Knowledge-and-Data-Driven Amplitude Spectrum Prediction for Hierarchical Neural Vocoders

Yang Ai, Zhen-Hua Ling

National Engineering Laboratory for Speech and Language Information Processing
University of Science and Technology of China, Hefei, P.R.China

ay8067@mail.ustc.edu.cn, zhling@ustc.edu.cn

Abstract

In our previous work, we have proposed a neural vocoder called HiNet which recovers speech waveforms by predicting amplitude and phase spectra hierarchically from input acoustic features. In HiNet, the amplitude spectrum predictor (ASP) predicts log amplitude spectra (LAS) from input acoustic features. This paper proposes a novel knowledge-and-data-driven ASP (KDD-ASP) to improve the conventional one. First, acoustic features (i.e., F0 and mel-cepstra) pass through a knowledge-driven LAS recovery module to obtain approximate LAS (ALAS). This module is designed based on the combination of STFT and source-filter theory, in which the source part and the filter part are designed based on input F0 and mel-cepstra, respectively. Then, the recovered ALAS are processed by a data-driven LAS refinement module which consists of multiple trainable convolutional layers to get the final LAS. Experimental results show that the HiNet vocoder using KDD-ASP can achieve higher quality of synthetic speech than that using conventional ASP and the WaveRNN vocoder on a text-to-speech (TTS) task.

Index Terms: neural vocoder, log amplitude spectrum, source-filter, TTS

1. Introduction

Nowadays, statistical parametric speech synthesis (SPSS) has become a popular text-to-speech (TTS) approach thanks to its flexibility and high quality. Both acoustic models which predict acoustic features (e.g., mel-cepstra and F0) from texts and vocoders [1] which reconstruct speech waveforms from predicted acoustic features are essential in SPSS. Early SPSS systems preferred to adopt conventional vocoders, such as STRAIGHT [2] and WORLD [3] as their vocoders. These vocoders are designed based on the source-filter model of speech production [4] and have some limitations, such as the loss of phase information and spectral details.

Recently, some autoregressive neural generative models such as WaveNet [5], SampleRNN [6] and WaveRNN [7] have been proposed and achieved good performance on generating raw audio signals. Their variants such as knowledge-distilling-based models (e.g., parallel WaveNet [8] and ClariNet [9]) and flow-based models (e.g., WaveGlow [10]) were also proposed to further improve the performance and generation efficiency. Based on these waveform generation models, neural vocoders have been developed [11–16], which reconstruct speech waveforms from various acoustic features for SPSS, voice conversion [17, 18], bandwidth extension [19], etc. Although these neural vocoders outperformed the conventional ones significantly, they still have some limitations. The autoregressive neural vocoders have low generation efficiency due to their point-by-point generation process. For knowledge-distilling-based vocoders and flow-based vocoders, it is difficult to train them due to their complicated training process and high complexity of model structures respectively.

Subsequently, some improved neural vocoders, such as glottal neural vocoder [20,21], LPCNet [22], and neural source-filter (NSF) vocoder [23–26], have been further proposed. These vocoders combine speech production mechanisms with neural networks and have also demonstrated impressive performance. In our previous work [27], we proposed a neural vocoder named HiNet, which consists of an amplitude spectrum predictor (ASP) and a phase spectrum predictor (PSP). HiNet produces speech waveforms by first predicting amplitude spectra from input acoustic features using ASP and then predicting phase spectra from amplitude spectra using PSP. The outputs of ASP and PSP are combined to recover speech waveforms by short-time Fourier synthesis (STFS). Besides, generative adversarial networks (GANs) [28] are also introduced into ASP and PSP to further improve their performance. Experimental results show that the proposed HiNet vocoder can generate waveforms with high quality and high efficiency.

