SAMPLERNN-BASED NEURAL VOCODER FOR STATISTICAL PARAMETRIC SPEECH SYNTHESIS

Yang Ai, Hong-Chuan Wu, Zhen-Hua Ling

National Engineering Laboratory for Speech and Language Information Processing
University of Science and Technology of China, Hefei, P.R.China
{ay8067,whc}@mail.ustc.edu.cn, zhling@ustc.edu.cn

ABSTRACT

This paper presents a SampleRNN-based neural vocoder for statistical parametric speech synthesis. This method utilizes a conditional SampleRNN model composed of a hierarchical structure of GRU layers and feed-forward layers to capture long-span dependencies between acoustic features and waveform sequences. Compared with conventional vocoders based on the source-filter model, our proposed vocoder is trained without assumptions derived from the prior knowledge of speech production and is able to provide a better modeling and recovery of phase information. Objective and subjective evaluations are conducted on two corpora. Experimental results suggested that our proposed vocoder can achieve higher quality of synthetic speech than the STRAIGHT vocoder and a WaveNet-based neural vocoder with similar run-time efficiency, no matter natural or predicted acoustic features are used as inputs.

Index Terms— SampleRNN, WaveNet, neural network, vocoder, statistical parametric speech synthesis

1. INTRODUCTION

Recently, speech synthesis technology [1, 2, 3] plays a more and more important role in people's daily life. A speech synthesis system with high intelligibility, naturalness and expressiveness is a goal pursued by speech synthesis researchers. In the early days, the approach of concatenative synthesis [4] was proposed and it can generate high-quality speech however its flexibility is limited due to the difficulty of constructing large corpus for unit selection. Later on, a new approach named statistical parametric speech synthesis (SPSS) was proposed, which provided a more flexible framework for speech synthesis by acoustic modeling and vocoder-based waveform generation. Hidden Markov models (HMMs) [1], deep neural networks (DNNs) [2], recurrent neural networks (RNNs) [3] and other deep learning models [5] have been applied to build the acoustic models for SPSS. Vocoders [6] also play an important role in SPSS. A vocoder is usually a digital filter which reconstructs speech waveforms from acoustic parameters. Its performance affects the quality of synthetic speech significantly.

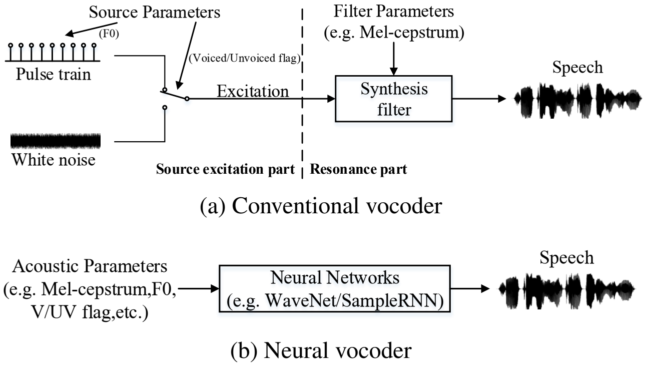
Various vocoders such as phase vocoder [7], channel vocoder [8] and spectral envelope estimation vocoder [9] have been proposed in previous work. More developed vocoders, such as STRAIGHT [10] and WORLD [11], have been popularly applied in current SPSS systems. All these existing vocoders are designed based on the source-filter model of voice production [12] as shown in Fig. 1(a). In the source excitation part, the generated excitation is usually a pulse train for voiced sounds and white noise for unvoiced sounds, according to the source parameters, such as F0 and V/UV flag. In the resonance part, the excitation signal passes through a synthesis filter which imitates the characteristics of vocal tract to generate the final speech waveforms. However, these vocoders still have some deficiencies. First, the source-filter model ignores the nonlinear effects in practical speech production. Second, representing vocal tract filter with low dimensional spectral features, such as mel-cepstra or line spectral pairs (LSP) leads to the loss of spectral details and phase information. These deficiencies constraint the performance of current vocoders and SPSS systems.

