Deep Acoustic-to-Articulatory Inversion Mapping with Latent Trajectory Modeling

Patrick Lumban Tobing*, Hirokazu Kameoka† and Tomoki Toda‡
* Graduate School of Information Science, Nagoya University, Japan
† NTT Communication Science Laboratories, NTT Corporation, Japan
‡ Information Technology Center, Nagoya University, Japan
E-mail: patrick.lumbantobing@g.sp.m.is.nagoya-u.ac.jp, kameoka.hirokazu@lab.ntt.co.jp, tomoki@icts.nagoya-u.ac.jp

***Abstract*—This paper presents a novel implementation of latent trajectory modeling in a deep acoustic-to-articulatory inversion mapping framework. In the conventional methods, i.e., the Gaussian mixture model (GMM)- and the deep neural network (DNN)-based inversion mappings, the frame interdependency can be considered while generating articulatory parameter trajectories with the use of an explicit constraint between static and dynamic features. However, in training these models, such a constraint is not considered, and therefore, the trained model is not optimum for the mapping procedure. In this paper, we address this problem by introducing a latent trajectory modeling into the DNN-based inversion mapping. In the latent trajectory model, the frame interdependency can be well considered, in both training and mapping, by using a soft-constraint between static and dynamic features. The experimental results demonstrate that the proposed latent trajectory DNN (LTDNN)-based inversion mapping outperforms the conventional and the state-of-the-art inversion mapping systems.**

I. Introduction

Articulators, such as tongue and lips, play a dominant role in determining the phonetic quality of a speech sound. Indeed, representations of the articulatory movements, e.g., articulatory parameters, have been used in various works, such as for speech recognition enhancement [1], for speech therapy/pronunciation learning [2], and for speech production/modification systems [3]. This infers that there is a rising need for a reliable acoustic-articulatory mapping framework.

In the recent years, in fact, there have been many notable works for developing robust statistical data-driven acoustic-to-articulatory inversion mapping frameworks. These include the codebook-based inversion mapping [4], the hidden Markov model (HMM)-based mapping [5], the Gaussian mixture model (GMM)-based mapping [6], and the artificial neural-network (ANN)-based mapping [7]. In this paper, due to its powerful predictive capability, we focus on the use of the deep neural-network (DNN)-based inversion mapping system.

In estimating articulatory parameter trajectories in the inversion mapping, it was reported that the use of temporal constraints reduces the errors of the estimated trajectory [8]. Furthermore, in the GMM-based [6] and the state-of-the-art mixture density network (MDN)-based inversion mappings [7], the use of maximum likelihood parameter generation (MLPG) [9], which explicitly uses a constraint between static and dynamic features, significantly improves the mapping accuracy. However, this constraint, which allows a consideration of frame interdependency, is not taken into account while training the corresponding models. In our previous work [10], based on the latent trajectory concept [11], we have proposed a latent trajectory training for the GMM-based inversion mapping, where the frame interdependency is considered in both training and mapping with a soft-constraint between a static feature sequence (regarded as an observed variable) and a static-dynamic feature sequence (regarded as a latent variable). Note that, this framework capable of using a well-formulated algorithm to optimize model parameters, such as the variational EM algorithm, is different from [12], in which such an algorithm is difficult to be used to optimize some of the model parameters.

In this paper, to make it possible to consider the frame interdependency in training while keeping the model parameter optimization as easy as in the conventional DNN-based inversion mapping, we propose a novel implementation of the latent trajectory modeling for the deep inversion mapping framework. Consistency between training and mapping is preserved by considering the frame interdependency within a latent space through the use of a soft-constraint between static and dynamic features. Moreover, almost the same model optimization procedure as in the conventional DNN training is available in the proposed latent trajectory DNN (LTDNN). The experimental results demonstrate the effectiveness of the proposed LTDNN-based inversion mapping, yielding superior mapping accuracy compared to the conventional methods.