In this paper, we propose a novel knowledge-and-data-driven ASP (KDD-ASP) to replace the conventional one in a HiNet vocoder. The aim of KDD-ASP is to integrate speech production and analysis knowledge into data-driven LAS prediction, expecting to improve the accuracy and generalization ability of ASP, especially when predicted acoustic features are used as input. KDD-ASP consists of a knowledge-driven LAS recovery module and a data-driven LAS refinement module. The first module is designed based on the combination of STFT and the source-filter theory of speech production, and generates approximate LAS (ALAS) from input acoustic features (i.e., F0 and mel-cepstra). We assume that the speech signal is produced via a source-filter process [4]. The source excitation signal and the filter are designed according to the input F0 and mel-cepstra respectively. Then, ALAS can be calculated by imitating the process of STFT which includes truncation, windowing and FFT. All operations are performed in the frequency domain. The second module predicts the final LAS from ALAS. This module consists of multiple trainable convolutional layers and is trained in a data-driven way. Experimental results confirm that the HiNet vocoder using KDD-ASP can achieve higher quality of synthetic speech than that using conventional ASP and the WaveRNN vocoder on a TTS task.

This paper is organized as follows. In Section 2, we briefly review the HiNet vocoder [27]. In Section 3, we describe the details of our proposed KDD-ASP. Section 4 reports our experimental results. Conclusions are given in Section 5.

This work was partially funded by the National Key R&D Program of China under Grant 2019YFF0303001 and the National Nature Science Foundation of China under Grant 61871358 and U1613211.

2. HiNet vocoder

HiNet [27] is a novel neural vocoder which recovers speech waveforms by predicting amplitude and phase spectra hierarchically from input acoustic features. Conventional neural vocoders usually employ single neural networks to generate speech waveforms directly. In contrast, the HiNet vocoder consists of an amplitude spectrum predictor (ASP) and a phase spectrum predictor (PSP). ASP uses acoustic features as input and predicts frame-level log amplitude spectra (LAS). Then PSP uses the predicted LAS and F0 as input and recovers the phase spectra. Finally, the outputs of ASP and PSP are combined to recover speech waveforms by short-time Fourier synthesis (STFS).

In our implement, ASP is a simple non-autoregressive DNN containing multiple feed-forward (FF) layers. It concatenates the acoustic features at current and previous frames as input to predict the LAS at current frame. At the training stage, the target LAS are extracted from natural waveforms by STFT. A GAN criterion is adopted to build ASP. The DNN model is used as the generator of GAN and its discriminator consists of multiple convolutional layers which operate along the frequency axis of the input LAS. A Wasserstein GAN [29] loss is combined with the mean square error (MSE) between the predicted LAS and natural ones to train the generator.

PSP is constructed by concatenating a neural waveform generator with a phase spectrum extractor. The neural waveform generator is built by adapting the NSF vocoder [23] from three aspects, 1) using LAS as the input, 2) pre-calculating the initial phase of the sine-based excitation signal for each voiced segment at the training stage and 3) adopting a combined loss function including MSE on amplitude spectra, waveform loss and correlation loss. GAN is also introduced into PSP. Here, the neural waveform generator of PSP is used as the generator of GAN and its discriminator is similar with that of ASP except that its input features are waveforms instead of LAS.

3. Knowledge-and-Data-Driven ASP

This paper proposes a novel knowledge-and-data-driven ASP (KDD-ASP) to replace the conventional one in a HiNet vocoder. The KDD-ASP is constructed by concatenating an LAS recovery module which refers to the knowledge of source-filtering speech production with an LAS refinement module which is trained using a corpus in a data-driven way as shown in Fig. 1.

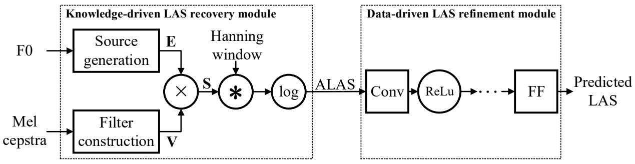

Figure 1: *Model structure of KDD-ASP. Here,* $\times$, $*$ *and log denote element-wise product, convolutional and log operation respectively,* FF *and* Conv *represent feed-forward and convolutional layers respectively and* ReLu *means rectified linear units.*

3.1. Knowledge-driven LAS recovery module

The equation for extracting LAS directly from a signal $\boldsymbol{s}$ by STFT can be written as follows,

$$\boldsymbol{LAS}_n = \log|\mathcal{F}(\boldsymbol{s}_n \odot \boldsymbol{w})|, \tag{1}$$

where $\boldsymbol{s}_n = [s_{n,1}, \ldots, s_{n,L}]^\top$ and $\boldsymbol{LAS}_n = [LAS_{n,1} \ldots, LAS_{n,K}]^\top$ are the framed signal of $\boldsymbol{s}$ and the LAS at the n-th frame respectively, and $\boldsymbol{w} = [w_1, \ldots, w_L]^\top$ denotes the Hanning window for short-time analysis. L is the frame number. $K = \frac{FN}{2} + 1$ represents the number of frequency bins and FN is the FFT point number. $\odot$ and $\mathcal{F}$ represent element-wise product and FFT, respectively.

