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

# Model integration for HMM- and DNN-based speech synthesis using Product-of-Experts framework

*Kentaro Tachibana*[1], *Tomoki Toda*[2,1], *Yoshinori Shiga*[1], *Hisashi Kawai*[1]

[1]National Institute of Information and Communications Technology, Japan
[2]Information Technology Center, Nagoya University, Japan

kentaro.tachibana@nict.go.jp, tomoki@icts.nagoya-u.ac.jp,
{yoshi.shiga, hisashi.kawai}@nict.go.jp

## Abstract

In this paper, we propose a model integration method for hidden Markov model (HMM) and deep neural network (DNN) based acoustic models using a product-of-experts (PoE) framework in statistical parametric speech synthesis. In speech parameter generation, DNN predicts a mean vector of the probability density function of speech parameters frame by frame while keeping its covariance matrix constant over all frames. On the other hand, HMM predicts the covariance matrix as well as the mean vector but they are fixed within the same HMM state, *i.e.*, they can actually vary state by state. To make it possible to predict a better probability density function by leveraging advantages of individual models, the proposed method integrates DNN and HMM as PoE, generating a new probability density function satisfying conditions of both DNN and HMM. Furthermore, we propose a joint optimization method of DNN and HMM within the PoE framework by effectively using additional latent variables. We conducted objective and subjective evaluations, demonstrating that the proposed method significantly outperforms the DNN-based speech synthesis as well as the HMM-based speech synthesis.

**Index Terms**: speech synthesis, deep neural network, hidden Markov model, model integration, product-of-experts

## 1. Introduction

Statistical parametric speech synthesis (SPSS) is a framework that generates synthetic speech based on statistical models. Heretofore, hidden Markov model- (HMM) based speech synthesis [1, 2, 3] has been actively studied for a long time in the SPSS. HMM-based speech synthesis is highly flexible with respect to voice variation and speaking style [4, 5], and it is comparatively easy to rectify problematic sounds. However, degradation in speech quality is caused by the state-by-state modeling and decision-tree based hard clustering [6]. The improvement of its speech quality is therefore a very important task.

Recently, deep neural network- (DNN) based speech synthesis [7, 8, 9] has attracted much attention. DNN-based speech synthesis generates high quality speech trajectory frame by frame, and has achieved significant improvements over HMM-based one [7, 8, 9]. However, there are mainly three problems in the DNN-based systems. Firstly, the controllability of voice variation and speaking style is still limited. Secondly, modification of generated speech parameter by the DNN-based systems is more difficult than that of the HMM-based systems because modifying network weights is harder than the decision trees [10]. Finally, experience with respect to the DNN-based systems is less than that of the HMM-based systems.

In contrast, methods integrating DNN and HMMs have been proposed to leverage advantages of both HMMs and DNN. For example, Chen *et. al.* have used decision tree question indicators as an input to DNN [11], and Hashimoto *et. al.* have used speech parameters generated by HMMs as an input to DNN [12]. In [11, 12], methods integrating HMM and DNN outperformed the DNN-based systems in the speech quality. These methods integrate DNN and HMMs in the training phase.

This paper proposes a new model integration method for both HMMs and DNN using a product-of-experts (PoE) framework [13] in the speech parameter generation phase to utilize advantages of them.[1] HMMs and DNN are trained individually, and probability density functions (*p.d.f.*s) output from them are multiplied. While conventional methods integrate HMMs and DNN serially, the proposed method integrates HMMs and DNN in parallel. Using the PoE framework enables *p.d.f.* to be so generated as to simultaneously satisfy constraints of individual models. Furthermore, we propose a method which jointly optimizes the individual models within the PoE framework using Expectation-Maximization (EM) algorithm.

This paper is organized as follows. Section 2 describes the SPSS framework and the details of the HMM- and the DNN-based speech synthesis. Section 3 describes our methods to integrate HMMs and DNN. The experimental conditions and results are presented in Section 4. Finally, Section 5 concludes the paper and discusses future works.

