Acoust. Sci. & Tech. **42**, 6 (2021)

PAPER

WaveNet modeling of distortion pedal using spectral features

Kento Yoshimoto*, Hiroki Kuroda†, Daichi Kitahara‡ and Akira Hirabayashi§

Graduate School of Information Science and Engineering, Ritsumeikan University, 1–1–1 Noji-higashi, Kusatsu, 525–8577 Japan

(*Received 25 February 2021, Accepted for publication 24 June 2021*)

Abstract: The present paper proposes a distortion pedal modeling method using the so-called WaveNet. A state-of-the-art method constructs a feedforward network by modifying the original autoregressive WaveNet, and trains it so that a loss function defined by the normalized mean squared error between the high-pass filtered outputs is minimized. This method works well for pedals with low distortion, but not for those with high distortion. To solve this problem, the proposed method exploits the same WaveNet, but a novel loss function, which is defined by a weighted sum of errors in time and time-frequency (T-F) domains. The error in the time domain is defined by the mean squared error without the high-pass filtering, while that in the T-F domain is defined by a divergence between spectral features computed from the short-time Fourier transform. Numerical experiments using a pedal with high distortion, the Ibanez SD9, show that the proposed method is capable of precisely reproducing high-frequency components without attenuation of low-frequency components compared to the state-of-the-art method.

Keywords: Distortion pedal, Black-box modeling, WaveNet, Loss function, Spectral features

1. INTRODUCTION

Electric guitars are used in connection with amplifiers through effects pedals. These devices not only produce a loud sound, but also put various effects on guitar sound, including distortion. Guitar players carefully choose particular ones from many different types of effects pedals and amplifiers to create their intended tone. Note that certain sounds can be produced only by ones so-called "vintage," whose production have already finished. Those products are in great demand and hard to obtain. To reproduce the sounds of such devices, digital modeling is required. Once digital modeling is completed, players do not need to carry around many heavy devices. Digital modeling is also useful to put favorite effects pedals into digital audio workstations (DAWs).

Digital modeling techniques can be classified into two types. Methods of the first type convert all electronic circuits in the devices into mathematical models [1,2]. This circuit-based approach is capable of generating high-quality sounds. Such modelings, however, require not only the circuit diagram of the target device but also characteristic curves of all nonlinear circuit parts including transistors, diodes, and vacuum tubes. Even worse, if the circuit diagram is not available, huge efforts for reverse engineering are required.

Methods of the second type exploit machine learning or system identification [3–12]. Using pairs of clean input and distorted output sounds of the target device, the mapping from the input to output is acquired. The cost to collect such data is much lower than that for the circuit-based approach, as long as the target device is available.

Methods of the second type are further classified into two subgroups. Those in the first subgroup are called block-oriented models [3–7]. Electronic circuits in the device typically consist of a linear filtering block followed by a nonlinear clipping block. The block-oriented models use this two-block structure. The pairs of the clean input and distorted output sounds of the target device are used to adjust parameters in the blocks. However, since distortion effects modeling is a problem of a nonlinear system identification, such a two-block structure reduces the model flexibility and causes degradation of reproduction quality.

Methods in the second subgroup use no information about the electronic circuits in the device. Instead, these methods exploit deep learning [8–12] and are called black-box modeling. Along this context, recurrent neural networks (RNNs) were first exploited. In particular, long short-term memory (LSTM) networks [13], which are one

*e-mail: is0383ip@ed.ritsumei.ac.jp
†e-mail: kuroda@media.ritsumei.ac.jp
‡e-mail: d-kita@media.ritsumei.ac.jp
§e-mail: akirahrb@media.ritsumei.ac.jp
[doi:10.1250/ast.42.305]

of the RNNs, were used in [8] and [9]. Even though these methods are capable of high-accuracy modeling, it takes a long time to train the networks due to their recursive structures.

For faster training, Damskägg *et al.* proposed a modeling method based on a feedforward variant of WaveNet [10–12], which was originally proposed to synthesize audio waveforms, including human voice and music, using a nonlinear autoregressive structure [14]. WaveNet does not use recursive structures and hence training is fast. Further, the so-called dilated causal convolution enables WaveNet to reproduce high-quality sounds with low computational cost. The modified WaveNet is trained so as to minimize the error-to-signal ratio (ESR) loss function, defined by the normalized mean squared error between high-pass filtered target and modeling sounds in the time domain. This method achieved better results with faster training than the LSTM methods. Nevertheless, low-frequency components are attenuated by the side effect of the high-pass filter. In particular, pedals with high distortion are not well modeled by this method in our experiments.

To solve the above problem, we propose a novel modeling method, in which the same modified WaveNet is used as in [11]. On the other hand, we exploit a novel loss function, which is defined by a weighted sum of errors in the time and time-frequency (T-F) domains.[1] The error in the time domain is the mean squared error without the high-pass filtering to avoid the attenuation of the low-frequency components. The error in the T-F domain is defined by a divergence between spectral features computed from the short-time Fourier transform (STFT). Numerical experiments using a pedal with high distortion, the Ibanez SD9, show that the proposed method reproduces the high-frequency components precisely without the attenuation of the low-frequency components compared with the conventional method [11].

The differences from the preliminary version [16] of this paper include (i) extensive simulations on the effect of the tuning parameter which controls the importance of the time and T-F domain errors and (ii) examination of the performance of the proposed approach with different divergences employed in the T-F domain error.

2. WAVENET FOR DISTORTION MODELING

Let $x[n]$ and $y[n]$ respectively be the input and the output signals of the distortion pedal at discrete time instant $n \in \{1, 2, \ldots, N\}$. As a black-box model, we adopt a state-of-the-art deep neural network (DNN) model based on WaveNet [10–12]. WaveNet was originally proposed in [14] as an autoregressive model which predicts a future sample from past samples, and was modified in [10–12] as a feedforward model that computes an output signal from input signals.

