Available online at www.sciencedirect.com

ScienceDirect

Computer Speech and Language 36 (2016) 260–273

COMPUTER SPEECH AND LANGUAGE

www.elsevier.com/locate/csl

Data driven articulatory synthesis with deep neural networks☆

Sandesh Aryal, Ricardo Gutierrez-Osuna*

Department of Computer Science and Engineering, Texas A&M University, United States

Received 2 July 2014; received in revised form 4 December 2014; accepted 24 February 2015
Available online 5 March 2015

Abstract

The conventional approach for data-driven articulatory synthesis consists of modeling the joint acoustic-articulatory distribution with a Gaussian mixture model (GMM), followed by a post-processing step that optimizes the resulting acoustic trajectories. This final step can significantly improve the accuracy of the GMM frame-by-frame mapping but is computationally intensive and requires that the entire utterance be synthesized beforehand, making it unsuited for real-time synthesis. To address this issue, we present a deep neural network (DNN) articulatory synthesizer that uses a tapped-delay input line, allowing the model to capture context information in the articulatory trajectory without the need for post-processing. We characterize the DNN as a function of the context size and number of hidden layers, and compare it against two GMM articulatory synthesizers, a baseline model that performs a simple frame-by-frame mapping, and a second model that also performs trajectory optimization. Our results show that a DNN with a 60-ms context window and two 512-neuron hidden layers can synthesize speech at four times the frame rate – comparable to frame-by-frame mappings, while improving the accuracy of trajectory optimization (a 9.8% reduction in Mel Cepstral distortion). Subjective evaluation through pairwise listening tests also shows a strong preference toward the DNN articulatory synthesizer when compared to GMM trajectory optimization.

Keywords: Articulatory synthesis; Electromagnetic articulography; Deep learning; Gaussian mixture models

1. Introduction

Certain speech modifications, such as changes in foreign/regional accents and articulatory styles are too difficult to perform in the acoustic domain, where voice quality information and speaking style interact in complex ways (Hermansky and Broad, 1989). By contrast, these two sources are readily decoupled in the articulatory domain through the position and dynamics of articulators (e.g., as measured via electromagnetic articulography). Following this intuition, in previous work (Felps et al., 2012; Aryal and Gutierrez-Osuna, 2014) we have shown how accent conversion can be performed by driving an articulatory synthesizer of a non-native speaker with articulatory gestures from a native speaker.

Our original articulatory-based method for accent conversion (Felps et al., 2012) was based on unit-selection synthesis. Given a corpus of articulatory-acoustic frames for a second-language (L2) speaker and a reference first-language

☆ This paper has been recommended for acceptance by S. Narayanan.

* Corresponding author. Tel.: +1 9798452942.
E-mail addresses: sandesh@cse.tamu.edu (S. Aryal), rgutier@cse.tamu.edu (R. Gutierrez-Osuna).

http://dx.doi.org/10.1016/j.csl.2015.02.003

(L1) utterance, the approach consisted of finding a sequence of L2 diphones with similar articulatory configurations as those in the reference L1 utterance. Unfortunately, the approach provided only modest improvements in accent reduction due to the small size of the L2 acoustic-articulatory corpus and the fact that unit-selection cannot produce sounds that do not already exist in the L2 corpus. For these reasons, our recent work (Aryal and Gutierrez-Osuna, 2014) has focused on parametric statistical techniques proposed by Toda et al. (2004). The approach uses a Gaussian mixture to model the joint acoustic-articulatory distribution, followed by an optimization stage to find the maximum-likelihood acoustic trajectory for an articulatory sequence. This GMM framework is better suited for the limited size of articulatory-acoustic corpora and can interpolate sounds that do not exist in the L2 inventory. However, it is computationally intensive and impractical for real-time synthesis since the trajectory-optimization stage requires that the entire utterance be present at synthesis time.

To address the limitations of the above GMM framework for real-time articulatory synthesis, this paper explores the use of deep neural networks (DNN) to perform the articulatory-to-acoustic mapping. Following prior work on DNN for speech recognition, articulatory inversion and text-to-speech synthesis (Hinton et al., 2012; Uria et al., 2012; Zen et al., 2013), our approach uses a tapped-delay line to contextualize features by forming an input vector of short-term articulatory sequences from which to predict acoustic observations. Because temporal information is encoded in the tapped-delay line, the resulting acoustic sequence does not require the costly trajectory optimization of GMM articulatory synthesis. We compare the proposed DNN against two GMM implementations, a static GMM that performs a frame-by-frame mapping from articulators to acoustics, and a dynamic GMM that uses trajectory optimization. Our results show that the DNN provides a more accurate mapping, measured as Mel cepstral distortion, than either GMM implementation. A second comparison between the DNN and the GMM with different tapped-delay lengths shows that DNN accuracy increases monotonically for contexts of up to 60 ms, whereas GMM accuracy degrades drastically for contexts larger than 20 ms. Compared to trajectory optimization, which requires an average of 39 s of synthesis time per second of speech, the DNN requires only 267 ms per second of speech, making it suitable for real-time synthesis. A final subjective assessment through pairwise listening tests shows a strong preference (73%) toward the DNN synthesizer.

The remaining sections of this paper are organized as follows. Section 2 reviews related work in articulatory-to-acoustic mappings. Section 3 presents the proposed DNN architecture and the two GMM synthesizers used for comparison. Section 4 describes the experimental setup used for model evaluation, followed by experimental results in Section 5. The article concludes with a discussion of these results and directions for future work.

2. Related work

2.1. *Articulatory-to-acoustic mappings*

A significant amount of research has been performed toward understanding the forward mapping from articulators to acoustics and developing methods to build such mappings. These efforts can be grouped into two broad categories: physics-based models and data-driven models. Physics-based models approximate vocal tract geometry using a stack of cylindrical tubes with different cross section areas. Speech waveforms are then generated by solving the wave propagation equation in the approximated tube model (Mermelstein, 1973; Browman et al., 1984; Maeda, 1990; Birkholz et al., 2006). In contrast, data-driven approaches use machine learning techniques to build a forward mapping from simultaneous recordings of articulators and acoustics; it is this latter group of models that our review will focus on.

In one of the earliest studies, Kaburagi and Honda (1998) proposed a codebook algorithm for data-driven articulatory synthesis. Given a target articulatory frame, its acoustic observation was estimated by finding the closest articulatory frames in the corpus, and then computing a weighted average of their acoustic observations. Unfortunately, the nearest-neighbors search makes the synthesis process computationally expensive, and synthesis quality is limited by the relatively small size of articulatory corpora. Over the past decade, codebook techniques have been replaced with parametric models (Hiroya and Honda, 2004; Toda et al., 2004; Nakamura et al., 2006), which are better suited when the size of the corpus is limited. In a landmark study, Toda et al. (2004) used Gaussian mixture models (GMM) to learn the joint distribution of articulatory and acoustic parameters. Given a trained GMM and a vector of articulatory parameters, estimating the corresponding acoustic parameters in a frame-by-frame fashion involves a fixed number of operations independent of the database size.