## 2. HMM and DNN speech synthesis

### 2.1. SPSS framework

SPSS trains the relationship between input text and waveforms using statistics. Since directly modeling the relationship is not easy, the input text is converted to contextual feature sequences and the waveforms are converted to acoustic feature sequences. Acoustic features comprise static and dynamic features. With the input contextual features, a statistical model that outputs acoustic features can be trained. Once trained, the model can generate acoustic feature sequences from the contextual features corresponding to a given arbitrary text. In doing so, it can estimate the probability distribution of the acoustic feature sequences. Acoustic sequences transitions are usually generated by utilizing the explicit relations between static and dynamic features [15]. Finally, synthetic speech is generated by inputting the estimated acoustic sequences into a vocoder.

[1]Model integration using the PoE framework in the HMM-based speech synthesis has been proposed [14].



http://dx.doi.org/10.21437/Interspeech.2016-1006

### 2.2. HMM speech synthesis

The HMM-based speech synthesis uses context-dependent phoneme HMMs to model the probability distribution of speech parameter sequences. The context-dependent state output *p.d.f.*s are predicted from given contextual features using decision trees.

Let $\boldsymbol{o}_t$ represent a speech parameter vector consisting of not only static features but also dynamic features at frame $t$ and $q_t$ be an HMM state sequence assigned to the frame $t$. The *p.d.f.* of a speech parameter vector sequence $\boldsymbol{o} = [\boldsymbol{o}_1^\top, \cdots, \boldsymbol{o}_t^\top, \cdots, \boldsymbol{o}_T^\top]^\top$ is modeled as follows:

$$P(\boldsymbol{o}|\boldsymbol{\lambda}^{(H)}) = \sum_{\text{all } \boldsymbol{q}} P(\boldsymbol{q}|\boldsymbol{\lambda}^{(H)}) \prod_{t=1}^{T} \mathcal{N}(\boldsymbol{o}_t; \boldsymbol{\mu}_{q_t}^{(H)}, \boldsymbol{U}_{q_t}^{(H)}), \quad (1)$$

where $\boldsymbol{q} = \{q_1, \cdots, q_t, \cdots, q_T\}$, $\boldsymbol{\lambda}^{(H)}$ is an HMM parameter set, and $\mathcal{N}(\cdot; \boldsymbol{\mu}_{q_t}^{(H)}, \boldsymbol{U}_{q_t}^{(H)})$ denotes a Gaussian distribution with a mean vector $\boldsymbol{\mu}_{q_t}^{(H)}$ and a covariance matrix $\boldsymbol{U}_{q_t}^{(H)}$.

In speech parameter generation, a sentence HMM is developed by concatenating the context-dependent phoneme HMMs corresponding to an input text, and then an HMM-state sequence is determined by maximizing the likelihood of the explicit duration model. Finally, a naturally varying speech parameter sequence is generated from the resulting *p.d.f.* sequence that varies state by state under some constraints, such as an explicit relationship between the static and dynamic features [15], and the global variance (GV) [16] or the modulation spectrum [17] of the speech parameter sequence.

### 2.3. DNN speech synthesis

In the DNN-based speech synthesis, the contextual features and speech parameter vectors are treated as the inputs and targets of a DNN, respectively. The contextual features are defined frame by frame by additionally including frame positions within a phoneme.

The DNN is trained to minimize the squared error between the targets and predicted speech parameter vectors, which are usually normalized as Z-scores (*i.e.*, zero means and unit variances). This is equal to modeling the *p.d.f.* of an original speech parameter vector before the normalization using a Gaussian distribution as follows:

$$P(\boldsymbol{o}|\boldsymbol{\lambda}^{(D)}) = \prod_{t=1}^{T} \mathcal{N}(\boldsymbol{o}_t; \boldsymbol{\mu}_t^{(D)}, \boldsymbol{U}^{(D)}), \quad (2)$$

where $\boldsymbol{\lambda}^{(D)}$ is a parameter set of the DNN, $\boldsymbol{U}^{(D)}$ is a global covariance matrix to be used for the normalization, and $\boldsymbol{\mu}_t^{(D)}$ is an unnormalized mean vector given by the speech parameter vector predicted by the DNN

In speech parameter generation, the mean vector is predicted frame by frame from the frame-wise contextual features by the DNN. The resulting *p.d.f.* sequence has a time-varying mean vector sequence and the constant covariance matrices over a sequence. A speech parameter sequence is generated in the same manner as in the HMM-based speech synthesis.

