A Probabilistic Interpretation for Artificial Neural Network-based Voice Conversion

Hsin-Te Hwang[1,3], Yu Tsao[2], Hsin-Min Wang[3], Yih-Ru Wang[1], Sin-Horng Chen[1]
[1]Dept. of Electrical and Computer Engineering, National Chiao Tung University, Hsinchu, Taiwan
[2]Research Center for Information Technology Innovation, Academia Sinica, Taipei, Taiwan
[3]Institute of Information Science, Academia Sinica, Taipei, Taiwan
E-mail: hwanght@iis.sinica.edu.tw, yu.tsao@citi.sinica.edu.tw, whm@iis.sinica.edu.tw, yrwang@cc.nctu.edu.tw, schen@mail.nctu.edu.tw

***Abstract*—Voice conversion (VC) using artificial neural networks (ANNs) has shown its capability to produce better sound quality of the converted speech than that using Gaussian mixture model (GMM). Although ANN-based VC works reasonably well, there is still room for further improvement. One of the promising ways is to adopt the successful techniques in statistical model-based parameter generation (SMPG), such as trajectory-based mapping approaches that are originally designed for GMM-based VC and hidden Markov model (HMM)-based speech synthesis. This study presents a probabilistic interpretation for ANN-based VC. In this way, ANN-based VC can easily incorporate the successful techniques in SMPG. Experimental results demonstrate that the performance of ANN-based VC can be effectively improved by two trajectory-based mapping techniques (maximum likelihood parameter generation (MLPG) algorithm and maximum likelihood-based trajectory mapping considering global variance (referred to as MLGV)), compared to the conventional ANN-based VC with frame-based mapping and the GMM-based VC with the MLPG algorithm. Moreover, ANN-based VC with the trajectory-based mapping techniques can achieve comparable performance when compared to the state-of-the-art GMM-based VC with the MLGV algorithm.**

I. INTRODUCTION

Voice conversion (VC) is a technique that transforms a source speaker's voice to that of a specific target speaker [1-12]. The overall VC includes two parts, namely spectral and prosody conversions. This study focuses on spectral conversion (SC). For simplicity, we use the term VC in the paper.

Numerous VC approaches [1-12] have been proposed based on various models. Among them, Gaussian mixture model (GMM) and artificial neural networks (ANNs) (also known as multilayer perceptron, MLP, or feed-forward neural networks) are two widely used models. In this study, we focus our attention on ANN-based VC.

Previous studies [6, 7] have shown that ANN-based VC can produce better sound quality of the converted speech when compared with GMM-based VC. The result can be attributed to three reasons: 1) Temporal information of acoustic features (such as using multiple neighboring frames as features) can be effectively used by ANN model for VC; 2) ANN is widely believed to be good at modeling features with strong inter-dimensional correlation, such as spectral envelopes directly calculated by fast Fourier transform (FFT); 3) ANN is good at modeling complex dependencies between input and output acoustic features by using a nonlinear mapping function. Although ANN-based VC works reasonably well, the quality of the converted speech using ANNs is still far from that of the natural speech, which leaves room for further improvement.

To improve the sound quality of ANN-based VC, there are two potential directions to follow. The first direction is based on the recent advances in the field of deep learning studies [17, 18], such as using pre-training for better initializing the model parameters of a neural network [17, 19] and using a better activation function [20]. In a recent study [7], it was found that a better initialization for the model parameters can be achieved by layer-wise backpropagation pre-training [19], thereby improving the conversion accuracy for ANN-based VC. The second direction is based on introducing the successful techniques that were originally developed for statistical model-based parameter generation (SMPG). A notable example is maximum likelihood parameter generation (MLPG) algorithm [2, 13] for ANN-based VC [7, 8] and deep neural network (DNN)-based speech synthesis (SS) [15, 16], with the aim of overcoming the discontinuity problem in the conventional ANN-based parameter generation process. Moreover, the minimum generation error (MGE) training criterion for hidden Markov model (HMM)-based SS [14] has also been adopted for ANN-based VC [7]. In this paper, we follow the second direction of using the successful techniques in SMPG to improve ANN-based VC.

The key idea of the proposed method is to use an ANN directly to model the conditional probability distribution of the target acoustic features given the source acoustic features and estimate the model parameters (including a conditional mean vector, associated with the parameters of the ANN, and a precision matrix) by a maximum likelihood (ML) criterion. In this way, the proposed method can easily adopt the successful techniques in SMPG, such as the MLPG and trajectory-based mapping considering global variance (referred to as MLGV [2]) algorithms, developed within a probabilistic framework. Experimental results demonstrate that the improvement of the converted speech can be achieved by the proposed method with the MLPG and MLGV algorithms, compared to the conventional ANN-based VC [6] and GMM-based VC [2].

The remainder of this paper is organized as follows. Section 2 reviews the conventional ANN-based VC [6]. Section 3 describes the proposed ANN-based VC. Section 4 presents our experimental results. Finally, the conclusion is given in Section 5.

II. Conventional ANN-based Voice Conversion

In the training phase, a parallel speech corpus composed by the source and target speakers' speeches is first prepared. After feature extraction, a preprocessing step is performed to time-align the spectral feature vector sequences of the source and target speeches. Finally, an ANN is employed to construct a nonlinear mapping between the source and target feature vectors.