Frame-by-frame mappings are computationally efficient but ignore the temporal nature of speech; as shown by a number of studies (Toda et al., 2004; Nakamura et al., 2006), the accuracy of the forward mapping can be increased by incorporating temporal information. As an example, Toda et al. (2004) compared the output of their frame-by-frame GMM mapping against an algorithm that also considered dynamical features to provide natural transition across adjacent frames. The latter method was more accurate and also resulted in more natural synthesis than the frame-by-frame mapping. In a later study, Nakamura et al. (2006) explored the use of dynamic information not only at the output (acoustics) but also at the input (articulators). For this purpose, the authors used context-dependent hidden Markov models (HMM) to capture the temporal characteristics of joint articulatory-acoustic observations. Given a text transcription and the articulatory trajectories for a test utterance, the HMMs were then used to estimate a maximum likelihood sequence of acoustic parameters. The authors found that increasing the amount of temporal information (number of HMM states) in the articulatory trajectory increased the accuracy of the articulatory-acoustic mapping.

Notwithstanding their accuracy (Toda et al., 2004; Nakamura et al., 2006), however, these methods are unsuited for real-time synthesis. In both cases, the increase in accuracy with respect to frame-by-frame mappings comes at the expense of much higher computational costs during synthesis because of the iterative estimation process. Moreover, these methods also require the complete sequence of articulatory frames from a test utterance before their corresponding acoustics can be estimated – a further limitation for real-time synthesis applications. Thus, exploiting the temporal structure of speech without adversely impacting articulatory-synthesis time remains challenging.

2.2. *Deep neural network in articulatory-acoustic mappings*

Connectionist models such as neural networks have rarely been used in forward-mapping problems, where GMMs are considered the de-facto standard. One notable exception is the work by Kello and Plaut (2004), who used a single-layer multilayer perceptron (MLP) to estimate acoustic features (Fourier transform coefficients) from electromagnetic articulography (EMA), electropalatograph and laryngograph measurements. In an intelligibility test, the authors reported a word identification rate of 84% for synthesized speech, only 8% lower than that of the actual recordings.

Compared to single-layer MLPs, DNNs can be expected to provide higher forward-mapping accuracy. First, the presence of multiple hidden layers makes DNNs more flexible models, allowing them to represent complex functions with fewer hidden units. Second, DNNs are pre-trained as generative models in an unsupervised mode, a step that has been shown to guide the learning process toward parameters that support better generalization (Erhan et al., 2010). For example, Nakashika et al. (2013) trained deep belief networks (DBNs) as generative models to project the spectral features into a higher-order eigenspace. The authors found that voice conversion in eigenspace outperformed the GMM-based spectral mappings in synthesis quality and conversion accuracy. These predictions have also been corroborated in several speech-related applications (Hinton et al., 2012; Uria et al., 2012; Zen et al., 2013; Xu et al., 2014), where DNN-based methods have surpassed the performance of state-of-the-art methods based on the HMM-GMM framework. Among these, a study on articulatory inversion by Uria et al. (2012) is particularly relevant here given the similarity between both problems. Using a DNN, the authors were able to estimate EMA pellet positions with an average root mean square error of 0.95 mm on the MNGU0 test dataset, an error that was not only lower than that of a single-layer MLP but also the lowest among all previously published results on that dataset. A recent study by Andrew et al. (2013) on joint articulatory-acoustic modeling also highlights the superiority of deep learning techniques in this domain. The authors proposed a deep architecture for canonical correlation analysis (CCA) and tested it on the Wisconsin X-ray Microbeam Database (Westbury, 1994). Their deep CCA method achieved significantly higher correlation between the transformed acoustic and articulatory spaces than conventional CCA and kernel-based CCA (Arora and Livescu, 2013) and also compared favorably against kernel-CCA in terms of flexibility and computational complexity. These results are encouraging and support our exploration of DNNs for real-time articulatory synthesis.

3. Methods

Consider a sequence of articulatory feature vectors[1] $\boldsymbol{x} = [\boldsymbol{x}_1, \boldsymbol{x}_2, \boldsymbol{x}_3, \ldots, \boldsymbol{x}_T]$ and the corresponding acoustic feature vectors $\boldsymbol{y} = [\boldsymbol{y}_1, \boldsymbol{y}_2, \boldsymbol{y}_3, \ldots, \boldsymbol{y}_T]$, where T is the number of frames in the utterance. A classical problem in

[1] Our notation assumes row vectors.

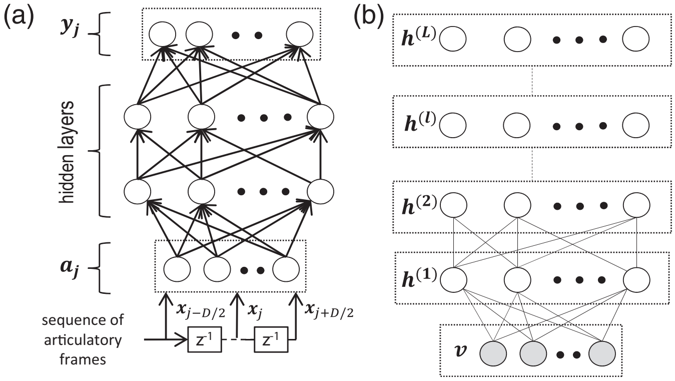


Fig. 1. (a) Forward mapping via a deep neural network (DNN) with a tapped-delay line input. (b) The Gaussian–Bernoulli deep Boltzmann machine as an undirected graphical model with real valued visible units v and binary hidden units h.

speech research is that of articulatory inversion (Richmond et al., 2003; Qin and Carreira-Perpinán, 2007; Ozbek et al., 2009; Rudzicz, 2010; Ghosh and Narayanan, 2011; Prabhavalkar et al., 2011), which seeks to predict articulatory configurations from speech acoustics: $\boldsymbol{x}=\boldsymbol{f}^{-1}(\boldsymbol{y})$. Articulatory inversion is a notoriously difficult problem since it requires a one-to-many mapping. Our study focuses on the forward mapping $\boldsymbol{y}=\boldsymbol{f}(\boldsymbol{x})$ of predicting acoustics from articulators; this is a well-behaved problem since the relationship becomes many-to-one. In this section, we present a method to build such a forward mapping using a Deep Neural Network (DNN) with tapped-delay line inputs. We also introduce the baseline GMM-based methods that will be used for comparison (Toda et al., 2004).

3.1. Deep forward mapping

As illustrated in Fig. 1(a), the DNN consists of an input layer, an output layer, and multiple layers of hidden units between them. In this particular topology, units in a layer are fully connected to units in the immediate layer above it, but there is no connection among units within a layer. The network contains a tapped-delay line to contextualize the input with features from past and future frames, resulting in the input vector $\boldsymbol{a}_j=\{\boldsymbol{x}_{j-D/2},\ldots,\boldsymbol{x}_j,\ldots,\boldsymbol{x}_{j+D/2}\}$, where $\boldsymbol{x}_j$ is the articulatory configuration at frame j, and D is the number of delay units in the tapped-delay line. When $D=0$, the input vector $\boldsymbol{a}_j$ becomes the articulatory feature vector x_j, and the DNN performs a *frame-by-frame* mapping. Increasing the value of D allows the DNN to include additional temporal context to aid in predicting the acoustic observation $\boldsymbol{y}_j$.