II. Conventional Deep Acoustic-to-Articulatory Inversion Mappings

In this section, we describe two conventional deep architectures for the use of acoustic-to-articulatory inversion mapping, i.e., the deep neural network (DNN) and the minimum generation error DNN (MGEDNN) [13].

A. Feature description

Let $\boldsymbol{x}_t$ and $\boldsymbol{y}_t$ be the D_x- and D_y-dimensional acoustic and articulatory feature vector at frame t, respectively. The segmental acoustic feature vector is denoted as $\boldsymbol{X}_t = [\boldsymbol{x}_{t-C}^\top, \ldots, \boldsymbol{x}_t^\top, \ldots, \boldsymbol{x}_{t+C}^\top]^\top$, while the joint static-dynamic articulatory feature vector is denoted as $\boldsymbol{Y}_t = [\boldsymbol{y}_t^\top, \Delta\boldsymbol{y}_t^\top]^\top$ at frame t. The number of contextual frames is $2C + 1$.

B. Deep neural network (DNN)-based inversion mapping

In the DNN-based inversion mapping, the conditional probability density function (pdf) of the joint static-dynamic articulatory feature vector $\boldsymbol{Y}_t$, at frame t, is defined as follows:

$$P(\boldsymbol{Y}_t|\boldsymbol{X}_t, \boldsymbol{D}, \boldsymbol{\lambda}) = \mathcal{N}(\boldsymbol{Y}_t; f_\lambda(\boldsymbol{X}_t), \boldsymbol{D}), \quad (1)$$

where $f_\lambda(\cdot)$ is a nonlinear function given by the network, i.e., the network output. The network parameters, i.e., weights and biases are given in a set of the parameters $\boldsymbol{\lambda}$. The diagonal covariance matrix of the articulatory training data is denoted as $\boldsymbol{D}$.

In the training procedure, the updated network parameters $\hat{\boldsymbol{\lambda}}$ are determined with

$$\begin{aligned}\hat{\boldsymbol{\lambda}} &= \operatorname*{argmax}_{\boldsymbol{\lambda}} \sum_{t=1}^{T} \log P(\boldsymbol{Y}_t|\boldsymbol{X}_t, \boldsymbol{D}, \boldsymbol{\lambda}) \\ &= \operatorname*{argmin}_{\boldsymbol{\lambda}} \frac{1}{2} \sum_{t=1}^{T} (\boldsymbol{Y}_t - f_\lambda(\boldsymbol{X}_t))^\top \boldsymbol{D}^{-1} (\boldsymbol{Y}_t - f_\lambda(\boldsymbol{X}_t)). \quad (2)\end{aligned}$$

A graphical representation of the training process is given in the left diagram of Fig. 1. Following the MLPG procedure [9], given a segmental acoustic feature vector sequence $\boldsymbol{X} = [\boldsymbol{X}_1^\top, \ldots, \boldsymbol{X}_t^\top, \ldots, \boldsymbol{X}_T^\top]^\top$, the corresponding articulatory parameter trajectory $\hat{\boldsymbol{y}}_X^{(\lambda)}$ is given by

$$\begin{aligned}\hat{\boldsymbol{y}}_X^{(\lambda)} &= \operatorname*{argmax}_{\boldsymbol{\lambda}} \sum_{t=1}^{T} \log P(\boldsymbol{Y}_t|\boldsymbol{X}_t, \boldsymbol{D}, \boldsymbol{\lambda}) \text{ s.t. } \boldsymbol{Y} = \boldsymbol{W}\boldsymbol{y} \\ &= \operatorname*{argmin}_{\boldsymbol{\lambda}} \frac{1}{2} (\boldsymbol{W}\boldsymbol{y} - \boldsymbol{M}_X^{(\lambda)})^\top \boldsymbol{U}^{-1} (\boldsymbol{W}\boldsymbol{y} - \boldsymbol{M}_X^{(\lambda)}) \\ &= (\boldsymbol{W}^\top \boldsymbol{U}^{-1} \boldsymbol{W})^{-1} \boldsymbol{W}^\top \boldsymbol{U}^{-1} \boldsymbol{M}_X^{(\lambda)}, \quad (3)\end{aligned}$$

