Recent Development of the DNN-based Singing Voice Synthesis System — Sinsy

Yukiya Hono*, Shumma Murata*, Kazuhiro Nakamura*†, Kei Hashimoto*,
Keiichiro Oura*, Yoshihiko Nankaku*, and Keiichi Tokuda*
* Department of Computer Science, Nagoya Institute of Technology, Japan
† Techno-Speech, Inc., Japan
E-mail: {hono, shunma37, nkazu, bonanza, uratec, nankaku, tokuda}@sp.nitech.ac.jp Tel: +81-52-735-5479

***Abstract*—This paper describes a singing voice synthesis system based on deep neural networks (DNNs) named Sinsy. Singing voice synthesis systems based on hidden Markov models (HMMs) have grown in the last decade. Recently, singing voice synthesis systems based on DNNs have been proposed. It has improved the naturalness of the synthesized singing voices. In this paper, we introduce several techniques, i.e., trajectory training, a vibrato model, and a time-lag model, into the DNN-based singing voice synthesis system to synthesize the high quality singing voices. Experimental results show that the DNN-based systems with these techniques outperformed the HMM-based systems. In addition, the present paper describes the details of the on-line service for singing voice synthesis.**

I. Introduction

A statistical parametric approach to text-to-speech (TTS) synthesis based on hidden Markov models (HMMs) has grown in popularity in the last decade [1], [2]. Context-dependent HMMs are estimated from speech databases in this approach, and speech waveforms are generated from the HMMs themselves. This framework makes it possible to model differences of voice characteristics, speaking styles, or emotions without recording large speech databases [3]–[5]. A singing voice synthesis system has also been proposed by applying the HMM-based approach [6]. In December 2009, we publicly released a free on-line singing voice synthesis service named "Sinsy" [7], [8]. It has been constructed using open-source software packages, e.g., HTS [9], hts engine API [10], SPTK [11], STRAIGHT [12], and the CrestMuseXML Toolkit [13]. Users can synthesize singing voices by uploading musical scores represented in MusicXML [14] to the website.

Recently, deep neural networks (DNNs) have attained significant improvement in various machine learning areas, e.g., speech recognition [15], speech synthesis [16], [17], and singing voice synthesis [18]. In DNN-based singing voice synthesis, a DNN works as an acoustic model that represents a mapping function from label sequences (e.g., phonetic, note key, and note length feature) to acoustic feature sequences. DNN-based acoustic models can represent complex dependencies between label sequences and acoustic feature sequences more efficiently than HMM-based acoustic models [19].

On the other hand, WaveNet [20], which is an autoregressive generative model that operates directly on audio waveforms, has been proposed. It has been shown that the WaveNet approach to the TTS system [21] has the potential to improve the naturalness of synthesized speech. Furthermore, singing voice synthesis based on a modified version of the WaveNet architecture, which models acoustic features instead of raw audio waveforms, has been proposed [22]. Because the models with a series structure that predicts excitation parameters and then predicts spectrum parameters using predicted excitation parameters were used, the prediction errors of excitation parameters cause negative impacts on the prediction of spectrum parameters in the latter part. There is no controllability of vibrato because vibrato modeling is enclosed in pitch modeling by WaveNet architecture. In addition, heuristic processing is used to make the total of phoneme durations fit note durations. Therefore, there is still room for improvement in singing voice synthesis systems.

In this paper, we propose a DNN-based singing voice synthesis system in which several techniques are used in order to improve the quality of the synthesized singing voices: i) trajectory training for DNN-based acoustic models, ii) a vibrato model based on DNNs, and iii) time-lag and duration models based on DNNs. These techniques can also address the problems mentioned above. Furthermore, the on-line service is updated so that the proposed DNN-based system can be used.

The rest of this paper is organized as follows. Section 2 gives an overview of the DNN-based singing voice synthesis system. Section 3 describes three techniques introduced in the proposed system. Experimental results in objective and subjective evaluations are given in Section 4. Details of the on-line service are presented in Section 5. Concluding remarks and future works are shown in Section 6.

II. DNN-based singing voice synthesis system

The DNN-based singing voice synthesis system is quite similar to the DNN-based TTS system [16]. However, there are distinct differences between them. This section overviews the conventional singing voice synthesis system [18] and then gives details of the differences between the DNN-based TTS synthesis system and the conventional DNN-based singing voice synthesis system.

A. System overview

Figure 1 gives an overview of the DNN-based singing voice synthesis system [18]. It consists of a training and synthesis

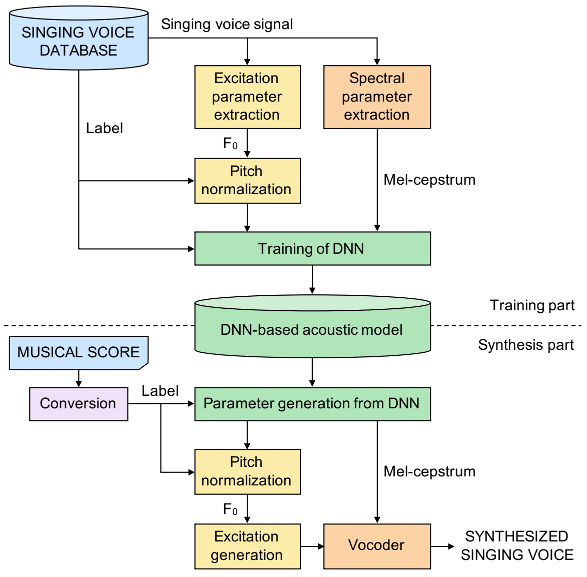


Fig. 1. Overview of the conventional DNN-based singing voice synthesis system.