We train the DNN using the conventional two-stage hybrid recipe (Hinton, 2012). During the first stage, model parameters (for all but the last layer) are learned in an unsupervised fashion; this pre-training stage makes it more likely to find a good local optimum than using randomly initialized parameters (Erhan et al., 2010). During the second stage, the pre-trained model (including the last layer) is fine-tuned in a supervised fashion via back-propagation (Rumelhart et al., 1986).

3.2. Pre-training the network as a generative model

During pre-training the network is operated as a Gaussian–Bernoulli deep Boltzmann machine (GDBM), an energy-based generative model that allows each unit to receive both top-down and bottom-up signals as shown in Fig. 1(b) (Cho et al., 2013). Unlike a generic deep Boltzmann machine (Salakhutdinov and Hinton, 2009), which has binary units in all of its layers, the GDBM has Gaussian units in the visible layer, making it better suited to handle real-valued inputs. We chose GDBM to pre-train the DNN – instead of the more common deep belief network (DBN), because DBM-based pretraining has been found to perform better in some standard tasks (Salakhutdinov and Hinton, 2009).

3.3. Building a DNN from a trained GDBM

Once the underlying GDBM is trained, we build a DNN as follows. First, a layer of output units is added to the topmost hidden layer of the GDBM, one output unit for each corresponding acoustic feature. Connection weights between the units at the topmost hidden layer and the newly added output layer are initialized randomly. The resulting multilayer neural network is then discriminatively fine-tuned using standard back-propagation (Rumelhart et al., 1986).

3.4. GMM-based baseline methods

We compared the proposed DNN against two GMM-based methods based on Toda et al. (2004). The first method, which we denote by sGMM, ignores dynamic information and serves as a baseline for real-time synthesis. Namely, sGMM performs a frame-by-frame mapping from articulatory positions onto *static* acoustic features (MFCCs). The second method, dGMM, incorporates the dynamics of acoustic features to improve the forward-mapping accuracy. Namely, dGMM predicts not only MFCCs but also delta-MFCCs, and then performs the computationally-intensive trajectory optimization (Toda et al., 2004). As such, dGMM is unsuited for real-time synthesis so it should be taken as an upper bound on accuracy.

The GMMs required for both methods are trained to model the joint distribution of articulatory and acoustic features $\boldsymbol{Z}_t = [\boldsymbol{x}_t, \boldsymbol{Y}_t]$, where $\boldsymbol{x}_t$ is the articulatory feature vector and $\boldsymbol{Y}_t = [\boldsymbol{y}_t, \Delta\boldsymbol{y}_t]$ is an acoustic feature vector containing both static and delta values at frame t. The joint distribution is given by:

$$p(\boldsymbol{Z}_t|\boldsymbol{\lambda}^{(z)}) = \sum_{m=1}^{M} \alpha_m N(\boldsymbol{Z}_t; \boldsymbol{\mu}_m^{(z)},\ \Sigma_m^{(z)}) \tag{1}$$

where α_m is the scalar weight of the mth mixture component and $N(;\boldsymbol{\mu}_m^{(z)}, \boldsymbol{\Sigma}_m^{(z)})$ is the Gaussian distribution with mean $\boldsymbol{\mu}_m^{(z)}$ and covariance matrix $\Sigma_m^{(z)}$:

$$\boldsymbol{\mu}_m^{(Z)} = \begin{bmatrix} \boldsymbol{\mu}_m^{(x)} & \boldsymbol{\mu}_m^{(Y)} \end{bmatrix}, \quad \boldsymbol{\Sigma}_m^{(Z)} = \begin{bmatrix} \boldsymbol{\Sigma}_m^{(xx)} & \boldsymbol{\Sigma}_m^{(xY)} \\ \boldsymbol{\Sigma}_m^{(Yx)} & \boldsymbol{\Sigma}_m^{(YY)} \end{bmatrix} \tag{2}$$

In what follows, we use the symbol $\boldsymbol{\lambda}^{(z)} = \{\alpha_m, \boldsymbol{\mu}_m^{(z)}, \boldsymbol{\Sigma}_m^{(z)}\}$ to denote the full parameter set for the GMM. Given a trained GMM and a test sequence of articulatory feature vectors $\boldsymbol{x} = [\boldsymbol{x}_1, \boldsymbol{x}_2, \boldsymbol{x}_3, \ldots, \boldsymbol{x}_T]$, we generate separate predictions of acoustic feature vectors $\boldsymbol{y} = [\boldsymbol{y}_1, \boldsymbol{y}_2, \ldots, \boldsymbol{y}_T]$ for the two GMM variants as follows:

(1) For sGMM, we ignore the acoustics dynamics and calculate the static acoustic feature vector at frame t as the minimum mean square error (MMSE[2]) estimate:

$$\hat{\boldsymbol{y}}_{t,MMSE} = \sum\nolimits_{m=1}^{M} P(m|\boldsymbol{x}_t, \boldsymbol{\lambda}^{(z)}) \boldsymbol{E}_{m,t}^{(y)} \tag{3}$$

where $\boldsymbol{E}_{m,t}^{(y)}$ is the subset of static features in the conditional expected value $\boldsymbol{E}_{m,t}^{(Y)}$, as given by

$$\boldsymbol{E}_{m,t}^{(Y)} = \boldsymbol{\mu}_m^{(Y)} + (\boldsymbol{x}_t - \boldsymbol{\mu}_m^{(x)}) \boldsymbol{\Sigma}_m^{xx^{-1}}\ \boldsymbol{\Sigma}_m^{(xY)}. \tag{4}$$

(2) For dGMM, we calculate the maximum likelihood estimate of the acoustic trajectory considering the dynamics, as given by:

$$\hat{\boldsymbol{y}} = arg \max_{y} P(\boldsymbol{Y}|\boldsymbol{x}, \boldsymbol{\lambda}^{(z)}) \tag{5}$$

where $\boldsymbol{Y} = [\boldsymbol{y}_1, \Delta\boldsymbol{y}_1, \boldsymbol{y}_2, \Delta\boldsymbol{y}_2, \ldots, \boldsymbol{y}_T, \Delta\boldsymbol{y}_T]$ is the time sequence of acoustic vectors (both static and dynamic).

[2] With static only features, we chose MMSE because Toda et al. (2004) reported lower Cepstral distortion compared to the ML estimate.