Recently, neural network-based speech waveform synthesizers, such as WaveNet [13] and SampleRNN [14], have been proposed and demonstrated impressive performance. In WaveNet [13], the distribution of each waveform sample conditioned on previous samples and additional conditions was represented using a neural network with dilated convolutional neural layers and residual architectures. Some variants such as fast WaveNet [15] and parallel WaveNet [16] were then proposed to improve the efficiency of generation. Different from WaveNet, SampleRNN [14] adopted recurrent neural layers with a hierarchical structure for unconditional audio generation. WaveNet-based speaker-dependent neural vocoders have been proposed and outperformed conventional vocoders, such as STRAIGHT [17, 18, 19]. However, building vocoders based on conditional SampleRNNs has not yet been thoroughly investigated.

Therefore, this paper presents a SampleRNN-based neural vocoder for SPSS. A conditional SampleRNN architecture composed of gated recurrent unit (GRU) layers and feed-forward (FF) layers is built to generate waveform sequences from input acoustic parameters. Different tiers in a conditional SampleRNN operate at different temporal resolution so as to efficiently capture long-span dependencies between input acoustic features and output waveform sequences. Similar to the WaveNet-based neural vocoder, our proposed vocoder is able to model the nonlinear effects during speech production without dependency on the conventional source-filter model, and to preserve the phase information of natural speech by using waveform samples directly for model training. Both of them can be considered as the instances of neural vocoders shown in Fig. 1(b). Experimental results show that our proposed vocoder can achieve higher quality of synthetic speech than STRAIGHT and a WaveNet-based neural vocoder with similar run-time efficiency.

This paper is organized as follows. In Section 2, we briefly review the basic unconditional SampleRNN model and describe the details of our proposed SampleRNN-based vocoder. Section 3 reports our experimental results. Conclusions are given in Section 4.

This work was partially funded by the Fundamental Research Funds for the Central Universities (Grant No. WK2350000001), the National Nature Science Foundation of China (Grant No. U1636201) and the Key Science and Technology Project of Anhui Province under Grant No. 17030901005.



(a) Conventional vocoder

(b) Neural vocoder

Fig. 1. The comparison between (a) a conventional vocoder and (b) a neural vocoder.

2. SAMPLERNN-BASED NEURAL VOCODER

2.1. Basic unconditional SampleRNN

An unconditional SampleRNN [14] is a waveform generator in an autoregressive manner which models the probability of a sequence of waveform samples $\boldsymbol{x} = [x_1, x_2, \ldots, x_T]$ as the product of the probabilities of each sample conditioned on all previous samples as

$$p(\boldsymbol{x}) = \prod_{t=1}^{T} p(x_t | x_1, \ldots, x_{t-1}). \tag{1}$$

The structure of a basic unconditional SampleRNN is composed of GRU layers and FF layers as shown by the solid lines in Fig. 2. These GRU and FF layers form a hierarchical structure of multiple tiers and each tier operates at a specific temporal resolution. The bottom tier (i.e. Tier 1 in Fig. 2) deals with individual samples and outputs sample-level predictions. Each higher tier operates on a lower temporal resolution (i.e. dealing with more samples per time step). Each tier conditions on the tier above it except the top tier.

Assume an unconditional SampleRNN has K tiers in total (e.g. $K = 3$ in Fig. 2). The k-th tier ($1 \leq k \leq K$) operates on *non-overlapping* frames composed of $L^{(k)}$ samples. In Fig. 2, we set $L^{(3)} = 8, L^{(2)} = 2$, and $L^{(1)} = 1$. The range of time step at the k-th tier, $t^{(k)}$, is determined by $L^{(k)}$. Denoting the input waveforms as $\boldsymbol{x} = [x_1, x_2, \ldots, x_T]$ and assuming that L represents the sequence length of $\boldsymbol{x}$ after zero-padding so that L is divisible by $L^{(K)}$, we can get

$$t^{(k)} \in T^{(k)} = \{1, 2, \ldots, \frac{L}{L^{(k)}}\}, 1 \leq k \leq K, \tag{2}$$

Furthermore, the relationship of temporal resolution between the m-th tier and the n-th tier ($1 \leq m < n \leq K$) can be described as

$$T^{(n)} = \{t^{(n)} | t^{(n)} = \lceil \frac{t^{(m)}}{L^{(n)}/L^{(m)}} \rceil, t^{(m)} \in T^{(m)}\}, \tag{3}$$

where $\lceil \cdot \rceil$ represents the operation of rounding up. It can be observed from (3) that one time step of the n-th tier corresponds to $L^{(n)}/L^{(m)}$ time steps of the m-th tier.