part. In the training part, the spectrum (e.g., mel-cepstral coefficients) and excitation (e.g., fundamental frequencies: F_0s) are extracted from a singing voice database, and they are then modeled by a DNN. The input and output features of the DNN are time-aligned frame-by-frame by well-trained HMMs. In the synthesis part, an arbitrarily given musical score including lyrics to be synthesized is first converted to a context-dependent label sequence. Second, the label sequence is mapped to an acoustic feature sequence by the trained DNN using forward propagation. Third, the spectrum and excitation parameters are generated by the speech parameter generation algorithm [23]. Finally, a singing voice is synthesized directly from the generated spectrum and excitation parameters by using Vocoder.

B. Pitch normalization

The performance of statistical parametric approaches to singing voice synthesis heavily depends on the training data because this system is “corpus-based.” It is difficult to express contextual factors that hardly ever appear in the training data. 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 correctly covered because generated F_0 trajectories greatly affect the quality of the synthesized singing voices.

To address this problem, a musical-note-level pitch normalization technique has been proposed for DNN-based singing voice synthesis systems [18]. In this technique, the differences between the log F_0 sequences extracted from waveforms and the pitch of musical notes are modeled. This technique makes it possible for DNN-based singing voice synthesis systems to generate variable singing voices including any pitch. However, modeling differences in log F_0 present a challenge: how to model log F_0 of singing voices including unvoiced frames and musical scores including musical rests. In [18], all unvoiced frames and musical rests in musical scores are linear-interpolated and modeled as voiced frames.

III. Techniques Introduced into the DNN-based Singing Voice Synthesis System

A. Trajectory training for DNN-based acoustic modeling

In singing voice synthesis systems using DNN-based acoustic models [18], a single DNN is trained to represent the mapping function from musical features to acoustic features. In order to generate smooth parameter trajectories, not only static features but also dynamic features are modeled by DNN.

The acoustic feature vector $\boldsymbol{o}_t$ is the acoustic feature vector consisting of a D-dimensional static-feature vector $\boldsymbol{c}_t = [c_t(1), \ldots, c_t(D)]^\top$ and their dynamic feature vectors.

$$\boldsymbol{o}_t = [\boldsymbol{c}_t^\top, \Delta^{(1)}\boldsymbol{c}_t^\top, \Delta^{(2)}\boldsymbol{c}_t^\top]^\top. \tag{1}$$

The sequences of the acoustic feature vectors $\boldsymbol{o}$ and the static feature vectors $\boldsymbol{c}$, which represent a song, can be written in vector forms as follows:

$$\boldsymbol{o} = [\boldsymbol{o}_1^\top, \ldots, \boldsymbol{o}_t^\top, \ldots, \boldsymbol{o}_T^\top]^\top, \tag{2}$$
$$\boldsymbol{c} = [\boldsymbol{c}_1^\top, \ldots, \boldsymbol{c}_t^\top, \ldots, \boldsymbol{c}_T^\top]^\top, \tag{3}$$

where T is the number of frames included in a song. The relation between $\boldsymbol{o}$ and $\boldsymbol{c}$ can be represented by $\boldsymbol{o} = \boldsymbol{W}\boldsymbol{c}$, where $\boldsymbol{W}$ is a window matrix extending $\boldsymbol{c}$ to $\boldsymbol{o}$. The optimal static feature vector sequence is obtained by

$$\hat{\boldsymbol{c}} = \arg\max_{\boldsymbol{c}} P(\boldsymbol{o} \mid \boldsymbol{\lambda}) = \arg\max_{\boldsymbol{c}} \mathcal{N}(\boldsymbol{W}\boldsymbol{c} \mid \boldsymbol{\mu}, \boldsymbol{\Sigma}), \tag{4}$$

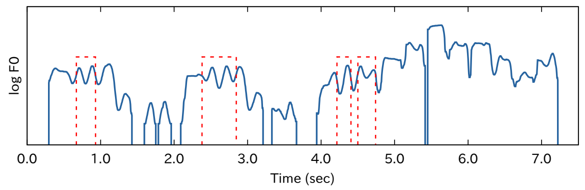
where $\boldsymbol{\lambda}$ is a parameter set and $\mathcal{N}(\cdot \mid \boldsymbol{\mu}, \boldsymbol{\Sigma})$ denotes the Gaussian distribution with a mean vector $\boldsymbol{\mu}$ and a covariance matrix $\boldsymbol{\Sigma}$. $\boldsymbol{\mu}$ is the output parameter from a trained neural network. The optimal static-feature sequence $\hat{\boldsymbol{c}}$ is given by

$$\hat{\boldsymbol{c}} = \boldsymbol{P}\boldsymbol{W}^\top\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}, \quad \boldsymbol{P} = \left(\boldsymbol{W}^\top\boldsymbol{\Sigma}^{-1}\boldsymbol{W}\right)^{-1}. \tag{5}$$