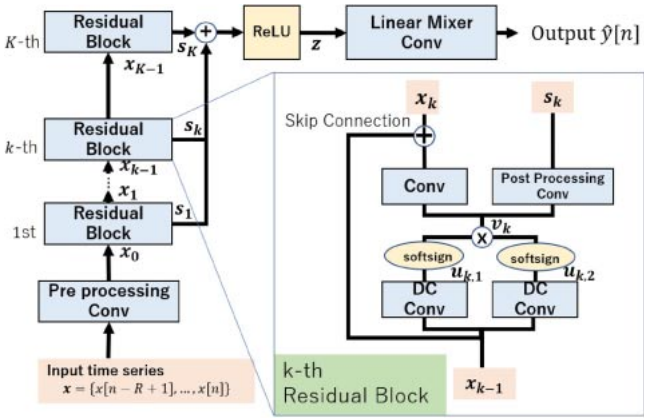


Fig. 1 Neural network model for distortion pedal.

The network structure is shown in Fig. 1, which consists of a pre-processing layer, K residual blocks, and a linear mixer after a ReLU conversion. The pre-processing layer converts the single-channel input signal $x[n]$ to an L-channel signal as

$$\boldsymbol{x}_0[n] = \boldsymbol{w}_0 x[n] + \boldsymbol{b}_0, \tag{1}$$

where $\boldsymbol{w}_0 \in \mathbb{R}^L$ represents a convolutional filter[2] of size 1, and $\boldsymbol{b}_0 \in \mathbb{R}^L$ is a bias term.

Then, the L-channel signal $\boldsymbol{x}_0[n]$ is passed to the residual blocks connected in sequence. The structure of each residual block is zoomed in a bottom right box in Fig. 1. For $k = 1, 2, \ldots, K$, the kth residual block first computes two dilated causal (DC) convolutions, as

$$\begin{cases} \boldsymbol{u}_{k,1}[n] = \sum_{m=0}^{M} W_{k,1}[m]\boldsymbol{x}_{k-1}[n - md_k] + \boldsymbol{b}_{k,1}, \\ \boldsymbol{u}_{k,2}[n] = \sum_{m=0}^{M} W_{k,2}[m]\boldsymbol{x}_{k-1}[n - md_k] + \boldsymbol{b}_{k,2}, \end{cases} \tag{2}$$

where $W_{k,1}[m], W_{k,2}[m] \in \mathbb{R}^{L \times L}$ $(m = 0, 1, \ldots, M)$ are convolutional filters of size $M + 1$, d_k is the dilation factor, and $\boldsymbol{b}_{k,1}, \boldsymbol{b}_{k,2} \in \mathbb{R}^L$ are bias terms. DC convolution is employed to enlarge the number, denoted by R, of input signals used by the network with keeping low computational complexity. The dilation factor is doubled as k increases and reset to 1 when it exceeds 256, i.e., $(d_1, d_2, \ldots . d_9, d_{10}, d_{11}, \ldots) = (1, 2, \ldots, 256, 1, 2, \ldots)$. Since R is given by $M(\sum_{k=1}^{K} d_k) + 1$, we can enlarge the number R of inputs while keeping the filter size $M + 1$ small.

[1] For the speech synthesis, the effectiveness of the use of time-frequency features was presented in [15].

[2] In this paper, L convolutional filters are collectively represented by a single vector or matrix.

After the DC convolutional layers, the vector $\boldsymbol{v}_k[n]$ is computed by

$$\boldsymbol{v}_k[n] = g(\boldsymbol{u}_{k,1}[n]) \odot g(\boldsymbol{u}_{k,2}[n]), \tag{3}$$

where $\odot$ denotes the component-wise multiplication, and g is the component-wise soft-sign activation function $u/(1+|u|)$ [11]. This part is called the gated activation unit because one of the term in the right-hand side of (3) denotes the flow of the signal while the other term can be regarded as a gate that controlls the signal flow [17].

The output $\boldsymbol{x}_k[n]$ of the kth residual block is obtained by mixing the output of the activation unit $\boldsymbol{v}_k[n]$ and the input $\boldsymbol{x}_{k-1}[n]$ to this block, as

$$\boldsymbol{x}_k[n] = W_{k,3}\boldsymbol{v}_k[n] + \boldsymbol{b}_{k,3} + \boldsymbol{x}_{k-1}[n], \tag{4}$$

where $W_{k,3} \in \mathbb{R}^{L\times L}$ is a convolutional filter of size 1, and $\boldsymbol{b}_{k,3} \in \mathbb{R}^L$ is a bias term. Another output $\boldsymbol{s}_k[n]$ is computed by

$$\boldsymbol{s}_k[n] = W_{k,4}\boldsymbol{v}_k[n] + \boldsymbol{b}_{k,4}, \tag{5}$$

where $W_{k,4} \in \mathbb{R}^{L\times L}$ and $\boldsymbol{b}_{k,4} \in \mathbb{R}^L$. Note that the Kth residual block computes only $\boldsymbol{s}_K[n]$, but not $\boldsymbol{x}_K[n]$.

The outputs $\boldsymbol{s}_1[n], \boldsymbol{s}_2[n], \ldots, \boldsymbol{s}_K[n]$ of the residual blocks are merged through the skip connections, and converted by rectified linear unit (ReLU), as

$$\boldsymbol{z}[n] = \mathrm{ReLU}\left(\sum_{k=1}^{K} \boldsymbol{s}_k[n]\right), \tag{6}$$

where $\mathrm{ReLU}(\boldsymbol{s})$ computes $\max(0, s)$ with s the component of $\boldsymbol{s}$. Finally, a single-channel modeling sound $\hat{y}[n]$ is obtained by

$$\hat{y}[n] = \boldsymbol{w}_{K+1}^{\top}\boldsymbol{z}[n], \tag{7}$$

where $\boldsymbol{w}_{K+1} \in \mathbb{R}^L$. Equation (7) completes the network computation for a time instance n.

The network has totally $2K(L^2(M+2)+2L) - L^2 + 2L$ parameters. These parameters are adjusted so that $\hat{y}[n]$ approximates $y[n]$ well under some loss function. The existing methods [10–12] use the error-to-signal ratio (ESR) for high-pass filtered target and modeling sounds, defined by

$$\mathrm{ESR} = \frac{\sum_{n=1}^{N}(\hat{y}_{\mathrm{f}}[n] - y_{\mathrm{f}}[n])^2}{\sum_{n=1}^{N} y_{\mathrm{f}}[n]^2}, \tag{8}$$

where $\hat{y}_{\mathrm{f}}[n]$ and $y_{\mathrm{f}}[n]$ are filtered modeling sounds and target sounds with a high-pass filter $H(z) = 1 - 0.95z^{-1}$ (see Fig. 2 for the frequency response of $H(z)$). Because of the high-pass filtering, the network trained by this strategy tends to disregard low-frequency components.