where

$$\boldsymbol{U} = \boldsymbol{I}_{T\times T} \otimes \boldsymbol{D}, \quad (4)$$

$$\boldsymbol{M}_X^{(\lambda)} = \left[f_\lambda(\boldsymbol{X}_1)^\top \ldots, f_\lambda(\boldsymbol{X}_t)^\top, \ldots, f_\lambda(\boldsymbol{X}_T)^\top \right]^\top, \quad (5)$$

and $\boldsymbol{W}$ is a transformation matrix to develop a joint static-dynamic articulatory feature vector sequence $\boldsymbol{Y} = [\boldsymbol{Y}_1^\top, \ldots, \boldsymbol{Y}_t^\top, \ldots, \boldsymbol{Y}_T^\top]^\top$ from $\boldsymbol{y} = [\boldsymbol{y}_1^\top, \ldots, \boldsymbol{y}_t^\top, \ldots, \boldsymbol{y}_T^\top]^\top$. The Kronecker product is denoted as $\otimes$.

C. Minimum generation error DNN (MGEDNN)-based inversion mapping

In the MGEDNN-based [13] inversion mapping, the conditional pdf of the articulatory feature vector sequence $\boldsymbol{y}$ is defined as follows:

$$P(\boldsymbol{y}|\boldsymbol{X}, \boldsymbol{D}, \boldsymbol{\lambda}) = \mathcal{N}(\boldsymbol{y}; \hat{\boldsymbol{y}}_X^{(\boldsymbol{\lambda})}, \boldsymbol{I}), \quad (6)$$

where $\hat{\boldsymbol{y}}_X^{(\lambda)}$ is given in (3).

In the training procedure, the updated network parameters $\hat{\boldsymbol{\lambda}}$ are determined with

$$\begin{aligned}\hat{\boldsymbol{\lambda}} &= \operatorname*{argmin}_{\boldsymbol{\lambda}} \frac{1}{2} (\boldsymbol{y} - \hat{\boldsymbol{y}}_X^{(\lambda)})^\top (\boldsymbol{y} - \hat{\boldsymbol{y}}_X^{(\lambda)}) \\ &= \operatorname*{argmax}_{\boldsymbol{\lambda}} \mathcal{N}(\boldsymbol{y}; \hat{\boldsymbol{y}}_X^{(\lambda)}, \boldsymbol{I}). \quad (7)\end{aligned}$$

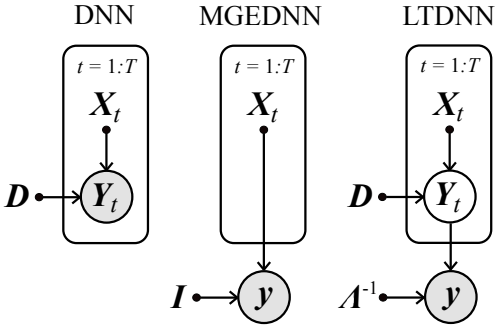

Fig. 1. Graphical representations for the training procedure of the conventional DNN, MGEDNN, and the proposed LTDNN. Unobserved variables are shaded.

A graphical representation of the training process is given in the middle diagram of Fig. 1. The network parameters need to be updated utterance by utterance, which is different from the frame-by-frame update available in the conventional DNN. Note that, due to the use of identity covariance matrix, it is difficult to apply articulatory parameter modification with covariance compensation as has been done in our previous work [10].

III. Proposed Deep Inversion Mapping with Latent Trajectory Modeling

In this section, we describe the proposed latent trajectory modeling [11] for the DNN-based inversion mapping, i.e., the latent trajectory DNN (LTDNN).