## 3. Model integration

In this section, we investigate a framework that integrates the estimated probability distributions in model space using individually trained models. An overview of this framework is shown in Fig. 1. HMMs and DNN are individually trained from acoustic and contextual features. Next, the trained HMMs and DNN using a PoE framework are integrated. Finally, by inputting contextual features to the integrated model, speech parameters are generated. In this study, we investigate whether the model integration using the PoE framework improves the final results.

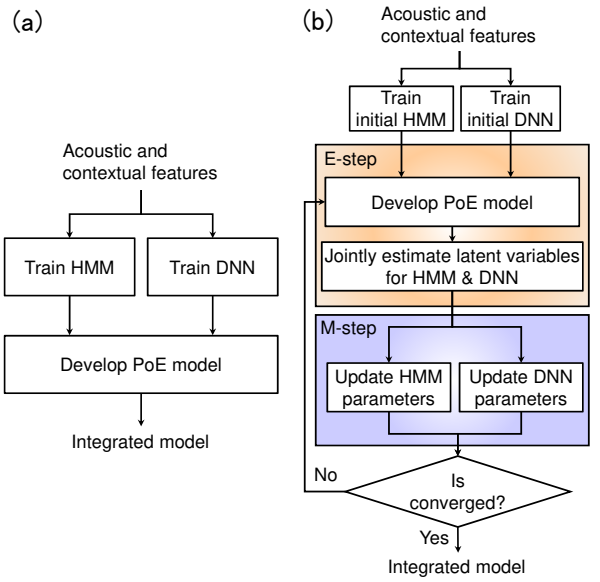


Figure 1: PoE-based model integration (a) without joint optimization and (b) with joint optimization process.

### 3.1. PoE-based model integration

We propose an integration method of multiple models within the PoE framework. The process of this method is shown in Fig. 1 (a). The PoE integrates multiple *p.d.f.*s into a single *p.d.f.* by taking a product of them. This product operation produces a shaper *p.d.f.* than the individual ones, making the integrated *p.d.f.* focus on a region overlapped over all *p.d.f.*s.

In this paper, we integrate the *p.d.f.* predicted by the DNN and that by the HMMs frame by frame. At frame $t$, the PoE-based *p.d.f.* is derived by

$$\mathcal{N}(\boldsymbol{o}_t; \boldsymbol{\mu}_t^{(I)}, \boldsymbol{U}_{q_t}^{(I)}) = \frac{1}{Z_{q_t}} \mathcal{N}(\boldsymbol{o}_t; \boldsymbol{\mu}_t^{(D)}, \boldsymbol{U}^{(D)}) \cdot \mathcal{N}(\boldsymbol{o}_t; \boldsymbol{\mu}_{q_t}^{(H)}, \boldsymbol{U}_{q_t}^{(H)}), \quad (3)$$

$$Z_{q_t} = \int \mathcal{N}(\boldsymbol{o}_t; \boldsymbol{\mu}_t^{(D)}, \boldsymbol{U}^{(D)}) \cdot \mathcal{N}(\boldsymbol{o}_t; \boldsymbol{\mu}_{q_t}^{(H)}, \boldsymbol{U}_{q_t}^{(H)}) \mathrm{d}\boldsymbol{o}_t, \quad (4)$$

where

$$\boldsymbol{\mu}_t^{(I)} = \boldsymbol{U}_{q_t}^{(I)} \left( \boldsymbol{U}^{(D)^{-1}} \boldsymbol{\mu}_t^{(D)} + \boldsymbol{U}_{q_t}^{(H)^{-1}} \boldsymbol{\mu}_{q_t}^{(H)} \right), \quad (5)$$

$$\boldsymbol{U}_{q_t}^{(I)} = \left( \boldsymbol{U}^{(D)^{-1}} + \boldsymbol{U}_{q_t}^{(H)^{-1}} \right)^{-1}. \quad (6)$$