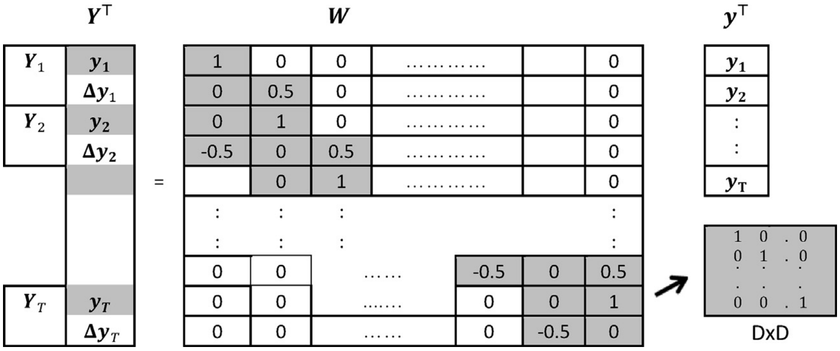
Calculating $\hat{\boldsymbol{y}}_{t,MMSE}$ in Eq. (3) is straightforward but there is no closed form solution for solving $\hat{\boldsymbol{y}}$ in Eq. (5). We follow the algorithm given by (Toda et al., 2004) to solve for $\hat{\boldsymbol{y}}$ in Eq. (5). The algorithm details are included here to illustrate the computational complexity of the algorithm. Following (Toda et al., 2004), we solve for $\hat{\boldsymbol{y}}$ in Eq. (5) iteratively via Expectation-Maximization. Namely, we define the auxiliary function with respect to $\hat{\boldsymbol{y}}$ as:

$$Q(\boldsymbol{Y},\hat{\boldsymbol{Y}}) = \sum_{m} P(m|\boldsymbol{x},\boldsymbol{Y},\boldsymbol{\lambda}^{(z)}) \log P(\hat{\boldsymbol{Y}},m|X,\boldsymbol{\lambda}^{(z)}) \tag{6}$$

At each M-step, we estimate the trajectory (static elements only) that maximizes auxiliary function $Q(\boldsymbol{Y},\hat{\boldsymbol{Y}})$ as:

$$\hat{\boldsymbol{y}} = (\boldsymbol{W}^T\overline{\boldsymbol{D}}\boldsymbol{W})^{-1}\boldsymbol{W}^T\bar{\Psi} \tag{7}$$

where $\boldsymbol{W}$ is the $2DT \times DT$ matrix that translates a trajectory of the static parameters to a trajectory of the complete acoustic feature vector as given by:



(8)

In Eq. (7), $\bar{\boldsymbol{D}}$ is a block-diagonal matrix whose diagonal consists of T covariance sub-matrices $\sum_{m=1}^{M} P(m|\boldsymbol{x}_t,\boldsymbol{Y}_t,\boldsymbol{\lambda}^{(v)})\,\boldsymbol{D}_m^{(Y)^{-1}}$; $t = 1\ldots T$ and $\bar{\Psi}$ is a row vector of length $2DT$ consisting of T sub-vectors of length $2D$ given by $\sum_{m=1}^{M} P(m|\boldsymbol{x}_t,\boldsymbol{Y}_t,\boldsymbol{\lambda}^{(v)})\,\boldsymbol{D}_m^{(Y)^{-1}}\,\boldsymbol{E}_{m,t}^{(Y)}$; $t = 1\ldots T$, where

$$\boldsymbol{D}_m^{(Y)} = \boldsymbol{\Sigma}_m^{(YY)} - \boldsymbol{\Sigma}_m^{(Yx)}\boldsymbol{\Sigma}_m^{(xx)^{-1}}\boldsymbol{\Sigma}_m^{(xY)} \tag{9}$$

The algorithm requires an initial estimate of the trajectory of the static acoustic features $\widehat{\boldsymbol{y}}$. In our implementation, we initialize with the minimum mean square error (MMSE) estimate as given by Eq. (3), which ignores the dynamics.

4. Experimental

We evaluated the three forward mappings (DNN, sGMM, dGMM) on a corpus of simultaneous recordings of acoustics and articulatory trajectories recorded via electromagnetic articulography (EMA) (Felps et al., 2012; Aryal and Gutierrez-Osuna, 2014). The corpus contained 344 phonetically-balanced sentences from the Glasgow Herald; 294 of them (approximately 1290 s of speech) were used in training and the remaining 50 sentences (~220 s) were used exclusively for testing.[3] The articulatory features available were the *x*–*y* coordinates of 6 articulatory flesh points (tongue tip, tongue body, tongue dorsum, upper lip, lower lip and lower incisor) in the midsagittal cross section of the vocal tract; see Fig. 2. The articulatory position data were sampled at 200 Hz.

We used STRAIGHT (Kawahara, 1997) to extract acoustic features from the acoustic recordings. First, we extracted *spectrum*, *fundamental frequency* (f_0) and *aperiodicity* for each utterance using STRAIGHT analysis. Then, we computed 25 *MFCCs* from the STRAIGHT *spectrum* (25 Mel frequency filterbanks with a cutoff frequency of 8 kHz). In order to match the articulatory frames, acoustic features were also extracted at 200 Hz. After all the features were extracted, a database of acoustic-articulatory feature vectors was created: the acoustic feature vector $\boldsymbol{y}_t$ consisting of $MFCC_{1\text{–}24}$, and the articulatory feature vector $\boldsymbol{a}_t$ consisting of the *x*–*y* coordinates of 6 articulatory fleshpoints, *frame energy* ($MFCC_0$), logarithm of fundamental frequency (i.e. $logf_0$), and *nasality*, for a total of 15 articulatory features.

[3] The models were evaluated on these test utterances, which were never seen by the model at any stage during training.

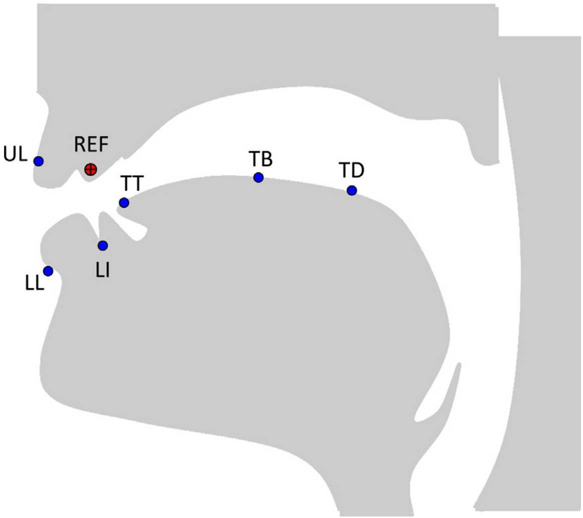


Fig. 2. Position of the 6 EMA pellets used in our study; UL: upper lip; LL: lower lip; LI: lower incisor; TT: tongue tip; TB: tongue blade; TD: tongue dorsum. An additional pellet (red cross-hair) was placed on the upper incisor and served as a reference. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

The *nasality* features were extracted from the transcription because EMA cannot measure velum position; without this feature, articulatory synthesis degrades significantly for nasals since they become indistinguishable from other consonants with similar place of articulation. All acoustic and articulatory features were normalized to zero-mean and unit-variance.

For the two GMM-based mappings, we trained GMMs with 128 mixture components[4] on the joint distribution of articulatory and acoustic features (including delta) using the Netlab toolbox (Nabney, 2002). Once the GMMs were trained, we estimated acoustic features using *sGMM* and *dGMM* methods as described by Eqs. (3) and (5), respectively. We also evaluated the performance of both GMM-based methods on different context sizes of articulatory input. For the DNN mapping, we used a tapped-delay line with delay units of 10 ms (≈2 frames), and evaluated tapped-delays with 2, 4, 6, and 8 delay units. As an example, for a delay line with 6 units the input vector contains features from 7 frames covering 60 ms of articulatory context (30 ms backward, 30 ms forward). DNNs were implemented using the Deepmat toolbox (Cho, 2013b).