A. Latent trajectory model

Let the articulatory feature vector sequence $\boldsymbol{y}$ be the observed variable and the joint static-dynamic articulatory feature vector sequence $\boldsymbol{Y}$ be the latent variable. The following soft-constraint is used between the observed and the latent variables:

$$\boldsymbol{Y} \simeq \boldsymbol{W}\boldsymbol{y}. \quad (8)$$

Considering an error-covariance matrix $\boldsymbol{\Sigma}$, the conditional pdf of the latent variable $\boldsymbol{Y}$ can be defined as follows:

$$P(\boldsymbol{Y}|\boldsymbol{y}, \boldsymbol{\Sigma}) \propto \exp\left\{ -\frac{1}{2} (\boldsymbol{Y} - \boldsymbol{W}\boldsymbol{y})^\top \boldsymbol{\Sigma}^{-1} (\boldsymbol{Y} - \boldsymbol{W}\boldsymbol{y}) \right\}. \quad (9)$$

By completing the square of the exponential part of the above pdf, the conditional pdf of the observed articulatory feature sequence $\boldsymbol{y}$ is then defined as follows:

$$P(\boldsymbol{y}|\boldsymbol{Y}, \boldsymbol{\Sigma}) = \mathcal{N}(\boldsymbol{y}; \boldsymbol{H}\boldsymbol{Y}, \boldsymbol{\Lambda}^{-1}), \quad (10)$$

where

$$\boldsymbol{H} = \boldsymbol{\Lambda}^{-1} \boldsymbol{W}^\top \boldsymbol{\Sigma}^{-1}, \quad (11)$$

$$\boldsymbol{\Lambda} = \boldsymbol{W}^\top \boldsymbol{\Sigma}^{-1} \boldsymbol{W}. \quad (12)$$

Then, by marginalizing out the latent variable $\boldsymbol{Y}$, the likelihood function of the observed articulatory feature vector sequence $\boldsymbol{y}$ can be written as

$$P(\boldsymbol{y}|\boldsymbol{\Sigma}, \boldsymbol{\lambda}) = \int P(\boldsymbol{y}|\boldsymbol{Y}, \boldsymbol{\Sigma}) P(\boldsymbol{Y}|\boldsymbol{\lambda}) \mathrm{d}\boldsymbol{Y}, \quad (13)$$

where we can simplify a pdf of the latent variable $P(\boldsymbol{Y}|\boldsymbol{\lambda})$, e.g., $P(\boldsymbol{Y}|\boldsymbol{\lambda}) = \prod_t P(\boldsymbol{Y}_t|\boldsymbol{\lambda})$.

B. Proposed latent trajectory DNN (LTDNN)-based inversion mapping

In the proposed LTDNN, following the likelihood function in (13), given an input segmental acoustic feature vector sequence $\boldsymbol{X}$, the conditional pdf of the articulatory feature vector sequence $\boldsymbol{y}$ is defined as follows:

$$\begin{aligned} P(\boldsymbol{y}|\boldsymbol{X},\boldsymbol{\Sigma},\boldsymbol{D},\boldsymbol{\lambda}) &= \int P(\boldsymbol{y}|\boldsymbol{Y},\boldsymbol{\Sigma})\prod_{t=1}^{T} P(\boldsymbol{Y}_t|\boldsymbol{X}_t,\boldsymbol{D},\boldsymbol{\lambda})\mathrm{d}\boldsymbol{Y} \\ &= \int \mathcal{N}(\boldsymbol{y};\boldsymbol{H}\boldsymbol{Y},\boldsymbol{\Lambda}^{-1})\mathcal{N}(\boldsymbol{Y};\boldsymbol{M}_X^{(\lambda)},\boldsymbol{U})\mathrm{d}\boldsymbol{Y} \\ &= \int \mathcal{N}\left(\begin{bmatrix}\boldsymbol{y}\\ \boldsymbol{Y}\end{bmatrix};\begin{bmatrix}\boldsymbol{H}\boldsymbol{M}_X^{(\lambda)}\\ \boldsymbol{M}_X^{(\lambda)}\end{bmatrix},\begin{bmatrix}\boldsymbol{\Lambda}^{-1}+\boldsymbol{H}\boldsymbol{U}\boldsymbol{H}^\top & \boldsymbol{H}\boldsymbol{U}\\ \boldsymbol{U}\boldsymbol{H}^\top & \boldsymbol{U}\end{bmatrix}\right)\mathrm{d}\boldsymbol{Y} \\ &= \mathcal{N}(\boldsymbol{y};\boldsymbol{H}\boldsymbol{M}_X^{(\lambda)},\boldsymbol{\Lambda}^{-1}+\boldsymbol{H}\boldsymbol{U}\boldsymbol{H}^\top). \end{aligned} \tag{14}$$