The mean vector of the PoE-based *p.d.f.* varies frame by frame as in the DNN and its covariance varies state by state as in the HMMs. We may also use integration weights to additionally control the effects of individual models on the integrated model as follows:

$$\boldsymbol{\mu}_t^{(I,w)} = \boldsymbol{U}_{q_t}^{(I,w)} \left( w^{(D)} \boldsymbol{U}^{(D)^{-1}} \boldsymbol{\mu}_t^{(D)} + w^{(H)} \boldsymbol{U}_{q_t}^{(H)^{-1}} \boldsymbol{\mu}_{q_t}^{(H)} \right), \quad (7)$$

$$\boldsymbol{U}_{q_t}^{(I,w)} = \left( w^{(D)} \boldsymbol{U}^{(D)^{-1}} + w^{(H)} \boldsymbol{U}_{q_t}^{(H)^{-1}} \right)^{-1}. \quad (8)$$

Although only the HMMs and the DNN are integrated in this paper, multiple models are straightforwardly integrated within this framework.

### 3.2. Joint optimization using EM algorithm

To make it possible to jointly optimize the DNN parameter set and the HMMs parameter set by maximizing a likelihood function based on PoE, which is given by

$$P(\boldsymbol{o}|\boldsymbol{q},\boldsymbol{\lambda}^{(I)}) = \prod_{t=1}^{T} \mathcal{N}(\boldsymbol{o}_t;\boldsymbol{\mu}_t^{(I)},\boldsymbol{U}_{q_t}^{(I)}), \tag{9}$$

where $\boldsymbol{\lambda}^{(I)}$ consists of both the DNN and HMM parameter sets, we propose a training method based on EM algorithm. The PoE-based likelihood function is reformulated by additionally using latent variables to model the observed speech parameter vector $\boldsymbol{o}_t$ as follows:

$$\boldsymbol{o}_t = \boldsymbol{U}_{q_t}^{(I)} \left[\boldsymbol{U}^{(D)^{-1}}, \boldsymbol{U}_{q_t}^{(H)^{-1}}\right] \left[\boldsymbol{o}_t^{(D)^\top}, \boldsymbol{o}_t^{(H)^\top}\right]^\top, \tag{10}$$

where $\boldsymbol{o}_t^{(D)}$ and $\boldsymbol{o}_t^{(H)}$ are the latent variables following the DNN-based *p.d.f.* and the HMM-based *p.d.f.*, respectively, as follows:

$$\boldsymbol{o}_t^{(D)} \sim \mathcal{N}(\boldsymbol{\mu}_t^{(D)},\boldsymbol{U}^{(D)}), \tag{11}$$

$$\boldsymbol{o}_t^{(H)} \sim \mathcal{N}(\boldsymbol{\mu}_{q_t}^{(H)},\boldsymbol{U}_{q_t}^{(H)}). \tag{12}$$

The PoE-based *p.d.f.* $\mathcal{N}(\boldsymbol{o}_t;\boldsymbol{\mu}_t^{(I)},\boldsymbol{U}_{q_t}^{(I)})$ can be derived by marginalizing a *p.d.f.* of the complete data, which is given by

$$\begin{bmatrix}\boldsymbol{o}_t\\ \boldsymbol{o}_t^{(D)}\\ \boldsymbol{o}_t^{(H)}\end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix}\boldsymbol{\mu}_t^{(I)}\\ \boldsymbol{\mu}_t^{(D)}\\ \boldsymbol{\mu}_{q_t}^{(H)}\end{bmatrix}, \begin{bmatrix}\boldsymbol{U}_{q_t}^{(I)} & \boldsymbol{U}_{q_t}^{(I)} & \boldsymbol{U}_{q_t}^{(I)}\\ \boldsymbol{U}_{q_t}^{(I)} & \boldsymbol{U}^{(D)} & \boldsymbol{0}\\ \boldsymbol{U}_{q_t}^{(I)} & \boldsymbol{0} & \boldsymbol{U}_{q_t}^{(H)}\end{bmatrix}\right), \tag{13}$$

