INTERSPEECH 2016
September 8–12, 2016, San Francisco, USA

Singing voice synthesis based on deep neural networks

Masanari Nishimura, Kei Hashimoto, Keiichiro Oura, Yoshihiko Nankaku, and Keiichi Tokuda

Department of Scientific and Engineering Simulation,
Nagoya Institute of Technology, Nagoya, Japan
{nishi02, bonanza, uratec, nankaku, tokuda}@sp.nitech.ac.jp

Abstract

Singing voice synthesis techniques have been proposed based on a hidden Markov model (HMM). In these approaches, the spectrum, excitation, and duration of singing voices are simultaneously modeled with context-dependent HMMs and waveforms are generated from the HMMs themselves. However, the quality of the synthesized singing voices still has not reached that of natural singing voices. Deep neural networks (DNNs) have largely improved on conventional approaches in various research areas including speech recognition, image recognition, speech synthesis, etc. The DNN-based text-to-speech (TTS) synthesis can synthesize high quality speech. In the DNN-based TTS system, a DNN is trained to represent the mapping function from contextual features to acoustic features, which are modeled by decision tree-clustered context dependent HMMs in the HMM-based TTS system. In this paper, we propose singing voice synthesis based on a DNN and evaluate its effectiveness. The relationship between the musical score and its acoustic features is modeled in frames by a DNN. For the sparseness of pitch context in a database, a musical-note-level pitch normalization and linear-interpolation techniques are used to prepare the excitation features. Subjective experimental results show that the DNN-based system outperformed the HMM-based system in terms of naturalness.

Index Terms: Singing voice synthesis, Neural network, DNN, Acoustic model

1. Introduction

Singing voice synthesis enables computers to "sing" any song. It has become especially popular in Japan since singing voice synthesis software Vocaloid [1] was released. There has also been a growing demand for more flexible systems that can sing songs with various voices. One approach to synthesize singing voices is hidden Markov model (HMM)-based singing voice synthesis [2, 3]. In this approach, the spectrum, excitation, and duration of the singing voices are simultaneously modeled by HMMs and singing voice parameter trajectories are generated from the HMMs by using a speech parameter generation algorithm [4]. However, the quality of the synthesized singing voices still has not reached that of natural singing voices.

Deep neural networks (DNNs) have largely improved on conventional approaches in various research areas, e.g., speech recognition [5], image recognition [6], and speech synthesis [7, 8, 9]. In a DNN-based text-to-speech (TTS) synthesis system, a single DNN is trained to represent a mapping function from linguistic features to acoustic features that is modeled by decision tree-clustered context dependent HMMs in HMM-based TTS systems. The DNN-based TTS synthesis can synthesize high quality and intelligible speech, and several studies have reported the performance of DNN-based methods [7, 8, 9].

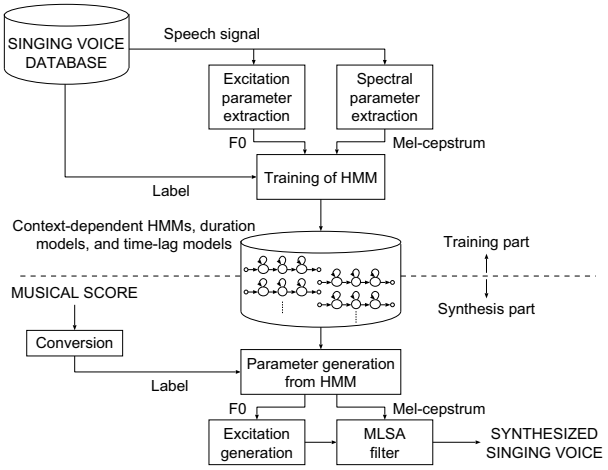

Figure 1: *Overview of the HMM-based singing voice synthesis system.*

In this paper, we propose singing voice synthesis based on DNNs and evaluate its effectiveness. In the proposed DNN-based singing voice synthesis, a DNN represents a mapping function from linguistic and musical-score features to acoustic features. Singing voice synthesis considers a larger number of contextual factors than standard TTS synthesis. Therefore, the strong mapping ability of DNNs is expected to largely improve singing voice quality. The reproducibility of each acoustic feature strongly depends on the training data because the DNN-based singing voice synthesis is a corpus-based approach. As for the pitch feature, which is one of the most important features in singing voice synthesis, it is difficult to generate a desirable F0 contour that closely follows the notes when the pitch contexts of the training data have poor coverage. This is a serious problem in singing voice synthesis systems. Therefore, a musical-note-level pitch normalization and linear-interpolation for both musical notes and extracted F_0 values for DNN-based singing voice synthesis are proposed to address the sparseness problem of pitch in a database.