3. LOSS FUNCTION USING SPECTRAL FEATURES

To reproduce the high-frequency components faithfully without sacrificing the accuracy for the low-frequency components, we propose to combine the time-frequency (T-F) domain error with the time domain error. As spectral features, we use the power spectrogram or the mel-frequency power spectrogram. The mel-frequency power spectrogram is employed because it is known to be effective in the related fields, e.g., speech processing [18], due to the hearing characteristics.

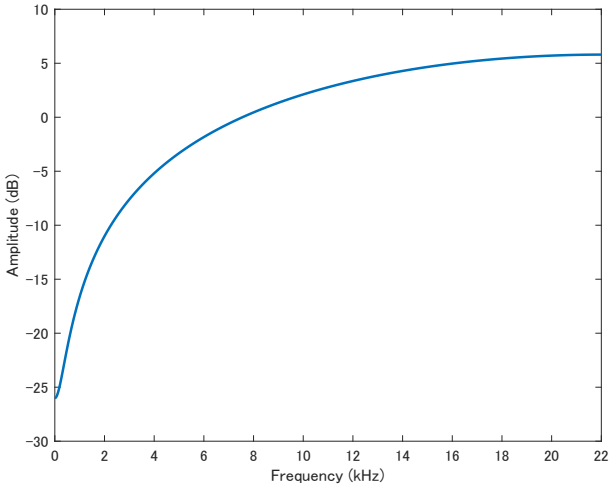


Fig. 2 Frequency response of $H(z)$ (sampling frequency 44.1 kHz).

The power spectrograms $\boldsymbol{Y}_{\mathrm{pow}} \in \mathbb{R}_+^{I\times J}$ and $\hat{\boldsymbol{Y}}_{\mathrm{pow}} \in \mathbb{R}_+^{I\times J}$ are computed from the target sound $y[n]$ and the modeling sound $\hat{y}[n]$ by the short-time Fourier transform (STFT), respectively, where $\mathbb{R}_+$ denotes the set of all nonnegative real numbers, I is the number of frequency bins, and J is the number of time frames. Specifically, we compute the target power spectrogram $\boldsymbol{Y}_{\mathrm{pow}} = (Y_{i,j}^{\mathrm{pow}}) \in \mathbb{R}_+^{I\times J}$ from the target sounds $y[1], y[2], \ldots, y[N]$ with the STFT, as

$$Y_{i,j}^{\mathrm{pow}} = \left|\sum_{n=1}^{N_{\mathrm{f}}} \psi[n]y[(j-1)\tau + n]e^{-2\sqrt{-1}\pi\frac{(i-1)(n-1)}{N_{\mathrm{f}}}}\right|^2, \tag{9}$$

for $i = 1, 2, \ldots, I$ and $j = 1, 2, \ldots, J$, where $\psi[n]$ is a window function, τ is the frame shift, and N_{f} is the frame length. Note that the number of frequencies is given as $I = \lceil (N_{\mathrm{f}}+1)/2 \rceil$ and the number of time frames is given as $J = \lceil N/\tau \rceil$, where $\lceil\cdot\rceil$ is the ceiling function, and zero padding is used for the outside parts $y[N+1], y[N+2], \ldots, y[(J-1)\tau + N_{\mathrm{f}}]$. The modeling power spectrogram $\hat{\boldsymbol{Y}}_{\mathrm{pow}}$ is also computed from the modeling sounds $\hat{y}[1], \hat{y}[2], \ldots, \hat{y}[N]$ in the same way.

The target mel-frequency power spectrogram $\boldsymbol{Y}_{\mathrm{mel}} = (Y_{b,j}^{\mathrm{mel}}) \in \mathbb{R}_+^{\tilde{I}\times J}$ is computed from $\boldsymbol{Y}_{\mathrm{pow}}$ by using a mel filter bank as

$$Y_{b,j}^{\mathrm{mel}} = \sum_{i=1}^{I} H_b[i]Y_{i,j}^{\mathrm{pow}}, \tag{10}$$

for $b = 1, 2, \ldots, \tilde{I}$ and $j = 1, 2, \ldots, J$, where $H_b[i]$ is the

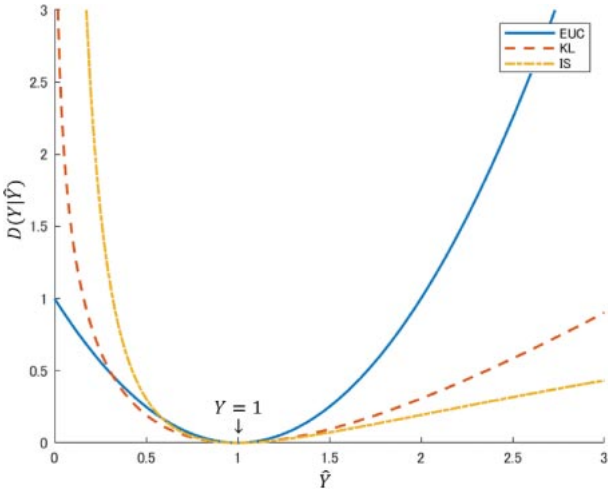


Fig. 3 Three error measures for spectrograms.

mel-scale band-pass filter [18, Section 6.5.2]. In the same way, the modeling mel-frequency power spectrogram $\hat{\boldsymbol{Y}}_{\text{mel}}$ is computed from $\hat{\boldsymbol{Y}}_{\text{pow}}$ with the same mel filter bank. In simulations of Sect. 4, we use the frequency band from 60 Hz to 22 kHz, which is divided into $\tilde{I} = 300$ bins.