Once a vector of MFCCs was predicted by either of the three mappings (DNN, sGMM, dGMM), we compute the least-squares estimate of the spectral envelope as $\hat{\boldsymbol{s}} = (\boldsymbol{F}^T\boldsymbol{F})^{-1}\boldsymbol{F}^T\boldsymbol{e}$, where $\boldsymbol{F}$ is the Mel Frequency Filter Bank (MFB) matrix used to extract MFCCs from the STRAIGHT spectrum, and $\boldsymbol{e}$ is the exponential of the inverse DCT of MFCCs. In a final step, we used the STRAIGHT synthesis engine to generate the waveform using the estimated spectral envelope $\hat{\boldsymbol{s}}$, signal aperiodicity and pitch. The overall process is summarized in Fig. 3.

Following Toda et al. (2004), we evaluated the forward mappings based on the Mel-Cepstral distortion between ground-truth and estimated acoustic features:

$$MCD = \frac{10}{\ln 10}\sqrt{2\sum_{d=1}^{24}(\boldsymbol{y}_t^{(d)} - \hat{\boldsymbol{y}}_t^{(d)})^2} \qquad (10)$$

where $\boldsymbol{y}_t^{(d)}$ is the dth component of the estimated acoustic feature vector (i.e., MFCC) at the t-th frame in a test utterance, and $\hat{\boldsymbol{y}}_t^{(d)}$ is the ground-truth value extracted from the acoustic recording. MCDs were calculated only on non-silent frames.

[4] As reported in (Toda et al., 2008), we also found 128 mixture components to be optimum in our preliminary experiment.

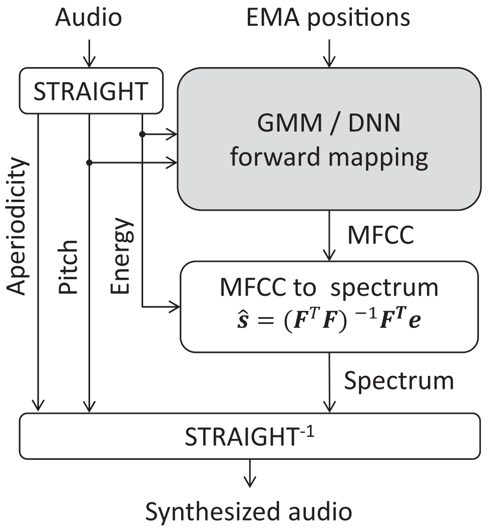


Fig. 3. Signal processing flow during articulatory synthesis.

5. Results

Acoustic predictions for the three forward mappings on a typical test utterance are illustrated in Fig. 4 alongside the ground truth. Because of the large number of inputs and outputs, we have only included trajectories for three articulatory coordinates (TT_x, TT_y, and LL_y) and two acoustic features ($MFCC_2$ and $MFCC_5$). Predictions from the sGMM display a number of unnatural transitions or glitches (see arrows in the figure), which are perceptible and have a detrimental effect on synthesis quality. Although the dGMM avoids such unnatural transitions by accounting for the dynamics of acoustic features in the trajectory optimization stage, it suffers from over-smoothing[5] effects, which are also perceptible and also clearly seen for $MFCC_2$ in the figure. By comparison, predictions from the DNN follow the target trajectory closely without introducing discontinuities in the derivative or over-smoothing.

We evaluated the forward mappings through a series of objective and subjective tests. In a first experiment, we compared the DNN against the two GMM mappings (sGMM, dGMM) in terms of their mapping accuracy (Mel-Cepstral distortion). Next, we evaluated the effect of tapped-delay length (experiment 2) and network depth (experiment 3) on Mel-Cepstral distortion, followed by a comparison of synthesis-time (experiment 4). In a final experiment, we compared the best performing DNN and GMM through a perceptual listening test.

5.1. Experiment 1: comparison of DNN vs. GMM

In the first experiment, we compared the accuracy of the DNN forward mapping against the two reference GMM methods. The DNN had a tapped-delay line with 2 delay units (a context window size of 20 ms) and two hidden layers of 512 units each. This simple architecture was selected to keep the number of model parameters comparable to that of the GMMs.

Fig. 5a summarizes the average MCDs of the three methods. The dGMM and DNN models achieve lower Mel-Cepstral distortion than the sGMM mapping. This is consistent with findings from previous studies (Toda et al., 2004; Nakamura et al., 2006), and shows that exploiting temporal information (as done by the dGMM and DNN) provides higher accuracy than a frame-by-frame mapping (sGMM). More importantly, the DNN reduces Mel-Cepstral distortion by 6% compared to the dGMM ($p < 0.001$, pairwise t-test), indicating that comparable (if not better) accuracy can be achieved at a fraction of the synthesis time required by the dGMM.

[5] A method known as global variance (Toda et al., 2007) has been suggested as a solution to the over-smoothing problem in the dGMM. However, the global variance method also increases prediction errors, so was not considered in this study as it would distort results from the objective tests.

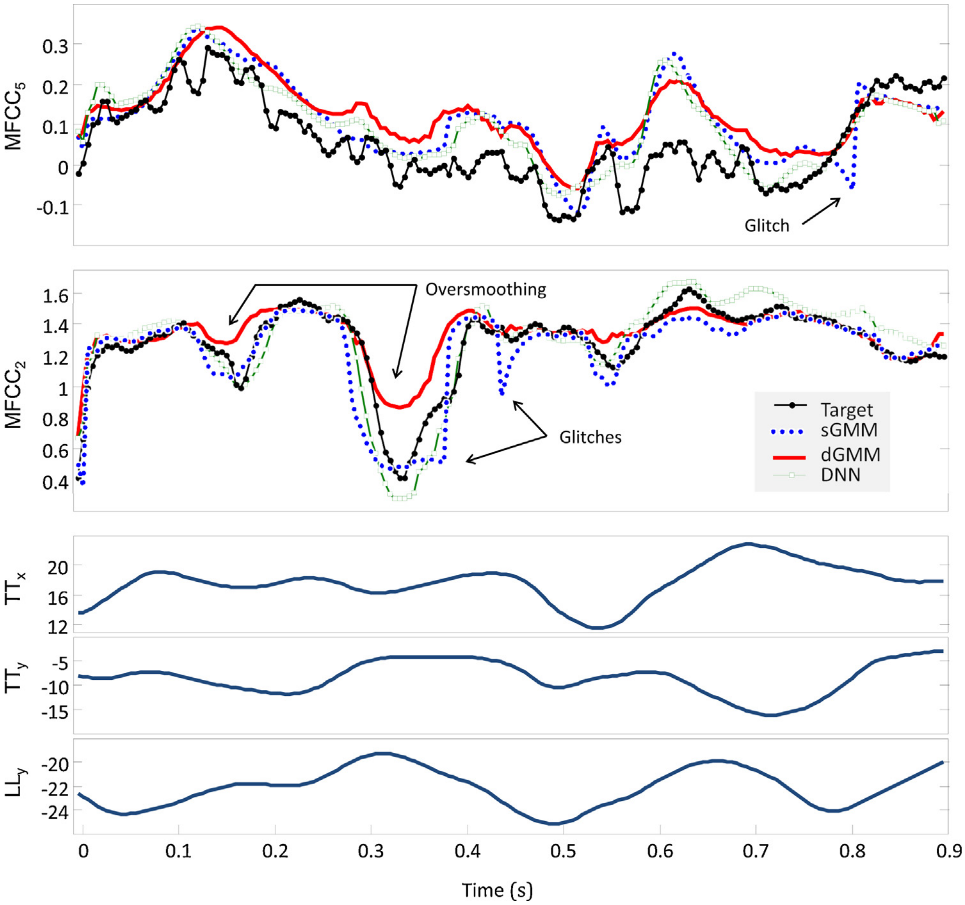


Fig. 4. Trajectories of selected acoustic and articulatory features from a typical test utterance. The top plots shows the second and fifth MFCCs predicted by the DNN, sGMM and dGMM alongside the target trajectory extracted from the audio recording of the same sentence. The bottom plots show the trajectories of a few articulatory input features for the same utterance. TT_x: anteroposterior position of the tongue tip, TT_y: height of the tongue tip, LL_y: height of the lower lip.