over the latent variables $\boldsymbol{o}_t^{(D)}$ and $\boldsymbol{o}_t^{(H)}$. To determine the model parameter set $\boldsymbol{\lambda}^{(I)}$ that maximizes the PoE-based likelihood function, the following auxiliary function is maximized with respect to the model parameter set:

$$\begin{aligned}&\mathcal{Q}(\boldsymbol{\lambda}^{(I)},\hat{\boldsymbol{\lambda}}^{(I)}) =\\ &\sum_{t=1}^{T}\left(\int P(\boldsymbol{o}_t^{(D)}|\boldsymbol{o}_t,q_t,\boldsymbol{\lambda}^{(I)}) \log \mathcal{N}(\boldsymbol{o}_t^{(D)};\hat{\boldsymbol{\mu}}_t^{(D)},\hat{\boldsymbol{U}}^{(D)})\mathrm{d}\boldsymbol{o}_t^{(D)}\right.\\ &\left.+\int P(\boldsymbol{o}_t^{(H)}|\boldsymbol{o}_t,q_t,\boldsymbol{\lambda}^{(I)}) \log \mathcal{N}(\boldsymbol{o}_t^{(H)};\hat{\boldsymbol{\mu}}_{q_t}^{(H)},\hat{\boldsymbol{U}}_{q_t}^{(H)})\mathrm{d}\boldsymbol{o}_t^{(H)}\right),\end{aligned} \tag{14}$$

where the posterior *p.d.f.*s calculated in E-step are given by

$$\begin{aligned}P(\boldsymbol{o}_t^{(D)}|\boldsymbol{o}_t,q_t,\boldsymbol{\lambda}^{(I)}) &= \int P(\boldsymbol{o}_t^{(D)},\boldsymbol{o}_t^{(H)}|\boldsymbol{o}_t,q_t,\boldsymbol{\lambda}^{(I)})\mathrm{d}\boldsymbol{o}_t^{(H)}\\ &= \mathcal{N}(\boldsymbol{o}_t^{(D)};\boldsymbol{o}_t+\boldsymbol{\mu}_t^{(D)}-\boldsymbol{\mu}_t^{(I)},\boldsymbol{U}^{(D)}-\boldsymbol{U}_{q_t}^{(I)}),\end{aligned} \tag{15}$$

$$\begin{aligned}P(\boldsymbol{o}_t^{(H)}|\boldsymbol{o}_t,q_t,\boldsymbol{\lambda}^{(I)}) &= \int P(\boldsymbol{o}_t^{(D)},\boldsymbol{o}_t^{(H)}|\boldsymbol{o}_t,q_t,\boldsymbol{\lambda}^{(I)})\mathrm{d}\boldsymbol{o}_t^{(D)}\\ &= \mathcal{N}(\boldsymbol{o}_t^{(H)};\boldsymbol{o}_t+\boldsymbol{\mu}_{q_t}^{(H)}-\boldsymbol{\mu}_t^{(I)},\boldsymbol{U}_{q_t}^{(H)}-\boldsymbol{U}_{q_t}^{(I)}).\end{aligned} \tag{16}$$

In M-step, the HMM parameter set and the DNN parameter set are separately updated by maximizing the auxiliary function in the usual manner.

Fig. 1 (b) shows the proposed joint optimization process. The DNN parameter set and the HMM parameter set are optimized as follows:

**Step 1** Separately train initial HMM and DNN using the speech parameter vectors and the contextual features.

**Step 2** Develop the PoE model using the trained HMM and DNN.

**Step 3** Jointly estimate the posterior *p.d.f.*s of the latent variables for the HMM and the DNN.

**Step 4** Update the HMM and DNN parameters separately using the corresponding posterior *p.d.f.*s as the observation vectors for individual models.