In the training procedure, the updated network parameters $\hat{\boldsymbol{\lambda}}$ are determined with

$$\hat{\boldsymbol{\lambda}} = \underset{\boldsymbol{\lambda}}{\operatorname{argmax}} \log P(\boldsymbol{y}|\boldsymbol{X},\boldsymbol{\Sigma},\boldsymbol{D},\boldsymbol{\lambda}). \tag{15}$$

A graphical representation of the training process is given in the right diagram of Fig. 1.

An auxiliary function to assist in finding the updated parameters $\hat{\boldsymbol{\lambda}}$ can be defined as follows:

$$Q(\hat{\boldsymbol{\lambda}},\boldsymbol{\lambda}) = \int P(\boldsymbol{Y}|\boldsymbol{y},\boldsymbol{X},\boldsymbol{\Sigma},\boldsymbol{D},\boldsymbol{\lambda})\log P(\boldsymbol{Y}|\boldsymbol{X},\boldsymbol{D},\hat{\boldsymbol{\lambda}})\mathrm{d}\boldsymbol{Y}, \tag{16}$$

where

$$P(\boldsymbol{Y}|\boldsymbol{y},\boldsymbol{X},\boldsymbol{\Sigma},\boldsymbol{D},\boldsymbol{\lambda}) = \mathcal{N}(\boldsymbol{Y};\hat{\boldsymbol{Y}}_{y,X}^{(\lambda)},\hat{\boldsymbol{\Sigma}}_{y,X}^{(\lambda)}) \tag{17}$$

$$\hat{\boldsymbol{Y}}_{y,X}^{(\lambda)} = \boldsymbol{M}_X^{(\lambda)} + \boldsymbol{U}\boldsymbol{H}^\top(\boldsymbol{\Lambda}^{-1}+\boldsymbol{H}\boldsymbol{U}^{-1}\boldsymbol{H}^\top)^{-1}(\boldsymbol{y}-\boldsymbol{H}\boldsymbol{M}_X^{(\lambda)}) \tag{18}$$

$$\hat{\boldsymbol{\Sigma}}_{y,X}^{(\lambda)} = \boldsymbol{U} - \boldsymbol{U}\boldsymbol{H}^\top(\boldsymbol{\Lambda}^{-1}+\boldsymbol{H}\boldsymbol{U}^{-1}\boldsymbol{H}^\top)^{-1}\boldsymbol{H}\boldsymbol{U}. \tag{19}$$

In this paper, an approximation of the posterior pdf of the above auxiliary function is employed with delta function as follows:

$$\begin{aligned} Q(\hat{\boldsymbol{\lambda}},\boldsymbol{\lambda}) &\approx \int \delta(\boldsymbol{Y}=\hat{\boldsymbol{Y}}_{y,X}^{(\lambda)})\log P(\boldsymbol{Y}|\boldsymbol{X},\boldsymbol{D},\hat{\boldsymbol{\lambda}})\mathrm{d}\boldsymbol{Y} \\ &= \log P(\hat{\boldsymbol{Y}}_{y,X}^{(\lambda)}|\boldsymbol{X},\boldsymbol{D},\hat{\boldsymbol{\lambda}}). \end{aligned} \tag{20}$$