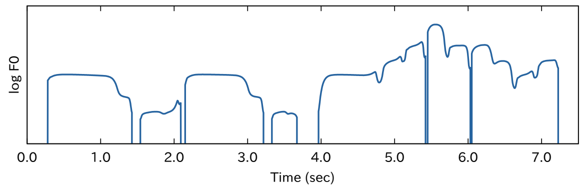
Training of the DNN aims to maximize the log likelihood function $\mathcal{L}$ as

$$\mathcal{L} = P(\boldsymbol{o} \mid \boldsymbol{\lambda}) = \mathcal{N}(\boldsymbol{o} \mid \boldsymbol{\mu}, \boldsymbol{\Sigma}) = \prod_{t=1}^{T} \mathcal{N}(\boldsymbol{o}_t \mid \boldsymbol{\mu}_t, \boldsymbol{\Sigma}_t). \tag{6}$$

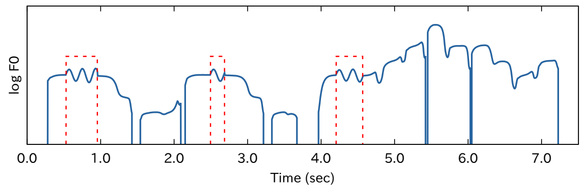
In the conventional DNN-based singing voice synthesis, although the frame-level objective function in (6) is used for training a DNN, the sequence-level objective function in (4) is used for parameter generation. To address this inconsistency between training and synthesis, a trajectory training method [24] is introduced into the training process of a DNN-based singing synthesis system. The conventional likelihood function in (6) can be reformulated as a trajectory likelihood function by imposing the explicit relationship between static



(a) Natural singing voice



(b) Synthesized singing voice without vibrato model



(c) Synthesized singing voice with vibrato model

Fig. 2. Example of F_0 sequence with vibrato.

and dynamic features, which is given by $\boldsymbol{o} = \boldsymbol{W}\boldsymbol{c}$ [25]. The trajectory likelihood function of $\boldsymbol{c}$ is then written as

$$\mathcal{L}_{Trj} = \frac{1}{Z} P(\boldsymbol{o} \mid \boldsymbol{\lambda}) = P(\boldsymbol{c} \mid \boldsymbol{\lambda}) = \mathcal{N}(\boldsymbol{c} \mid \bar{\boldsymbol{c}}, \boldsymbol{P}), \quad (7)$$

where Z is a normalization term. Inter-frame correlation is modeled by the covariance matrix $\boldsymbol{P}$ that is generally full. Note that the mean vector $\bar{\boldsymbol{c}}$ is equivalent to the generated static feature sequence shown by (5). The parameter set $\boldsymbol{\lambda}$ is estimated by maximizing the trajectory likelihood $\mathcal{L}_{Trj}$.

B. Vibrato model

Vibrato is one of the important singing techniques that should be modeled, even though it is not included in the musical score. Vibrato has been assumed as periodic fluctuations of only F_0 for the sake of simplicity, and it is modeled by sinusoid [26]. The vibrato $v(\cdot)$ of the t frame in the i-th vibrato section $[t_i^{(s)}, t_i^{(e)}]$ can be defined as

$$v\big(m_a(t), m_f(t), i\big) = m_a(t) \sin\Big(2\pi m_f(t) f_s \big(t - t_i^{(s)}\big)\Big), \quad (8)$$

where $m_a(t)$, $m_f(t)$, and f_s correspond to the F_0 amplitude of vibrato in cents, the F_0 frequency of vibrato in Hz, and frame shift. Two dimensional parameters, $m_a(t)$ and $m_f(t)$, are added to the acoustic feature vector. Figure 2 shows examples of the F_0 sequence extracted from the natural singing voice and synthesized singing voice. The area bounded by red broken lines shows the extracted and generated vibrato areas. It can be seen from the figure that the vibrato is accurately trained.

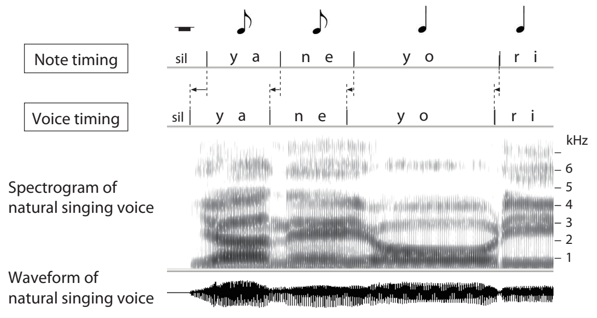

Note timing
Voice timing
sil
y a
n e
y o
r i
kHz
6
5
4
3
2
1
Spectrogram of natural singing voice
Waveform of natural singing voice

Fig. 3. Example of time-lag.