This paper is organized as follows. Section 2 describes the HMM-based singing voice synthesis framework. Section 3 describes the DNN-based singing voice synthesis framework. Experiments are presented in Section 4. Concluding remarks are shown in Section 5.

2. HMM-based singing voice synthesis system

HMM-based singing voice synthesis is quite similar to HMM-based TTS synthesis [10, 11]. Figure 1 illustrates an overview

http://dx.doi.org/10.21437/Interspeech.2016-1027

of the HMM-based singing voice synthesis system [2, 3]. This approach consists of training and synthesis parts. In the training part, spectrum and excitation parameters (e.g. mel-cepstral coefficients and log F_0) are extracted from a singing voice database and then modeled by context-dependent HMMs. Context-dependent models of state durations are also estimated simultaneously [12]. The amount of available training data is normally not sufficient to robustly estimate all context-dependent HMMs because there is rarely enough data to cover all the context combinations. To address these problems, top-down decision-tree-based context clustering is widely used [13]. In this technique, the states of the context-dependent HMMs are grouped into "clusters" and the distribution parameters within each cluster are shared. HMMs are assigned to clusters by examining the context combination of each HMM through a binary decision tree, where one context-related binary question is associated with each non-terminal node. The decision tree is constructed by sequentially selecting the questions that yield the largest log likelihood gain of the training data. By using context-related questions and state parameter sharing, the unseen contexts and data sparsity problems are effectively addressed.

In the synthesis part, an arbitrarily given musical score including the lyrics to be synthesized is first converted into a context-dependent label sequence. Next, a state sequence corresponding to the song is constructed by concatenating the context-dependent HMMs in accordance with the label sequence. The state durations of the song HMMs are then determined by the state duration models. Finally, the speech parameters (spectrum and excitation) are generated from the HMMs by using a speech parameter generation algorithm [4], and a singing voice is synthesized from the generated singing voice parameters by using a vocoder.

3. DNN-based singing voice synthesis system

An overview of the proposed framework based on a DNN is shown in Fig. 2. In DNN-based singing voice synthesis, decision tree-clustered context dependent HMMs are replaced by a DNN. In the training part, a given musical score is first converted into a sequence of input features for the DNN. The input features consist of binary and numeric values representing linguistic contexts (e.g. the current phoneme identity, the number of phonemes in the current syllable, and durations of the current phoneme) and musical contexts (e.g. the key of the current measure and the absolute pitch of the current musical note). Output features of a DNN consist of spectral and excitation parameters and their dynamic features [14]. The input and output features are time-aligned frame-by-frame by well-trained HMMs. The weights of the DNN can be trained using pairs of the input and output features extracted from training data.

The quality of the synthesized singing voices strongly depends on training data because DNN-based singing voice synthesis systems are "corpus-based." Therefore, DNNs corresponding to contextual factors that rarely appear in training data cannot be well-trained. Although databases including various contextual factors should be used in DNN-based singing voice synthesis systems, it is almost impossible to cover all possible contextual factors because singing voices involve a huge number of them, e.g., keys, lyrics, dynamics, note positions, durations, and pitch. Pitch should be properly covered because it greatly affects the subjective quality of the synthesized singing

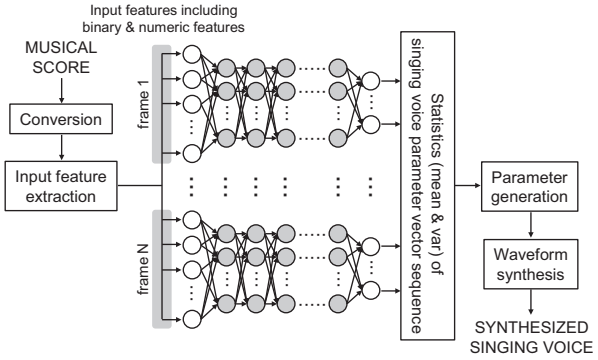

Figure 2: *Singing voice synthesis framework based on DNN. Note that phoneme alignments are given by well-trained HMMs in the training/synthesis part.*