The frame input at the k-th tier ($1 < k \leq K$) and the t-th time step can be written by framing operation as

$$\boldsymbol{f}_t^{(k)} = [x_{L^{(K)}+(t-2)L^{(k)}+1}, \ldots, x_{L^{(K)}+(t-1)L^{(k)}}], t \in T^{(k)}. \tag{4}$$

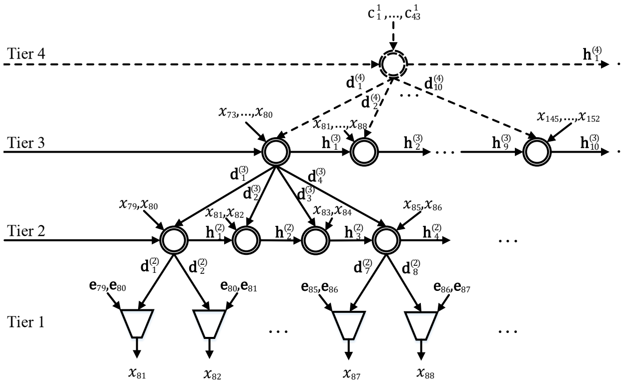


Fig. 2. The structure of a SampleRNN-based neural vocoder, where concentric circles represent GRU layers and inverted trapezoids represent FF layers. The solid lines represent the basic unconditional SampleRNN structure [14] and the dotted lines represent the conditional tier added for building a SampleRNN-based neural vocoder.

Particularly, for the bottom tier (i.e. $k = 1$), the individual sample x_t is first mapped into a real-valued vector $\boldsymbol{e}_t$ by an embedding layer and then used as the frame input. The *non-overlapping* frame size $L^{(1)}$ is set to 1 fixedly and the frame input is

$$\boldsymbol{f}_t^{(1)} = [\boldsymbol{e}_{L^{(K)}-L^{(2)}+t}, \ldots, \boldsymbol{e}_{L^{(K)}+t-1}], t \in T^{(1)}. \tag{5}$$

For the non-top tiers (i.e., $1 \leq k < K$), the input of GRU layers or FF layers is a linear combination of the frame input $\boldsymbol{f}_t^{(k)}$ and the conditioning vector $\boldsymbol{d}_t^{(k+1)}$ coming from the output of the GRU layers in the above tier. Then, the GRU units update their hidden states $\boldsymbol{h}_t^{(k)}$ based on the hidden states of previous time step $\boldsymbol{h}_{t-1}^{(k)}$ and the input at current time step. At last, the FF layers with a softmax activation function at the last layer generate a probability distribution of the current sample conditioned on the previous samples. At synthesis time, this conditional distribution is used to determine the sample value at each time step.

2.2. SampleRNN-based neural vocoder

The proposed SampleRNN-based neural vocoder is built based on the conditional form of a SampleRNN which models the probability of a sequence of waveform samples $\boldsymbol{x} = [x_1, x_2, \ldots, x_T]$ as the product of the probabilities of each sample conditioned on all previous samples and a sequence of acoustic features $\boldsymbol{c}$ as

$$p(\boldsymbol{x}|\boldsymbol{c}) = \prod_{t=1}^{T} p(x_t | x_1, \ldots, x_{t-1}, \boldsymbol{c}). \tag{6}$$

