Interspeech 2018
2-6 September 2018, Hyderabad

Korean Singing Voice Synthesis System based on an LSTM Recurrent Neural Network

Juntae Kim[1], Heejin Choi[1], Jinuk Park[1], Minsoo Hahn[1], Sangjin Kim[2] and Jong-Jin Kim[2]

[1]School of Electrical Engineering,
Korea Advanced Institute of Science and Technology (KAIST), Daejeon, Korea
[2]SK telecom, Seoul, Korea

{jtkim, change, windclay, mshahn2}@kaist.ac.kr, {kimsangjin, kimjj.geek}@sk.com

Abstract

Singing voice synthesis (SVS) systems generate the singing voice from a musical score. Similar to the text-to-speech synthesis (TTS) field, SVS systems have also been greatly improved since the deep neural network (DNN) framework was introduced. Although they share many parts of the framework, the main difference between TTS and SVS systems is that the feature composing method, between linguistic and musical features, is important for SVS systems. In this paper, we propose a Korean SVS system based on a long-short term memory recurrent neural network (LSTM-RNN). At the feature composing stage, we propose a novel composing method, based on Korean syllable structure. At the synthesis stage, we adopt LSTM-RNN for the SVS. According to our experiments, our composed feature improved the naturalness of the voice, specifically in any part that has to be pronounced for a long time. Furthermore, LSTM-RNN outperformed the DNN based SVS system in both quantitative and qualitative evaluations.

Index Terms: Singing voice synthesis, speech synthesis

1. Introduction

Singing voice synthesis (SVS) systems generate the singing voice from a musical score which contains both linguistic (lyrics) and musical features (notes, tempo, etc.). This is different from ordinary speech synthesis systems, which are dependent on only the linguistic features. Therefore, SVS systems are composed of two parts: composing the linguistic and musical features, and synthesizing the singing voice from the model learned by using the composed input features. Traditionally, SVS systems are based on the hidden Markov model (HMM) [1, 2], which can model several acoustic features of a singing voice simultaneously. While HMM needs fluent composed features (such as quinphone, the number of phonemes in a current syllable, and musical information of the note), its sound quality is known to be unnatural because HMM has a chronic over-smoothing problem in both the frequency and time domains [3].

Recently, many types of neural network based models have been proposed for text-to-speech synthesis (TTS) systems: deep neural network (DNN) [4], recurrent neural network with long-short term memory (LSTM-RNN) [5], Wavenet [6], and Tacotron [7]. These models significantly outperformed the traditional HMM-based TTS systems [8], [9], while they need fewer linguistic features than HMM-based TTS systems. In the same manner, some of the neural network based models were adopted for SVS systems. In [10] and [3], DNN and Wavenet were adopted for their acoustic models, respectively. Additionally, due to the importance of pitch for a natural sounding singing voice, both [3] and [10] proposed a post-processing method for the pitch, based on the heuristic signal processing method. This implies that learning the expressive pitch from the model itself is still a challenge. This is because the amount of singing voice data recorded in the studio, with the corresponding musical score, is quite limited, whereas there exists a number of musical factors (such as melody, note, and accent) that make pitch variation complicated. In addition to the post-processing of the predicted acoustic features, the deficiency in the amount of data also exhibits the importance of the feature composing method, considered to the feature engineering, which can reduce the complexity of the data distribution that the model needs to learn. As in [11], the feature composing method is dependent on the language used, because of their individual syllable structures. So far, the feature composing methods are mainly based on Japanese and English, although Spanish has also recently been considered [3].

In this paper, we provide the method of how to compose Korean lyrics with musical features, such as note and slur, based on the syllable structure of Korean at syllable, phoneme, and frame level. From the available methods which use frame-level input features, this paper adopts the LSTM-RNN for SVS system, with some post-processing proposed to synchronize the duration of synthesized singing voice with the musical note duration. According to our experiments, we found that: (a) our proposed feature helped in handling the parts of the musical score that contained long slurs, (b) LSTM-RNN outperformed the DNN based SVS system in both quantitative and qualitative evaluations.