The error between two spectrograms $\boldsymbol{Y} = (Y_{i,j})$ and $\hat{\boldsymbol{Y}} = (\hat{Y}_{i,j})$ is normally measured by the Euclidean distance, generalized Kullback–Leibler (KL) divergence, or Itakura–Saito (IS) divergence, which are defined by

$$\text{EUC}(\boldsymbol{Y}\|\hat{\boldsymbol{Y}}) = \frac{1}{IJ}\sum_{i=1}^{I}\sum_{j=1}^{J}(\hat{Y}_{i,j} - Y_{i,j})^2, \tag{11}$$

$$\text{KL}(\boldsymbol{Y}\|\hat{\boldsymbol{Y}}) = \frac{1}{IJ}\sum_{i=1}^{I}\sum_{j=1}^{J}\left(Y_{i,j}\log\frac{Y_{i,j}}{\hat{Y}_{i,j}} - Y_{i,j} + \hat{Y}_{i,j}\right), \tag{12}$$

or

$$\text{IS}(\boldsymbol{Y}\|\hat{\boldsymbol{Y}}) = \frac{1}{IJ}\sum_{i=1}^{I}\sum_{j=1}^{J}\left(\frac{Y_{i,j}}{\hat{Y}_{i,j}} - \log\frac{Y_{i,j}}{\hat{Y}_{i,j}} - 1\right), \tag{13}$$

respectively. For the case of $I = J = 1$, the values of these functions in terms of $\hat{Y} > 0$ with $Y = 1$ are shown in Fig. 3. The Euclidean distance is symmetric with respect to $Y = 1$. On the other hand, the generalized KL and IS divergences are asymmetric and penalize more for $\hat{Y} < 0.3$ and less for $\hat{Y} > 1$ than the Euclidean distance. Using one of these metrics, we are going to define the following loss function:

$$\text{Loss}(\boldsymbol{\vartheta}) = l_{\text{time}}(\boldsymbol{\vartheta}) + \lambda l_{\text{freq}}(\boldsymbol{\vartheta}), \tag{14}$$

where $\boldsymbol{\vartheta}$ denotes the adjustable parameters of the network,

$$l_{\text{time}}(\boldsymbol{\vartheta}) = \frac{1}{N}\sum_{n=1}^{N}(\hat{y}[n] - y[n])^2, \tag{15}$$

and $l_{\text{freq}}(\boldsymbol{\vartheta})$ is one of Eqs. (11)–(13). The parameter $\lambda > 0$ controls the relative importance of the time domain waveform and the spectral features. In the loss function, the time-frequency (T-F) domain term $l_{\text{freq}}(\boldsymbol{\vartheta})$ evaluates only the power of the spectrogram, but not the phase, while the time domain term $l_{\text{time}}(\boldsymbol{\vartheta})$ compensates it. Taking the balance between them by λ, the proposed loss function appropriately evaluates the errors between the network outputs and the target sounds. The proposed approach uses the spectral features in the loss function for the training, but the network is kept in the time domain. Since the low latency is important for real-time playing, this approach is suitable for the distortion pedal modeling, while methods completely in the T-F domain, e.g., [19], have a delay due to the T-F conversion.

4. NUMERICAL EXPERIMENTS

4.1. Experimental Setup

We used a high distortion pedal, the Ibanez SD9, in this experiment. In our preliminary experiment, we judged the Ibanez SD9 as the high distortion pedal through the comparison with the Ibanez Tube Screamer. The Ibanez SD9 has three knobs, each of which controls level of distortion, tone, and volume. They were set to the direction of 12 o'clock, or the middle position.

The modified WaveNet structure was set as follows: channel number $L = 16$, residual block number $K = 18$, and filter size $M + 1 = 3$, thus $M = 2$. Hence, we have $2K(L^2(M + 2) + 2L) - L^2 + 2L = 37{,}792$ adjustable parameters. Further, R is given by $M(\sum_{k=1}^{K} d_k) + 1 = 2{,}045$, which approximately corresponds to 46.4 ms when the sampling frequency is 44.1 kHz.

The training process was implemented with Tensorflow 2.3.0 in Python 3.8.5 on Windows 10 Pro, Intel Core i9-7980X, 128 GB main memory, GeForce GTX1080Ti GPU.

4.2. Training and Test Data

For training data, we exploited the IDMT dataset [20,21], where the sampling frequency is 44.1 kHz and the bit depth is 16 bits. A total of 300 s of data (150 s of guitar sounds and 150 s of bass sounds) were randomly selected from the dataset. They were used as clean input signals and sent to the pedal through a reamper (Radial ProRMP [22]) to generate the corresponding distorted output signals. These data were trimmed at every 100 ms so that $D = 3{,}000$ subgroups $\{\boldsymbol{t}_{\text{train}}^{(1)}, \boldsymbol{t}_{\text{train}}^{(2)}, \ldots, \boldsymbol{t}_{\text{train}}^{(D)}\}$ of $N = 4{,}410$ input and output sequences were obtained. Each subgroup $\boldsymbol{t}_{\text{train}}^{(d)}$ consists of N elements $\{t_{\text{train}}^{(d,1)}, t_{\text{train}}^{(d,2)}, \ldots, t_{\text{train}}^{(d,N)}\}$, where $t_{\text{train}}^{(d,n)}$ consists of a single output value $y[n]$ and R input values $x[n - R + 1], x[n - R + 2], \ldots, x[n]$, from which $\hat{y}[n]$ is computed. For $n < R$, zeros were filled into $x[n - R + 1], x[n - R + 2], \ldots, x[0]$.

For each subgroup $\boldsymbol{t}_{\text{train}}^{(d)}$, we compute the spectral features used in the proposed loss function as follows. From the target sounds $y[1], y[2], \ldots, y[N]$ in $\boldsymbol{t}_{\text{train}}^{(d)}$, we

compute the target power spectrogram $\boldsymbol{Y}_{\mathrm{pow}} \in \mathbb{R}_{+}^{I \times J}$ as (9), where we set the frame shift to $\tau = 256$, and the frame length to $N_{\mathrm{f}} = 1{,}024$. Note that this setting implies that the number I of frequency bins is 513 and the number J of time frames is 18. In (9), we use the hann window

$$\psi[n] = \frac{1}{2} - \frac{1}{2}\cos\left(2\pi \frac{n-1}{N_{\mathrm{f}}-1}\right) \quad (n = 1, 2, \ldots, N_{\mathrm{f}}). \quad (16)$$

Similarly, we obtain the modeling power spectrogram $\hat{\boldsymbol{Y}}_{\mathrm{pow}}$ from the modeling sounds $\hat{y}[1], \hat{y}[2], \ldots, \hat{y}[N]$ computed from the input values in $\boldsymbol{t}_{\mathrm{train}}^{(d)}$. The mel-frequency power spectrograms $\boldsymbol{Y}_{\mathrm{mel}}$ and $\hat{\boldsymbol{Y}}_{\mathrm{mel}}$ are computed as (10).