voices. To address this problem, pitch adaptive training (PAT) has been proposed in HMM-based singing voice synthesis systems [15]. In PAT, the differences between log F_0 sequences extracted from waveforms and the pitch of musical notes can be modeled. Therefore, PAT enables singing voices including any pitch to be generated. However, PAT is difficult to directly apply to DNN-based singing voice synthesis systems. Therefore, we propose a musical-note-level pitch normalization technique for DNN-based singing voice synthesis. In the proposed pitch normalization technique, the differences between log F_0 extracted from waveforms and one calculated from musical notes are used as training data. By modeling the difference in log F_0 with a DNN, DNN-based singing voice synthesis systems can generate variable singing voices including any pitch. However, modeling differences in log F_0 presents a challenge: how to model log F_0 of singing voices including unvoiced frames and musical scores including musical rests. To appropriately define the differences in log F_0 in such unvoiced frames and musical rests, we introduce the zero-filling and linear interpolation techniques. Figures 3, 4, 5, and 6 illustrate the musical-note-level pitch normalization with the combinations of the linear-interpolation for the unvoiced frames of the singing voice and the musical rest on the musical score. Blue-colored regions of figures mean that it can not model the difference without linear interpolation. Figure 3 illustrates musical-note-level pitch normalization without interpolation. In this approach, the differences in the voiced frames on musical rests and unvoiced frames on musical notes are filled with zero. Therefore, log F_0 values in these frames cannot be effectively used. The linear-interpolation of log F_0 values can avoid the zero-filling (Figures 4, 5, and 6).

In the same fashion as the HMM-based approach, by setting the predicted output features from the DNN as mean vectors and pre-computed variances of the output features from all training data as covariance matrices, the speech parameter generation algorithm [4] can generate smooth trajectories of singing voice parameter features that satisfy both the statistics of static and dynamic features. Finally, a singing voice is synthesized directly from the generated parameters by using a vocoder. Note that the parameter generation and waveform synthesis modules of the DNN-based system can be shared with the HMM-based one, i.e. only the mapping module from context-dependent labels to statistics needs to be replaced.

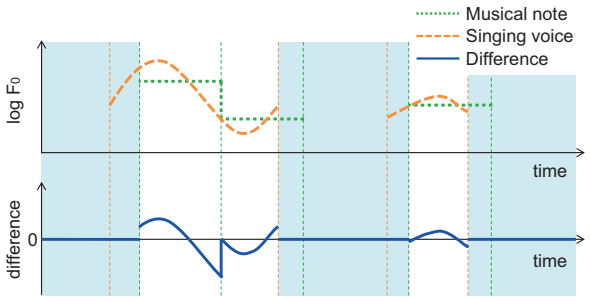

Figure 3: *Musical-note-level pitch normalization without interpolation.*

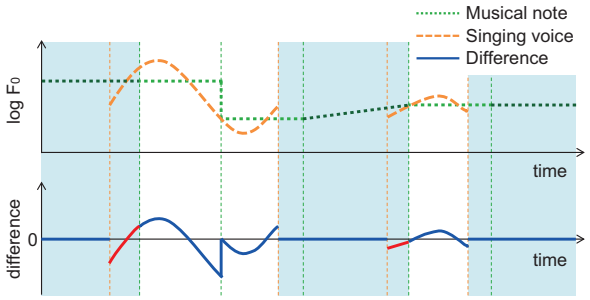

Figure 4: *Musical-note-level pitch normalization with linear-interpolation of the pitch of the musical note.*

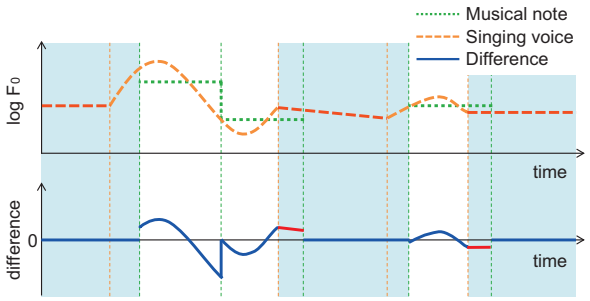

Figure 5: *Musical-note-level pitch normalization with linear-interpolation of the pitch of the singing voice.*