2. LSTM-RNN based Korean singing voice synthesis system

The main components of our system are feature composing and synthesis. For the feature composing part, we examine ways to compose the Korean lyrics and musical features at syllable, phoneme, and frame levels. For the synthesis part, we present our synthesis model which consists of duration and acoustic models with a vocoder, based on the framework in [5]. The duration model predicts the duration of the target phoneme from the input features, at phoneme level. This predicted duration is used to convert the phoneme level input features to the frame level. The acoustic model predicts the acoustic features such as mel-generalized cepstral features (MGC), log fundamental frequency (LF0), mean band aperiodicity (BAP), and voicing decision (VUV) from the input features at frame level. Finally, a vocoder generates the singing voice using predicted acoustic features from the

10.21437/Interspeech.2018-1575

acoustic model. The details of each part are described as follows.

2.1. Feature composing

In this section, we formally described the feature composing step from the musical score. Let $\mathbf{x} = (x_1, \ldots, x_N)$ be the input sequence corresponding to the musical score, and $x_i = [l_i, \mathrm{M1}_i, \mathrm{M2}_i, \mathrm{M3}_i]$ where: N is the number of musical notes in the musical score, $[\cdot]$ is the concatenation operation over feature dimension, $l_i \in \{SYL, -, X\}$ (in which the *SYL* is Korean syllable), '–' is the slur, and *X* is the rest. $\mathrm{M1}_i, \ldots, \mathrm{M3}_i$ are musical features, as described in Table 1. M4 and M5 are excluded from x_i because M2 represents those features, which will be described in the following section. Note that except for the musical features in Table 1, we have also tried to use other musical features that can be extracted from the musical score such as beat, the number of measures, etc. However, using these features added some noise sounds to the singing voice of which deteriorates the quality.

From $\mathbf{x}$, the acoustic model should generate $\mathbf{y} = (y_1, \ldots, y_T)$ where $y_i = [\mathrm{MGC}_i, \mathrm{LF0}_i, \mathrm{BAP}_i, \mathrm{VUV}_i]$ corresponding to the acoustic features, and T is the number of frames. However, as T is much larger than N, this can be a problem because we adopt vanilla LSTM-RNN which predicts the outputs at each time step of the inputs, so that both the length of the input and output sequences must be matched. In order to solve this problem, the length of $\mathbf{x}$ is extended by sequentially going through the syllable, phoneme, and frame level feature composing steps as follows.

2.1.1. Syllable level

A musical note has two important features: pitch (M1) and duration (M4). We encode the duration of the note into categorical features, and for the pitch, we encode it into numerical features that represent frequency. For example, if a pitch of note is C with octave 1, it was encoded to 32 Hz. This numerical representation is beneficial compared to categorical representation as it reduces both the feature dimension and the number of model parameters. The tempo (M5) is encoded by numerical features, and this determines the speed of the music. Because we have unified the unit of tempo into a quarter note, from now on we will define "tempo" as the "number of quarter notes per minute". We can extract the physical time of a note from the tempo and duration information. For example, if the tempo is set to 100, the time corresponding to a quarter note is 0.6 s. We defined it as M2, and this has been included in our feature set. Note that the musical rest also has a duration, although it does not have a pitch. If a rest occurred, we assigned a value of zero for the pitch and we assigned '*X*' for l_i.