Let $\mathbf{X}_t$ and $\hat{\mathbf{Y}}_t$ be the source and converted feature vectors at frame t, respectively. For a 3-layer (two hidden layers) neural network, its mapping function is given by

$$\hat{\mathbf{Y}}_t = f_{NN}(\mathbf{X}_t) = f_3(\mathbf{w}^{(3)} f_2(\mathbf{w}^{(2)} f_1(\mathbf{w}^{(1)} \tilde{\mathbf{X}}_t))), \quad (1)$$

where $\tilde{\mathbf{X}}_t = \left[\mathbf{X}_t^{\mathrm{T}}, 1\right]^{\mathrm{T}}$ is composed by the source feature vector and an additional value (set to one); $\mathbf{w}^{(1)}$, $\mathbf{w}^{(2)}$, and $\mathbf{w}^{(3)}$ are the weight matrices of the first, second, and third layer, respectively; and f_1, f_2, and f_3 are the activation functions of the first, second, and third layer, respectively. Note that the bias vectors have been absorbed into the weight matrices. Typically, when ANN is used for a regression problem, a nonlinear activation function (a hyperbolic tangent or logistic sigmoid function) is used for the hidden layer (i.e., f_1 and f_2 in a 3-layer ANN) while a linear activation function is used for the output layer (i.e., f_3 in a 3-layer ANN). To train an ANN for the regression case, the parameter set of the ANN $\lambda = \left\{\mathbf{w}^{(1)}, \mathbf{w}^{(2)}, \mathbf{w}^{(3)}\right\}$ is optimized by minimizing the following objective function

$$e = \frac{1}{2T}\sum_{t=1}^{T}\left\|\mathbf{Y}_t - \hat{\mathbf{Y}}_t\right\|^2, \quad (2)$$

where $\mathbf{Y}_t$ are the target feature vector at frame t. Typically, a stochastic gradient descent algorithm with mini-batches is applied to update the model parameters as

$$\lambda(\tau+1) = \lambda(\tau) - \alpha \cdot \frac{\partial e}{\partial \lambda(\tau)}, \quad (3)$$

where τ denotes the τ-th iteration and α is the learning rate. The gradient $\partial e / \partial \lambda(\tau)$ in (3) can be obtained by the backpropagation algorithm.

In the conversion stage, each feature vector $\mathbf{X}_t$ of the source speaker is transformed into the converted feature vector $\hat{\mathbf{Y}}_t$ using the mapping function in (1) in a frame by frame manner.

III. Proposed ANN-based Spectral Conversion

A. Objective Function for Training

In the proposed ANN-based VC, the objective of ANN training is to maximize the following log-likelihood

$$L = \frac{1}{T}\sum_{t=1}^{T}\log P(\mathbf{Y}_t \mid \mathbf{X}_t, \lambda^{(\mathbf{Y}|\mathbf{X})}), \quad (4)$$

where $\lambda^{(\mathbf{Y}|\mathbf{X})}$ is the model parameter set; $\mathbf{X}_t = [\mathbf{x}_t^{\mathrm{T}}, \Delta\mathbf{x}_t^{\mathrm{T}}]^{\mathrm{T}}$ and $\mathbf{Y}_t = [\mathbf{y}_t^{\mathrm{T}}, \Delta\mathbf{y}_t^{\mathrm{T}}]^{\mathrm{T}}$ are the $2D$-dimensional source and target feature vectors, each consisting of D static and D dynamic (delta) features, at frame t. Here, we assume that the conditional probability density function (PDF) in (4) is modeled by a Gaussian distribution as

$$P(\mathbf{Y}_t \mid \mathbf{X}_t, \lambda^{(\mathbf{Y}|\mathbf{X})}) = N(\mathbf{Y}_t; \boldsymbol{\mu}_t^{(\mathbf{Y}|\mathbf{X})}, \boldsymbol{\Lambda}^{(\mathbf{Y}|\mathbf{X})-1}), \quad (5)$$

where the mean vector $\boldsymbol{\mu}_t^{(\mathbf{Y}|\mathbf{X})}$ at frame t and the precision matrix $\boldsymbol{\Lambda}^{(\mathbf{Y}|\mathbf{X})}$ (assumed to be diagonal in this work) are given by

$$\boldsymbol{\mu}_t^{(\mathbf{Y}|\mathbf{X})} = \hat{\mathbf{Y}}_t = f_{NN}(\mathbf{X}_t), \quad (6)$$

$$\boldsymbol{\Lambda}^{(\mathbf{Y}|\mathbf{X})} = \mathrm{diag}[\beta(1), \cdots, \beta(d), \cdots, \beta(2D)], \quad (7)$$

where $\beta(d)$ is the d-th diagonal element of the precision matrix $\boldsymbol{\Lambda}^{(\mathbf{Y}|\mathbf{X})}$. It is worth noting that the mean vector $\boldsymbol{\mu}_t^{(\mathbf{Y}|\mathbf{X})}$ is given by the output of the ANN as shown in (6) while the mean vector of the conditional PDF for GMM-based VC is given by the output of a set of linear regression functions, each for one mixture component [1-5]. As a result, the model parameters (including the weight matrices of the ANN and the precision matrix, i.e., $\lambda^{(\mathbf{Y}|\mathbf{X})} = \{\mathbf{w}, \boldsymbol{\Lambda}^{(\mathbf{Y}|\mathbf{X})}\}$) can be estimated by maximizing the objective function in (4). Note that if the precision matrix $\boldsymbol{\Lambda}^{(\mathbf{Y}|\mathbf{X})}$ is set to an identity matrix, maximizing the proposed objective function in (4) is equivalent to minimizing the conventional objective function in (2), when estimating the weight matrices of the ANN.