Therefore, the updated network parameters $\hat{\boldsymbol{\lambda}}$ are determined with

$$\begin{aligned} \hat{\boldsymbol{\lambda}} &= \underset{\boldsymbol{\lambda}}{\operatorname{argmax}} \log P(\hat{\boldsymbol{Y}}_{y,X}^{(\lambda)}|\boldsymbol{X},\boldsymbol{D},\hat{\boldsymbol{\lambda}}) \\ &= \underset{\boldsymbol{\lambda}}{\operatorname{argmin}} \frac{1}{2}\sum_{t=1}^{T}(\hat{\boldsymbol{Y}}_{y,X,t}^{(\lambda)}-f_\lambda(\boldsymbol{X}_t))^\top\boldsymbol{D}^{-1}(\hat{\boldsymbol{Y}}_{y,X,t}^{(\lambda)}-f_\lambda(\boldsymbol{X}_t)). \end{aligned} \tag{21}$$

Note that this equation is almost the same as (2). Therefore, the frame-by-frame network parameter update is still available although a mean vector of the posterior pdf needs to be estimated utterance by utterance as shown in (18).

In the mapping procedure, the estimated articulatory feature vector sequence $\hat{\boldsymbol{y}}_X^{(\lambda)}$ is given by

$$\begin{aligned} \hat{\boldsymbol{y}}_X^{(\lambda)} &= \underset{\boldsymbol{y}}{\operatorname{argmax}} P(\boldsymbol{y}|\boldsymbol{X},\boldsymbol{\Sigma},\boldsymbol{D},\boldsymbol{\lambda}) = \boldsymbol{H}\boldsymbol{M}_X^{(\lambda)} \\ &= (\boldsymbol{W}^\top\boldsymbol{\Sigma}^{-1}\boldsymbol{W})^{-1}\boldsymbol{W}^\top\boldsymbol{\Sigma}^{-1}\boldsymbol{M}_X^{(\lambda)}. \end{aligned} \tag{22}$$

Thanks to the use of the temporal covariance matrix, it would be straightforward to apply articulatory parameter modification as in [10] with the proposed LTDNN framework.

IV. Experimental evaluation

A. Experimental conditions

We used the multichannel articulatory (MOCHA) data [14] provided by CSTR, University of Edinburgh. From this dataset, the speech and the articulatory data, recorded with electromagnetic articulograph (EMA), were used. In this experiment, we used the female speaker dataset (fsew0). The total number of utterances was 460.

As the spectral envelope parameters, we used the first through 24^{th} mel-cepstral coefficients converted from the spectral envelope, which was extracted frame-by-frame using STRAIGHT [15] analysis. As the articulatory parameters, we used the 14-dimensional EMA data, consisting of the time-varying positions of seven articulators: lower incisor, upper lip, lower lip, tongue tip, tongue body, tongue dorsum, and velum; on x- and y-coordinate in the mid-sagittal plane. These articulatory parameters were converted into Z-scores. The speech data were sampled at 16 kHz. The frame shift was set to 5 ms. Starting and ending silence frames were included, 20 frames for each side, which were smoothed toward starting or ending zero values, respectively, using half-hanning windows.

We conducted experiments to evaluate the accuracy of the proposed inversion mapping system by comparing the generated articulatory trajectory with the measured ones. The proposed LTDNN-based inversion mapping system was compared to three baseline systems, i.e., GMM [6], DNN, and MGEDNN. The inversion mapping accuracy was measured by computing the root-mean-square error (RMSE). The number of training uterrances was varied to 46, 92, 184, and 368. The validation set contained 46 utterances, consisting of the files ending with 2, e.g.,"fsew0_002". The evaluation set contained also 46 utterances, but consisting of the files ending with 6.