Since the temporal resolution of the acoustic features is much lower than waveform samples, this paper proposes to construct the conditional SampleRNN by adding a conditional tier on the top of an unconditional SampleRNN as shown by the dotted lines in Fig. 2. With the help of the hierarchical structure, the SampleRNN-based neural vocoder can avoid the temporal resolution adjustment in WaveNet-based neural vocoders [17, 18]. In Fig. 2, we set $K = 4$ and $L^{(4)} = 80$ which corresponds to 5ms frame shift of acoustic features for 16kHz sampling rate. Equation (2) and (3) can also be applied here. For the conditional tier, $L^{(K)}$ donates the frame shift of acoustic features and the frame input is

$$\boldsymbol{f}_t^{(K)} = [c_1^t, \ldots, c_d^t], t \in T^{(K)}, \tag{7}$$

where $[c_1^t, \ldots, c_d^t], t \in T^{(K)}$ are the t-th frame acoustic features for predicting waveform samples $[x_{tL^{(K)}+1}, \ldots, x_{(t+1)L^{(K)}}]$ and d represents the dimension of the acoustic features ($d = 43$ in Fig. 2). Equation (4) and (5) can also be applied for intermediate tiers $(1 < k < K)$ and bottom tier $(k = 1)$ respectively.

2.3. Model training and waveform generation

At the training stage, the waveform samples are first quantized to discrete values by μ-law [20]. The sequence of quantized waveform samples are used as the output of the network. The sequence of acoustic features together with history waveform samples are used as the input of the network. The network is trained to minimize the cross-entropy between the natural and predicted distributions of waveform samples.

At the generation stage, the waveform samples are generated in an autoregressive manner. Each sample is generated based on its corresponding acoustic features and previous samples by stochastically sampling its predicted conditional distribution. After one sample is predicted, it is fed back into the network to predict the next one. At last, the generated samples are processed by inverse μ-law mapping to get the final waveforms.

3. EXPERIMENTS

3.1. Experimental conditions

Two speech synthesis corpora were used in our experiments. One was a Chinese corpus with 1000 utterances from a female speaker. Another was an English corpus with 1000 utterances randomly selected from the recordings of the male speaker *bdl* in CMU-ARCTIC databases [21]. The waveforms of both corpora had 16kHz sampling rate and 16bits resolution. For each speaker, we chose 800 and 100 utterances to construct the training set and validation set respectively, and the remaining 100 utterances were used as the test set. The acoustic features at each frame were 43-dimensional including 40-dimensional mel-cepstra, an energy, an F0 and a V/UV flag. The natural acoustic features were extracted by STRAIGHT and the window size was 25ms and the window shift was 5ms. For SPSS, a bidirectional LSTM-RNN acoustic model [3] having 2 hidden layers with 1024 units per layer (512 forward units and 512 backward units) was trained to predict acoustic features from linguistic features for experiments. The input linguistic context features were 566-dimension for Chinese and 425-dimension English. The output of the acoustic model contained the acoustic features together with their delta and acceleration counterparts, which were total 127 dimensions (the V/UV flag had no dynamic components). Finally, the predicted acoustic features were generated from the output by maximum likelihood parameter generation (MLPG) algorithm.

Three vocoders were compared in our experiments. The descriptions of these vocoders are as follows and all settings below were determined by model performance on the validation set.

- ***STRAIGHT***: The conventional STRAIGHT vocoder. At synthesis time, the spectral envelope at each frame was first reconstructed from input mel-cepstra and frame energy, and then used to generate speech waveforms together with input source parameters (i.e., F0 and V/UV flag) [10].

Table 1. Comparison of classification accuracy (denoted by ACC) and cross entropy (denoted by CE) between the WaveNet-based and the SampleRNN-based neural vocoder on the test set of two corpora.

	Chinese female		English male	
	WaveNet	***SampleRNN***	***WaveNet***	***SampleRNN***
ACC(%)	19.77	**20.59**	14.16	**14.51**
CE	2.7427	**2.6983**	3.2304	**3.1570**