Inspired by this process, the knowledge-driven LAS recovery module constructs approximate LAS (ALAS) from F0 and mel-cepstra based on the frequency-domain representation of Eq. (1). We assume that the speech signal at the n-th frame $\boldsymbol{s}_n$ is obtained by the convolution between a source excitation signal $\boldsymbol{e}_n$ and a filter impulse response $\boldsymbol{v}_n$. In frequency domain, this process can be represented as

$$\boldsymbol{S}_n = \boldsymbol{E}_n \odot \boldsymbol{V}_n, \tag{2}$$

where $\boldsymbol{S}_n = [S_{n,1}, \ldots, S_{n,K}]^\top$, $\boldsymbol{E}_n = [E_{n,1}, \ldots, E_{n,K}]^\top$ and $\boldsymbol{V}_n = [V_{n,1}, \ldots, V_{n,K}]^\top$ are the Fourier transform of $\boldsymbol{s}_n$, $\boldsymbol{e}_n$ and $\boldsymbol{v}_n$ respectively.

Let f_n denote the F0 value of the n-th frame when it is voiced and $f_n = 0$ when the frame is unvoiced. For voiced frames ($f_n > 0$), $\boldsymbol{E}_n$ is produced as a pulse train with equal frequency interval $K_0 = Round(\frac{f_n}{F_s} \cdot FN)$, which corresponds to constructing all the harmonics below the Nyquist frequency, where F_s is the sampling rate. For unvoiced frames ($f_n = 0$), we set $E_n \equiv 1$, meaning that the excitation signal is a Gaussian white noise. The equation for producing $\boldsymbol{E}_n$ based on F0 values can be written as

$$E_{n,k} = \begin{cases} 1, & f_n > 0, k = i \cdot K_0 \\ 0, & f_n > 0, k \neq i \cdot K_0 \text{ or } f_n = 0 \end{cases}, \tag{3}$$

where $i = 1, 2, \ldots, \lceil \frac{K}{K_0} \rceil$.

$\boldsymbol{V}_n$ is calculated by transforming mel-cepstra to amplitude spectra [30]. The mel-cepstral coefficients at the n-th frame (with energy as the first order) are first padded with zeros to form a K-dimensional vector $\boldsymbol{m}_n = [m_{n,1}, \ldots, m_{n,K}]^\top$. Then, the cepstral coefficients $c_{n,k}, k = 1, \ldots, K$ are calculated by the following iterative formulas

$$c_{n,k}(i) = \begin{cases} m_{n,i} - \alpha \cdot c_{n,1}(i+1), & k=1 \\ (1-\alpha^2) \cdot c_{n,1}(i+1) - \alpha \cdot c_{n,2}(i+1), & k=2 \\ c_{n,k-1}(i+1) - \alpha \cdot [c_{n,k}(i+1) - c_{n,k-1}(i)], & k>2 \end{cases}, \tag{4}$$

where i iterates from K to 1 with the initial value $c_{n,k}(K+1) = 0, k = 1, \ldots, K$. α is the mel-frequency warping coefficient, which is 0.42 for $F_s = 16000$. After the iteration, we can obtain the cepstra vector $\boldsymbol{c}_n = [c_{n,1}(1), \ldots, c_{n,K}(1)]^\top$, which is further transformed to the amplitude spectra $\boldsymbol{V}_n$ by

$$\boldsymbol{V}_n = \exp[\mathcal{F}(\boldsymbol{c}_n)]. \tag{5}$$

Finally, ALAS can be calculated as

$$\boldsymbol{ALAS}_n = \log|\boldsymbol{S}_n * \boldsymbol{W}|, \tag{6}$$

where $\boldsymbol{ALAS}_n = [ALAS_{n,1} \ldots, ALAS_{n,K}]^\top$ is the n-th frame ALAS and $\boldsymbol{W} = [W_1, \ldots, W_K]^\top$ is the Fourier transform of the analysis window $\boldsymbol{w}$. The operation $*$ represents convolution. It is worth mentioning that the elements in the vectors of $\boldsymbol{S}_n$ and $\boldsymbol{W}$ should be rearranged by complementing their mirror-symmetric parts and shifting the zero-frequency component to the center before convolution.