**Step 5** Return to Step 2 unless an increase of the PoE-based likelihood converges.

In this paper, we approximate the posterior *p.d.f.*s with the maximum to *a posteriori* estimates, *i.e.*, their mean vectors.

It is also possible to handle an HMM state sequence as a hidden variable. Moreover, we can easily extend the frame-wise integration process used in this paper to a sequence-wise integration process by further introducing the latent trajectory modeling technique [18].

## 4. Experimental evaluation

### 4.1. Experimental conditions

We conducted experiments to confirm the performance of the proposed method. A Japanese corpus recorded using a female speaker was used for the experiments. The training set consisted of phonetically balanced 503 sentences. Another 100 and 93 sentences were used as validation and test data, respectively. The speech data was downsampled from 48 kHz to 16 kHz. The spectrum and aperiodicity (AP), analyzed by STRAIGHT [19] every 5 ms, were represented by 40 Mel-cepstral coefficients (from the 0th to the 39th). Logarithmic fundamental frequency (log $F_0$) values were calculated by integrating the results of multiple $F_0$ extractors [20, 21, 22], and micro-prosody was removed to smooth the results. In addition, when training the DNN model, we used the log $F_0$ pattern which was interpolated during unvoiced and silent periods. Acoustic features were composed of the Mel-cepstral coefficients, AP and log $F_0$ and their delta and delta-delta features. Five-state, left-to-right, no-skip hidden semi-Markov models (HSMMs) were used. The sizes of decision trees in the HMM-based system was controlled by the scaling factor $\alpha$ for the model complexity penalty term of the minimum description length (MDL) criterion [23] ($\alpha = 1$).

In the HMM-based system, the above acoustic features and phoneme duration were trained using HSMMs and contextual features in the phonemes. In the DNN-based system, frame-by-frame contextual features were used that includes frame positions calculated from the same phoneme duration used in the HMM-based system. The output vector consisted of the above acoustic features and a voiced/unvoiced (V/UV) binary value. The total numbers of input and output vectors in the DNN training were 483 and 244, respectively. Both the input and output vectors were so normalized as to have zero means and unit variances. The structure of the DNN had six layers of 1024 units, and the weights of the DNN were initialized by random values. The mini-batch size was 18, the number of epochs was 30, the learning rate was $1.0 \times 10^{-5}$, and activation function was tanh. Estimated output vector in the DNN was unnormalized using global means and variances that were calculated from training data, and the probability distribution was generated so that the unnormalized output vector was mean vector and variance was the global variance.

Table 1: Results of objective evaluation.

| | MCD (dB) | AP distortion (dB) | V/UV error rate (%) | RMSE in log $F_0$ (oct) |
|---|---|---|---|---|
| HMM | 5.83 | 4.24 | 7.72 | **0.39** |
| DNN | 5.64 | 4.10 | 4.59 | 0.45 |
| PoE w/o opt. | 5.53 | 4.10 | **4.47** | 0.41 |
| PoE w opt. | **5.45** | **4.09** | 4.53 | **0.39** |

In PoE-based model integration, integration weights are set to $w^{(D)} = 0.9$ and $w^{(H)} = 0.1$ based on the likelihood of the Gaussian distribution of the PoE-based model from preliminary experiments. In the joint optimization process, models trained using the above HMM- and DNN-based systems were used as the initial models. In the parameter update process of the integrated model, in the case of DNN, the learning rate was $1.0 \times 10^{-7}$, the other conditions were the same as those of DNN-base system and, in the case of the HMM, all conditions were the same as those of the HMM-based system; the state sequences of HMMs were fixed. For approximation, only the mean vectors of both the HMMs and DNN were updated. The number of iteration in the joint optimization was two.