B. Model Parameter Optimization

Similar to the conventional ANN training process described in Section II, the stochastic gradient accent algorithm with mini-batches is employed to optimize the model parameters as

$$\begin{aligned}\lambda^{(\mathbf{Y}|\mathbf{X})}(\tau+1) &= \lambda^{(\mathbf{Y}|\mathbf{X})}(\tau) + \alpha \cdot \frac{\partial L}{\partial \lambda^{(\mathbf{Y}|\mathbf{X})}(\tau)} \\ &= \lambda^{(\mathbf{Y}|\mathbf{X})}(\tau) + \alpha \cdot \frac{1}{T}\sum_{t=1}^{T}\frac{\partial L_t}{\partial \lambda^{(\mathbf{Y}|\mathbf{X})}(\tau)}, \end{aligned} \quad (8)$$

where $L_t = \log P(\mathbf{Y}_t \mid \mathbf{X}_t, \lambda^{(\mathbf{Y}|\mathbf{X})})$. Let $\mathbf{w}_{i,j}^{(k)}$ be the weight associated with the connection between the *j*-th node of the $(k-1)$-th layer and the *i*-th node of the *k*-th layer. By applying the partial derivative on L_t with respect to the weight $\mathbf{w}_{i,j}^{(k)}$, we have

$$\frac{\partial L_t}{\partial \mathbf{w}_{i,j}^{(k)}} = \frac{\partial L_t}{\partial z_i^{(k)}} \cdot \frac{\partial z_i^{(k)}}{\partial a_i^{(k)}} \cdot \frac{\partial a_i^{(k)}}{\partial \mathbf{w}_{i,j}^{(k)}}, \tag{9}$$

where

$$a_i^{(k)} = \sum_j \mathbf{w}_{i,j}^{(k)} \cdot z_j^{(k-1)}, \quad z_i^{(k)} = f_k(a_i^{(k)}), \tag{10}$$

$$\frac{\partial a_i^{(k)}}{\partial \mathbf{w}_{i,j}^{(k)}} = z_j^{(k-1)}. \tag{11}$$

In (10) and (11), $a_i^{(k)}$ is the input of the *i*-th node in the *k*-th layer; $z_i^{(k)}$ is output of the *i*-th node of the *k*-th layer; $z_j^{(k-1)}$ is the output of the *j*-th node in the $(k-1)$-th layer; and $f_k(\cdot)$ is the activation function of the *k*-th layer. In this work, a logistic sigmoid function and a linear function are used for the hidden and output layers, respectively, and they are given by

$$\begin{aligned} f_k(\theta) &= \frac{1}{1+\exp(-\theta)}, \text{ for } k = 1 \sim (K\text{-}1), \\ f_k(\theta) &= \theta, \text{ for } k = K, \end{aligned} \tag{12}$$

where *K* is the number of layers in the ANN. By defining $\delta_i^{(k)}$ (referred to as error in the neural networks literature) as

$$\delta_i^{(k)} \equiv \frac{\partial L_t}{\partial a_i^{(k)}} = \frac{\partial L_t}{\partial z_i^{(k)}} \cdot \frac{\partial z_i^{(k)}}{\partial a_i^{(k)}}, \tag{13}$$

(9) can be rewritten as

$$\frac{\partial L_t}{\partial \mathbf{w}_{i,j}^{(k)}} = \delta_i^{(k)} \cdot z_j^{(k-1)}. \tag{14}$$

The error of the output layer $\delta_i^{(k)}$ (for $k = K$, and $i = 1 \sim 2D$) is given by

$$\begin{aligned} \delta_i^{(k)} &= \left(\partial - \frac{1}{2} \sum_{i=1}^{2D} \beta(i) \left(\mathbf{Y}_t(i) - \boldsymbol{\mu}_t^{(\mathbf{Y}|\mathbf{X})}(i) \right)^2 / \partial z_i^{(k)} \right) \cdot \frac{\partial f_k(a_i^{(k)})}{\partial a_i^{(k)}} \\ &= \left(\partial - \frac{1}{2} \sum_{i=1}^{2D} \beta(i) \left(\mathbf{Y}_t(i) - \boldsymbol{\mu}_t^{(\mathbf{Y}|\mathbf{X})}(i) \right)^2 / \partial \boldsymbol{\mu}_t^{(\mathbf{Y}|\mathbf{X})}(i) \right) \cdot \frac{\partial a_i^{(k)}}{\partial a_i^{(k)}} \\ &= \beta(i) \cdot \left(\mathbf{Y}_t(i) - \boldsymbol{\mu}_t^{(\mathbf{Y}|\mathbf{X})}(i) \right), \end{aligned} \tag{15}$$