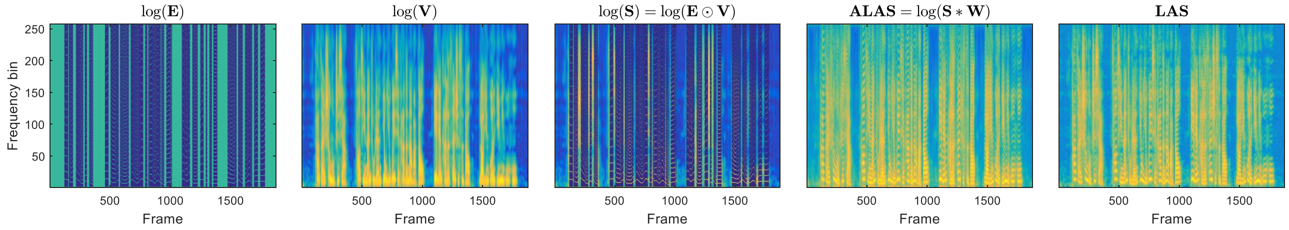

Figure 2: *The visualization of* $\log(\boldsymbol{E}_n)$*,* $\log(\boldsymbol{V}_n)$*,* $\log(\boldsymbol{S}_n)$*,* $\boldsymbol{ALAS}_n$ *and* $\boldsymbol{LAS}_n$ *for an example utterance. Here, the input F0 and mel-cepstra are natural ones.*

3.2. Data-driven LAS refinement module

The data-driven LAS refinement module converts ALAS to final LAS by a trainable neural network. In our implement, this module adopts the ASP model in Section 2 but has two structural improvements. First, convolutional layers are used instead of FF layers in the generator and the input is the ALAS at current frame instead of the concatenated ones as shown in Fig. 1. Second, another discriminator which operates along with the time axis of the input LAS is added[1].

4. Experiments

4.1. Experimental conditions

A Chinese speech synthesis corpus with 13334 utterances ($\sim$20 hours) was used in our experiments. The speaker was a female and the waveforms had 16 kHz sampling rate with 16 bits resolution. The training, validation and test sets contained 13134, 100 and 100 utterances, respectively. The natural acoustic features were extracted with a frame length and shift of 25 ms and 5 ms respectively. The acoustic features at each frame were 43-dimensional including 40-dimensional mel-cepstra, an energy, an F0 and a V/UV flag. For SPSS, a bidirectional LSTM-RNN acoustic model [31] having 2 hidden layers with 1024 units per layer (512 forward units and 512 backward units) was trained as the acoustic model, which predicted acoustic features from 566-dimensional linguistic features. The output of the acoustic model was 127-dimensional including 43-dimensional acoustic features together with their delta and acceleration counterparts (the V/UV flag had no dynamic components). Then, the predicted acoustic features were generated from the output by maximum likelihood parameter generation (MLPG) [32] considering global variance (GV) [33]. Since this paper focuses on vocoders, natural durations obtained by HMM-based forced alignment were used at synthesis time.

Three vocoders were compared in our experiments[2]. The descriptions of these vocoders are as follows.

1) ***WaveRNN*** A 16-bit WaveRNN-based neural vocoder using acoustic features as input. This vocoder was implemented by ourselves and the efficiency optimization strategies [7] were not adopted here. Its structure was the same as ***WaveRNN*** in our previous work [27] which performed better than the 16-bit WaveNet vocoder using open source implementation[3]. The waveform samples were quantized to discrete values by 16-bit linear quantization and the model had one hidden layer with 1024 nodes where 512 nodes for coarse outputs and another 512 nodes for fine outputs. Models were trained and evaluated on a single Nvidia 1080Ti GPU using TensorFlow [34].

