TRAJECTORY TRAINING CONSIDERING GLOBAL VARIANCE FOR SPEECH SYNTHESIS BASED ON NEURAL NETWORKS

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

Department of Scientific and Engineering Simulation, Nagoya Institute of Technology, Nagoya, Japan

ABSTRACT

This paper proposes a new training method of deep neural networks (DNNs) for statistical parametric speech synthesis. DNNs are recently used as acoustic models that represent mapping functions from linguistic features to acoustic features in statistical parametric speech synthesis. There are problems to be solved in conventional DNN-based speech synthesis: 1) the inconsistency between the training and synthesis criteria; and 2) the over-smoothing of the generated parameter trajectories. In this paper, we introduce the parameter trajectory generation process considering the global variance (GV) into the training of DNNs. A unified framework which consistently uses the same criterion in both training and synthesis can be obtained and the model parameters are optimized for parameter generation considering the GV in the proposed method. Experimental results show that the proposed method outperforms the conventional method in the naturalness of synthesized speech.

Index Terms— Speech synthesis, statistical model, neural network, trajectory model, global variance

1. INTRODUCTION

Statistical parametric speech synthesis has grown in popularity in the last decade [1]. In statistical parametric speech synthesis, relation between acoustic features (e.g., spectral and excitation features) and linguistic features (e.g., phonetic, syllabic, and grammatical features) is modeled by statistical models, which are generally called acoustic models. Effective acoustic modeling is one of the most critical problems for statistical parametric speech synthesis.

Hidden Markov models (HMMs) have been widely used as acoustic models in statistical parametric speech synthesis [2]. Acoustic features and duration of speech are simultaneously modeled with HMMs [3] and decision tree based context clustering [4] is widely used to effectively handle linguistic features in HMM-based speech synthesis. Smooth speech parameter sequences, i.e., trajectories, can be generated by using dynamic features as constraints [5]. However, synthesized speech sounds muffled and the quality of the synthesized speech still does not reached that of natural speech. Recently, deep neural networks (DNNs), which are feed-forward artificial neural networks with many hidden layers, have achieved significant improvement in automatic speech recognition [6]. Motivated by the success of DNNs in speech recognition, DNNs have been introduced to statistical parametric speech synthesis [7, 8, 9]. A single DNN is trained to represent the mapping function from linguistic features to acoustic features, which is modeled by decision tree-clustered context dependent HMMs in HMM-based approach. In the generation process of DNN-based speech synthesis, the linguistic features extracted from given text to be synthesized are mapped to acoustic features by the trained DNN. DNN-based acoustic models can represent complex mapping functions from linguistic features to acoustic features and DNN-based speech synthesis shows the potential to produce more naturally-sounding synthesized speech.

This paper focuses on two problems in DNN-based speech synthesis: 1) inconsistency between the training and synthesis criteria; and 2) over-smoothing of the generated parameter trajectories. In the training process of DNNs, a frame-by-frame independence is generally assumed and frame-level objective functions are widely used to train DNNs. However, in the synthesis process, objective functions with respect to static feature sequences is used to generate speech parameter trajectories. Consequently, there is an inconsistency between the training and synthesis criteria and DNNs are not optimized for parameter generation. In addition, the static feature vectors generated by the traditional generation process are usually over-smoothed and this is one of the main factors causing the muffled effect in statistical parametric synthesized speech. For improving the synthetic speech quality, Toda and Tokuda [10] have introduced a new criterion on a higher order moment called the global variance (GV), which is the variance of the static feature vectors calculated over a time sequence (e.g., over an utterance), into the parameter generation process. It has been reported that synthetic speech quality can be significantly improved by generating the parameter trajectory while keeping its GV close to the natural one [10, 11].

To address these problems, in this paper, we introduce a trajectory training method considering the GV, which has been proposed for HMM-based speech synthesis [12], into the DNN training. DNNs are optimized in the sense of maximum likelihood subject to a constraint on the GV of the generated parameter trajectory. Consequently, a unified framework which consistently uses the same criterion in both training and synthesis is obtained and the over-smoothing problem is alleviated. In this paper, the proposed method is compared with the conventional DNN training method on objective and subjective evaluations.