Since the amount of data is huge ($DN \approx 1.3 \times 10^7$), it is not appropriate to compute the gradient of the loss function for the overall training data $\{\boldsymbol{t}_{\mathrm{train}}^{(1)}, \boldsymbol{t}_{\mathrm{train}}^{(2)}, \ldots, \boldsymbol{t}_{\mathrm{train}}^{(D)}\}$. Thus, we utilize the so-called mini-batch training. The overall training data is randomly divided into $P = 16$ mini-batch data $\{\boldsymbol{t}_{\mathrm{train}}^{(d_p[1])}, \boldsymbol{t}_{\mathrm{train}}^{(d_p[2])}, \ldots, \boldsymbol{t}_{\mathrm{train}}^{(d_p[D_p])}\}$ $(p = 1, 2, \ldots, P)$, where $d_p[1], \ldots, d_p[D_p]$ indicate the randomly selected subgroups in the pth mini-batch data, and $D_p = D/P$. The gradient of the loss function is computed for each mini-batch data. We say that an "epoch" is completed when all of P mini-batch data are used for training. An iterative optimization algorithm, Adam [23], is exploited to minimize the loss function. The iteration is repeated for 1,000 epochs, or until an early stopping condition is met. The early stopping condition is checked for other two 300 subgroups of guitar and bass sounds randomly selected from the IDMT dataset.

To validate whether the trained model can accurately reproduce the sounds which have characteristics different from the IDMT dataset used for the training, we prepared original four sound sources for evaluation.[3] We recorded the sound sources played on the guitar, *Fender JM66*, using the audio interface, *Focusrite scarlett 6i6*. In sound source 1 (S1), a $\mathrm{B}\flat_{\mathrm{add9}}$ chord was played with strong attacks. In sound source 2 (S2), a $\mathrm{D}_{\mathrm{sus2}}$ chord was played with weak attacks. In sound source 3 (S3), a chromatic scale from E2 to A♭3 was played with strong attacks. In sound source 4 (S4), two tones of D3 and D4 were played with weak attacks.

4.3. Experimental Results

The proposed methods using the power spectrogram with the Euclidean distance, generalized KL divergence, and IS divergence are referred to as PS-EUC, PS-KL, and PS-IS, respectively. Further, the proposed methods using the mel-frequency power spectrogram with the Euclidean distance, generalized KL divergence, and IS divergence are called MFS-EUC, MFS-KL, and MFS-IS, respectively. The conventional method that uses only the mean square error in the time domain without the high-pass filter $H(z)$ is called Method MSE. Table 1 shows the averages of quality measures including ESRs with and without high-pass filter $H(z)$ and the normalized mean square error (NMSE) of the power spectrogram obtained by the method MFS-KL for various values of λ. We can see that the two ESRs are the best when λ is 10^{-3} while NMSE is so when λ is 10^{-1}. Since the ESRs change more than NMSE, we adopt $\lambda = 10^{-3}$ for MFS-KL. Through similar experiments, we determined λ for MFS-EUC, MFS-IS, PS-EUC, PS-KL,

[3]You have access to the sound sources from our web site [24].

Table 1 Averages of quality measures [%] for four sound sources obtained by MFS-KL.

λ	10^{-1}	10^{-2}	10^{-3}	10^{-4}	0
ESR	0.91	0.93	**0.84**	0.95	1.14
ESR (high-pass)	10.33	10.29	**9.32**	10.74	12.13
NMSE (PS)	**0.11**	0.13	0.14	0.15	0.17

Table 2 ESR [%] without the high-pass filter.

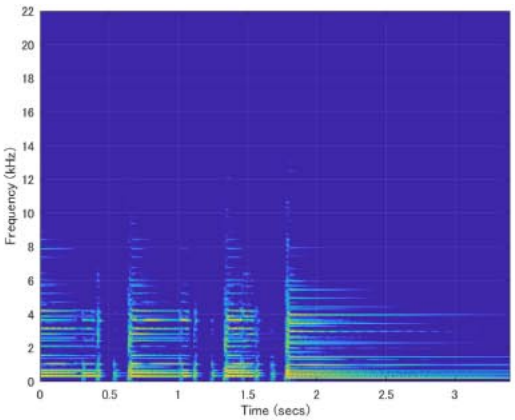
Method (λ)	S1	S2	S3	S4	Ave.
PS-EUC (10^{-6})	**1.98**	**0.55**	**0.45**	**0.27**	**0.81**
PS-KL (10^{-5})	2.23	0.63	0.54	0.30	0.92
PS-IS (10^{-7})	2.39	0.59	0.48	0.29	0.94
MFS-EUC (10^{-7})	2.51	0.74	0.60	0.35	1.05
MFS-KL (10^{-3})	2.07	0.58	0.46	**0.27**	0.84
MFS-IS (10^{-4})	2.11	0.57	0.49	0.29	0.87
MSE (0)	2.90	0.75	0.50	0.40	1.14
Conv. [11]	2.52	1.19	1.58	0.87	1.54

Table 3 ESR [%] with the high-pass filter.

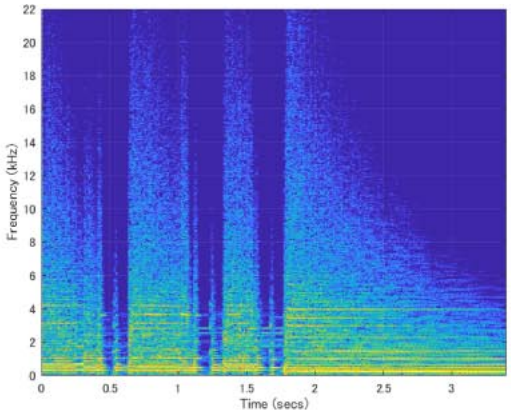
Method (λ)	S1	S2	S3	S4	Ave.
PS-EUC (10^{-6})	18.40	7.38	4.31	7.33	9.36
PS-KL (10^{-5})	18.79	7.84	4.08	7.63	9.59
PS-IS (10^{-7})	21.08	8.37	4.17	7.29	10.23
MFS-EUC (10^{-7})	22.96	8.84	4.97	8.24	11.25
MFS-KL (10^{-3})	18.68	7.65	**4.03**	**6.91**	9.32
MFS-IS (10^{-4})	18.52	7.70	4.38	7.35	9.49
MSE (0)	26.06	9.57	4.49	8.40	12.13
Conv. [11]	**16.86**	**7.29**	5.03	7.15	**9.08**

Table 4 NMSE of the power spectrogram.