### 4.2. Objective evaluation

To objectively evaluate the performance of all the methods, Mel-cepstral distortion (MCD) (dB), AP distortion (dB), V/UV error rate (%), and root mean squared error (RMSE) in log $F_0$ (oct) were used. Tab. 1 shows the results of the objective evaluation. **PoE w/o opt.** and **PoE w opt.** mean the model integration in Sec. 3.1 and Sec. 3.2 respectively.

First, we compared with the HMM- and the DNN-based systems. The DNN-based system outperformed the HMM-based ones in all measurements except for RMSE in log $F_0$. **PoE w/o opt.** achieved same or higher performances than the HMM- and the DNN-based systems except for RMSE in log $F_0$. **PoE w opt.** outperformed **PoE w/o opt.** in all the measurements except for RMSE in log $F_0$. Therefore **PoE w opt.** comprehensively outperformed the others.

### 4.3. Subjective evaluation

We compared the performances of all the methods by carrying out a subjective preference listening test. The subjectives were 5 males. The number of test sentences was 12. In the HMM-based system, the parameter generation considering GV [16] is used to enhance the dynamics within each speech utterance. We applied a postfilter for Mel-cepstral coefficients [24] (PF1) to the speech parameters generated from the DNN-based system and the proposed method. However, in preliminary experiments, we found that the dynamic range of speech generated from the proposed method tends to be narrower than the HMM-based system with GV and the DNN-based system with the PF1. To enhance the dynamic range of synthesized speech, we applied another postfilter (PF2) for Mel-cepstral coefficients generated by the proposed method. The PF2 was designed to recover the variance in each utterance to the global variance. The PF2 was not applied to DNN-based system. Because the effect of the PF2 was almost not confirmed due to using the global variance as the variance of each frame in the parameter generation phase. As a result, the PF1 was applied to DNN-based system in this subjective evaluation. Furthermore, we found another problem about the proposed method, that abnormal sounds were observed in the proposed method when results of V/UV in the HMM- and the DNN-based systems were different. To avoid such sounds, when the results of V/UV in the HMM- and the DNN-based systems were same, we integrated HMMs and DNN using the PoE in the proposed method. On the other hand, when the results were different we used the speech parameter generated from the DNN-based system in the proposed method.

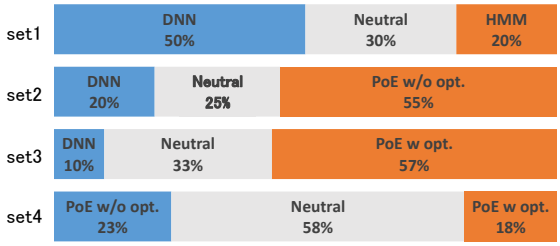


Figure 2: Preference score between chosen two systems.

Subjectives evaluated the naturalness of synthesized speech with the same the test set. After listening to each pair of samples, the subjects were asked to choose their preferred one, whereas they were able to choose "neutral" if they did not have any preference. Fig. 2 shows the results of the subjective evaluation. We confirmed that significantly better preferences at $p > 0.01$ were observed in each of set1, 2 and 3. As for set1, the DNN-based system outperformed the HMM-based one. We can also find that the proposed method outperformed the DNN-based system in set2 and 3. The differences of significant preference between the proposed methods could not be found in set4.

### 4.4. Analysis

The postfilter considering the global variance worked well in the proposed method. The proposed method strongly weights the DNN-based system in this experimental condition. The mean vectors of the proposed method varied frame by frame as in the DNN and its covariance matrix slightly varied state by state as in the HMMs. This suggests that the variation of the covariance matrix results in speech quality improvement.

## 5. Conclusion

In this paper, we investigated model integration in statistical parametric speech synthesis. The HMM- and the DNN-based systems were integrated based on a PoE framework, which are the two main systems in SPSS. Moreover, this paper provided joint optimization within the PoE. The integrated model based on the PoE achieved significant improvements over the HMM- and the DNN-based systems in both objective and subjective evaluation. In the proposed method, joint optimization comprehensively achieved performance improvement in objective evaluation; however, preference was not shown in subjective evaluation.