The rest of this paper is organized as follows. Section 2 and 3 describe statistical parametric speech synthesis based on DNNs and the proposed training method, respectively. The experimental conditions and results are given in Section 4. Concluding remarks and future work are presented in Section 5.

2. STATISTICAL PARAMETRIC SPEECH SYNTHESIS USING NEURAL NETWORKS

In statistical parametric speech synthesis using DNN-based acoustic models [7, 8, 9], a single DNN is trained to represent a mapping function from linguistic features to acoustic features including spectral and excitation parameters with their dynamic features. In the generation process, the linguistic features extracted from given text to be synthesized are mapped to acoustic features by the trained DNN using forward propagation. Figure 1 shows an overview of generation procedures in DNN-based speech synthesis. Although it

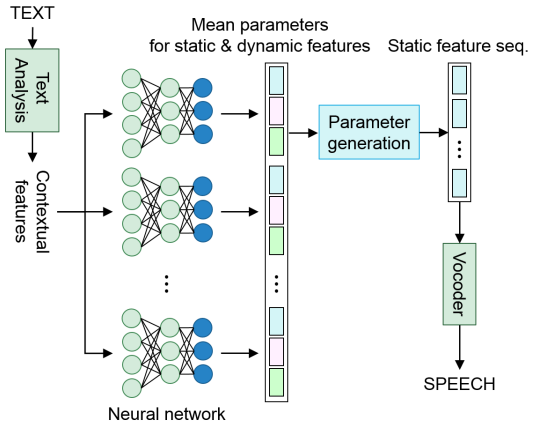

Fig. 1. An overview of generation procedures in statistical parametric speech synthesis based on neural networks.

is possible to generate static acoustic features directly by the DNN, it has been reported that the speech parameter trajectories generated by the parameter generation algorithm considering explicit relationship between static and dynamic features shows better performance [13]. Therefore, in this work, the parameter generation is applied for generating smooth speech parameter trajectories.

A speech parameter vector $\boldsymbol{o}_t$ consists 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 speech parameter vectors and the static feature vectors, which represent an utterance, 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 an utterance. 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 the static feature vector sequence $\boldsymbol{c}$ to the speech parameter vector sequence $\boldsymbol{o}$. The optimal static feature vector sequence is obtained by

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

where $\boldsymbol{\lambda}$ is a parameter set and $\mathcal{N}(\cdot|\boldsymbol{\mu}, \boldsymbol{\Sigma})$ denotes the Gaussian distribution with a mean vector $\boldsymbol{\mu}$ and a covariance matrix $\boldsymbol{\Sigma}$. The mean vector $\boldsymbol{\mu}$ and the covariance matrix $\boldsymbol{\Sigma}$ are given by

$$\boldsymbol{\mu} = \left[\boldsymbol{\mu}_1^\top, \ldots, \boldsymbol{\mu}_t^\top, \ldots, \boldsymbol{\mu}_T^\top\right]^\top \tag{5}$$

$$\boldsymbol{\Sigma} = \mathrm{diag}\left[\boldsymbol{\Sigma}_1, \ldots, \boldsymbol{\Sigma}_t, \ldots, \boldsymbol{\Sigma}_T\right] \tag{6}$$

The optimal static feature sequence $\hat{\boldsymbol{c}}$ is given by

$$\hat{\boldsymbol{c}} = \left(\boldsymbol{W}^\top\boldsymbol{\Sigma}^{-1}\boldsymbol{W}\right)^{-1}\boldsymbol{W}^\top\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu} = \boldsymbol{P}\boldsymbol{r} \tag{7}$$

where

$$\boldsymbol{P} = \left(\boldsymbol{W}^\top\boldsymbol{\Sigma}^{-1}\boldsymbol{W}\right)^{-1}, \qquad \boldsymbol{r} = \boldsymbol{W}^\top\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu} \tag{8}$$

As a result, smooth static feature trajectories can be obtained by using dynamic features as constraints. In DNN-based speech synthesis, the mean vector at frame t, $\boldsymbol{\mu}_t$, is obtained from a trained neural network and a linguistic feature vector at time t, $\boldsymbol{x}_t$, as follows:

$$\boldsymbol{\mu}_t = g(\boldsymbol{x}_t|\boldsymbol{\lambda}_{NN}) \tag{9}$$

where $g(\cdot|\boldsymbol{\lambda}_{NN})$ is a non-linear mapping function represented by a neural network $\boldsymbol{\lambda}_{NN}$. A covariance matrix is usually independent of linguistic features, i.e., a globally tied covariance matrix $\boldsymbol{\Sigma}_g$ is used, in DNN-based speech synthesis.