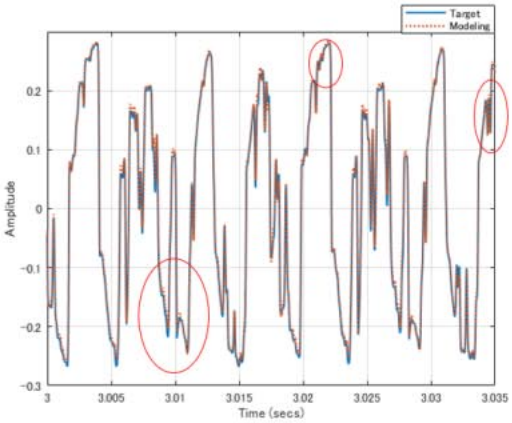
Method (λ)	S1	S2	S3	S4	Ave.
PS-EUC (10^{-6})	0.23	0.10	0.15	0.07	**0.14**
PS-KL (10^{-5})	0.27	0.12	0.17	**0.04**	0.15
PS-IS (10^{-7})	0.30	0.10	**0.15**	0.06	0.15
MFS-EUC (10^{-7})	0.36	0.17	0.19	**0.10**	0.20
MFS-KL (10^{-3})	**0.22**	**0.09**	0.18	0.05	**0.14**
MFS-IS (10^{-4})	0.24	0.10	0.16	0.06	**0.14**
MSE (0)	0.35	0.14	0.16	0.06	0.17
Conv. [11]	0.77	0.50	1.01	0.42	0.68



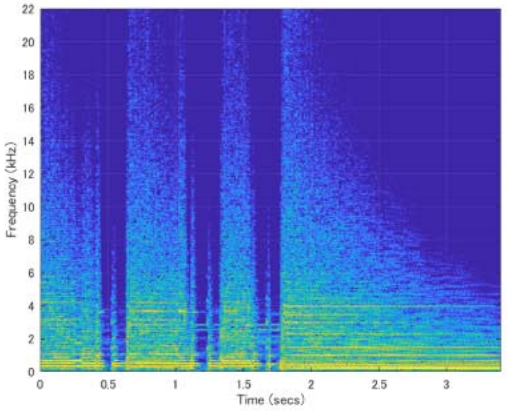
(a) Power spectrogram of the input sound $x[n]$.



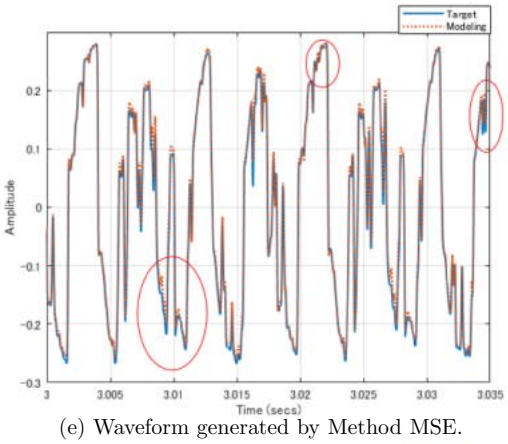
(b) Power spectrogram of the target sound $y[n]$.



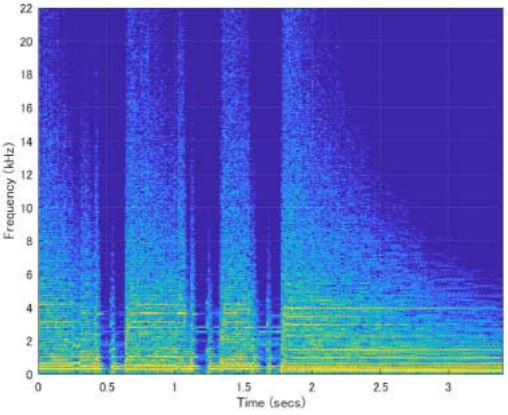
(c) Waveform generated by the proposed method MFS-KL.



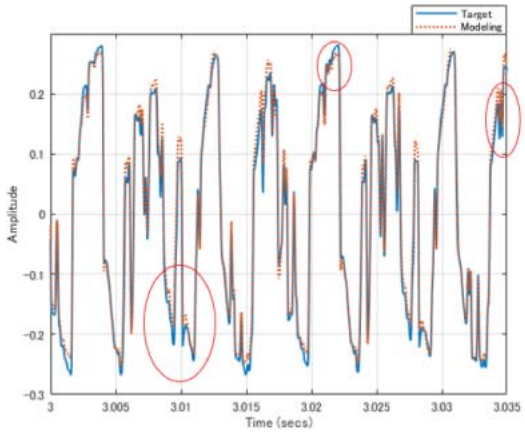
(d) Power spectrogram of the waveform in (c).



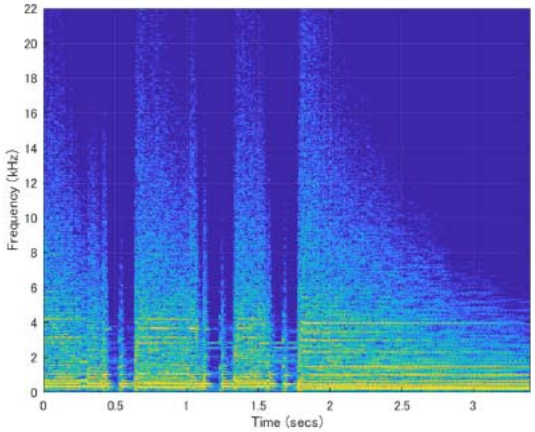
(e) Waveform generated by Method MSE.