where $\mathbf{Y}_t(i)$ and $\boldsymbol{\mu}_t^{(\mathbf{Y}|\mathbf{X})}(i)$ are the *i*-th element of the target feature vector and the conditional mean vector at frame *t*, respectively; $\beta(i)$ is the *i*-th diagonal element of the precision matrix $\boldsymbol{\Lambda}^{(\mathbf{Y}|\mathbf{X})}$. $\boldsymbol{\mu}_t^{(\mathbf{Y}|\mathbf{X})}(i)$ can be calculated by (6).

The error of a hidden layer $\delta_i^{(k)}$ (for $k = 1 \sim (K-1)$) is given by

$$\begin{aligned} \delta_i^{(k)} &= \left(\sum_j \frac{\partial L_t}{\partial a_j^{(k+1)}} \cdot \frac{\partial a_j^{(k+1)}}{\partial z_i^{(k)}} \right) \cdot \frac{\partial f_k(a_i^{(k)})}{\partial a_i^{(k)}} \\ &= \left(\sum_j \delta_j^{(k+1)} \cdot \mathbf{w}_{j,i}^{(k+1)} \right) \cdot f_k(a_i^{(k)}) \cdot \left(1 - f_k(a_i^{(k)}) \right). \end{aligned} \tag{16}$$

Finally, by applying the partial derivative on L_t with respect to the precision $\beta(d)$ (for $d = 1 \sim 2D$), we have

$$\frac{\partial L_t}{\partial \beta(d)} = \frac{1}{2\beta(d)} - \frac{1}{2} \left(\mathbf{Y}_t(d) - \boldsymbol{\mu}_t^{(\mathbf{Y}|\mathbf{X})}(d) \right)^2. \tag{17}$$

Note that, in the implementation, we optimize the $\beta(d)$ in the logarithm domain in order to keep the precision matrix positive definite. Moreover, the initial value of $\beta(d)$ is set to the target speaker's precision.

C. Parameter Generation

It has been noted that the conventional ANN-based VC [6] still suffers from two major problems, namely, the discontinuity problem and the over-smoothing problem. To handle these two problems, we propose to adopt the MLPG and MLGV for ANN-based VC.

1) MLPG Algorithm: The aim of the MLPG algorithm is to tackle the discontinuity problem existing in frame-based conversion by performing trajectory-based conversion with the dynamic-feature constraint. The MLPG algorithm for the proposed method is given by

$$\hat{\mathbf{y}} = \arg\max P(\mathbf{Y} \mid \mathbf{X}, \lambda^{(\mathbf{Y}|\mathbf{X})}), \text{ s.t. } \mathbf{Y} = \mathbf{M}\mathbf{y}, \tag{18}$$

where $\mathbf{M}$ in (18) is a weighting matrix for appending the dynamic features to the static ones [2]. The solution of (18) is as follows

$$\hat{\mathbf{y}} = (\mathbf{M}^{\mathrm{T}} \mathbf{U}^{(\mathbf{Y}|\mathbf{X})} \mathbf{M})^{-1} \mathbf{M}^{\mathrm{T}} \mathbf{U}^{(\mathbf{Y}|\mathbf{X})} \mathbf{E}^{(\mathbf{Y}|\mathbf{X})}, \tag{19}$$

where

$$\mathbf{E}^{(\mathbf{Y}|\mathbf{X})} = [(\boldsymbol{\mu}_1^{(\mathbf{Y}|\mathbf{X})})^{\mathrm{T}}, \cdots, (\boldsymbol{\mu}_t^{(\mathbf{Y}|\mathbf{X})})^{\mathrm{T}}, \cdots, (\boldsymbol{\mu}_T^{(\mathbf{Y}|\mathbf{X})})^{\mathrm{T}}]^{\mathrm{T}}, \tag{20}$$

$$\mathbf{U}^{(\mathbf{Y}|\mathbf{X})} = \mathrm{daig}[\boldsymbol{\Lambda}_1^{(\mathbf{Y}|\mathbf{X})}, \cdots, \boldsymbol{\Lambda}_t^{(\mathbf{Y}|\mathbf{X})}, \cdots, \boldsymbol{\Lambda}_T^{(\mathbf{Y}|\mathbf{X})}]. \tag{21}$$

The mean vector $\boldsymbol{\mu}_t^{(\mathbf{Y}|\mathbf{X})}$ and precision matrix $\boldsymbol{\Lambda}_t^{(\mathbf{Y}|\mathbf{X})}$ of the conditional PDF can be calculated by (6) and (7), respectively. It is noted that the MLPG algorithm has been employed in

ANN-based VC in [7] and [8]. We adopt different ways to estimate $\boldsymbol{\mu}_t^{(\mathbf{Y}|\mathbf{X})}$ (associated with the weight matrices of the ANN) and the precision matrix $\boldsymbol{\Lambda}_t^{(\mathbf{Y}|\mathbf{X})}$. We optimize $\boldsymbol{\mu}_t^{(\mathbf{Y}|\mathbf{X})}$ by maximizing a log-likelihood function in (4), while [8] used the conventional objective function in (2) and [7] used a sequence error criterion. Moreover, both previous studies ([7] and [8]) approximated the precision matrix $\boldsymbol{\Lambda}_t^{(\mathbf{Y}|\mathbf{X})}$ simply from the target speaker's variance, while we estimate it using a ML criterion as shown in (7) and (17).