Assuming that outputs of a neural network are used as mean parameters in a statistical model, a objective function can be defined as

$$\mathcal{L} = P(\boldsymbol{o}|\boldsymbol{\lambda}) = \mathcal{N}(\boldsymbol{o}|\boldsymbol{\mu}, \boldsymbol{\Sigma}) = \prod_{t=1}^{T}\mathcal{N}(\boldsymbol{o}_t|\boldsymbol{\mu}_t, \boldsymbol{\Sigma}_g) \tag{10}$$

The parameter set $\boldsymbol{\lambda}$, which consists of the parameter of the neural network and the covariance matrix $\boldsymbol{\Sigma}$, is optimized in the sense of maximum likelihood as follows:

$$\hat{\boldsymbol{\lambda}} = \arg\max_{\boldsymbol{\lambda}} P(\boldsymbol{o}|\boldsymbol{\lambda}) = \prod_{t=1}^{T}\mathcal{N}(\boldsymbol{o}_t|\boldsymbol{\mu}_t, \boldsymbol{\Sigma}_g) \tag{11}$$

If an identity matrix is used as the covariance matrix, maximization of the objective function $\mathcal{L}$ is equivalent to minimization of the conventional frame-level mean square errors. The neural network can be trained by standard back-propagation using the gradient of the mean vector.

3. TRAJECTORY TRAINING METHOD CONSIDERING GLOBAL VARIANCE FOR DNNS

3.1. Trajectory training

In the conventional DNN-based speech synthesis framework, although the frame-level objective function is used for DNN training, the sequence-level objective function is used for parameter generation. To address this inconsistency between training and synthesis, a trajectory training method is introduced into the training process of DNNs.

The traditional likelihood function in Eq. (10) can be reformulated as a trajectory likelihood function by imposing explicit relationship between static and dynamic features, which is given by $\boldsymbol{o} = \boldsymbol{W}\boldsymbol{c}$ [14]. The trajectory likelihood function of $\boldsymbol{c}$ is then written as

$$\mathcal{L}_{Trj} = \frac{1}{Z}P(\boldsymbol{o}|\boldsymbol{\lambda}) = P(\boldsymbol{c}|\boldsymbol{\lambda}) = \mathcal{N}(\boldsymbol{c}|\bar{\boldsymbol{c}}, \boldsymbol{P}) \tag{12}$$

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 Eq. (7).

The parameter set $\boldsymbol{\lambda}$ is estimated by maximizing the trajectory likelihood $\mathcal{L}_{Trj}$. The gradients of mean vector $\boldsymbol{\mu}$ and covariance matrix $\boldsymbol{\Sigma}$ can be calculated as follows

$$\frac{\partial\mathcal{L}_{Trj}}{\partial\boldsymbol{\mu}} = \boldsymbol{\Sigma}^{-1}\boldsymbol{W}(\boldsymbol{c} - \bar{\boldsymbol{c}}) \tag{13}$$

$$\frac{\partial\mathcal{L}_{Trj}}{\partial\boldsymbol{\Sigma}^{-1}} = \frac{1}{2}\mathrm{diag}\big[\boldsymbol{W}(\boldsymbol{P} + \bar{\boldsymbol{c}}\bar{\boldsymbol{c}}^\top - \boldsymbol{c}\boldsymbol{c}^\top)\boldsymbol{W}^\top - 2\boldsymbol{\mu}(\bar{\boldsymbol{c}} - \boldsymbol{c})^\top\boldsymbol{W}^\top\big] \tag{14}$$

The parameters of neural network are updated by the back-propagation algorithm using the gradient in Eq. (13). The computation of gradients for the parameters of the neural network in lower layers is the same as the calculation of gradients for standard neural networks. The covariance matrix $\mathbf{\Sigma}$ is iteratively updated using the gradient in Eq. (14).