(f) Power spectrogram of the waveform in (e).



(g) Waveform generated by the conventional method [11].



(h) Power spectrogram of the waveform in (g).

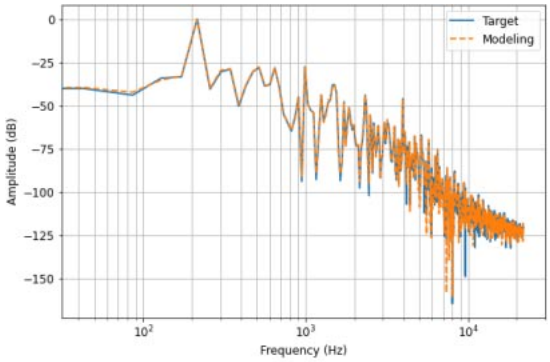
Fig. 4 Simulation results for the Ibanez SD9 with sound source 1.

and PS-IS as 10^{-7}, 10^{-4}, 10^{-6}, 10^{-5}, and 10^{-7}, respectively, as shown in Tables 2, 3 and 4. Note that when $\lambda = 0$, all the proposed methods reduce to Method MSE. The average of each quality measure is always better when λ is not equal to zero than when λ is zero, as shown in Table 1. This confirms the effectiveness of the addition of the spectral features to the loss function in the proposed methods.

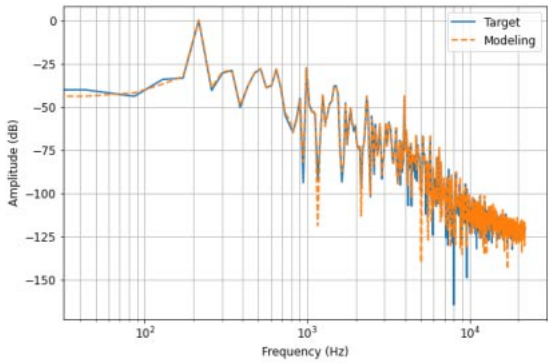
We compare the ESR in (8) for each method and each sound source. ESR without the high-pass filter is shown in Table 2. We can see that PS-EUC and MFS-KL improved the ESR of the conventional method in [11] by 47.4% and 45.4% in average. ESR with the high-pass filter is shown in Table 3. Since the conventional method minimizes this measure, it shows the best value in average. Nevertheless, the measure obtained by PS-EUC and MFS-KL indicated mostly same value as the conventional method. Table 4 shows NMSE of each method. We can see that all of PS-EUC, MFS-KL and MFS-IS improved NMSE by 79.4%. These results confirm the effectiveness of the proposed methods.

Figure 4 shows simulation results for sound source 1. Figures 4(a) and 4(b) are the power spectrograms of the clean input sound and the distorted target sound. The waveform generated by the proposed method MFS-KL is indicated in Fig. 4(c) by a red dotted line with the target waveform indicated by a blue solid line. Figure 4(d) shows the corresponding power spectrogram. Figure 4(e) shows the waveform generated by Method MSE with red as well as the target waveform with blue. Figure 4(f) shows the corresponding power spectrogram. Figures 4(g) and 4(h) indicates the waveform generated by the conventional method in [11] and the corresponding power spectrogram, respectively. By comparing the parts indicated by the circles in Figs. 4(c), 4(e) and 4(g), we can see that the proposed method MFS-KL reproduced the target sound more accurately than Method MSE and the method in [11]. Even though the differences of the overall spectrograms in Figs. 4(d), 4(f) and 4(h) are not clear, from the power spectra shown in Figs. 5(a), 5(b) and 5(c) which correspond to the waveforms shown in Figs. 4(c), 4(e) and 4(g) respectively, we can see that the proposed method MFS-KL improved the modeling accuracy of low-frequency components while keeping the accuracy of high-frequency components. We also noticed by listening to the results that MFS-KL reproduces the original sound more accurately than the conventional method [11] and Method MSE in particular for the attack parts, i.e., the beginnings of the sounds.

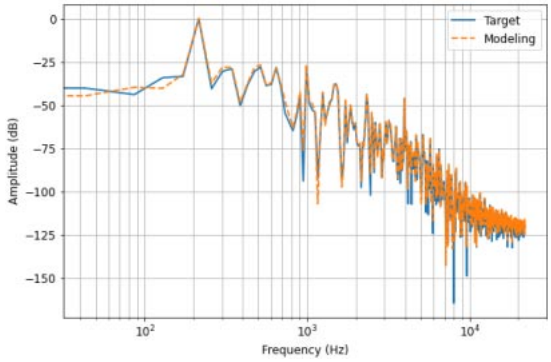
For the real-time performances, we implemented the trained model using Tensorflow 2.3.0 in Python 3.8.5 on an Apple MacBook Pro with Intel Core i7 3.5 GHz processor. We used a processing buffer of 512 samples, where the sampling frequency is 44.1 kHz. This setting implies that the 512 samples should be processed approximately within 11.6 ms. We confirmed that our model satisfied this condition, as the average processing time is 10.5 ms for 10,000 trials of processing sounds.



(a) Power spectrum generated by the proposed method MFS-KL.



(b) Power spectrum generated by Method MSE.



(c) Power spectrum generated by the conventional method [11].

Fig. 5 Simulation results for the Ibanez SD9 with the power spectra of sound source 1.

5. CONCLUSIONS

This paper proposed a method for distortion pedal modeling based on the modified WaveNet. We adopted the same network structure as the conventional method [11]. To train the network, we proposed a novel loss function,

which is defined by a weighted sum of errors in the time and time-frequency (T-F) domains. The error in time domain is the mean squared error without high-pass filtering. The error in the T-F domain is defined by a divergence between spectral features computed from the short-time Fourier transform of target and modeling sounds. Numerical experiments using a pedal with high distortion, the Ibanez SD9, showed that the proposed method reproduced high-quality sounds more than the conventional method.

ACKNOWLEDGMENT

The authors would like to thank the anonymous reviewers for their helpful comments on the original version of the manuscript.

REFERENCES

[1] D. T. Yeh, J. Abel and J. O. Smith, "Simulation of the diode limiter in guitar distortion circuits by numerical solution of ordinary differential equations," *Proc. Int. Conf. Digital Audio Effects* (*DAFx*), Bordeaux, France, pp. 197–203 (2007).