We define the symbol '–' as a slur, which means the successive phonation of a preceding syllable. The slur was encoded into the categorical feature (M3). When the slur occurs, it is necessary to substitute '–' for the syllable by considering the actual pronunciation, depending on the preceding syllable. We used the substitution rule from the Korean syllable structure, which consists of phonemes that have a beginning consonant (BC), a middle vowel (MV), and an optional final consonant (FC). If the syllable begins with a vowel sound, i.e., silent BC (S-BC), 'o' is assigned. In other words, the combination of BC, MV, and optional FC form a Korean syllable. Note that a Korean syllable is composed of a minimum of two phonemes, i.e., BC and MV must be included in a Korean syllable. The substitution rule for l_{i+1} which is set to '–', when the preceding syllable l_i is set to *SYL*: composed of (S-BC+MV), (BC+MV), (S-BC+MV+FC) and (BC+MV+FC) is described as follows in sequence:

$$\begin{aligned}
&l'_i = SYL(\text{S-BC} + \text{MV}),\ l'_{i+1} = SYL(\text{S-BC} + \text{MV}),\\
&l'_i = SYL(\text{BC} + \text{MV}),\ l'_{i+1} = SYL(\text{S-BC} + \text{MV}),\\
&l'_i = SYL(\text{S-BC} + \text{MV}),\ l'_{i+1} = SYL(\text{S-BC} + \text{MV+FC}),\\
&l'_i = SYL(\text{BC} + \text{MV}),\ l'_{i+1} = SYL(\text{S-BC} + \text{MV+FC}),
\end{aligned} \tag{1}$$

where '+' is the combination operator in a Korean syllable. The following results are the examples of (1) derived when l_{i+1} is set to '–', and l_i is sequentially set to 이 (/ih/), 기 (/g ih/), 잉 (/ih ng/) and 깅 (/g ih ng/):

$$\begin{aligned}
&l'_i = \text{ㅇ}|(/\text{ih}/),\ l'_{i+1} = \text{ㅇ}|(/\text{ih}/),\\
&l'_i = \text{ㄱ}|(/\text{g ih}/),\ l'_{i+1} = \text{ㅇ}|(/\text{ih}/),\\
&l'_i = \text{ㅇ}|(/\text{ih}/),\ l'_{i+1} = \text{ㅇ}\underline{\text{ㅇ}}(/\text{ih ng}/),\\
&l'_i = \text{ㄱ}|(/\text{g ih}/),\ l'_{i+1} = \text{ㅇ}\underline{\text{ㅇ}}(/\text{ih ng}/),
\end{aligned} \tag{2}$$

where the phonemes with respect to the pronunciation of Korean syllables are represented in the brackets, by referring to the CMU pronouncing dictionary [12].

According to our experiments, this substitution helped the model by indicating what the model actually phonates in response to the slur. If we treated the slur as a single symbol, like a rest, the model would learn numerous cases of phonating responses to preceding syllables, which made the learning of the model difficult. However, the synthetic singing voice after substitution was only natural in the case where M1 of x_i and x_{i+1} were different. In the case where M1 of x_i and x_{i+1} are the same, our substitution caused discontinuous phonation, because in this case the actual phonation is like phonating the preceding syllable for a long time, rather than phonating the preceding syllable and substituted syllable from (1) discretely. Therefore, in the case that M1 of x_i and x_{i+1} are the same, we combined x_i and x_{i+1} into x'_i, which has the same feature as x_i and x_{i+1}, except for the M2. The M2 of x'_i was calculated as follows:

$$\mathrm{M2}_i \text{ of } x'_i = \mathrm{M2}_i \text{ of } x_i + \mathrm{M2}_i \text{ of } x_{i+1}. \tag{3}$$

As a result, we can get the $\mathbf{x}_{SYL} = (x'_1, \ldots, x'_{N\text{-}K})$ from $\mathbf{x}$ with $x'_i = [l'_i, \mathrm{M1}_i, \mathrm{M2}_i, \mathrm{M3}_i]$ where: K is the number of combinations from (3), and $l'_i \in \{SYL, X\}$.