3.2. Trajectory training considering GV

To address the over-smoothing problem of generated parameter trajectories, the concept of parameter generation considering the GV is introduced into the training of DNNs. The proposed objective function $\mathcal{L}_{GVTrj}$ is given by

$$\begin{aligned}\mathcal{L}_{GVTrj} &= P(\boldsymbol{c}|\boldsymbol{\lambda})P(\boldsymbol{v}(\boldsymbol{c})|\boldsymbol{\lambda},\boldsymbol{\lambda}_v)^{wT} \\ &= \mathcal{N}(\boldsymbol{c}|\bar{\boldsymbol{c}},\boldsymbol{P})\mathcal{N}(\boldsymbol{v}(\boldsymbol{c})|\boldsymbol{v}(\bar{\boldsymbol{c}}),\mathbf{\Sigma}_v)^{wT} \end{aligned} \tag{15}$$

where $\boldsymbol{v}(\boldsymbol{c}) = [v(1),\ldots,v(D)]^\top$ is a GV vector of the static feature vector sequence $\boldsymbol{c}$. The GV vector is calculated utterance by utterance as follows:

$$v(d) = \frac{1}{T}\sum_{t=1}^{T}(c_t(d) - \langle c(d)\rangle)^2 \tag{16}$$

$$\langle c(d)\rangle = \frac{1}{T}\sum_{t=1}^{T} c_t(d) \tag{17}$$

where d is an index of the feature dimension. The mean vector of the probability density for the GV, $\boldsymbol{v}(\bar{\boldsymbol{c}})$, is defined as the GV of the mean vector of the trajectory likelihood function in Eq. (12), which is equivalent to the GV of the generated parameters shown by Eq. (7). The GV likelihood $P(\boldsymbol{v}(\boldsymbol{c})|\boldsymbol{\lambda},\boldsymbol{\lambda}_v)$ works as a penalty term to make the GV of the generated parameters close to that of the natural ones. The balance between the two likelihoods $P(\boldsymbol{c}|\boldsymbol{\lambda})$ and $P(\boldsymbol{v}(\boldsymbol{c})|\boldsymbol{\lambda},\boldsymbol{\lambda}_v)$ is controlled by the GV weight w.

The parameter set, which consists of the parameter of the neural network and the covariance matrix $\mathbf{\Sigma}$, is estimated by maximizing the proposed objective function $\mathcal{L}_{GVTrj}$. The gradients of the mean vector $\boldsymbol{\mu}$ and the covariance matrix $\mathbf{\Sigma}$ can be calculated as follows:

$$\frac{\partial\mathcal{L}_{GVTrj}}{\partial\boldsymbol{\mu}} = \mathbf{\Sigma}^{-1}\boldsymbol{W}(\boldsymbol{c} - \bar{\boldsymbol{c}} + w\boldsymbol{P}\bar{\boldsymbol{x}}) \tag{18}$$

$$\begin{aligned}\frac{\partial\mathcal{L}_{GVTrj}}{\partial\mathbf{\Sigma}^{-1}} =& \frac{1}{2}\mathrm{diag}\big[\boldsymbol{W}(\boldsymbol{P} + \bar{\boldsymbol{c}}\bar{\boldsymbol{c}}^\top - \boldsymbol{c}\boldsymbol{c}^\top)\boldsymbol{W}^\top \\ &- 2\boldsymbol{\mu}(\bar{\boldsymbol{c}} - \boldsymbol{c})^\top\boldsymbol{W}^\top + 2w\boldsymbol{W}\boldsymbol{P}\bar{\boldsymbol{x}}(\boldsymbol{\mu} - \boldsymbol{W}\boldsymbol{c})^\top\big]\end{aligned} \tag{19}$$

$$\bar{\boldsymbol{x}} = -2\boldsymbol{P}_v(\bar{\boldsymbol{c}} - \langle\bar{\boldsymbol{c}}\rangle) \tag{20}$$

$$\boldsymbol{P}_v = \mathrm{diag}\left[\boldsymbol{I}_{T\times T}\otimes\left(\mathbf{\Sigma}_v^{-1}(\boldsymbol{v}(\bar{\boldsymbol{c}}) - \boldsymbol{v}(\boldsymbol{c}))\right)\right] \tag{21}$$

where $\otimes$ is a Kronecker product. The neural network can be updated trained by the back-propagation algorithm using the gradient in Eq. (18). The computation of gradients for the parameters in lower layers is the same as the calculation of gradients for standard neural networks. The parameters are optimized so that the GVs of generated trajectories get close to the natural ones.