In future work, we plan to compare the proposed method with other model integration methods such as Dropout [25].

# 6. References


[1] K. Tokuda, T. Mausko, N. Miyazaki, T. Kobayashi, "Multi-space probability distribution HMM", *IEICE Trans. Inf. & Syst.*, vol.E85-D, no.3, pp. 455–464, 2002.

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

[3] K. Tokuda, Y. Nankaku, T. Toda, H. Zen, J. Yamagishi and K. Oura "Speech synthesis based on hidden Markov models," *Proc. of IEEE*, vol. 101, no. 5, pp. 1234-1252, 2013.

[4] M. Tamura, T. Masuko, K. Tokuda, and T. Kobayashi, "Adaptation of pitch and spectrum for HMM-based speech synthesis using MLLR," *in Proc. ICASSP*, pp. 805–808, 2001.

[5] T. Nose, M. Tachibana, T. Kobayashi, "HMM-based style control for expressive speech synthesis with arbitrary speakerś voice using model adaptation," *IEICE Trans. Inf. & Syst.*, vol.E92-D, 3, pp. 489–497, Mar. 2009.

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

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

[8] Y. Fan, Y. Qian, F. Xie and F. K. Soong, "TTS Synthesis with Bidirectional LSTM based Recurrent Neural Networks," *in Proc. Interspeech*, pp. 1964–1968, 2014.

[9] Z. Wu, C. Valentini-Botinhao, O. Watts and S. King, "Deep neural networks employing multi-task learning and stacked bottleneck features for speech synthesis," *in Proc. ICASSP*, pp. 4460–4464, 2015.

[10] H. Zen, "Acoustic Modeling in Statistical Parametric Speech Synthesis - From HMM to LSTM-RNN," *Proc. MLSLP*, 2015.

[11] B. Chen, Z. Chen, J. Xu and K. Yu, "An investigation of context clustering for statistical speech synthesis with deep neural network," *in Proc. ICASSP*, pp. 2212–2216, 2015.

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

[13] G. Hinton, "Product of experts," *in Proc. ICANN*, vol. 1, pp. 1–6, 1999.

[14] H. Zen, M. Gales, Y. Nankaku and K. Tokuda, "Product of experts for statistical parametric speech synthesis." *Audio, Speech, and Language Processing, IEEE Transactions on*, 20(3) pp. 794–805, 2012.

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

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

[17] S. Takamichi, T. Toda, A. W. Black and S. Nakamura, "Parameter generation algorithm considering modulation spectrum for HMM-based speech synthesis," *In Proc. ICASSP*, pp. 4210–4214, 2015.

[18] H. Kameoka, "Modeling speech parameter sequences with latent trajectory Hidden Markov model," *In Proc. MLSP*, 2015.

[19] H. Kawahara, I. Masuda-Katsuse, and A. Cheveigne, "Restructing speech representations using a pitch-adaptive timefrequency 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] A. Camacho, "SWIPE: A Sawtooth Waveform Inspired Pitch Estimator for Speech And Music," Ph.D. Thesis, University of Florida, 2007.

[21] D. Talkin, "A Robust Algorithm for Pitch Tracking (RAPT)," *in Speech Coding & Synthesis*, W. B. Kleijn and K. K. Pailwal (Eds.), Elsevier, pp. 495–518, 1995.

[22] M. Morise, "An attempt to develop a singing synthesizer by collaborative creation," *Proc. the Stockholm Music Acoustics Conference 2013 (SMAC2013)*, pp. 287–292, 2013.

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

[24] T. Yoshimura, K. Tokuda, T. Masuko, T. Kobayashi, and T. Kitamura, "Incorporation of mixed excitation model and postfilter into HMM-based text-to-speech synthesis," *IEICE Trans. Inf. & Syst. (Japanese Edition)*, J87–D–II(8), pp. 1563–1571, 2004.

[25] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A Simple Way to Prevent Neural Networks from Overfitting," *The Journal of Machine Learning Research*, Vol. 15, pp. 1929–1958, 2014.