5.2. Experiment 2: context length

In the second experiment, we trained DNNs with tapped-delay line lengths of 0, 2, 4, 6 and 8 units, corresponding to temporal window sizes of 0, 20, 40, 60 and 80 ms, respectively. In each of these DNNs, we kept the same number of hidden layers and hidden units used in the first experiment. Fig. 5b summarizes results in terms of the Mel-Cepstral distortion, including that of the dGMM as a reference. Regardless of context length, the DNNs result in lower Mel-Cepstral distortion than the dGMM, the difference being statistically significant except for a context window size of 0 ms (i.e., a frame-by-frame mapping). More importantly, the Mel-Cepstral distortion decreases as the context window size increases, reaching a minimum with a 60 ms context window – a 9.8% reduction compared to the dGMM.

As part of this experiment we also sought to answer whether the same improvements in performance could be achieved by a GMM with a tapped-delay line. For this purpose we trained four GMMs with tapped-delay lines of 0, 20, 40 and 60 ms, respectively. Results are shown in Fig. 5b; GMM mappings had higher Mel-Cepstral distortion than the corresponding DNN regardless of context window size. More importantly, whereas the DNN is able to take advantage of the added information in the tapped-delay line (up to 60 ms), the GMM accuracy decreases markedly for context window sizes larger than 20 ms. This result may be explained by the fact that the tapped-delay features tend to be highly correlated, which may lead to near-singular covariance matrices in the GMM.

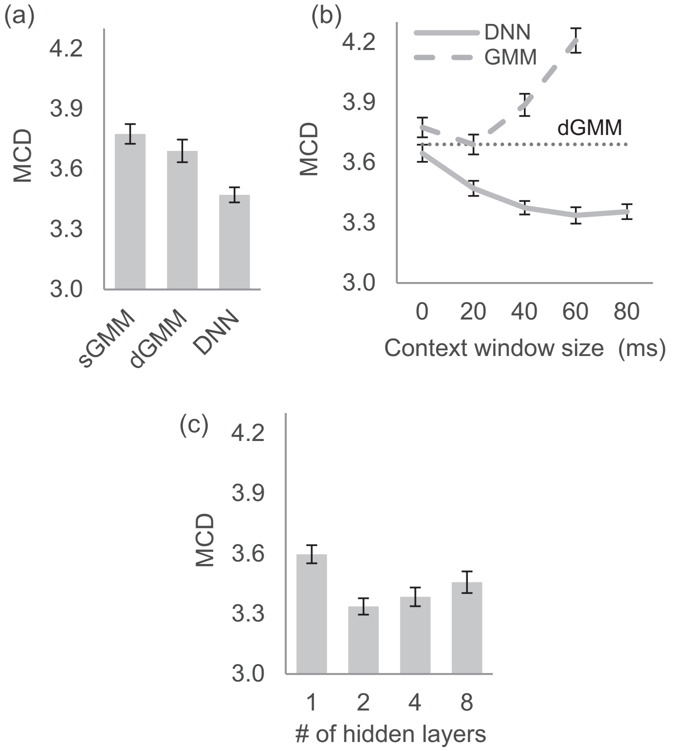


Fig. 5. (a) Experiment 1: Mel cepstral distortion (MCD) for the DNN, sGMM and dGMM mappings. (b) Experiment 2: MCD for the DNN and GMM as a function of the input articulatory context window. (c) Experiment 3: MCD for the DNN as a function of the number of hidden layers; error bars denote standard errors of means.

5.3. *Experiment 3: network depth*

In the third experiment, we sought to determine whether the complexity of the forward mapping justifies the use of a DNN; a DNN can model complex nonlinear functions with fewer parameters than a single-hidden MLP, but requires considerably longer training times. To answer this question, we trained four models: a single-layer MLP with 1024 hidden nodes, and three DNNs with 2, 4 and 8 hidden layers; the numbers of hidden units per layer in the DNN were adjusted so that the total number of hidden units remained constant across models (i.e. 1024). The tapped-delay line was fixed to 60 ms, the optimal context length found in the previous experiment. The MLP was trained using standard back-propagation (Rumelhart et al., 1986).

Fig. 5c summarizes the average Mel-Cepstral distortion for the four architectures; the three DNNs outperformed the MLP (pairwise t-test $p \ll 0.01$), which suggests that a single-layer network is insufficient to model the articulatory-to-acoustic mapping. The minimum Mel-Cepstral distortion – a 7% reduction compared to a single-layer MLP, was obtained for a DNN with 2 hidden layers.

5.4. *Experiment 4: synthesis time*

In the fourth experiment, we compared the synthesis time of the DNN and dGMM mappings. Both models were run on a Windows 7 Enterprise machine with an Intel Core i7-2600@3.4 GHz processor; models were implemented and run under Matlab v.7.14.

On average, the dGMM method required 39 s of synthesis time for each second of speech, rendering it unsuited for real-time synthesis (results not shown). In the case of the DNN, synthesis time depended on the network size, but increased linearly with the number of connections in the network. Fig. 6(a) shows the relationship between Mel-Cepstral distortion and synthesis time for five DNN structures, three from the third experiment (2 × 512, 4 × 256 and 8 × 128 hidden units, 60 ms context) and two relatively larger networks (3 × 512 and 4 × 512 hidden units) trained

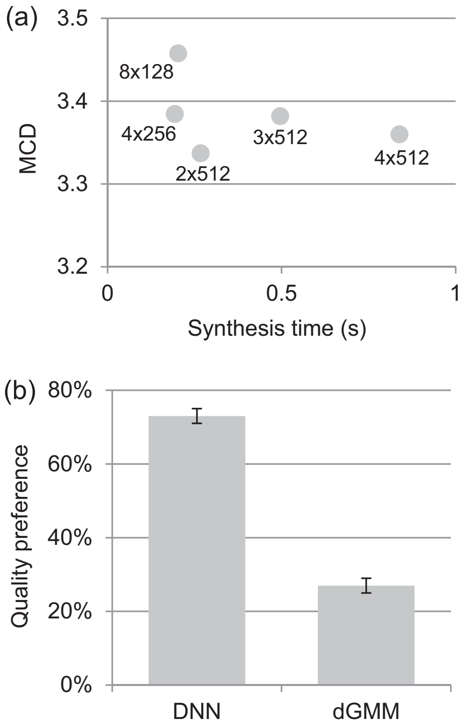


Fig. 6. (a) Experiment 4: Synthesis time of a DNN mapping increases with the size of the network. (b) Experiment 5: Pairwise comparison between DNN and dGMM synthesis; error bars denote standard errors of means.

specifically for this experiment. The largest among them, a DNN with 4 layers of 512 hidden units, required 838 ms for each second of speech, suitable for real-time synthesis. Smaller networks are even more efficient: a DNN with 2 layers of 512 hidden units required only 267 ms for each second of speech, and achieved the lowest Mel-Cepstral distortion.