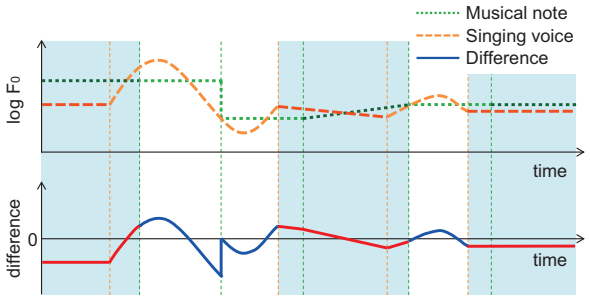

Figure 6: *Musical-note-level pitch normalization with linear-interpolation of the pitch of both the musical note and the singing voice.*

4. Experiments

4.1. Experimental conditions

To evaluate the effectiveness of the proposed method, objective and subjective experiments were conducted. A database consisting of 70 Japanese children's songs sung by a female singer was used. Sixty songs were used for training data, and the other 10 songs were used for evaluation. Singing voice signals were sampled at a rate of 48 kHz, and the number of quantization bits was 16. The acoustic feature vectors consisted of spectrum and excitation parameters. The spectrum parameter vectors consisted of 0th-49th STRAIGHT [16] mel-cepstral coefficients, their delta, and delta-delta coefficients. The excitation parameter vectors consisted of log F_0, its delta, and delta-delta.

For the baseline system based on HMMs, seven-state (including the beginning and ending null states), left-to-right, no-skip hidden semi-Markov models (HSMMs) [17] were used. To model log F_0 sequences consisting of voiced and unvoiced observations, a multi-space probability distribution (MSD) was used [18]. PAT was applied to cover possible pitch. The number of questions for the decision tree-based context clustering was 11440.

For the proposed system based on the DNN, the input features including 561 binary features for categorical contexts (e.g. the current phoneme identity, the key of the current measure) and 86 numerical features for numerical contexts (e.g. the number of phonemes in the current syllable, the absolute pitch of the current musical note) were used. In addition to the contexts-related input features, three numerical features for the position of the current frame in the current phoneme were used. The input and output features were time-aligned frame-by-frame by well-trained HMMs. The output features were basically the same as those used in HMM-based systems. To model log F_0 sequences by the DNN, the continuous F_0 with explicit voicing modeling approach [19] was used; voiced/unvoiced binary values were added to output features. The weights of the DNN were initialized randomly and then optimized to minimize the mean squared error between the output features of the training data and predicted values using a minibatch stochastic gradient descent (SGD)-based back-propagation algorithm. Both input and output features in the training data for the DNN were normalized; the input features were normalized to be within 0.00–1.00 on the basis of their minimum and maximum values in the training data, and the output features were normalized to be within 0.01–0.99 on the basis of their minimum and maximum values in the training data. The sigmoid activation function was used for hidden and output layers.

Singing voice parameters for the evaluation were generated from the HMMs/DNNs using the speech parameter generation algorithm [4]. From the generated singing voice parameters, singing voice waveforms were synthesized using the MLSA filter [20].

To objectively evaluate the performance of the HMM and DNN-based systems, mel-cepstral distortion (Mel-cd) [21] and root mean squared error of log F_0 (F_0-RMSE) were used. Combinations of the number of hidden layers (1, 2, 3, 4, or 5) and units per layer (128, 256, 512, 1024, or 2048) were decided by calculating Mel-cd and F_0-RMSE for each method.

4.2. Comparison of the pitch interpolation techniques

We compared the combinations of the presence or absence of linear-interpolation for the unvoiced frame of the singing voice and the musical rest on the musical score. The number of hidden

Table 1: *Comparison results of linear-interpolation method of log* F_0. ✓ *represents used linear-interpolation methods.*

Song F_0 interp			✓	✓
Score F_0 interp		✓		✓
F_0-RMSE [logHz]	0.04851	0.04847	**0.04777**	0.04784

Table 2: *Comparative approaches and combinations of the number of hidden layers and units per layer.*

		Hidden layers	Units per layer
HMM			
DNN (tuned for mgc)		3	1024
DNN (tuned for lf0)		4	1024
Separated DNN	lf0	1	1024
	mgc	3	1024

layers and units per layer that showed the smallest F_0-RMSE were 4 and 1024 in all combinations.