2) MLGV Algorithm: The aim of the MLGV algorithm is to overcome the over-smoothing problem existing in SMPG by performing trajectory-based conversion with global variance (GV). The log-scaled likelihood function of the MLGV algorithm is given by

$$L_{(MLGV)} = \omega \cdot \log P(\mathbf{Y} \mid \mathbf{X}, \lambda^{(\mathbf{Y}|\mathbf{X})}) + \log P(v(\mathbf{y}) \mid \lambda^{(v)}), \quad (22)$$

where the power weight ω controls the balance between the two log-likelihoods; $v(\mathbf{y}) = [v(1), \cdots, v(d), \cdots, v(D)]^{\mathrm{T}}$ is a GV vector of the target static feature sequence calculated in an utterance by utterance manner as

$$v(d) = \frac{1}{T}\sum_{t=1}^{T}\left((y_t(d) - \langle y(d)\rangle)\right)^2, \quad (23)$$

$$\langle y(d)\rangle = \frac{1}{T}\sum_{t=1}^{T} y_t(d), \quad (24)$$

where $v(d)$ is the GV of the d-th dimension; and $v(\mathbf{y})$ is modeled by a single Gaussian as

$$P(v(\mathbf{y}) \mid \lambda^{(v)}) = N(v(\mathbf{y}); \boldsymbol{\mu}_v, \boldsymbol{\Sigma}^{(vv)}). \quad (25)$$

The mean vector $\boldsymbol{\mu}_v$ and the covariance matrix $\boldsymbol{\Sigma}^{(vv)}$ of the GV probability density function can be obtained using the GVs of the target feature sequences calculated from individual utterances in the training data. Finally, the converted static feature sequence can be obtained by maximizing the objective function in (22). Because there is no closed form solution, a gradient-based approach is typically applied as described in [2]. We omit the derivation of the gradient $\partial L_{(MLGV)} / \partial \mathbf{y}$ because it is very similar to that given by [2].

IV. Experiments

A. Experimental Setup

In our experiments, a parallel Mandarin speech corpus composed by two speakers (one female and one male) was prepared for evaluating the proposed methods. Eighty parallel sentences were selected from both speakers. Among the 80 sentences, 40 sentences were used to establish the conversion system and the remaining 40 sentences were used for evaluation. Speech signals were firstly recorded in a 20kHz sampling rate, and then down-sampled to 16kHz. The resolution per sample was 16 bits. The spectral features were the first through 24 Mel-cepstral coefficients (MCCs). The STRAIGHT toolkit [23] was employed for parameter extraction and speech synthesis. A dynamic time warping (DTW) algorithm is performed within each syllable boundary to perform time-alignment on the MCC sequences of the source and target speakers. The following five VC systems were constructed for evaluation:

- ***GMM+MLPG*** (baseline): Conventional GMM-based VC using the MLPG algorithm [2]. The number of mixture components was set to 64, which yielded the best performance on the test set using the measurement of conversion accuracy (described in the following section); static and delta features were used for constructing the system. We use the same setting as described in [5].
- ***GMM+MLGV*** (baseline): The system is based on ***GMM+MLPG***, where the MLGV algorithm [2] was additionally involved in parameter generation.
- ***ANN*** (baseline): Conventional ANN-based VC [6]. We used a 4-layer ANN (consisting of three hidden layers, and 250 nodes in each hidden layer) architecture; a logistic sigmoid function and a linear function were used for the hidden and output layers, respectively. Static and delta features were used for constructing the system.
- ***ANN+MLPG*** (**proposed**): The system is based on ANN-based VC using the MLPG algorithm. The architecture, activation functions, and features used for constructing the system are the same as those of ***ANN.***
- ***ANN+MLGV*** (**proposed**): The system is based on ***ANN+MLPG***, where the MLGV algorithm was additionally involved in parameter generation.

For the conventional and proposed system (***ANN***, ***ANN+MLPG***), the setting of the architecture of an ANN (described above) was based on which architectures would give the best performance in terms of the conversion accuracy. Other settings for training an ANN include using random initialization for the weights (drawn from a zero-mean normal distribution with standard deviation 0.1), normalizing the input and output features to have zero-mean unit-variance, setting the constant learning rate to as 0.01, using the momentum method to speed up learning, and using the conventional regularization methods to avoid over-fitting such as weight decay and early stopping. The size of mini-batches for stochastic gradient descent algorithm was set to 110. We reported both the objective and subjective evaluation results on the female to male VC task.

B. Objective Evaluations

In the objective tests, we evaluate the VC systems with the converison accuray and the degree of the over-smoothing of the converted Mel-cepstra.