The optimal static feature vector sequence is determined by maximizing the objective function $\mathcal{L}_{GVTrj}$ as follows:

$$\hat{\boldsymbol{c}} = \arg\max_{\boldsymbol{c}} P(\boldsymbol{c}|\boldsymbol{\lambda})P(\boldsymbol{v}(\boldsymbol{c})|\boldsymbol{\lambda},\boldsymbol{\lambda}_v) \tag{22}$$

Since this estimate is equivalent to the ML estimate by the basic parameter generation algorithm shown by Eq. (4), the basic parameter generation algorithm can be employed for the proposed framework. Note that the basic algorithm is computationally much more efficient compared to the parameter generation algorithm considering the GV [10] that requires an iterative process.

3.3. Related work

The several training methods incorporated the parameter trajectory generation have been proposed [15, 16, 17]. In these methods, neural networks are optimized by minimizing sequence-level error between parameter trajectories extracted from natural speech and generated parameter trajectories. It is reported that the naturalness of synthesized speech is significantly improved by employing the training method using sequence-level error functions. Although these training methods are similar to the proposed training method, the different objective function is employed for parameter optimization. The objective function for the trajectory training $\mathcal{L}_{Trj}$ is defined as the likelihood function of the static feature trajectory $\boldsymbol{c}$ and parameter generation process is included to represent the mean parameter. In addition, inter-frame correlation is modeled by the covariance matrix $\boldsymbol{P}$. However, the training methods proposed in [15, 16, 17] do not model covariance matrices. If an identity matrix is set to the covariance matrix $\boldsymbol{P}$ in the trajectory likelihood Eq. (12), maximization of the trajectory likelihood is equivalent to minimization of the sequence errors used in these works. Additionally, in the proposed method, the parameter trajectory generation considering the GV, rather than the traditional parameter trajectory generation, is incorporated into the training process. Thus, it is expected that the proposed method alleviates the over-smoothing problem of generated parameter trajectories.

4. EXPERIMENTS

4.1. Experimental conditions

Japanese 503 utterances, which can be downloaded from HTS web site[1], were used in these experiments. The contents of the data were the same as the B-set of the ATR phonetically balanced Japanese speech database [18]. The 450 utterances were used for training and the remaining 53 utterances were used for testing. Speech signals were sampled at 48 kHz. Feature vectors were extracted with a 5-ms shift and the feature vector consisted of the 0-th through 49-th mel-cepstral coefficients and a log F_0 value. Mel-cepstral coefficients were extracted from the smoothed spectrum analyzed by the STRAIGHT [19].

In these experiments, four systems were compared.

- **HMM**: Conventional HMM-based speech synthesis system
- **DNN**: Speech synthesis based on DNN trained by maximizing the objective function in Eq. (10)
- **TrjDNN**: Speech synthesis based on DNN trained by maximizing the objective function in Eq. (12)
- **GVTrjDNN**: Speech synthesis based on DNN trained by maximizing the objective function in Eq. (15)

In **HMM**, HMMs modeled observation vectors consisting of 50 mel-cepstral coefficients, log F_0 values, and their dynamic features (delta and delta-delta). Five-state, left-to-right, no-skip hidden semi-Markov models (HSMMs) were used. To model log F_0 sequences consisting of voiced and unvoiced observations, a multi-space probability distribution (MSD) was used. The minimum description

[1] http://hts.sp.nitech.ac.jp/

Table 1. Global variance distances and Mel-cepstral distortions (dB) on test data.