Table 1 shows the experimental results. It can be seen from the table that the musical-note-level pitch normalization with linear-interpolation of log F_0 sequences extracted from the singing voice achieved the lowest F_0-RMSE. The results also show that the linear-interpolation of log F_0 sequences extracted from the singing voices more strongly affects F_0-RMSE than the linear-interpolation of log F_0 sequences calculated from the musical note. That is, the difference between linear-interpolated log F_0 sequences and musical notes appropriately represents the singer's characteristics and the normalization using such difference is effective to generate songs that are not included in the pitch range of the training data.

4.3. Objective experiments

To compare the performance of the DNN-based systems with the HMM-based ones, objective experiments were conducted. Table 2 shows comparative systems and combinations of the number of hidden layers and units per layer. **HMM** is a conventional HMM-based singing voice synthesis system. **DNN (tuned for mgc)** is a method that uses the combination of the number of hidden layers and units per layer that indicated the smallest Mel-cd. **DNN (tuned for lf0)** is a method that uses the combination of the number of hidden layers and units per layer that indicated the smallest F_0-RMSE. **Separated DNN** is a method by which the spectrum DNN and the excitation DNN were trained individually. In all the DNN-based systems, the musical-note-level normalization that achieved the lowest F_0-RMSE in section 4.2 was applied to the output features of the excitation.

Table 3 shows the experimental results for Mel-cd and F_0-RMSE. The results show that the DNN-based systems consistently outperformed the HMM-based ones in terms of Mel-cd but obtained worse results in terms of log F_0 prediction.

4.4. Subjective experiments

To evaluate the naturalness of synthesized singing voices, a subjective listening test was conducted. In this subjective evaluation, the four systems compared in section 4.3 were evaluated. Ten Japanese subjects were asked to evaluate the naturalness of the synthesized singing voices on a mean opinion score (MOS)

Table 3: *Objective evaluation results: comparison of HMM-based and DNN-based singing voice synthesis.*

	HMM	DNN (tuned for mgc)	DNN (tuned for lf0)	Separated DNN
Mel-cd [dB]	5.162	5.027	5.054	**4.997**
F_0-RMSE [logHz]	**0.04423**	0.04856	0.04777	0.04729

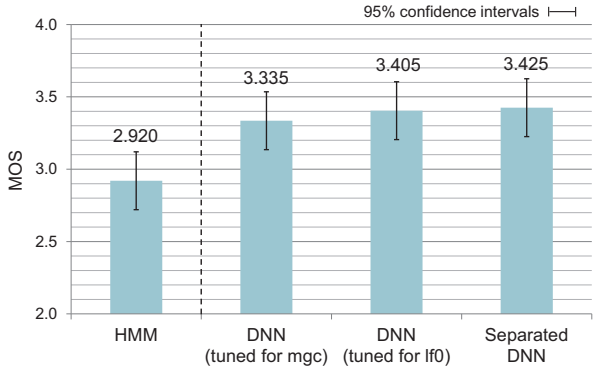

Figure 7: *Subjective evaluation results: comparison of HMM-based and DNN-based singing voice synthesis.*

on a scale from 1 (poor) to 5 (good). The subjects used headphones. Each subject was presented 20 musical phrases randomly selected from 10 songs.

Figure 7 shows the experimental results. This figure shows that all the DNN-based systems achieved significantly higher MOS than the HMM-based ones although there was no significant difference among the three DNN-based systems. The better prediction of mel-cepstral coefficients by the DNN-based systems seems to have contributed to their higher MOS. This result clearly shows the effectiveness of the proposed DNN-based singing voice synthesis.

5. Conclusions

DNN based singing voice synthesis was proposed and its effectiveness was evaluated in this paper. The relationship between musical scores and their acoustic features was modeled by a DNN in each frame. The objective experimental results show that the difference between the interpolated log F_0 sequences extracted from the waveform and the non-interpolated pitch of the musical note was effective for the excitation features of the DNN-based systems. Furthermore, the DNN-based systems outperformed the HMM-based systems in the subjective listening test. Future work will include the comparison with other architecture such as LSTM-RNN.

6. Acknowledgements

The research leading to these results was partly funded by the Hoso Bunka Foundation (HBF) and the Core Research for Evolutionary Science and Technology (CREST) from the Japan Science and Technology Agency (JST).