C. Time-lag and duration model

One of the unique features of the singing voice synthesis system is the time-lag model [6]. Because the rhythm or tempo of the music must not be ignored when the singing voice is synthesized, the start timing of the notes or phoneme durations in each note must be determined according to the musical score. In human singing voice, however, there are differences between the start timing of the notes and the singing voices as shown in Fig. 3. The start timing of the singing voice is often earlier than that of a corresponding note. Because this could be an important factor to the naturalness of the synthesized singing voice, they are modeled explicitly by time-lag models.

In the HMM-based singing voice synthesis system, the timings of each note are modeled with Gaussian distributions to overcome this problem. Hidden semi-Markov model (HSMM) [27]-based phoneme alignments are used to find the start timing of each note in the singing voice database. The time-lag models are then trained as context-dependent models, and decision tree-based context-clustering is applied to them in the same manner as the other models in the HMM-based system. In this paper, we propose the DNN-based time-lag model and duration model. Periods between the start timings of musical notes and the singing voice are modeled by DNNs. The phoneme durations at each note are also modeled by DNNs. In the synthesis part, first we determine the duration of each musical note from the given score including the lyrics to be synthesized. Next, the time-lags of each musical note are predicted using the time-lags model and the boundary of each musical note is determined. Then, the each phoneme duration is predicted by the duration model with consideration of the length of musical notes. The duration of the k-th phoneme in the n-th musical note is determined as follows:

$$d_{nk}^{(\mathrm{DNN})} = L_n \cdot \mu_{nk} \Big/ \sum_{k=1}^{K_n} \mu_{nk}, \quad (9)$$

where L_n is the length of the n-th musical note considering time-lag, K_n is the number of the phoneme in the n-th musical note, μ_{nk} is the output value of the DNN-based duration model at the k-th phoneme in the n-th musical note.

It is known that there is a difference in phoneme duration for each kind of phoneme (e.g., vowel, consonant, and pause).

Thus, we use a mixture density network (MDN) [28] as duration modeling to consider the phoneme duration variances. It should be noted that the MDN with 1 mixture component was used so that the MDN can represent a single Gaussian distribution like the state duration distributions in an HMM-based system. The phoneme durations considering variances are obtained by

$$d_{nk}^{(\mathrm{MDN})} = \mu_{nk} + \rho \cdot \sigma_{nk}^2, \tag{10}$$

$$\rho = \left(L_n - \sum_{k=1}^{K_n} \mu_{nk} \right) \Bigg/ \sum_{n=1}^{K} \sigma_{nk}^2, \tag{11}$$

where σ_{nk} is the variance corresponding to k-th phoneme in the n-th musical note, which is predicted by a trained MDNs.

IV. Experiment

A. Experimental Conditions

70 Japanese children's songs (total: 70 min) by female singer f001 were used for training. 60 songs were used for training, and the others were used for testing. Singing voice singles were sampled at 48 kHz and windowed with a 5-ms shift. The feature vectors consisted of 0-th through 49-th STRAIGHT mel-cepstral coefficients, log F_0 value, 24 dimension mel-cepstral analysis aperiodicity measures, and 2-dimension vibrato parameters. Mel-cepstral coefficients were extracted by the STRAIGHT [29]. The vibrato parameter vectors consisted of amplitude (cent) and frequency (Hz).

Five-state, left-to-right, no-skip HSMMs were used. The decision tree-based context clustering technique was separately applied to distributions for the spectrum, excitation, state duration, and time-lag. The spectrum stream was modeled with single multivariate Gaussian distributions. The excitation stream was modeled with multi-space probability distributions HSMMs (MSD-HSMMs) [30], each of which consisted of a Gaussian distribution for "voiced" frames and a discrete distribution for "unvoiced" frames. The vibrato stream was also modeled with MSD-HSMMs, each of which consisted of a Gaussian distribution for "vibrato" frames and a discrete distribution for "no-vibrato" frames. The MDL criterion [31] was used to control the size of the decision trees.

B. Objective evaluation of time-lag and duration modeling

To evaluate the accuracy of predicted time-lag and duration, root mean squared error in phoneme boundary (Boundary-RMSE) was used.

- **DT**: Conventional method that consists of decision tree-based clustered context-dependent time-lag model and state duration distribution of the HSMM [6]
- **DNN**: Time-lags and phoneme durations are modeled by DNN
- **MDN**: Time-lags are modeled by DNN and phoneme durations are modeled by MDN

The input features of the DNN-based time-lag and duration model was an 822-dimensional feature vector, consisting of 734 binary features for categorical linguistic contexts (e.g., the current phoneme identity) and 88 numerical features for numerical contexts (e.g., the number of phonemes in the current syllable). The output feature for the DNN-based time-lag model was a 1-dimensional time-lag value, and the output feature of the DNN-based duration model was a 1-dimensional phoneme duration value. The output feature for MDN-based duration model was a 1-dimensional phoneme duration mean and variance value. The architecture of the DNN was 3 hidden layers with 64 units per layer for time-lag modeling, and 3 hidden layers with 256 units per layer for duration modeling. The sigmoid activation function was used in the hidden layers, and the linear activation function was used in the output layer. The weights of the DNN and the MDN were initialized randomly, then the DNN was optimized by minimizing the mean squared error, and the MDN was optimized by maximizing the likelihood. For training the DNN and the MDN, a mini-batch stochastic gradient descent (SGD)-based back-propagation algorithm was used.