	HMM	DNN	TrjDNN	GVTrjDNN
GVD	0.701	0.687	0.442	0.407
MCD	5.123	4.831	4.897	4.981

length (MDL) criterion was employed to determine the size of decision tree for context clustering [20]. The input feature for the DNN used in **DNN**, **TrjDNN**, and **GVTrjDNN** was a 411-dimensional feature vector, consisting of 408 linguistic features including binary features and numerical features for contexts and three duration features including duration of the current phoneme and the position of the current frame. The output feature was a 154-dimensional acoustic feature vector, consisting of 50 mel-cepstral coefficients, a log F_0 value, their dynamic features (delta and delta-delta), and a voiced/unvoiced binary value. The input features were normalized to be within 0.0-1.0 based on their minimum and maximum values in the training data, and the output features were normalized to have zero-mean unit-variance. The input and output features were time-aligned frame-by-frame by well-trained HMMs. A single network which modeled both spectral and excitation parameters was trained. The architecture of the DNNs used in **DNN**, **TrjDNN**, and **GVTrjDNN** was 3-hidden-layer with 1024 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 used in **DNN** were initialized randomly, then they were optimized by maximizing the objective function $\mathcal{L}$ in Eq. (10). The weights of the DNN used in **TrjDNN** were initialized by the trained DNN in **DNN**, then they were optimized by maximizing the objective function $\mathcal{L}_{Trj}$. The trained DNN in **TrjDNN** was used as initial model for **GVTrjDNN**. The weights of **GVTrjDNN** were optimized by maximizing the objective function $\mathcal{L}_{GVTrj}$ in Eq. (15). For training the DNNs, a mini-batch stochastic gradient descent (SGD)-based back-propagation algorithm was used. For **TrjDNN** and **GVTrjDNN**, one utterance was used as one mini-batch in SGD-based training. The GV weight w for **GVTrjDNN** was set to 0.001[2]. The basic parameter generation algorithm was applied to generate parameter trajectories for all systems.

4.2. Experimental results

To objectively evaluate the performance of the systems, the GV distance (GVD) for mel-cepstrum coefficients and the mel-cepstral distortion (MCD) were used. The GVDs were calculated by

$$GVD = \frac{1}{T}\sqrt{\sum_{d=1}^{D}(v(d) - \bar{v}(d))} \tag{23}$$

Table 1 lists the objective evaluation results. The GVD results show that **TrjDNN** achieved significantly lower GVD than **HMM**, though **DNN** gave similar GVD to **HMM**. Additionally, **GVTrjDNN** further improved the GVD from **TrjDNN**. This result shows that the over-smoothing problem was alleviated by employing the trajectory training method considering the GV. Comparing the systems on the MCD, the DNN-based systems, **DNN**, **TrjDNN**, and **GVTrjDNN**, outperformed **HMM**. However, **GVTrjDNN** showed slightly worse MCD than **TrjDNN** and **DNN**. This result suggests that the prediction of acoustic features was affected by using the proposed training method. This is because the DNN in **GVTrjDNN** was optimized not only for the acoustic features but for the global variance.

[2]The GV weight was decided from preliminaly experiments.

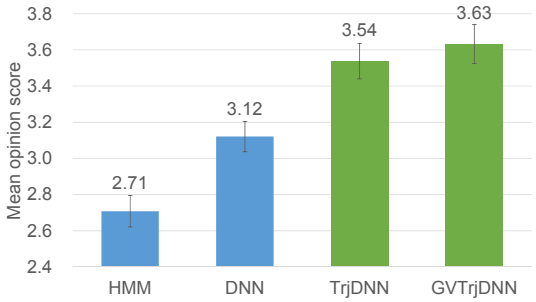

Fig. 2. Mean opinion scores of the four speech synthesis systems.

To evaluate the naturalness of the synthesized speech, a subjective listening test was conducted. The naturalness of the synthesized speech was assessed by the mean opinion score (MOS) test method. The subjects were twelve Japanese students in our research group. Twenty sentences were chosen at random from the test sentences. Speech samples were presented in random order for each test sentence. 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 2 shows the subjective evaluation results. The DNN-based systems, **DNN**, **TrjDNN**, and **GVTrjDNN**, outperformed **HMM**, as shown in Figure 2. Comparing the DNN-based systems, **TrjDNN** and **GVTrjDNN** gave significantly higher MOS than **DNN**. These results indicate that the naturalness of synthesized speech is drastically improved by introducing the parameter generation process into the training of DNNs. Comparing **TrjDNN** to **GVTrjDNN**, **GVTrjDNN** showed better score though the difference from **TrjDNN** is not statistically significant. This could be because the covariance matrix is independent of the linguistic features. It is known that the covariance matrix affects the parameter generation considering the GV. Therefore, more improvement is expected by modeling covariance matrices depending on linguistic features with mixture density networks [21].