1) Conversion Accuracy: To evaluate the conversion accuracy, we used the Mel-cepstral distortion (MCD) to compute the difference of the target and converted Mel-cepstra in the evaluation set, which is given by

TABLE I
Conversion accuracy of the five VC systems. The MCD before the conversion is 9.37 dB.

Method	*GMM+ MLPG*	*ANN*	*ANN+ MLPG*	*GMM+ MLGV*	*ANN+ MLGV*
MCD [dB]	5.12	5.08	**5.05**	5.67	5.48

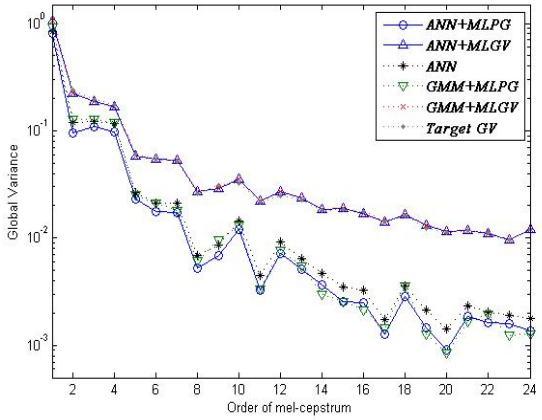

Fig. 1: The means of GV of the converted Mel-cepstra over all test utterances.

$$D_{MCD}(y_t,\hat{y}_t)\ [\text{dB}] = \frac{10}{\ln 10}\sqrt{2\sum_{d=1}^{D}(y_t(d)-\hat{y}_t(d))^2}\ , \qquad (26)$$

where y_t and $\hat{y}_t$ are the target and converted static feature vectors at frame t, respectively. A lower MCD value indicates a more accurate conversion. Table 1 shows the MCD results of the five VC systems.

From Table 1, we first note that when GV is not involved in parameter generation, both the two ANN-based VC methods (***ANN+MLPG*** and ***ANN***) give lower MCD values when compared to GMM-based VC (***GMM+MLPG***). On the other hand, when GV is involved in parameter generation, ANN-based VC (***ANN+MLGV***) achieves a significant lower MCD value than GMM-based VC (***GMM+MLGV***). This set of results suggests that ANN-based VC can achieve better conversion accuracy than GMM-based VC with and without involving GVs in the parameter generation process.

Next, when comparing the two ANN-based VC systems with and without the MLPG algorithm, we noticed that ***ANN+MLPG*** gives a slightly lower MCD value than ***ANN***. The improvement of conversion accuracy reveals the potential advantage of involving the trajectory-based conversion (the MLPG algorithm) over the conventional frame-based conversion. A similar trend has been reported in a previous study [7].

It is also noted that ***ANN+MLGV*** and ***GMM+MLGV*** give higher MCD values than ***ANN+MLPG*** and ***GMM+MLPG***, respectively, suggesting that by involving GV in parameter generation, the conversion accuracy may decrease, thereby degrading the quality of the converted speech. Notably, previous studies have shown that the MCD values may not perfectly reflect the real subjective evaluation results when introducing the GV for parameter generation [2, 5, 21]. Similar results are also observed in this study, which will be shown later in the subjective test.

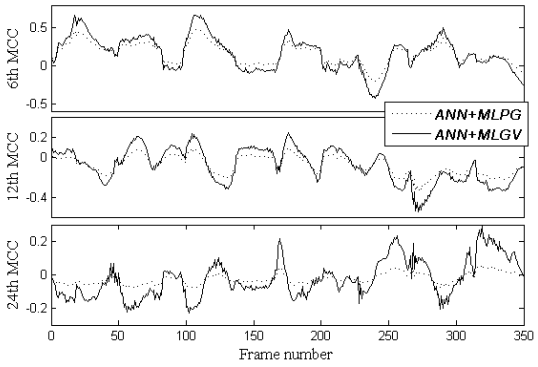

Fig. 2: Example trajectories of MCC features converted by the proposed ANN-based VCs.

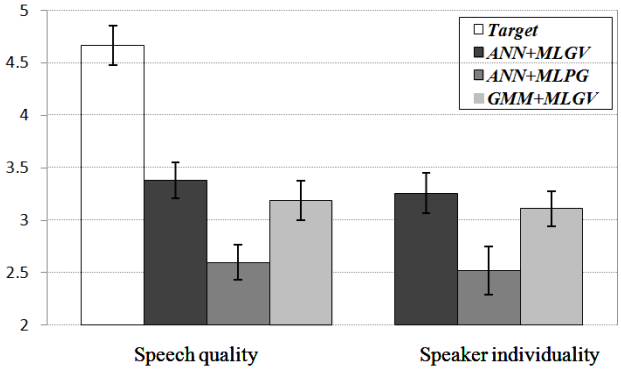

Fig. 3: Subjective test results of the speech converted by ***ANN+MLGV***, ***ANN+MLPG***, and ***GMM+MLGV***. Error bars indicate 95% confidence intervals. "Target" shown in the speech quality evaluation denotes the analysis-synthesized target speech.