2) ***HiNet*** A HiNet vocoder using conventional ASP. The structure of ASP is the same with that of the data-driven LAS refinement module introduced in Section 3.2. When extracting natural LAS, the frame length and frame shift of STFT were 20ms (i.e., $L = 320$) and 5ms respectively and FFT point number was 512 (i.e., $K = 257$). There were 3 convolutional layers with 2048 nodes per layer (filter width=7), and a 257-dimensional linear output layer which predicted the LAS. For each training step, ASP used 128 frames of acoustic features as input and outputted corresponding 128 frames of LAS. GANs were also used in ASP. Discriminator #1 operated along with the frequency axis and consisted of 6 convolutional layers (filter width=9, stride size=2) and their channels were 16, 32, 64, 128 and 256 respectively. The resulting dimensions per layer, being it frequency bins × channels, were 257×1, 129×16, 65×32, 33×64, 17×128 and 9×256. Finally, two FF layers with 256 and 9 nodes respectively were used to map the 9×256 convolutional results into a value for loss calculation. Discriminator #2 operated along with the time axis and consisted of 4 convolutional layers (filter width=9, stride size=2) and their channels were 64, 128, 256 and 512 respectively. The resulting dimensions per layer, being it frequency bins × channels, were 128×257, 64×64, 32×128, 16×256 and 8×512. Finally, two FF layers with 512 and 8 nodes respectively were used to map the 8×512 convolutional results into a value for loss calculation. Remaining settings of ASP and all the settings of PSP are the same as the ***HiNet-S-GAN*** vocoder in our previous work [27]. ASP and PSP models were both trained and evaluated on a single Nvidia 1080Ti GPU using TensorFlow framework [34].

3) ***HiNet-KDD*** A HiNet vocoder using the KDD-ASP proposed in this paper. For KDD-ASP, the knowledge-driven LAS generation module adopted the same settings with that of extracting natural LAS (i.e., $L = 320$ and $K = 257$) and the settings of the data-driven LAS refinement module were the same as the ASP of ***HiNet***. The settings of PSP and the implementation conditions were all the same as that of ***HiNet***. Fig. 2 shows the visualization of $\boldsymbol{E}_n$, $\boldsymbol{V}_n$, $\boldsymbol{S}_n$, $\boldsymbol{ALAS}_n$ and $\boldsymbol{LAS}_n$ for all frames in an example utterance. We can see that the recovered ALAS is close to the reference LAS with analogous harmonic and formant structures, meaning that the input and output of the data-driven LAS refinement module are similar, expecting to facilitate the model learning and to improve the performance of predicting amplitude spectra.

4.2. Objective evaluation

We first compared the performance of these three vocoders using objective evaluations. Five objective metrics used in our

[1] Discriminators are not shown in Fig. 1 for simplification.

[2] Examples of generated speech can be found at `http://home.ustc.edu.cn/~ay8067/Interspeech2020/demo.html`.

[3] `https://github.com/r9y9/wavenet_vocoder`.

Table 1: *Objective evaluation results of* **WaveRNN**, **HiNet** *and* **HiNet-KDD** *on the test set. "AS" stands for analysis-synthesis task and "TTS" stands for TTS task.*

		WaveRNN	***HiNet***	***HiNet-KDD***
AS	SNR(dB)	4.6631	**5.2587**	5.0152
	LAS-RMSE(dB)	4.9623	**4.2602**	4.5659
	MCD-V(dB)	1.0702	**0.7686**	0.8583
	F0-RMSE(cent)	13.2365	9.3345	**9.0960**
	V/UV error(%)	4.2515	2.0116	**2.0041**
TTS	MCD-V(dB)	1.0702	1.0939	**0.9488**
	F0-RMSE(cent)	12.4645	7.0877	**6.4970**
	V/UV error(%)	3.5247	**1.7983**	2.0194

