in *Proc. Digital Audio Effects (DAFx-10)*, Graz, Austria, Sept. 6–10, 2010.

[11] R.C.D. de Paiva, J. Pakarinen, and V. Välimäki, "Reduced-complexity modeling of high-order nonlinear audio systems using swept-sine and principal component analysis," in *Audio Engineering Society Conference: 45th International Conference: Applications of Time-Frequency Processing in Audio*, Mar. 2012.

[12] F. Eichas and U. Zölzer, "Block-oriented modeling of distortion audio effects using iterative minimization," in *Proc. Digital Audio Effects (DAFx-15)*, Trondheim, Norway, Nov 30 – Dec 3 2015.

[13] "Linear Technologies SPICE Simulator," Website, Available at `http://http://www.linear.com/designtools/software/` — accessed March 17th 2016.

[14] D.T. Yeh, J.S. Abel, and J.O. Smith, "Simulation of the diode limiter in guitar distortion circuits by numerical solution of ordinary differential equations," in *Proc. Digital Audio Effects (DAFx-07)*, Bordeaux, France, Sep. 10 – 15 2007.

[15] D.T. Yeh, *Digital implementation of musical distortion circuits by analysis and simulation*, Ph.D. thesis, Stanford University, 2009.

[16] D.T. Yeh, J.S. Abel, and J.O. Smith, "Simplified, physically-informed models of distortion and overdrive guitar effects pedals," in *Proc. Digital Audio Effects (DAFx-07)*, Bordeaux, France, Sep. 10 – 15 2007.

[17] R.C.D. Paiva, S. D'Angelo, J. Pakarinen, and V. Välimäki, "Emulation of operational amplifiers and diodes in audio distortion circuits," in *IEEE Trans. on Circuits and Systems*, Bordeaux, France, Oct. 2012, vol. 59.

[18] K.J. Werner, V. Nangia, A. Bernardini, J.O. Smith, and A. Sarti, "An improved and generalized diode clipper model for wave digital filters," in *Audio Engineering Society Convention 139*. Audio Engineering Society, 2015.

[19] A. Farina, "Simultaneous measurement of impulse response and distortion with a swept-sine technique," in *Audio Engineering Society Convention 108*, Paris, France, 2000.

[20] J. Pakarinen and D.T. Yeh, "A review of digital techniques for modeling vacuum-tube guitar amplifiers," *Computer Music Journal*, vol. 33, no. 2, pp. 85–100, 2009.

[21] K. Levenberg, "A method for the solution of certain problems in least squares," *Quarterly of applied mathematics*, vol. 2, pp. 164–168, 1944.

[22] D.W. Marquardt, "An algorithm for least-squares estimation of nonlinear parameters," *Journal of the Society for Industrial & Applied Mathematics*, vol. 11, no. 2, pp. 431–441, 1963.

[23] T. Strutz, *Data Fitting and Uncertainty: A practical introduction to weighted least squares and beyond*, Vieweg and Teubner, 2010.

[24] "Listening Examples," Website, Available at `http://www.hermelinband.de/felix/dafx16/audio_examples_local_version.html` — accessed March 16th 2016.