2) Degree of the Over-Smoothing: To evaluate the degree of the over-smoothing of the converted Mel-cepstra, we adopted the GV measurement, computed by (23)-(24), with the speech data from the evaluation set. Fig. 1 shows GV measurements of the converted Mel-cepstra provided by the five VC systems and the natural Mel-cepstra of the target speech (referred to as the ***Target GV*** hereafter).

From Fig. 1, it is easily noted that the GV measurements of ***GMM+MLGV*** are clearly larger than ***GMM+MLPG***, and ***ANN+MLGV*** are also notably larger than ***ANN*** and ***ANN+MLPG***, in particular for higher order Mel-cepstra. Meanwhile, the GV measurements of both ***GMM+MLGV*** *and* ***ANN+MLGV*** are rather close to ***Target GV***. This set of results implies that the converted Mel-cepstra obtained by the three systems without using GV (***GMM+MLPG***, ***ANN***, and ***ANN+MLPG***) are overly smoothed, and the other two VC systems using GV (***GMM+MLGV*** and ***ANN+MLGV***) can effectively alleviate the over-smoothing problem.

As a reference, Fig. 2 demonstrates the trajectories of the 6^{th}, 12^{th}, and 24^{th} MCCs of a speech utterance generated by the proposed ***ANN+MLPG*** and ***ANN+MLGV*** (without and

with involving GVs in mappings, respectively). From Fig. 2, we can observe that the MCC features generated by ***ANN+MLPG*** present over-smoothed trajectories while ***ANN+MLGV*** compensates such over-smoothing effect by effectively enhancing the dynamic range of the trajectory movements closer to match that of target trajectory counterparts, particularly for higher order Mel-cepstra.

C. Subjective Evaluations

In our preliminary experiments, we found that the over-smoothing effect for the three VC systems without using GV (***ANN***, ***ANN+MLGV***, and ***GMM+MLPG***) severely degrades the quality of converted speech. Moreover, the advantage of using the MLPG algorithm for NN-based speech parameter generation has already been reported in previous studies ([7-9] and [15, 16]); similar trends are also observed in our preliminary experiments. Thus in the following discussion, we focus on evaluating the effectiveness of using GV for the proposed method.

We conducted a formal listening test to evaluate the speech quality and speaker individuality of the converted speech. The mean opinion score (MOS) method was performed for the analysis-synthesized target speech and the following three VC systems, namely, two VC systems using GVs for mappings (***ANN+MLGV*** and ***GMM+MLGV***) and one VC system without using GV for mapping (***ANN+MLPG***, which achieves the best conversion accuracy as shown in Table1). Twenty test sentences were randomly chosen from the test set and presented in a random order to eight subjects. Since this study focuses on spectral conversion, a simple linear transformation method [2] was used for F_0 conversion for all of the VC systems. Fig. 3 shows the MOS test result.

From Fig. 3, we first note that ***ANN+MLGV*** yields a significant gain on both the speech quality and speaker individuality tests over ***ANN+MLPG***. This result indicates that the MLGV algorithm can effectively alleviate the over-smoothing effect, thereby improving the quality of converted speech. Notably, although ***ANN+MLGV*** outperforms ***ANN+MLPG*** in terms of MOS results, the conversion accuracy achieved by ***ANN+MLGV*** is worse than that by ***ANN+MLPG*** (see Table 1). Similar trends were also noted in previous studies of GMM- and HMM-based speech parameter generations [2, 5, 21]. We also note that ***GMM+MLGV*** outperforms ***ANN+MLPG***, even though ***ANN+MLPG*** yields higher conversion accuracy than ***GMM+MLGV*** (see Table 1), again confirming the effectiveness of using GV for overcoming the over-smoothing problem.

Finally, when comparing the two GV-based VC systems, we can see that ***ANN+MLGV*** is as good as ***GMM+MLGV***. Although ***ANN+MLGV*** yields a better average mean score in speech quality test, the result is not significant (p value of t-test > 0.05). For a further analysis, it is interesting to note that the converted speech by ***ANN+MLGV*** sounds slightly clear and bright then that of ***GMM+MLGV*** according to the responses of the subjects. This may be attributed by that ***ANN+MLGV*** yields a better conversion accuracy than ***GMM+MLGV*** (see Table 1). Based on this observation, it is potentially to further improve the ANN-based VC with the MLGV algorithm by using context features (in order to achieve a better conversion accuracy) rather than dynamic features as reported by [6].

V. Conclusions

In this paper, we have presented a probabilistic interpretation for ANN-based VC. The major advantage of the proposed interpretation is that the successful techniques that were originally derived for SMPG can be suitably incorporated into the ANN-based VC system. The experimental results from both objective and subjective evaluations demonstrate that ANN-based VC with trajectory-based mapping techniques outperforms conventional ANN-based VC and provides comparable performance to GMM-based VC with the MLGV algorithm [2]. In the future, we will first work on expanding the developed ANN-based VC systems by incorporating advanced deep learning techniques. Moreover, it is noted that the probabilistic model for the proposed ANN-based VC may encounter an inconsistence between training and conversion when using a dynamic-feature constraint; such inconsistence can degrade the quality of converted speech [3, 4, 22]. Therefore, we will also explore suitable trajectory models for the proposed ANN-based VC.