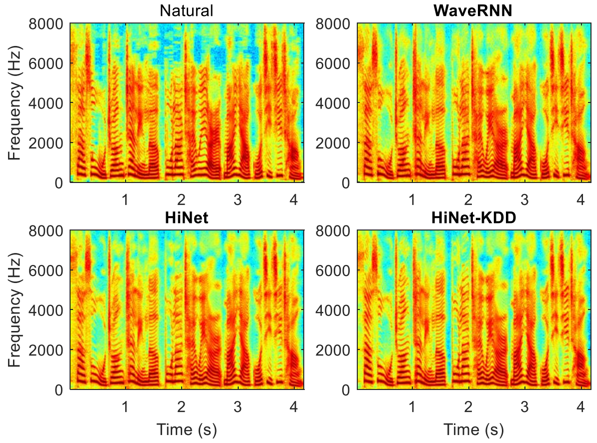

Figure 3: *The spectrograms of natural speech and the speech generated by* **WaveRNN**, **HiNet** *and* **HiNet-KDD** *on TTS task for an example sentence in the test set.*

previous work [27] were adopted here, including signal-to-noise ratio (SNR), root MSE (RMSE) of LAS (denoted by LAS-RMSE), mel-cepstrum distortion for voiced frames (denoted by MCD-V), MSE of F0 (denoted by F0-RMSE) and V/UV error. For the analysis-synthesis (AS) task, the references are natural waveforms or the acoustic features extracted from natural waveforms. For the TTS task, the references are the mel-cepstra and F0 predicted by the acoustic model and only MCD-V, F0-RMSE and V/UV error were adopted since the calculation of SNR and LAS-RMSE relied on natural speech waveforms.

The objective results on the test set are listed in Table 1. It is obvious that both ***HiNet*** and ***HiNet-KDD*** outperformed ***WaveRNN*** on most metrics for both AS and TTS tasks. By comparing ***HiNet*** and ***HiNet-KDD***, we can find that ***HiNet-KDD*** performed better on F0-RMSE than ***HiNet*** for both AS and TTS tasks, which indicated that ***HiNet-KDD*** is better at restoring harmonics for voiced frames. Considering the SNR, LAS-RMSE and MCD-V for AS task, ***HiNet-KDD*** was not as good as ***HiNet***. However, for TTS task, ***HiNet-KDD*** achieved better MCD-V than ***HiNet***. This advantage can be attributed to that using ALAS as the input to train the ASP model improves its generalization ability when dealing with unseen acoustic features. We also draw the spectrograms extracted from natural waveforms and from the waveforms generated by these three vocoders on TTS task in Fig. 3. We can see that ***HiNet-KDD*** can restore more clear harmonics (e.g., 0.7∼1.0s and 1.7∼2.0s) especially in the high-frequency band than the other two vocoders.

Table 2: *Average preference scores (%) on naturalness among different vocoders, where N/P stands for "no preference" and p denotes the p-value of a t-test between two vocoders. "AS" stands for analysis-synthesis task and "TTS" stands for TTS task.*

	WaveRNN	***HiNet***	***HiNet-KDD***	*N/P*	p
AS	2.73	**72.73**	–	24.54	< 0.01
	–	21.36	15.45	63.19	0.15
TTS	16.82	**57.27**	–	25.91	< 0.01
	10.91	–	**66.82**	22.27	< 0.01
	–	14.55	**53.64**	31.81	< 0.01

4.3. Subjective evaluation

Five groups of ABX preference tests were conducted to compare the subjective performance of different vocoders. In each subjective test, 20 utterances generated by two comparative vocoders were randomly selected from the test set. Each pair of generated speech were evaluated in random order. 11 Chinese native speakers were asked to judge which utterance in each pair had better naturalness or there was no preference. The p-value of a t-test was also calculated to measure the significance of the difference between two comparative vocoders.

The subjective results are shown in Table 2. We can see that ***HiNet*** outperformed ***WaveRNN*** very significantly ($p < 0.01$) on both AS and TTS tasks. However, the preference difference between these two vocoders became weaker on TTS task than on AS task. Comparing ***HiNet*** with ***HiNet-KDD***, we can see that there was no significant difference ($p > 0.05$) between these two vocoders on AS task but ***HiNet-KDD*** outperformed ***HiNet*** significantly ($p < 0.01$) on TTS task. We also conducted a group of ABX test between ***WaveRNN*** and ***HiNet-KDD*** for TTS task and ***HiNet-KDD*** also outperformed ***HiNet*** significantly ($p < 0.01$). Besides, the preference score difference between ***HiNet-KDD*** and ***WaveRNN*** was larger than that between ***HiNet*** and ***WaveRNN***. These results all indicated that using KDD-ASP in HiNet vocoder was helpful for improving the quality of reconstructed speech waveforms when the input acoustic features were predicted for TTS.