- ***WaveNet***: A WaveNet-based neural vocoder. The built model had 40 dilated casual convolution layers which were divided into 4 convolution blocks. Each block contained 10 layers and their dilation coefficients were $\{2^0, 2^1, 2^2, \ldots, 2^9\}$. For the residual architectures, the number of residual channels was 128 and the number of skip channels was 256. The waveform samples were quantized by 10-bit μ-law. An *Adam* optimizer [22] was used to update the parameters to minimize the cross-entropy. The average time for generating one second speech was 101.29s on a single Tesla K40 GPU using the TensorFlow [23] framework for implementation.
- ***SampleRNN***: Our proposed SampleRNN-based neural vocoder. The built model was composed of 4 tiers with two FF layers in Tier 1 and one GRU layer in Tier 2,3 and 4. Both the GRU layers and the FF layers had 1024 hidden units and the embedding size was 256. We set $L^{(4)} = 80, L^{(3)} = 8, L^{(2)} = 2$ and $L^{(1)} = 1$ as shown in Fig. 2. The optimization method and waveform quantization method were the same as that of the WaveNet-based neural vocoder mentioned above. Truncated back propagation through time (TBPTT) algorithm was employed to improve the efficiency of model training and the truncated length was set to 480. Under the same hardware and software environments as the WaveNet-based neural vocoder, the average time consumed for generating one second speech was 91.89s for the SampleRNN-based neural vocoder, which was slightly faster than the WaveNet-based one.

3.2. Objective evaluation

We first compared the performance of the trained models in the WaveNet-based and the SampleRNN-based vocoders using two metrics. One was the accuracy of classifying waveform sample into quantization levels, which was calculated by imitating the training process and assuming that the historical input samples were all natural and each output sample was obtained by selecting the quantization level with maximum posterior probability. Another was the average cross entropy calculated between the original and predicted conditional distributions of output samples. The results of these two metrics calculated on the test sets of the two corpora are listed in Table 1. We can see that our proposed SampleRNN-based neural vocoder was slightly better than the WaveNet-based one in both accuracy and cross entropy.

Then, we compared the distortions between natural speech and the speech reproduced by the three vocoders listed in Section 3.1. Four metrics in previous work [17] were adopted here, including signal-to-noise ratio (SNR) which reflected the distortion of waveforms, mel-cepstrum distortion (MCD) which described the distortion of mel-cepstra, RMSE of F0 which reflected the distortion of F0 (denoted by F0-RMSE), and V/UV error which was the ratio of the number of unmatched V/UV frames between original and synthesized speech to the number of total frames.

Table 2. Comparison of distortion among STRAIGHT, WaveNet-based neural vocoder and SampleRNN-based neural vocoder on the test set of the Chinese corpus.

	STRAIGHT	***WaveNet***	***SampleRNN***
SNR(dB)	2.4994	4.7093	**5.1987**
MCD(dB)	1.5744	1.6919	**1.4950**
F0-RMSE(cent)	20.6821	14.9475	**11.4926**
V/UV error(%)	**2.9172**	3.5552	3.1725

Table 3. Comparison of distortion among STRAIGHT, WaveNet-based neural vocoder and SampleRNN-based neural vocoder on the test set of the English corpus.

	STRAIGHT	***WaveNet***	***SampleRNN***
SNR(dB)	1.3858	**4.3741**	4.3404
MCD(dB)	**1.5239**	1.7491	1.9512
F0-RMSE(cent)	24.9766	20.1244	**19.9917**
V/UV error(%)	**4.3922**	5.2266	4.5458

STRAIGHT was used to extract acoustic features from both original and reproduced speech waveforms for calculating all these metrics. The results on the test sets of the two corpora are listed in Table 2 and 3 respectively. It is obvious that the STRAIGHT vocoder achieved the lowest SNR for both speakers due to the neglect of phase information. On the other hand, the two neural vocoders restored the shape of waveforms much better because they modeled and predicted waveform samples directly. Besides, our proposed SampleRNN-based vocoder achieved the lowest F0-RMSE among the three vocoders. Regarding with MCD, the results on these two speakers were inconsistent which needed further investigation. We should also notice that the performance of the two neural vocoders on generating waveforms with correct V/UV flags was still not as good as the STRAIGHT vocoder.