TABLE I
Objective evaluation results: comparison of three systems

	DT	**DNN**	**MDN**
Boundary-RMSE (frame)	12.74	11.51	11.21

Table I shows the experimental results. It can be seen that both **DNN** and **MDN** performed better in predicting time-lags and durations than **DT**. This result indicates that replacing the decision tree-based clustered models into DNN-based models is effective. Also, comparing **MDN** with **DNN**, **MDN** outperformed **DNN**. This suggests that considering variances is effective in estimating the phoneme boundary.

C. Objective evaluation of acoustic modeling

To objectively evaluate the performance of the systems, the mel-cepstral distortion (MCD), and root mean squared error in log F_0 (F_0-RMSE) were used. In this experience, the following three systems were compared.

- **HMM**: Conventional HMM-based singing voice synthesis system [32]
- **DNN**: Singing voice synthesis based on DNN trained by maximizing the objective function in (6)
- **TrjDNN**: Singing voice synthesis based on DNN trained by maximizing the trajectory function in (7)

The input features for the DNN-based systems was an 842-dimensional feature vector consisting of 734 binary features for categorical linguistic contexts, 108 numerical features for numerical contexts, and duration features including the duration of the current phoneme and the position of the current frame. The output feature was a 236-dimensional feature vector consisting of 50 mel-cepstral coefficients, log F_0 value, 25 dimensional mel-cepstral analysis aperiodicity measures, 2 dimensional vibrato parameters and their dynamic features (delta and delta-delta), and a voiced/unvoiced binary value and a vibrato/no-vibrato binary value. A single network that modeled every spectral, excitation, aperiodicity parameters, and vibrato parameters was trained. The architecture of the

TABLE II
OBJECTIVE EVALUATION RESULTS: COMPARISON OF THREE SYSTEMS

	HMM	**DNN**	**TrjDNN**
MCD (dB)	5.188	5.061	5.169
F_0-RMSE (cent)	78.46	79.74	81.34

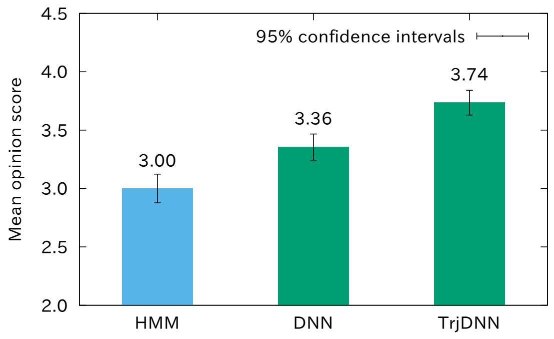


Fig. 4. Mean opinion scores of three singing voice synthesis systems.

DNN-based acoustic models was 3 hidden layer with 2048 units per layer. The sigmoid activation function was used in the hidden layers, and the linear activation function was used in the output layer. The weights of the DNN in **DNN** were initialized randomly, then they were optimized by maximizing the objective function $\mathcal{L}$ in (6). The weights of the DNN in **TrjDNN** were initialized by a trained DNN in **DNN**, then they were optimized by maximizing the objective function $\mathcal{L}_{Trj}$ in (7). For training the DNNs, a mini-batch SGD-based back-propagation algorithm was used. For **TrjDNN**, one musical phrase was used as one mini-batch in SGD-based training.

Table II shows the results of objective evaluation. From this table, it can be seen that the DNN-based system shows the better results than the HMM-based system **HMM** in MCD. This result indicates the effectiveness of the use of DNN for modeling spectrum parameters. However, the DNN-based system showed slightly worse F_0-RMSE than **HMM**. This is because linear-interpolation of log F_0 in pitch normalization has an influence on predicted F_0 parameter sequences.

D. Subjective evaluation

To evaluate the naturalness of the synthesized singing voice, a subjective listening test was conducted. In this experiment, **HMM**, **DNN**, and **TrjDNN** were compared. Time-lags and phoneme durations are modeled by **DT** in **HMM** and **MDN** in **DNN** and **TrjDNN**. The naturalness of the synthesized singing voice was assessed by the mean opinion score (MOS) test method. The subjects were ten Japanese students in our research group. Twenty musical phrases were chosen at random from the test songs. In the MOS test, after listening to each test sample, the subjects were asked to assign the sample a five-point naturalness score (5: natural – 1: poor).

Figure 4 shows the results of subjective evaluation scores. The DNN-based systems, **DNN** and **TrjDNN**, outperformed **HMM**. This result indicates the effectiveness of the use of DNNs for modeling singing voices. Comparing **DNN** and **TrjDNN**, **TrjDNN** performs better than **DNN**. This result indicates that the naturalness of a synthesized singing voice is improved by introducing the parameter generation process into the training of DNNs.