5. Conclusion

In this paper, we have proposed a novel knowledge-and-data-driven amplitude spectrum predictor (KDD-ASP) to replace the conventional one in HiNet, a hierarchical neural vocoder. KDD-ASP consists of a knowledge-driven LAS recovery module and a data-driven LAS refinement module. The first module is designed based on the combination of STFT and source-filter theories in order to convert F0 and mel-cepstra into approximate log amplitude spectra (ALAS). The input F0 values are used to produce the source signal and the filter part is calculated from mel-cepstra. The second module is a convolutional neural network which adopts GANs and predicts the final LAS from input ALAS. Experimental results show that the HiNet vocoder using KDD-ASP can achieve higher quality of synthetic speech than the HiNet vocoder using conventional ASP and the WaveRNN vocoder on a TTS task. To explore other knowledge-driven methods for ASP and further improve the performance of phase spectrum prediction will be the tasks of our future research.

6. References

[1] H. Dudley, "The vocoder," *Bell Labs Record*, vol. 18, no. 4, pp. 122–126, 1939.

[2] H. Kawahara, I. Masuda-Katsuse, and A. De Cheveigne, "Restructuring speech representations using a pitch-adaptive time–frequency smoothing and an instantaneous-frequency-based f0 extraction: Possible role of a repetitive structure in sounds," *Speech communication*, vol. 27, no. 3, pp. 187–207, 1999.

[3] M. Morise, F. Yokomori, and K. Ozawa, "WORLD: A vocoder-based high-quality speech synthesis system for real-time applications," *IEICE Transactions on Information and Systems*, vol. 99, no. 7, pp. 1877–1884, 2016.

[4] F. Gunnar, *The acoustic theory of speech production*. The Hague, The Netherlands: Mouton, 1960.

[5] A. v. d. 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," in *9th ISCA Speech Synthesis Workshop*, 2016, pp. 125–125.

[6] S. Mehri, K. Kumar, I. Gulrajani, R. Kumar, S. Jain, J. Sotelo, A. Courville, and Y. Bengio, "SampleRNN: An unconditional end-to-end neural audio generation model," in *Proc. ICLR*, 2017.

[7] N. Kalchbrenner, E. Elsen, K. Simonyan, S. Noury, N. Casagrande, E. Lockhart, F. Stimberg, A. Oord, S. Dieleman, and K. Kavukcuoglu, "Efficient neural audio synthesis," in *Proc. ICML*, 2018, pp. 2410–2419.

[8] A. v. d. Oord, Y. Li, I. Babuschkin, K. Simonyan, O. Vinyals, K. Kavukcuoglu, G. v. d. Driessche, E. Lockhart, L. C. Cobo, F. Stimberg *et al.*, "Parallel WaveNet: Fast high-fidelity speech synthesis," in *Proc. ICML*, 2018, pp. 3918–3926.

[9] W. Ping, K. Peng, and J. Chen, "ClariNet: Parallel wave generation in end-to-end text-to-speech," in *Proc. ICLR*, 2019.

[10] R. Prenger, R. Valle, and B. Catanzaro, "WaveGlow: A flow-based generative network for speech synthesis," in *Proc. ICASSP*, 2019, pp. 3617–3621.

[11] A. Tamamori, T. Hayashi, K. Kobayashi, K. Takeda, and T. Toda, "Speaker-dependent WaveNet vocoder," in *Proc. Interspeech*, 2017, pp. 1118–1122.

[12] T. Hayashi, A. Tamamori, K. Kobayashi, K. Takeda, and T. Toda, "An investigation of multi-speaker training for WaveNet vocoder," in *Proc. ASRU*, 2017, pp. 712–718.

[13] N. Adiga, V. Tsiaras, and Y. Stylianou, "On the use of WaveNet as a statistical vocoder," in *Proc. ICASSP*, 2018, pp. 5674–5678.

[14] Y. Ai, H.-C. Wu, and Z.-H. Ling, "SampleRNN-based neural vocoder for statistical parametric speech synthesis," in *Proc. ICASSP*, 2018, pp. 5659–5663.