The hyperparameters of the network were set as follows: the learning rate was set to 0.0006; the number of hidden units was set to 1024; the rectified linear unit (ReLU) was used as the non-linear activation function for the hidden units; the number of hidden layers was set to 5; the number of iteration limit for early-stopping was set to 20; and utterance mini-batch was used. The Adam [16] optimization algorithm was used to train the network parameters. The weights were randomly initialized with Xavier [17] initialization method. The biases were initialized with zero values. The trained DNN were used as the initial model for both the MGEDNN and the LTDNN. As for the GMM, the number of mixture components was set to 256, 512, 1024, and 2048, with respect to each number of training utterances, using a tied-covariance matrix.

TABLE I
AVERAGE OF ROOT-MEAN-SQUARE ERROR (RMSE) [MM] FOR ALL MODELS AND NUMBER OF TRAINING UTTERANCES

Model	Number of Training Utterances			
	46	92	184	368
GMM	1.655	1.558	1.498	1.423
DNN	1.576	1.464	1.391	1.326
MGEDNN	1.589	1.462	1.379	1.318
LTDNN	**1.575**	**1.450**	**1.371**	**1.302**

B. Experimental results

The results of the average RMSE from 14 articulatory dimensions for all combinations of models and number of training utterances are shown in Table I. The lowest RMSEs for all number of training utterances were achieved by the proposed LTDNN, yielding 1.575 mm, 1.450 mm, 1.371 mm, and 1.302 mm, for the 46, 92, 184, and 368 training utterances, respectively. Note that, as a related work, the best result with a similar data configuration was achieved using the state-of-the-art mixture density network (MDN)-based inversion mapping in [7] with 1.370 mm RMSE using 368 training utterances.

Considering that the movements of tongue and jaw are ones of the most dominant in speech production mechanism, we present also the RMSE for five important articulators, i.e., lower incisor (LI), lower lip (LL), tongue tip (TT), tongue body (TB), and tongue dorsum (TD), on y-coordinate in the mid-sagittal plane. The RMSEs computed from each of the optimum models are given in Table II. The lowest RMSEs for all articulatory dimensions, i.e., LI_y, LL_y, TT_y, TB_y, and TD_y, are achieved by the proposed LTDNN yielding 1.067 mm, 2.040 mm, 1.823mm, 1.738 mm, and 1.843 mm, respectively. Note that in the related work [7], the lowest achieved RMSEs are respectively given as 1.030 mm, 2.200 mm, 1.940 mm, 1.730 mm, and 1.850 mm.

These experimental results show that the proposed latent trajectory modeling for the deep acoustic-to-articulatory inversion mapping improves the mapping accuracy compared to the conventional baseline systems. Moreover, although it cannot be compared directly, the proposed LTDNN achieves higher accuracy than that of the state-of-the-art MDN-based inversion mapping method with the same dataset configuration.

TABLE II
RMSE [MM] FOR FIVE IMPORTANT ARTICULATORY DIMENSIONS: LOWER INCISOR (LI), LOWER LIP (LL), TONGUE TIP (TT), TONGUE BODY (TB), AND TONGUE DORSUM (TD) ON y-COORDINATE IN THE MID-SAGITTAL PLANE (USING THE OPTIMUM MODEL FOR EACH SYSTEM)

Model	Articulatory Dimension				
	LI_y	LL_y	TT_y	TB_y	TD_y
GMM	1.085	2.258	2.068	1.890	1.910
DNN	1.078	2.121	1.831	1.773	1.886
MGEDNN	1.124	2.095	1.831	1.791	1.872
LTDNN	**1.067**	**2.040**	**1.823**	**1.738**	**1.843**

V. CONCLUSIONS

We have proposed the latent trajectory modeling in deep acoustic-to-articulatory inversion mapping systems. The latent trajectory model allows frame-interdependency to be considered in training the model by utilizing a soft-constraint between static and dynamic features in the latent space. The experimental results show that the proposed LTDNN yields higher inversion mapping accuracy compared to the conventional inversion mapping systems. In the future, we would like to combine the latent trajectory modeling with deep mixture density network.

ACKNOWLEDGMENT

Part of this work was supported by JST, PRESTO Grant Number JPMJPR1657 and JSPS KAKENHI Grant Number 17H01763.