5.5. Experiment 5: subjective assessment

In the final experiment, we evaluated the best-performing DNN (2×512 hidden units and 60 ms context window) against the conventional dGMM of Toda et al. (2004) through a listening test. Our goal was to determine whether the improvement in Mel-Cepstral distortion achieved by the DNN (a reduction of 9.8%) was perceptually significant.

Following our previous studies (Felps et al., 2012; Aryal and Gutierrez-Osuna, 2013, 2014), we recruited participants through Mechanical Turk, Amazon's online crowdsourcing tool. Participants listened to pairs of synthesis of the same sentence (one from the DNN, another from the dGMM) and were asked to select the utterance with the best quality in terms of naturalness, distortion, and intelligibility. 30 listeners participated in this test, each participant rating 30 pairs of utterances. Order of presentation within a pair (DNN vs. dGMM) was randomized to avoid order bias. Shown in Fig. 6(b), DNN syntheses were rated as more natural than dGMM syntheses in 73% of the cases, which is significantly higher than 50% chance level (pairwise t-test, $p \ll 0.001$). This result corroborates the objective comparisons, and indicates that the DNN mapping can synthesize utterances of higher perceptual quality than the conventional dGMM.

6. Discussion

We have presented a real-time articulatory synthesis method that exploits dynamic information in the articulatory trajectories to increase the accuracy of the forward mapping. Namely, our approach uses a tapped-delay line to concatenate articulatory feature vectors (EMA positions) from nearby frames, and a DNN to map the concatenated articulatory input vector into the corresponding acoustic observations (MFCCs). We compared the DNN against two GMM-based articulatory synthesizers, one that performs a frame-by-frame mapping (sGMM) and one that also incorporates speech dynamics (dGMM) as proposed by Toda et al. (2004). As our results show, the DNN is able to take advantage of the additional temporal information in the articulatory input features while keeping synthesis time below frame rate, surpassing the accuracy of both GMM-based methods through objective evaluations (Mel Cepstral distortion) and the subjective quality of the dGMM through listening tests.

Though GMMs are easier to train than DNNs, our results show they are unable to exploit the added temporal information via a tapped-delay line. This is partly due to the fact that the number of model parameters in a GMM increases quadratically with the number of input features, which can lead to over-fitting given the limited amount of training data. More importantly, tapped-delay features are likely to be correlated since they are time-delayed versions of the same signal, which may lead to near-singular covariance matrices in the GMMs. Though linear dimensionality reduction techniques (e.g., principal components analysis) may be used to decorrelate the input features, research in speech recognition (Bao et al., 2012) indicates that such techniques cannot compete with the capabilities of DNNs.

The dGMM and DNN articulatory synthesizers represent two distinct alternatives to incorporate speech dynamics. dGMMs can be trained relatively fast, but have long synthesis times due to the trajectory optimization post-processing stage; in our experiments, each second of speech required an average of 39 s of synthesis time on a contemporary desktop computer. By contrast, training a DNN is time consuming, but this is usually a one-time process that can be done offline. Once trained, the DNN has a short synthesis time[6] (e.g., 267 ms for our best-performing DNN). This makes the DNN ideally suited for other real-time applications of articulatory synthesis such as silent speech interfaces (Denby et al., 2010).

A few low-delay implementations of GMM mappings may be suitable for estimating maximum likelihood trajectory of spectral parameters in real time (Muramatsu et al., 2008; Toda et al., 2012). These studies, however, report tradeoffs between speed and mapping accuracy. Although these methods are yet to be evaluated in articulatory-acoustic mappings, the existence of these tradeoffs suggests that low-delay GMM mappings may result in lower accuracy than the DNN mapping. In agreement with several prior studies on DBM pre-training (Salakhutdinov and Hinton, 2009; Zhang et al., 2012; You et al., 2013; Hu et al., 2014), we found that performance degrades when the numbers of hidden layers are increased beyond two. These observations may at first suggest that DBMs (unlike DBNs) are not appropriate for training networks deeper than two hidden layers but there is evidence to the contrary. As an example, Cho (2013a) showed that a deep network with four hidden layer significantly outperforms networks with two hidden layers when pre-trained as a DBM. It may be possible that the flexibility offered by two-layer DBM-based networks is sufficient to model the complexity of articulatory-acoustic mappings. DBMs allow uncertainty to flow in bottom-up and top-down directions, which may lead to more efficient use of hidden layers than DBNs. This possibility is also supported by findings in a phone recognition task, where a two layer DBM-based network outperformed a seven layer DBN-based network, the best performing configuration among DBNs (You et al., 2013).

The acoustic quality of the articulatory synthesizers is limited by the fact that EMA only provides partial information about articulatory positions (6 pellets in our case). In particular, inner structures such as the velum, posterior part of the tongue or pharynx are difficult to measure with EMA, in part due to the natural gag reflex. Additional articulatory information may be obtained from the phonetic transcription if one is available, as we did in our study to generate a binary feature for nasality. A promising direction to improve articulatory synthesis quality is provided by the recent availability of rtMRI speech corpora (Narayanan et al., 2011), which can capture the full tongue contour, pharynx and larynx, and as well as other lingual, labial and jaw motions. DNNs are particularly well-suited in this case since the higher input dimensionality afforded by rtMRI (and the possible collinearity among features) may pose problems for GMM-based articulatory synthesizers.

Our immediate goal is to apply the DNN to the problem of accent conversion. This can be achieved by building a DNN model for the non-native speaker, then driving the model with articulatory trajectories from a native speaker; see (Felps et al., 2012; Aryal and Gutierrez-Osuna, 2014). A critical step in this process is bringing the two articulatory spaces into registration; several techniques have been proposed for this purpose, including z-score normalization (Toth and Black, 2005), global Procrustes transforms (Geng and Mooshammer, 2009) as well as pellet-specific transforms (Felps et al., 2014). A distinct advantage of a DNN in this regard is the possibility to perform the unsupervised training phase on articulatory data from multiple speakers with different articulatory styles, followed by a final supervised tuning on a small articulatory-acoustic corpus from the non-native speaker. Training the articulatory synthesizer on multiple

[6] Although the DNN uses a tapped-delay line that extends 30 ms into the future, this latency time (<200 ms) is considered acceptable for real-time communication (ITU-T, 2003).

articulatory styles would be especially beneficial for accent conversion, where the synthesis is driven by articulators different from those of the non-native speaker. Such strategies have been found beneficial in speech recognition tasks (Hinton et al., 2012), but have not been possible in articulatory accent conversion due to the scarcity of joint acoustic-articulatory data from non-native speakers. As an example, the MOCHA and X-ray Microbeam corpora only contain data for native speakers (Westbury, 1994; Wrench, 2000). An interesting new resource in this regard is the Marquette University Electromagnetic Articulography Mandarin Accented English (EMA-MAE), which contains a large EMA corpus from multiple Mandarin second-language speakers of American English (Ji et al., 2014). This new resource makes it possible to validate our articulatory synthesis and accent conversion methods across multiple speakers.