2.1.2. Phoneme level

At the phoneme level, we decomposed the syllable in l'_i into either {BC, MV} or {BC, MV, FC}. For example, if the syllable in l'_i consisted of {BC, MV, FC}, the decomposition result became as follows:

Table 1: *Musical feature description (C: Categorical feature, N: Numerical feature).*

Description	Symbol	Example	Type
Pitch of note	M1	C4, Bb3	N
Duration from tempo	M2	0.6 s	N
Slur	M3	True, False	C
Duration of note	M4	Quarter, 16th	C
Tempo	M5	100, 120	N

$$Decompose(x_i') = (x_j'', x_{j+1}'', x_{j+2}''), \quad (4)$$

where $(x_j'', x_{j+1}'', x_{j+2}'')$ is the sequence of decomposed elements, which are as follows:

$$x_j'' = [\text{BC of } l_i', \text{M1}_i, \text{M2}_i, \text{M3}_i]. \quad (5)$$

$$x_{j+1}'' = [\text{MV of } l_i', \text{M1}_i, \text{M2}_i, \text{M3}_i]. \quad (6)$$

$$x_{j+2}'' = [\text{FC of } l_i', \text{M1}_i, \text{M2}_i, \text{M3}_i]. \quad (7)$$

As described in (5)–(7), all musical features were set to the same as the syllable level. In the case where l_i' is set to X (rest), all features were the same as the syllable level. As a result, we obtained a sequence of phoneme level input features $\mathbf{x}_P=(x_1'', \ldots, x_L'')$, where L is the number of phonemes.

2.1.3. Frame level

At the frame level, using the duration at the phoneme level and $\mathbf{x}_P$, we composed a sequence of frame level input features $\mathbf{x}_F=(x_1''', \ldots, x_T''')$, where x_k''' is the input features at frame level. We used the ground truth phoneme level duration in the training phase, and the predicted result of our duration model in the synthesis phase. Note that if $(x_k''', \ldots, x_{k+l}''')$ are extracted from x_j'' , they share input features with x_j'' with an additional numerical feature for determining the position of the current frame in the current phoneme. The position values were normalized to have a value of zero to one.

2.2. Synthesis model

The duration model comprised 2 steps: prediction and post-processing. The prediction step was conducted by LSTM-RNN, which was trained with $\mathbf{x}_P$ and the ground truth phoneme level duration with the mean squared error loss (MSE). Then the post-processing normalized the predicted duration using M2_i which is the summed result of the decomposed phoneme-level durations, corresponding to the ith syllable. For example, if the syllable-level acoustic features x_i' are decomposed as in (4), the normalization is conducted as follows.

$$dur_m' = \frac{dur_m \cdot \text{M2}_i}{\sum_{k=j}^{j+2} dur_k}, m = j, j+1, j+2, \quad (8)$$

where $dur_{j:j+2}$ is the predicted duration of $x_{j:j+2}''$, and $dur_{j:j+2}'$ is the normalized duration. It should be noted that the normalized duration was used to transform $\mathbf{x}_P$ to the $\mathbf{x}_F$ in the synthesis phase.

The acoustic model consisted of 3 parts: MGC and BAP (MB), LF0, and the VUV feature. Each part adopted the LSTM-RNN, used $\mathbf{x}_F$ as its inputs, and was trained separately with its individual loss. For the LF0 and MB parts, the loss was set to negative log-likelihood (NLL) with the Gaussian mixture model (GMM) as shown in this sequence:

$$loss_p = -\sum_{i=1}^{T} \log \sum_{n=1}^{N_p} \pi_{p,n} \mathcal{N}\left(y_{p,i}^{target} \mid \mu_{p,n}, \Sigma_{p,n}\right), \; p = 1, 2, \quad (9)$$