[15] Y. Ai, J.-X. Zhang, L. Chen, and Z.-H. Ling, "DNN-based spectral enhancement for neural waveform generators with low-bit quantization," in *Proc. ICASSP*, 2019, pp. 7025–7029.

[16] J. Lorenzo-Trueba, T. Drugman, J. Latorre, T. Merritt, B. Putrycz, R. Barra-Chicote, A. Moinet, and V. Aggarwal, "Towards achieving robust universal neural vocoding," in *Proc. Interspeech*, 2019, pp. 181–185.

[17] L.-J. Liu, Z.-H. Ling, Y. Jiang, M. Zhou, and L.-R. Dai, "WaveNet vocoder with limited training data for voice conversion," in *Proc. Interspeech*, 2018, pp. 1983–1987.

[18] K. Kobayashi, T. Hayashi, A. Tamamori, and T. Toda, "Statistical voice conversion with WaveNet-based waveform generation," in *Proc. Interspeech*, 2017, pp. 1138–1142.

[19] Z.-H. Ling, Y. Ai, Y. Gu, and L.-R. Dai, "Waveform modeling and generation using hierarchical recurrent neural networks for speech bandwidth extension," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 26, no. 5, pp. 883–894, 2018.

[20] Y. Cui, X. Wang, L. He, and F. K. Soong, "A new glottal neural vocoder for speech synthesis," in *Proc. Interspeech*, 2018, pp. 2017–2021.

[21] L. Juvela, V. Tsiaras, B. Bollepalli, M. Airaksinen, J. Yamagishi, and P. Alku, "Speaker-independent raw waveform model for glottal excitation," in *Proc. Interspeech*, 2018, pp. 2012–2016.

[22] J.-M. Valin and J. Skoglund, "LPCNet: Improving neural speech synthesis through linear prediction," in *Proc. ICASSP*, 2019, pp. 5891–5895.

[23] X. Wang, S. Takaki, and J. Yamagishi, "Neural source-filter-based waveform model for statistical parametric speech synthesis," in *Proc. ICASSP*, 2019, pp. 5916–5920.

[24] X. Wang and J. Yamagishi, "Neural harmonic-plus-noise waveform model with trainable maximum voice frequency for text-to-speech synthesis," in *Proc. SSW*, 2019, pp. 1–6.

[25] X. Wang, S. Takaki, and J. Yamagishi, "Neural source-filter waveform models for statistical parametric speech synthesis," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 28, pp. 402–415, 2019.

[26] Y. Zhao, X. Wang, L. Juvela, and J. Yamagishi, "Transferring neural speech waveform synthesizers to musical instrument sounds generation," in *Proc. ICASSP*, 2020, pp. 6269–6273.

[27] Y. Ai and Z.-H. Ling, "A neural vocoder with hierarchical generation of amplitude and phase spectra for statistical parametric speech synthesis," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 28, pp. 839–851, 2020.

[28] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Advances in neural information processing systems*, 2014, pp. 2672–2680.

[29] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, and A. C. Courville, "Improved training of Wasserstein GANs," in *Advances in neural information processing systems*, 2017, pp. 5767–5777.

[30] K. Tokuda, T. Kobayashi, T. Masuko, and S. Imai, "Mel-generalized cepstral analysis-a unified approach to speech spectral estimation," in *Proc. ICSLP*, 1994, pp. 1043–1046.

[31] Y. Fan, Y. Qian, F.-L. Xie, and F. K. Soong, "TTS synthesis with bidirectional LSTM based recurrent neural networks," in *Proc. Interspeech*, 2014, pp. 1964–1968.

[32] K. Tokuda, T. Yoshimura, T. Masuko, T. Kobayashi, and T. Kitamura, "Speech parameter generation algorithms for HMM-based speech synthesis," in *Proc. ICASSP*, vol. 3, 2000, pp. 1315–1318.

[33] T. Toda and K. Tokuda, "A speech parameter generation algorithm considering global variance for HMM-based speech synthesis," *IEICE Transactions on Information and Systems*, vol. 90, no. 5, pp. 816–824, 2007.

[34] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin *et al.*, "Tensorflow: Large-scale machine learning on heterogeneous distributed systems," 2015. [Online]. Available: http://download.tensorflow.org/paper/whitepaper2015.pdf