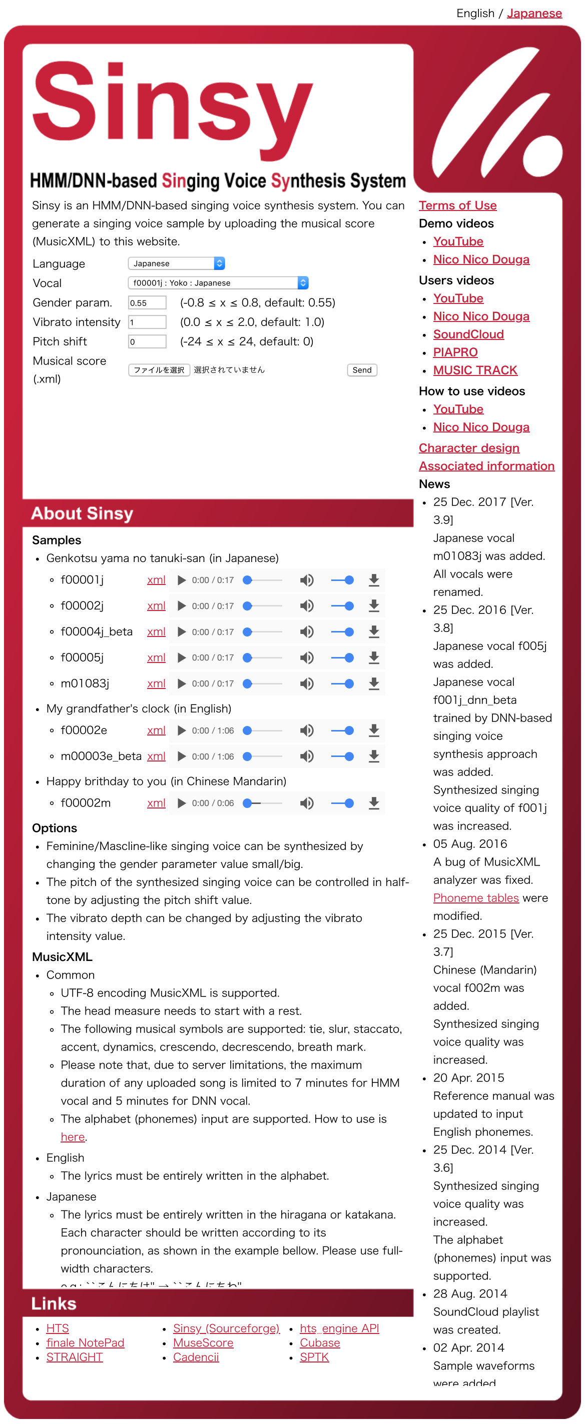
TABLE III
DIFFERENCE OF SYSTEMS

	2009	**2018**
Number of singers	Japanese: 1	Japanese: 5 English: 2 Mandarin: 1
Supported pitch range	G3 to F5	All pitches
Maximum length of musical score	5 minutes	HMM: 7 minutes DNN: 5 minutes

V. DETAILS OF ON-LINE SINGING SYNTHESIS SERVICE

The on-line singing synthesis service "Sinsy" was released in December 2009. Users can easily change the timbre, pitch, and strength of the vibrato. A web-based user interface [7] was used for Sinsy (Fig. 5). One of the reasons for this was that Sinsy could be frequently updated. We have updated various functions since the first release. Table III shows the differences of the system between 2009 and 2018. One Japanese singer's voice was offered at the beginning of the service in 2009. In 2018, 5 Japanese, 2 English, and 1 Mandarin singer were provided. There was a restriction of the pitch range because a pitch that barely ever appeared in the training data could not be synthesized in the statistical parametric method. Therefore, MusicXML files that exceeded the range of pitches from G3 to F5 were rejected in 2009. The limitation of the pitch range was abolished because all pitches can be synthesized by pitch adaptive training in an HMM-based system [32] and pitch normalization in the DNN-based system that was described in Section II-A. Another limitation was the length of the synthesized singing voice. One of the most attractive features of HMM-based singing voice synthesis is its small computational cost in the synthesis part. However, this system is vulnerable to frequent access or long songs because singing voices are synthesized on the web server. Therefore, MusicXML files that exceed 7 minutes for HMM vocal and 5 minutes for DNN vocal are rejected.

The number of posts per year was 4,045 in 2010; however, it was 21,921 in 2017. The rate at which waveforms were properly synthesized be utilizing users' MusicXML files that were uploaded to Sinsy in 2010 was about 70%. The other 30% included errors, other than those created by the restrictions, that prevented conversion into MusicXML files because of the differences in MusicXML files generated by various tools. On the other hand, the rejection rate was about 5% in 2017. This is because of improvement of analysis accuracy of MusicXML and the relaxation of restrictions on musical scores.



English / Japanese

Sinsy

HMM/DNN-based Singing Voice Synthesis System

Sinsy is an HMM/DNN-based singing voice synthesis system. You can generate a singing voice sample by uploading the musical score (MusicXML) to this website.