7. References

[1] H. Kenmochi and H. Ohshita, "VOCALOID-commercial singing synthesizer based on sample concatenation," *Proc. of Interspeech*, 2007.

[2] K. Saino, H. Zen, Y. Nankaku, A. Lee, and K. Tokuda, "An HMM-based Singing Voice Synthesis System," *Proc. of ICSLP*, pp. 1141–1144, 2006.

[3] K. Oura, A. Mase, T. Yamada, S. Muto, Y. Nankaku, and K. Tokuda, "Recent Development of the HMM-based Singing Voice Synthesis System - Sinsy," *Proc. of Speech Synthesis Workshop*, pp. 211–216, 2010.

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

[5] G. Hinton, L. Deng, D. Yu, G. E. Dahl, A. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. N. Sainath, B. Kingsbury, "Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups," *Signal Processing Magazine, Proc. of IEEE*, vol. 29, no. 6, pp. 82–97, 2012.

[6] A. Krizhevsky, I. Sutskever, and G. Hinton, "Imagenet classification with deep convolutional neural networks," *Advances in neural information processing systems*, 2012.

[7] H. Zen, A. Senior, and M. Schuster, "Statistical parametric speech synthesis using deep neural networks," *Proc. of ICASSP 2013*, pp. 7962–7966, 2013.

[8] H. Lu, S. King, and O. Watts, "Combining a vector space representation of linguistic context with a deep neural network for text-to-speech synthesis," *Proc. of ISCA SSW8*, pp. 281–285, 2013.

[9] Y. Qian, Y. Fan, H. Wenping, and F. K. Soong, "On the training aspects of deep neural network (DNN) for parametric TTS synthesis," *Proc. of ICASSP 2014*, pp. 3857–3861, 2014.

[10] T. Masuko, K. Tokuda, T. Kobayashi, and S. Imai, "Speech synthesis from HMMs using dynamic features," *Proc. of ICASSP*, pp. 389–392, 1996.

[11] T. Yoshimura, K. Tokuda, T. Masuko, T. Kobayashi, and T. Kitamura, "Simultaneous modeling of spectrum, pitch and duration in HMM-based speech synthesis," *Proc. of Eurospeech*, pp. 2347–2350, 1999.

[12] H. Zen, T. Masuko, K. Tokuda, T. Kobayashi, and T. Kitamura, "A Hidden Semi-Markov Model-Based Speech Synthesis System," *IEICE transactions on information and systems*, vol. 90, no. 5, pp. 825–834, 2007.

[13] S. Young, J. Odell, and P. Woodland, "Tree-based state tying for high accuracy acoustic modelling," *Proc. of the workshop on Human Language Technology, Association for Computational Linguistics*, pp. 307–312, 1994.

[14] S. Furui, "Speaker independent isolated word recognition using dynamic features of speech spectrum," *IEEE Transactions, Acoustics, Speech and Signal Processing*, vol. 34, no. 1, pp. 52–59, 1986.

[15] K. Oura, A. Mase, Y. Nankaku, and K. Tokuda, "Pitch adaptive training for HMM-based singing voice synthesis," *Proc. of ICASSP*, pp. 5377–5380, 2012.

[16] H. Kawahara, M. K. Ikuyo, and A. 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.

[17] H. Zen, K. Tokuda, T. Masuko, T. Kobayashi, and T. Kitamura, "A hidden semi-Markov model-based speech synthesis system," *IEICE Transactions on information and systems*, vol. 90, no. 5, pp. 825–834, 2007.

[18] K. Tokuda, T. Masuko, N. Miyazaki, and T. Kobayashi, "Multi-space probability distribution HMM," *IEICE Transactions on Information and Systems*, vol. 85, no. 3, pp. 455–464, 2002.

[19] K. Yu and S. Young, "Continuous F0 modelling for HMM based statistical parametric speech synthesis," *IEEE Transactions, Audio, Speech, and Language Processing*, vol. 19, no. 5, pp. 1071–1079, 2011.

[20] S. Imai, "Cepstral analysis synthesis on the mel frequency scale," *Proc. of ICASSP*, pp. 93–96, 1983.

[21] T. Toda, A. Black, and K. Tokuda, "Voice conversion based on maximum-likelihood estimation of spectral parameter trajectory," *IEEE Transactions, Audio, Speech, and Language Processing*, vol. 15, no. 8, pp. 2222–2235, 2007.