where $p = 1$ corresponds to the LF0 part, $p = 2$ corresponds to the MB part, N_p is the number of mixtures, $\pi_{p,n}$ is the mixing coefficient, $\mu_{p,n}$ is the mean vector, and $\Sigma_{p,n}$ is the diagonal covariance matrix of nth mixture. Note that $\{\pi_{p,n}, \mu_{p,n}, \Sigma_{p,n}\}$ are predicted from the softmax, linear output layer and exponential activation function followed by linear output layer, respectively [13]. The $y_{p,i}^{target}$ is from the ground truth acoustic features. The $y_{1,i}^{target}$ is set to LF0_i and $y_{2,i}^{target}$ is set to $[\text{MGC}_i, \text{BAP}_i]$. For the VUV part, the softmax function was applied to the output of VUV LSTM-RNN, and then cross-entropy loss was used. This separation is necessary for effective learning, because acoustic features are known to be independent of each other [3]. Therefore, if acoustic features are trained in one model, the impact of the learning signal (i.e. the gradient) from an acoustic feature (e.g. LF0) can deteriorate due to the learning signal from other acoustic features which makes the model underfit. This fact is also supported from the result in [3], [10] which showed that the separation is helpful for increasing the subjective evaluation score. However, according to our experiments, the separation of BAP and MGC didn't show any outstanding improvement, so these features were used simultaneously for learning one model to reduce the learning time.

At the synthesis phase, the mean vector of the most probable mixture component was used for prediction purposes [14]. Finally, the predicted acoustic features were used as inputs for the vocoder, to synthesize the waveform.

3. Experiments

3.1. Experimental setup

3.1.1. Dataset preparation

For the dataset, we recorded 52 Korean children's songs, performed by a female singer. Of these, 44 songs were used to train, 3 songs were used to validate, and 5 songs were used to evaluate our model. Note that the length of each song was too long to train or evaluate all at once, due to memory issues, so we first decomposed the songs into segments, and each segment had four measures, which could be extracted from the musical score. The average length of each segment was 7.2 s. The sampling rate for the singing voice was originally 44.1 kHz, then reduced to 22.05 kHz. Out of the acoustic features, 25 MGC and 2 BAP coefficients were extracted based on a WORLD vocoder [15], and one LF0 was extracted from RAPT [16] implemented in SPTK [17] with a 5 ms hop time. In addition, 2 VUVs (one-hot coded) were extracted by applying a threshold to the LF0 feature. Therefore, 30 acoustic features in total were used as the outputs of our acoustic model. The input features had 50 dimensions (each feature is described in Table 2), and the phoneme identity was encoded as a one-hot vector. The input features were z-score-normalized for training. To normalize the input features for evaluation, the same normalization factors used in training were applied. The BAP and MGC were also normalized in the same way. The ground truth phoneme alignment was firstly found by applying the forced alignment with a well-trained HMM [18]. In some cases when the acoustic context of the singing voice is quite different from the ordinary speech, the forced alignment didn't work well; hence, we manually found the phoneme alignment for those cases.

Table 2: Input feature description at frame level (C= Categorical feature, N= Numerical feature).

Description	Dimension	Type
Phoneme identity	46	C
Pitch of note	1	N
Slur	1	C
Position	1	N
Duration from tempo	1	N

3.1.2. *Parameter setting and baseline methods*

All LSTM-RNNs which consist of acoustic and duration models have one forward-directed hidden LSTM layer, with 128 memory blocks. For the baseline methods, a DNN was adopted instead of LSTM-RNN for the acoustic and duration models, which is similar to the framework used in [10]. All DNNs of the duration and acoustic models each had 2 hidden layers, and 1024 hidden units. Furthermore, because a DNN cannot inherently model the context information, DNN inputs included both past and future, one context at slur and pitch level, and two contexts at phoneme level. Therefore, the input features for the DNN had a dimension of 238. Furthermore, to verify the effect of the number of mixtures, we compared our model results when the number of mixtures was set to one, two and four. The DNN had four mixtures. All the aforementioned parameters were found from our validation set.

3.1.3. *Training methods*

All acoustic and duration models, including the baseline methods, were trained using a mini-batch stochastic gradient descent (SGD) method, with Adam-based learning rate scheduling [19]. Also, the exponential decay of the learning rate was applied. The initial learning rate was set to 0.001, then decayed with a base of 0.95 per 50 learning steps. The initial weights were initialized randomly. For regularization purposes, dropout [20] was applied with a rate of 0.5. The early stopping method [21] was applied, to decide on the proper number of learning steps.