Language	Japanese	
Vocal	f00001j : Yoko : Japanese	
Gender param.	0.55	(-0.8 ≤ x ≤ 0.8, default: 0.55)
Vibrato intensity	1	(0.0 ≤ x ≤ 2.0, default: 1.0)
Pitch shift	0	(-24 ≤ x ≤ 24, default: 0)
Musical score (.xml)	ファイルを選択 選択されていません	Send

About Sinsy

Samples

- Genkotsu yama no tanuki-san (in Japanese)
  - f00001j xml 0:00 / 0:17
  - f00002j xml 0:00 / 0:17
  - f00004j_beta xml 0:00 / 0:17
  - f00005j xml 0:00 / 0:17
  - m01083j xml 0:00 / 0:17
- My grandfather's clock (in English)
  - f00002e xml 0:00 / 1:06
  - m00003e_beta xml 0:00 / 1:06
- Happy brithday to you (in Chinese Mandarin)
  - f00002m xml 0:00 / 0:06

Options

- Feminine/Mascline-like singing voice can be synthesized by changing the gender parameter value small/big.
- The pitch of the synthesized singing voice can be controlled in half-tone by adjusting the pitch shift value.
- The vibrato depth can be changed by adjusting the vibrato intensity value.

MusicXML

- Common
  - UTF-8 encoding MusicXML is supported.
  - The head measure needs to start with a rest.
  - The following musical symbols are supported: tie, slur, staccato, accent, dynamics, crescendo, decrescendo, breath mark.
  - Please note that, due to server limitations, the maximum duration of any uploaded song is limited to 7 minutes for HMM vocal and 5 minutes for DNN vocal.
  - The alphabet (phonemes) input are supported. How to use is here.
- English
  - The lyrics must be entirely written in the alphabet.
- Japanese
  - The lyrics must be entirely written in the hiragana or katakana. Each character should be written according to its pronounciation, as shown in the example bellow. Please use full-width characters.

Links

- HTS
- finale NotePad
- STRAIGHT
- Sinsy (Sourceforge)
- MuseScore
- Cadencii
- hts_engine API
- Cubase
- SPTK

Terms of Use

Demo videos

- YouTube
- Nico Nico Douga

Users videos

- YouTube
- Nico Nico Douga
- SoundCloud
- PIAPRO
- MUSIC TRACK

How to use videos

- YouTube
- Nico Nico Douga

Character design

Associated information

News

- 25 Dec. 2017 [Ver. 3.9]
  Japanese vocal m01083j was added.
  All vocals were renamed.
- 25 Dec. 2016 [Ver. 3.8]
  Japanese vocal f005j was added.
  Japanese vocal f001j_dnn_beta trained by DNN-based singing voice synthesis approach was added.
  Synthesized singing voice quality of f001j was increased.
- 05 Aug. 2016
  A bug of MusicXML analyzer was fixed.
  Phoneme tables were modified.
- 25 Dec. 2015 [Ver. 3.7]
  Chinese (Mandarin) vocal f002m was added.
  Synthesized singing voice quality was increased.
- 20 Apr. 2015
  Reference manual was updated to input English phonemes.
- 25 Dec. 2014 [Ver. 3.6]
  Synthesized singing voice quality was increased.
  The alphabet (phonemes) input was supported.
- 28 Aug. 2014
  SoundCloud playlist was created.
- 02 Apr. 2014
  Sample waveforms

Fig. 5. Sinsy demonstration website

VI. Conclusions

This paper described recent developments in the DNN-based singing voice synthesis system (Sinsy). To obtain natural singing voices, we introduced several specific techniques for DNN-based singing voice synthesis: trajectory training, a vibrato model, and a time-lag and duration model. Experimental results show that the proposed system gives a more natural synthesized singing voice. In addition, the details of the on-line service with the proposed system were described. Future work includes conducting the model structure of recurrent units and the expansion of singing voice synthesis to other languages.

Acknowledgments

This research was partly funded by JSPS KAKENHI Grant Number JP18K11163, CASIO SCIENCE PROMOTION FOUNDATION, and Microsoft Development Co., Ltd.

References

[1] K. Tokuda, Y. Nankaku, T. Toda, H. Zen, J. Yamagishi, and K. Oura, “Speech synthesis based on hidden Markov models,” *Proceedings of the IEEE*, vol. 101, no. 5, pp. 1234–1252, 2013.

[2] H. Zen, K. Tokuda, and A. W. Black, “Statistical parametric speech synthesis,” *Speech Communication*, vol. 51, no. 11, pp. 1039–1064, 2009.

[3] J. Yamagishi and T. Kobayashi, “Average-voice-based speech synthesis using HSMM-based speaker adaptation and adaptive training,” *IEICE Transactions on Information and Systems*, vol. E90-D, no. 2, pp. 533–543, 2007.

[4] T. Yoshimura, T. Masuko, K. Tokuda, T. Kobayashi, and T. Kitamura, “Speaker interpolation in HMM-based speech synthesis system,” *Proceedings of Eurospeech 1997*, pp. 2523–2526, 1997.

[5] K. Shichiri, A. Sawabe, T. Yoshimura, K. Tokuda, T. Masuko *et al.*, “Eigenvoices for hmm-based speech synthesis,” *Proceedings of ICSLP 2002*, 2002.