3.3. Subjective evaluation

Several groups of ABX preference tests were conducted on both corpora to compare the subjective performance of different vocoders.[1] Not only the acoustic features extracted from natural recordings, but also the acoustic features predicted from an acoustic model, were used to reconstruct speech waveforms for evaluation. Here, the acoustic model was a bidirectional LSTM-RNN which predicted acoustic features from corresponding linguistic features. In each subjective test, 20 utterances synthesised by two comparative vocoders were randomly selected from the test set. Each pair of generated speech were evaluated in random order. For the tests on the Chinese corpus, 10 Chinese native speakers were asked to be the listeners. For the tests on the English corpus, each pair of synthetic sentences were evaluated by at least 15 English native listeners on the crowdsourcing platform of Amazon Mechanical Turk (`https://www.mturk.com`). The listeners were asked to judge which utterance in each pair had better speech quality or there was no preference. In addition to calculating the average preference scores, the p-value of a t-test was used to measure the significance of the difference between two vocoders. The subjective evaluation results are listed in Table 4 and 5.

Table 4. Average preference scores (%) on speech quality among three vocoders using the Chinese corpus, where N/P stands for "no preference" and p denotes the p-value of a t-test between two vocoders. "R" stands for using natural acoustic features as input and "P" stands for using predicted acoustic features as input.

	STRAIGHT	***WaveNet***	***SampleRNN***	N/P	p
R	10.55	–	**55.05**	34.40	< 0.001
	–	9.17	**37.16**	53.67	< 0.001
P	9.13	–	**54.80**	36.07	< 0.001
	–	10.18	**38.89**	50.93	< 0.001

Table 5. Average preference scores (%) on speech quality among three vocoders using the English corpus, where N/P stands for "no preference" and p denotes the p-value of a t-test between two vocoders. "R" stands for using natural acoustic features as input and "P" stands for using predicted acoustic features as input.

	STRAIGHT	***WaveNet***	***SampleRNN***	N/P	p
R	17.06	–	**65.29**	17.65	< 0.001
	–	23.91	**39.35**	36.74	< 0.001
P	10.88	–	**67.35**	21.77	< 0.001
	–	17.90	**42.89**	39.21	< 0.001

From these two tables, we can see that our proposed SampleRNN-based neural vocoder outperformed the conventional STRAIGHT vocoder significantly in terms of the subjective quality of synthetic speech on both corpora and using both kinds of input acoustic features. This result shows the effectiveness of neural vocoders for improving the naturalness of SPSS. Furthermore, the SampleRNN-based vocoder also achieved better subjective performance than the WaveNet-based one. This is consistent with the objective performance shown in Table 1. One possible reason is that the SampleRNN-based neural vocoder can make use of all history information to generate current sample according to the characteristics of RNNs. However, the receptive field (i.e. the number of previous samples which can be used as conditions to generate the current sample) of the WaveNet-based neural vocoder is fixed and limited. Increasing its receptive field always needs more layers and leads to higher complexity of model training and waveform generation.

4. CONCLUSION

In this paper, we have proposed a SampleRNN-based neural vocoder, which utilizes conditional SampleRNN model to convert the acoustic features into speech waveforms directly. Different from conventional vocoders following the source-filter model, this proposed vocoder adopts nonlinear neural networks with hierarchical and recurrent architectures to describe the conditional distribution of waveform samples. At synthesis time, the waveform samples are generated in an autoregressive manner. Experimental results show that our proposed vocoder outperforms both the STRAIGHT vocoder and a WaveNet-based vocoder with similar run-time efficiency in terms of the subjective quality of synthetic speech no matter natural or predicted acoustic features are used as inputs. An important goal of our future work is to improve the efficiency of the SampleRNN-based neural vocoder since our current implementation still runs about 90 times slower than real-time. To study more variants of the conditional SampleRNN structures and to investigate speaker-independent (SI) modeling of neural vocoder will also be the tasks of our future research.

[1] Examples of generated speech can be found at `http://home.ustc.edu.cn/~ay8067/ICASSP_2018/demo.html`.

5. REFERENCES

[1] Keiichi Tokuda, Yoshihiko Nankaku, Tomoki Toda, Heiga Zen, Junichi Yamagishi, and Keiichiro Oura, "Speech synthesis based on hidden Markov models," *Proceedings of the IEEE*, vol. 101, no. 5, pp. 1234–1252, 2013.

[2] Heiga Zen, Andrew Senior, and Mike Schuster, "Statistical parametric speech synthesis using deep neural networks," in *Proc. ICASSP*, 2013, pp. 7962–7966.