3.2. Investigation to the effect of proposed features

In order to verify the effect of our proposed features in the lyrics with slur, specifically, in the case of [살 (/s aa l/), –], we compared a spectrogram of the original waveform and synthesized waveform with our proposed features that applied (1), and a synthesized waveform with the features that did not apply (1), as shown in Figure 1. As expected, if (1) was not applied, the spectral energy drastically decreased in the segment of '–', corresponding to 0.2–0.7 s in (c). This implies that learning the slurred segments from the model itself is difficult, so that it is necessary to inform the model what actually should be pronounced for a '–'. In (b), the spectral energy was remained in 0.2–0.7 s, while the shape of the harmonic spectral envelope was different from (a). This is because there is a difference between natural pronunciation and the artificially injected pronunciation from (1).

Furthermore, we investigated 20 musical phrases including slurred segments to verify the perceptual aspects. For all slurred segments in 20 musical phrases, without the proposed features, the phonation was quite unnatural because it suddenly stopped although it should be kept. In contrast, with proposed features, the naturalness of phonation was improved while the slurred segments were somewhat monotonous compared to the original phonation because some musical context such as vibrato was not modeled.

3.3. Objective experiments

To evaluate the performance of the proposed system objectively, we used Mel-cepstral Distortion (MCD), Band Aperiodicity Distortion (BAPD), F0 Root Mean Squared Error (RMSE), FPR (False Positive Rate), and FNR (False Negative Rate). FPR and FNR represented the voiced/unvoiced (VUV) error rate. The evaluation results are described in Table 3

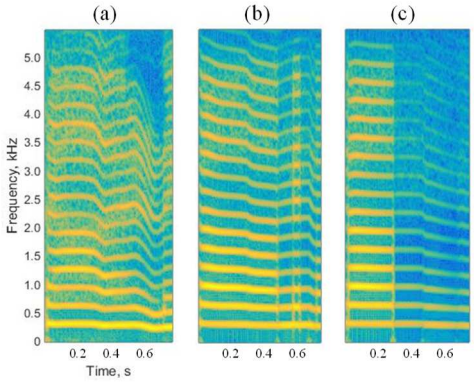

Figure 1: *The spectrogram corresponding to lyrics '살 (/s aa l/)' (0-0.2 s), '–' (0.2-0.7 s). (a) is from the ground truth waveform, (b) is from our synthesis model with proposed features, and (c) is from our synthesis model with features not applying the substitution rule, as described in (1).*

Table 3: *Evaluation results. The number besides the LSTM is the number of mixtures.*

	DNN	LSTM-1	LSTM-2	LSTM-4
MCD (dB)	8.61	6.01	5.57	**5.43**
F0-RMSE	62.00	39.56	39.56	**39.40**
BAPD (dB)	21.77	15.44	14.64	**14.58**
FPR (%)	54.15	20.34	20.21	**18.10**
FNR (%)	**1.87**	5.94	6.81	5.43
MOS	2.87±0.12	3.04±0.15	3.15±0.11	**3.24±0.10**

which shows that LSTM-4 outperformed all the baseline methods in almost all metrics, except for the FNR. DNN showed an outstanding result for the FNR measure, however, it cannot be concluded that the DNN is better than the LSTM-RNN based VUV classifier because the DNN showed quite higher FPR than the LSTM-RNN, which implies that the DNN classified most of inputs as voice sounds. This is because, while DNN learned from the singing voice which has much more voiced sounds than unvoiced sounds [3], has many parameters (~1294k) than LSTM-RNN (~91k), which can easily lead to overfitting to voiced sounds.

3.4. Subjective experiments

The subjective listening test was carried out to evaluate the naturalness of the synthesized songs. A total of 20 Korean subjects participated in the evaluation, to measure the mean opinion score (MOS) on a scale from one (poor) to five (good). All synthesized 37 musical phrases, which were decomposed from five songs, were presented to the subjects using headphones. The last row in Table 3 shows the experimental results. As described in Table 3, all LSTM based approaches outperformed the DNN based approach. Among the LSTM-based systems, there was no significant difference according to the number of mixtures, while the objective measure was slightly better as the number of mixtures were increasing.