REFERENCES

[1] K. Kirchhoff, G. A. Fink, and G. Sagerer., "Combining acoustic and articulatory feature information for robust speech recognition," *Speech Commun.*, vol. 37, no. 3, pp. 303–319, 2002.
[2] B. J. Kröger, V. Graf-Borttscheller, and A. Lowit, "Two- and three-dimensional visual articulatory models for pronunciation training and for treatment of speech disorders," in *Proc. INTERSPEECH*, Brisbane, Australia, Sep. 2008, pp. 2639–2642.
[3] P. L. Tobing, K. Kobayashi, T. Toda, G. Neubig, S. Sakti, and S. Nakamura, "Articulatory controllable speech modification based on Gaussian mixture models with direct waveform modification using spectrum differential," in *Proc. INTERSPEECH*, Dresden, Germany, Sep. 2015, pp. 3350–3354.
[4] J. Hogden, A. Lofqvist, V. Gracco, I. Zlokarnik, P. Rubin, and E. Saltzman, "Accurate recovery of articulator positions from acoustics: New conclusions based on human data," *J. Acoust. Soc. Am.*, vol. 100, no. 3, pp. 1819–1834, 1996.
[5] S. Hiroya and M. Honda, "Estimation of articulatory movements from speech acoustics using an HMM-based speech production model," *IEEE Speech Audio Process.*, vol. 12, no. 2, pp. 175–185, 2004.
[6] T. Toda, A. W. Black, and K. Tokuda, "Statistical mapping between articulatory movements and acoustic spectrum using a Gaussian mixture model," *Speech Commun.*, vol. 50, no. 3, pp. 215–227, 2008.
[7] K. Richmond, "Trajectory mixture density networks with multiple mixtures for acoustic-articulatory inversion," in *Proc. NOLISP*, Paris, France, May 2007, pp. 263–272.
[8] S. Suzuki, T. Okadome, and M. Honda, "Determination of articulatory positions from speech acoustics by applying dynamic articulatory constraints," in *Proc. ICSLP*, Sydney, Australia, 1998, pp. 2251–2254.
[9] K. Tokuda, T. Kobayashi, and S. Imai, "Speech parameter generation from HMM using dynamic features," in *Proc. ICASSP*, Detroit, USA, May 1995, pp. 660–663.
[10] P. L. Tobing, T. Toda, H. Kameoka, and S. Nakamura, "Acoustic-to-articulatory inversion mapping based on latent trajectory Gaussian mixture model," in *Proc. INTERSPEECH*, San Francisco, USA, Sep. 2016, pp. 953–957.
[11] H. Kameoka, "Modeling speech parameter sequences with latent trajectory hidden Markov model," in *Proc. MLSP*, Boston, USA, Sep. 2015, pp. 1–6.
[12] L. Zhang and S. Renals, "Acoustic-articulatory modeling with the trajectory HMM," *IEEE Signal Processing Letters*, vol. 15, pp. 245–248, 2008.
[13] Z. Wu and S. King, "Minimum trajectory error training for deep neural networks, combined with stacked bottleneck features," in *Proc. INTERSPEECH*, Dresden, Germany, Sep. 2015, pp. 309–313.
[14] A. Wrench. (1999) The MOCHA-TIMIT articulatory database. Queen Margaret University College. [Online]. Available: http://www.cstr.ed.ac.uk/artic/mocha.html
[15] H. Kawahara, I. Masuda-Katsuse, and A. de Cheveigné, "Restructuring speech representation using a pitch-adaptive time frequency smoothing and an instantaneous-frequency-based F0 extraction: Possible role of a repetitive structure in sounds," *Speech Commun.*, vol. 27, no. 3–4, pp. 187–207, 1999.
[16] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *CoRR*, vol. abs/1412.6980, 2014. [Online]. Available: http://arxiv.org/abs/1412.6980
[17] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," in *AISTATS*, vol. 9, 2010, pp. 249–256.