[3] Yuchen Fan, Yao Qian, Feng-Long Xie, and Frank K Soong, "TTS synthesis with bidirectional LSTM based recurrent neural networks," in *Proc. Interspeech*, 2014, pp. 1964–1968.

[4] Andrew J Hunt and Alan W Black, "Unit selection in a concatenative speech synthesis system using a large speech database," in *Proc. ICASSP*, 1996, vol. 1, pp. 373–376.

[5] Zhen-Hua Ling, Shi-Yin Kang, Heiga Zen, Andrew Senior, Mike Schuster, Xiao-Jun Qian, Helen M Meng, and Li Deng, "Deep learning for acoustic modeling in parametric speech generation: A systematic review of existing techniques and future trends," *IEEE Signal Processing Magazine*, vol. 32, no. 3, pp. 35–52, 2015.

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

[7] James L Flanagan and RM Golden, "Phase vocoder," *Bell Labs Technical Journal*, vol. 45, no. 9, pp. 1493–1509, 1966.

[8] Bernard Gold and C Rader, "The channel vocoder," *IEEE Transactions on Audio and Electroacoustics*, vol. 15, no. 4, pp. 148–161, 1967.

[9] D Paul, "The spectral envelope estimation vocoder," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 29, no. 4, pp. 786–794, 1981.

[10] Hideki Kawahara, Ikuyo Masuda-Katsuse, and Alain 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.

[11] Masanori Morise, Fumiya Yokomori, and Kenji 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.

[12] Fant Gunnar, "The acoustic theory of speech production," *SGravenhage, Mouton*, 1960.

[13] Aaron van den Oord, Sander Dieleman, Heiga Zen, Karen Simonyan, Oriol Vinyals, Alex Graves, Nal Kalchbrenner, Andrew Senior, and Koray Kavukcuoglu, "WaveNet: A generative model for raw audio," *arXiv preprint arXiv:1609.03499*, 2016.

[14] Soroush Mehri, Kundan Kumar, Ishaan Gulrajani, Rithesh Kumar, Shubham Jain, Jose Sotelo, Aaron Courville, and Yoshua Bengio, "SampleRNN: An unconditional end-to-end neural audio generation model," *arXiv preprint arXiv:1612.07837*, 2016.

[15] Tom Le Paine, Pooya Khorrami, Shiyu Chang, Yang Zhang, Prajit Ramachandran, Mark A Hasegawa-Johnson, and Thomas S Huang, "Fast WaveNet generation algorithm," *arXiv preprint arXiv:1611.09482*, 2016.

[16] Aaron van den Oord, Yazhe Li, Igor Babuschkin, Karen Simonyan, Oriol Vinyals, Koray Kavukcuoglu, George van den Driessche, Edward Lockhart, Luis C Cobo, Florian Stimberg, et al., "Parallel WaveNet: Fast high-fidelity speech synthesis," *arXiv preprint arXiv:1711.10433*, 2017.

[17] Akira Tamamori, Tomoki Hayashi, Kazuhiro Kobayashi, Kazuya Takeda, and Tomoki Toda, "Speaker-dependent WaveNet vocoder," *Proc. Interspeech 2017*, pp. 1118–1122, 2017.

[18] Ya-Jun Hu, Chuang Ding, Li-Juan Liu, Zhen-Hua Ling, and Li-Rong Dai, "The USTC system for Blizzard Challenge 2017.," in *Proc. Blizzard Challenge Workshop*, 2017.

[19] Hong-Chuan Wu, Yu Gu, and Zhen-Hua Ling, "Speech parameter synthesizer based on deep convolutional neural network," *Proc. of the 14th National Conference on Man-Machine Speech Communication (in Chinese)*, 2017.

[20] ITUT Recommendation, "G. 711: Pulse code modulation (PCM) of voice frequencies," *International Telecommunication Union*, 1988.

[21] John Kominek and Alan W Black, "The CMU Arctic speech databases," in *Proc. ISCA Workshop*, 2004.

[22] Diederik Kingma and Jimmy Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[23] Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, et al., "Tensorflow: Large-scale machine learning on heterogeneous distributed systems," *arXiv preprint arXiv:1603.04467*, 2016.