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

[3] A. Novak, L. Simon, P. Lotton and J. Gilbert, "Chebyshev model and synchronized swept sine method in nonlinear audio effect modeling," *Proc. Int. Conf. Digital Audio Effects* (*DAFx*), Graz, Austria, 4 pages (2010).

[4] 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," *Proc. AES Int. Conf. Appl. Time-Freq. Process. Audio*, Helsinki, Finland, 10 pages (2012).

[5] F. Eichas and U. Zölzer, "Black-box modeling of distortion circuits with block-oriented models," *Proc. Int. Conf. Digital Audio Effects* (*DAFx*), Brno, Czech Republic, pp. 39–45 (2016).

[6] F. Eichas, S. Möller and U. Zölzer, "Block-oriented gray box modeling of guitar amplifiers," *Proc. Int. Conf. Digital Audio Effects* (*DAFx*), Edinburgh, UK, pp. 184–191 (2017).

[7] F. Eichas, *System Identification of Nonlinear Audio Circuits*, Ph.D. thesis, Helmut Schmidt University (2019).

[8] 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," *Proc. IEEE SoutheastCon*, St. Petersburg, FL, USA, 5 pages (2018).

[9] Y. Matsunaga, N. Aoki, Y. Dobashi and T. Yamamoto, "A digital modeling technique for distortion effect based on a machine learning approach," *Proc. Asia-Pac. Signal Inf. Process. Assoc. Annu. Summit Conf.* (*APSIPA ASC*), Honolulu, HI, USA, pp. 1888–1892 (2018).

[10] E.-P. Damskägg, L. Juvela and V. Välimäki, "Deep learning for tube amplifier emulation," *Proc. IEEE Int. Conf. Acoust. Speech Signal Process.* (*ICASSP*) *2019*, Brighton, UK, pp. 471–475 (2019).

[11] E.-P. Damskägg, L. Juvela and V. Välimäki, "Real-time modeling of audio distortion circuits with deep learning," *Proc. Sound Music Comput. Conf.* (*SMC*), Málaga, Spain, pp. 332–339 (2019).

[12] A. Wright, E.-P. Damskägg, L. Juvela and V. Välimäki, "Real-time guitar amplifier emulation with deep learning," *Appl. Sci.*, **10**(3), 18 pages (2020).

[13] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, **9**, 1735–1780 (1997).

[14] A. van den Oord, S. Dieleman, H. Zen, K. Simonyan, O. Vinyals, A. Graves, N. Kalchbrenner, A. Senior and K. Kavukcuoglu, "WaveNet: A generative model for raw audio," *arXiv:1609.03499*, 15 pages (2016).

[15] R. Yamamoto, E. Song and J. Kim, "Parallel WaveGAN: A fast waveform generation model based on generative adversarial networks with multi-resolution spectrogram," *Proc. IEEE Int. Conf. Acoust. Speech Signal Process.* (*ICASSP*) *2020*, Barcelona, Spain, pp. 6199–6203 (2020).

[16] K. Yoshimoto, H. Kuroda, D. Kitahara and A. Hirabayashi, "Deep neural network modeling of distortion stomp box using spectral features," *Proc. Asia-Pac. Signal Inf. Process. Assoc. Annu. Summit Conf.* (*APSIPA ASC*), Auckland, New Zealand, pp. 339–345 (2020).

[17] Y. N. Dauphin, A. Fan, M. Auli and D. Grangier, "Language modeling with gated convolutional networks," *arXiv: 1612.08083*, Sep. (2017).

[18] X. Huang, A. Acero and H.-W. Hon, *Spoken Language Processing: A Guide to Theory, Algorithm, and System Development* (Prentice Hall, Upper Saddle River, NJ, 2001).

[19] S. Ö. Arık, H. Jun and G. Diamos, "Fast spectrogram inversion using multi-head convolutional neural networks," *IEEE Signal Process. Lett.*, **26**, 94–98 (2019).

[20] https://www.idmt.fraunhofer.de/en/business_units/m2d/smt/guitar.html (accessed 20 Jan. 2020).

[21] https://www.idmt.fraunhofer.de/en/business_units/m2d/smt/bass_lines.html (accessed 20 Jan. 2020).

[22] https://www.radialeng.com/product/prormp (accessed 29 Jun. 2020).

[23] D. P. Kingma and J. L. Ba, "Adam: A method for stochastic optimization," *Proc. Int. Conf. Learn. Represent.* (*ICLR*), San Diego, CA, USA, 15 pages (2015).

[24] https://mediasensinghp.github.io/wavenet_modeling/sound_source.html (accessed 24 Feb. 2021).

Kento Yoshimoto received his B.E. degrees in computer science in 2019 from the Ritsumeikan University, Shiga, Japan. His research interest includes musical audio signal processing.

Hiroki Kuroda received his B.E. degree in computer science, and his M.E. and Ph.D. degrees in information and communications engineering from the Tokyo Institute of Technology, Tokyo, Japan, in 2013, 2015 and 2019, respectively. Currently, he is an Assistant Professor with the College of Information Science and Engineering, Ritsumeikan University, Shiga, Japan. His current research interests include signal processing and its applications, nonlinear inverse problem, and sparse optimization.

Daichi Kitahara received his B.E. degree in computer science in 2012, and his M.E. and Ph.D. degrees in communications and computer engineering in 2014 and 2017 from the Tokyo Institute of Technology, Tokyo, Japan, respectively. Since 2017, he has been an Assistant Professor with the College of Information Science and Engineering, Ritsumeikan University, Shiga, Japan. His current research interest includes signal processing and its applications, inverse problem, optimization theory, and multivariate spline theory.

Akira Hirabayashi received his D.E. degrees in computer science in 1999, from the Tokyo Institute of Technology, Tokyo, Japan. From 2013, he is a Professor of the College of Information Science and Engineering, Ritsumeikan University, Shiga, Japan. His research interests include compressed sensing, sampling theory, and signal and image processing. Prof. Hirabayashi is a member of the Acoustical Society of Japan (ASJ), the Institute of Electronics, Information and Communication Engineers (IEICE) and the Institute of Electrical and Electronics Engineers (IEEE).