4. Conclusions

In this paper, we proposed a feature-composing method to combine linguistic and musical features including how to deal with the slur to use as the input features for a Korean SVS system. We adopted the LSTM-RNN as our synthesis model, and showed that our proposed SVS system outperformed the baseline systems in both objective and subjective evaluations.

5. References

[1] Saino, Keijiro, et al., "An HMM-based singing voice synthesis system," *Ninth International Conference on Spoken Language Processing*, 2006.

[2] Oura, Keiichiro, et al., "Recent development of the HMM-based singing voice synthesis system—Sinsy," *Seventh ISCA Workshop on Speech Synthesis*, 2010.

[3] Blaauw, Merlijn, and Jordi Bonada, "A Neural Parametric Singing Synthesizer Modeling Timbre and Expression from Natural Songs," *Applied Sciences*, 2017.

[4] Zen, Heiga, and Andrew Senior, "Deep mixture density networks for acoustic modeling in statistical parametric speech synthesis," in *ICASSP*, 2014.

[5] Zen, Heiga, and Haşim Sak, "Unidirectional long short-term memory recurrent neural network with recurrent output layer for low-latency speech synthesis," in *ICASSP*, 2015.

[6] Van Den Oord, Aaron, et al., "Wavenet: A generative model for raw audio," *arXiv preprint arXiv:1609.03499*, 2016.

[7] Wang, Yuxuan, et al., "Tacotron: A fully end-to-end text-to-speech synthesis model," *arXiv preprint arXiv:1703.10135*, 2017.

[8] Tokuda, Keiichi, Takao Kobayashi, and Satoshi Imai, "Speech parameter generation from HMM using dynamic features," in *ICASSP*, 1995.

[9] Yoshimura, Takayoshi, et al., "Simultaneous modeling of spectrum, pitch and duration in HMM-based speech synthesis," *Sixth European Conference on Speech Communication and Technology*, 1999.

[10] Nishimura, Masanari, et al., "Singing Voice Synthesis Based on Deep Neural Networks," in *Proc. Interspeech*, 2016.

[11] Nakamura, Kazuhiro, et al., "HMM-based singing voice synthesis and its application to Japanese and English," in *ICASSP*, 2014.

[12] "CMU pronouncing dictionary," [Online] Available: http://www.speech.cs.cmu.edu/cgi-bin/cmudict/

[13] Zen, Heiga, and Andrew Senior, "Deep mixture density networks for acoustic modeling in statistical parametric speech synthesis," in *ICASSP*, 2014.

[14] Wang, Xin, Shinji Takaki, and Junichi Yamagishi, "An autoregressive recurrent mixture density network for parametric speech synthesis," in *ICASSP*, 2017.

[15] Morise, Masanori, Fumiya Yokomori, and Kenji Ozawa, "WORLD: a vocoder-based high-quality speech synthesis system for real-time applications," *IEICE TRANSACTIONS on Information and Systems*, 2016.

[16] Talkin, David, "A robust algorithm for pitch tracking (RAPT)," *Speech coding and synthesis*, 1995.

[17] Tokuda, K., et al., "Speech signal processing toolkit (SPTK)," [Online], 2012.

[18] Ueda, Naonori, and Ryohei Nakano, "Deterministic annealing EM algorithm," *Neural networks*, 1998.

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

[20] Srivastava, Nitish, et al., "Dropout: A simple way to prevent neural networks from overfitting," *The Journal of Machine Learning Research*, 2014.

[21] Caruana, Rich, Steve Lawrence, and C. Lee Giles, "Overfitting in neural nets: Backpropagation, conjugate gradient, and early stopping," *Advances in neural information processing systems*, 2001.