[6] K. Saino, H. Zen, Y. Nankaku, A. Lee, and K. Tokuda, “An HMM-based singing voice synthesis system,” *Proceedings of Interspeech 2006*, pp. 1141–1144, 2006.

[7] “HMM/DNN-based singing voice synthesis system (Sinsy),” http://www.sinsy.jp/.

[8] K. Oura, A. Mase, T. Yamada, S. Muto, Y. Nankaku, and K. Tokuda, “Recent development of the HMM-based singing voice synthesis system–Sinsy,” *Proceedings of ISCA SSW7*, pp. 211–216, 2010.

[9] “HMM-based speech synthesis system (HTS),” http://hts.sp.nitech.ac.jp/.

[10] “HMM-based speech synthesis engine (hts_engine API),” http://hts.sp.nitech.ac.jp/.

[11] “Speech signal processing toolkit (SPTK),” http://sp-tk.sourceforge.net/.

[12] “A speech analysis, modication and synthesis system (STRAIGHT),” http://www.wakayama-u.ac.jp/kawahara/STRAIGHTadv/index_e.html.

[13] “CrestMuseXML toolkit (CMX),” http://cmx.sourceforge.jp.

[14] “MusicXML definition,” http://musicxml.org.

[15] G. Hinton, L. Deng, D. Yu, G. E. Dahl, A. Mohamed *et al.*, “Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups,” *IEEE Signal Processing Magazine*, vol. 29, no. 6, pp. 82–97, 2012.

[16] H. Zen, A. Senior, and M. Schuster, “Statistical parametric speech synthesis using deep neural networks,” *Proceedings of ICASSP 2013*, pp. 7962–7966, 2013.

[17] Y. Qian, Y. Fan, W. Hu, and F. K. Soong, “On the training aspects of deep neural network (DNN) for parametric TTS synthesis,” *Proceedings of ICASSP 2014*, pp. 3829–3833, 2014.

[18] M. Nishimura, K. Hashimoto, K. Oura, Y. Nankaku, and K. Tokuda, “Singing voice synthesis based on deep neural networks,” *Proceedings of Interspeech 2016*, pp. 2478–2482, 2016.

[19] O. Watts, G. E. Henter, T. Merritt, Z. Wu, and S. King, “From HMMs to DNNs: where do the improvements come from?” *Proceedings of ICASSP 2016*, pp. 5505–5509, 2016.

[20] A. van den Oord, S. Dieleman, H. Zen, K. Simonyan, O. Vinyals *et al.*, "Wavenet: A generative model for raw audio," *arXiv preprint arXiv:1609.03499*, 2016.
[21] S. Ö. Arik, M. Chrzanowski, A. Coates, G. Diamos, A. Gibiansky *et al.*, "Deep voice: Real-time neural text-to-speech," *arXiv preprint arXiv:1705.08947*, 2017.
[22] M. Blaauw and J. Bonada, "A neural parametric singing synthesizer modeling timbre and expression from natural songs," *Applied Sciences*, vol. 7, no. 12, 2017.
[23] K. Tokuda, T. Yoshimura, T. Masuko, T. Kobayashi, and T. Kitamura, "Speech parameter generation algorithms for HMM-based speech synthesis," *Proceedings of ICASSP 2000*, vol. 3, pp. 1315–1318, 2000.
[24] K. Hashimoto, K. Oura, Y. Nankaku, and K. Tokuda, "Trajectory training considering global variance for speech synthesis based on neural networks," *Proceedings of ICASSP 2016*, pp. 5600–5604, 2016.
[25] H. Zen, K. Tokuda, and T. Kitamura, "Reformulating the HMM as a trajectory model by imposing explicit relationships between static and dynamic feature vector sequences," *Computer Speech & Language*, vol. 21, no. 1, pp. 153–173, 2007.
[26] T. Yamada, S. Muto, Y. Nankaku, S. Sako, and K. Tokuda, "Vibrato modeling for HMM-based singing voice synthesis," *Proceedings of Information Processing Society of Japan*, vol. 2009-NUS-80, no. 5, pp. 1–6, 2009 (in Japanese).
[27] H. Zen, K. Tokuda, T. Masuko, T. Kobayasih, and T. Kitamura, "A hidden semi-Markov model-based speech synthesis system," *IEICE Transactions on Information and Systems*, vol. E90-D, no. 5, pp. 825–834, 2007.
[28] C. M. Bishop, "Mixture density networks," Neural Computing Research Group, Aston University, Tech. Rep. NCRG/94/004, 1994.
[29] 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.
[30] K. Tokuda, T. Masuko, N. Miyazaki, and T. Kobayashi, "Hidden Markov models based on multi-space probability distribution for pitch pattern modeling," *Processings of ICASSP 1999*, vol. 1, pp. 229–232, 1999.
[31] K. Shinoda and T. Watanabe, "MDL-based context-dependent subword modeling for speech recognition." *The Journal of The Acoustical Society of Japan (e)*, vol. 21, no. 2, 2000.
[32] K. Oura, A. Mase, Y. Nankaku, and K. Tokuda, "Pitch adaptive training for HMM-based singing voice synthesis," *Proceedings of ICASSP 2012*, pp. 5377–5380, 2012.