5. CONCLUSIONS

In this paper, a trajectory training method considering the GV is proposed for DNN-based speech synthesis. The proposed method solve the inconsistency between training and synthesis criteria and the over-smoothing problem. Experimental results show the proposed method can alleviate the over-smoothing problem and improve the naturalness of synthesized speech from a conventional DNN-based system.

Future work will include some extensive experiments to compare the proposed method with the parameter generation method considering the GV and the other trajectory training methods [15, 16, 17]. In addition, we will apply the proposed training method to speech synthesis based on mixture density networks [21].

6. ACKNOWLEDGEMENTS

This work was supported by the Core Research for Evolutionary Science and Technology (CREST) program from the Japan Science and Technology Agency (JST).

7. REFERENCES

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

[2] 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.

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

[4] S. Young, J.J. Odell, and P. Woodland, "Tree-based state tying for high accuracy acoustic modelling," *Proceedings of ARPA Workshop on Human Language Technology*, pp. 307–312, 1994.

[5] K. Tokuda, T. Yoshimura, T. Masuko, T. Kobayashi, and T. Kitamura, "Speech parameter generation algorithms for HMM-based speech synthesis," *Proceedings of ICASSP 2000*, pp. 936–939, 2000.

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

[7] H. Zen, A. Senior, and M. Schuster, "Statistical parametric speech synthesis using deep neural networks," *Proceedings 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," *Proceedings 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," *Proceedings of ICASSP 2014*, pp. 3857–3861, 2014.

[10] T. Toda and K. Tokuda, "A speech parameter generation algorithm considering global variance for HMM-based speech synthesis," *IEICE Transactions on Information & Systems*, vol. E90-D, no. 5, pp. 816–824, 2007.

[11] M. Shannon and W. Byrne, "Fast, low-artifact speech synthesis considering global variance," *Proceedings of ICASSP 2013*, pp. 7869–7873, 2013.

[12] T. Toda and S. Young, "Trajectory training considering global variance for HMM-based speech synthesis," *Proceedings of ICASSP 2009*, pp. 4025–4028, 2009.

[13] K. Hashimoto, K. Oura, Y. Nankaku, , and K. Tokuda, "The effect of neural networks in statistical parametric speech synthesis," *Proceedings of ICASSP 2015*, pp. 4455–4459, 2015.

[14] H. Zen, K. Tokuda, and T. Kitamura, "Reformulating the HMM as a trajectory model by imposing explicit relationships between static and dynamic features," *Computer Speech and Language*, vol. 21, no. 1, pp. 153–173, 2007.

[15] Z. Wu and S. King, "Minimum trajectory error training for deep neural networks, combined with stacked bottleneck features," *Proceedings of Interspeech 2015*, pp. 309–313, 2015.

[16] Y. Fan, Y. Qian, F.K. Soong, and L. He, "Sequence generation error (SGE) minimization based deep neural networks training for text-to-speech synthesis," *Proceedings of Interspeech 2015*, pp. 864–868, 2015.

[17] F.L. Xie, Y. Qian, Y. Fan, F.K. Soong, and H. Li, "Sequence error SE minimization training of neural network for voice conversion," *Proceedings of Interspeech 2014*, pp. 2283–2287, 2014.

[18] A. Kurematsu, K. Takeda, Y. Sagisaka, S. Katagiri, H. Kuwabara, and K. Shikano, "ATR Japanese speech database as a tool of speech recognition and synthesis," *Speech Communication*, vol. 9, pp. 357–363, 1990.

[19] H. Kawahara, I. Masuda-Katsuse, 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, pp. 187–207, 1999.

[20] K. Shinoda and T. Watanabe, "Acoustic modeling based on the MDL criterion for speech recognition," *Proceedings of Eurospeech 1997*, pp. 99–102, 1997.

[21] H. Zen and A. Senior, "Deep mixture density networks for acoustic modeling in statistical parametric speech synthesis," *Proceedings of ICASSP 2014*, pp. 3872–3876, 2014.