Acknowledgment

The authors would like to thank Prof. H. Kawahara of Wakayama University, Japan, for the permission to use the STRAIGHT method.

References

[1] Y. Stylianou, O. Cappé, and E. Moulines, "Continuous probabilistic transform for voice conversion," *IEEE Trans. Speech Audio Process.*, vol. 6, no. 2, pp.131-142, Mar. 1998.

[2] T. Toda, A. W. Black, and K. Tokuda, "Voice conversion based on maximum-likelihood estimation of spectral parameter trajectory," *IEEE Trans. Audio, Speech, Lang., Process.,* vol. 15, no. 8, pp. 2222-2235, Nov. 2007.

[3] H. Zen, Y. Nankaku, and K. Tokuda, "Continuous stochastic feature mapping based on trajectory HMMs," *IEEE Trans. Audio, Speech, Lang., Process.,* vol. 19, no. 2, pp. 417-430, Feb. 2011.

[4] S. Takamichi, T. Toda, A. W. Black, and S. Nakamura, "Modulation spectrum-constrained trajectory training algorithm for GMM-based voice conversion," *Proc. ICASSP*, 2015.

[5] H. T. Hwang, Y. Tsao, H. M. Wang, Y. R. Wang and S. H. Chen, "Incorporating global variance in the training phase of GMM-based voice conversion," *Proc. APSIPA*, 2013.

[6] S. Desai, A. W. Black, B. Yegnanarayana, and K. Prahallad, "Spectral mapping using artificial neural networks for voice conversion," *IEEE Trans. Audio, Speech, Lang., Process.,* vol. 18, no. 5, pp. 954-964, 2010.

[7] F. L. Xie, Y. Qian, F. K. Soong, and H. Li, "Sequence error (SE) minimization training of neural network for voice conversion," *Proc. INTERSPEECH*, 2014.

[8] F. L. Xie, Y. Qian, F. K. Soong, and H. Li, "Pitch transformation in neural network based voice conversion," *Proc. ISCSLP*, 2014.

[9] L. H. Chen, Z. H. Ling, L. J. Liu, and L. R. Dai, "Voice conversion using deep neural networks with layer-wise generative training," *IEEE/ACM Trans. Audio, Speech, Lang., Process.,* vol. 22, no. 12, pp.1859-1872, 2014

[10] T. Nakashika, R. Takashima, T. Takiguchi, and Y. Ariki, "Voice conversion in high-order eigen space using deep belief nets," *Proc. INTERSPEEH*, 2013.

[11] D. Erro, A. Moreno, and A. Bonafonte, "Voice conversion based on weighted frequency warping," *IEEE Trans. Audio, Speech, Lang., Process.,* vol. 18, no. 5, pp. 922-931, July. 2010.

[12] Z. Wu, T. Virtanen, E. S. Chng, and H. Li, "Exemplar-based sparse representation with residual compensation for voice conversion," *IEEE/ACM Trans. Audio, Speech, Lang., Process.,* vol. 22, no. 10, pp.1506-1521, 2014.

[13] K. Tokuda, T. Yoshimura, T. Masuko, T. Kobayashi, and T. kitamura, "Speech parameter generation algorithms for HMM-based speech synthesis," *Proc. ICASSP*, 2000.

[14] Y. J. Wu, and R. H. Wang, "Minimum generation error training for HMM-based speech synthesis" *Proc. ICASSP*, 2006.

[15] H. Zen, A. Senior, and M. Schuster, "Statistical parametric speech synthesis using deep neural networks," *Proc. ICASSP*, 2013.

[16] Y. Qian, Y. Fan, W. Hu, and F. k., Soong, "On the training aspects of deep neural network (DNN) for parametric TTS synthesis", *Proc. ICASSP*, 2014.

[17] G. Hinton, S. Osindero, and Y. Teh, "A fast learning algorithm for deep belief nets", *Neural Computation*, vol. 18, pp. 1527-1554, 2006.

[18] Y. Bengio, "Learning deep architectures for AI," *Foundations and trends in Machine Learning*, vol. 2, no.1, pp.1-127, 2009.

[19] F. Seide, G. Li, X. Chen, and D. Yu, "Feature engineering in context dependent deep neural networks for conversational speech transcription," *Proc. ASRU*, 2011.

[20] M. Zeiler, M. Ranzato, R. Monga, M. Mao, K. Yang, Q.-V. Le, P. Nguyen, A. Senior, V. Vanhoucke, J. Dean, and G. Hinton, "On rectified linear units for speech processing," *Proc. ICASSP*, 2013.

[21] T. Toda, and K. Tokuda, "A speech parameter generation algorithm considering global variance for HMM-based speech synthesis," *IEICE Trans. Inf, Syst.,* vol. E90D, no. 5, pp. 816-824, 2007.

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

[23] H. Kawahara, I. Masuda-Katsuse, and A. de Cheveigné, "Restructuring speech representations using a pitch-adaptive time-frequency smoothing and an instantaneous-frequency-based F_0 extraction: Possible role of a repetitive structure in sounds," *Speech Commun.*, vol. 27, no. 3-4, pp.187-207, 1999.