Acknowledgements

This work was supported by NSF award 0713205. We are grateful to Prof. Steve Renals and the Scottish Informatics and Computer Science Alliance (SICSA) for their support during RGO's sabbatical stay at CSTR (University of Edinburgh), and to Dr. Christian Geng for his assistance in performing the EMA recordings. We are also grateful to Prof. Kawahara for permission to use the STRAIGHT analysis–synthesis method. We would also like to thank anonymous reviewers whose suggestions and comments helped us improve the quality of the paper.

Appendix A. Supplementary data

Supplementary data associated with this article can be found, in the online version, at http://dx.doi.org/10.1016/j.csl.2015.02.003.

References

Andrew, G., Arora, R., Bilmes, J., Livescu, K., 2013. Deep canonical correlation analysis. In: Proceedings of ICML, pp. 1247–1255.

Arora, R., Livescu, K., 2013. Multi-view CCA-based acoustic features for phonetic recognition across speakers and domains. In: Proceedings of ICASSP, pp. 7135–7139.

Aryal, S., Gutierrez-Osuna, R., 2013. Articulatory inversion and synthesis: towards articulatory-based modification of speech. In: Proceedings of ICASSP, pp. 7952–7956.

Aryal, S., Gutierrez-Osuna, R., 2014. Accent conversion through cross-speaker articulatory synthesis. In: Proceedings of ICASSP, pp. 7744–7748.

Bao, Y., Jiang, H., Liu, C., Hu, Y., Dai, L., 2012. Investigation on dimensionality reduction of concatenated features with deep neural network for LVCSR systems. In: Proceedings of ICSP, pp. 562–566.

Birkholz, P., Jackèl, D., Kroger, B., 2006. Construction and control of a three-dimensional vocal tract model. In: Proceedings of ICASSP, pp. 873–876.

Browman, C.P., Goldstein, L., Kelso, J.A.S., Rubin, P., Saltzman, E., 1984. Articulatory synthesis from underlying dynamics. J. Acoust. Soc. Am. 75, S22–S23.

Cho, K., 2013a. Boltzmann machines and denoising autoencoders for image denoising. arXiv:1301.3468.

Cho, K.H., 2013b. Matlab code for restricted/deep Boltzmann machines and autoencoders. https://github.com/kyunghyuncho/deepmat.

Cho, K.H., Raiko, T., Ilin, A., 2013. Gaussian–Bernoulli deep Boltzmann machine. In: Proceedings of IJCNN, pp. 1–7.

Denby, B., Schultz, T., Honda, K., Hueber, T., Gilbert, J.M., Brumberg, J.S., 2010. Silent speech interfaces. Speech Commun. 52, 270–287.

Erhan, D., Bengio, Y., Courville, A., Manzagol, P.-A., Vincent, P., Bengio, S., 2010. Why does unsupervised pre-training help deep learning? J. Mach. Learn. Res. 11, 625–660.

Felps, D., Aryal, S., Gutierrez-Osuna, R., 2014. Normalization of articulatory data through Procrustes transformations and analysis-by-synthesis. In: Proceedings of ICASSP, pp. 3051–3055.

Felps, D., Geng, C., Gutierrez-Osuna, R., 2012. Foreign accent conversion through concatenative synthesis in the articulatory domain. IEEE Trans. Audio Speech Lang. Process. 20, 2301–2312.

Geng, C., Mooshammer, C., 2009. How to stretch and shrink vowel systems: results from a vowel normalization procedure. J. Acoust. Soc. Am. 125, 3278.

Ghosh, P.K., Narayanan, S.S., 2011. A subject-independent acoustic-to-articulatory inversion. In: Proceedings of ICASSP, pp. 4624–4627.

Hermansky, H., Broad, D.J., 1989. The effective second formant F2′ and the vocal tract front-cavity. Proceedings of ICASSP, 480–483.

Hinton, G., Deng, L., Yu, D., Dahl, G.E., Mohamed, A.R., Jaitly, N., Senior, A., Vanhoucke, V., Nguyen, P., Sainath, T.N., 2012. Deep neural networks for acoustic modeling in speech recognition: the shared views of four research groups. IEEE Signal Process. Mag. 29, 82–97.

Hinton, G.E., 2012. A practical guide to training restricted Boltzmann machines. In: Neural Networks: Tricks of the Trade. Springer, pp. 599–619.

Hiroya, S., Honda, M., 2004. Estimation of articulatory movements from speech acoustics using an HMM-based speech production model. IEEE Trans. Speech Audio Process. 12, 175–185.

Hu, A., Li, H., Zhang, F., Zhang, W., 2014. Deep Boltzmann machines based vehicle recognition. In: Proceedings of CCDC, pp. 3033–3038.

ITU-T, 2003. Recommendation G.114: One-way transmission time.

Ji, A., Berry, J., Johnson, M.T., 2014. The electromagnetic articulography mandarin accented english (EMA-MAE) corpus of acoustic and 3D articulatory kinematic data. In: Proceedings of ICASSP, pp. 7769–7773.

Kaburagi, T., Honda, M., 1998. Determination of the vocal tract spectrum from the articulatory movements based on the search of an articulatory-acoustic database. In: Proceedings of ICSLP, pp. 433–436.

Kawahara, H., 1997. Speech representation and transformation using adaptive interpolation of weighted spectrum: vocoder revisited. In: Proceedings of ICASSP, pp. 1303–1306.

Kello, C.T., Plaut, D.C., 2004. A neural network model of the articulatory-acoustic forward mapping trained on recordings of articulatory parameters. J. Acoust. Soc. Am. 116, 2354–2364.

Maeda, S., 1990. Compensatory articulation during speech: evidence from the analysis and synthesis of vocal tract shapes using an articulatory model. In: Hardcastle, W., Marchal, A. (Eds.), Speech Production and Speech Modelling. Kluwer Academic Publisher, Amsterdam, pp. 131–149.

Mermelstein, P., 1973. Articulatory model for the study of speech production. J. Acoust. Soc. Am. 53, 1070–1082.

Muramatsu, T., Ohtani, Y., Toda, T., Saruwatari, H., Shikano, K., 2008. Low-delay voice conversion based on maximum likelihood estimation of spectral parameter trajectory. In: Proceedings of INTERSPEECH, pp. 1076–1079.

Nabney, I.T., 2002. NETLAB: Algorithms for Pattern Recognition. Springer.

Nakamura, K., Toda, T., Nankaku, Y., Tokuda, K., 2006. On the use of phonetic information for mapping from articulatory movements to vocal tract spectrum. In: Proceedings of ICASSP, I-I.

Nakashika, T., Takashima, R., Takiguchi, T., Ariki, Y., 2013. Voice conversion in high-order eigen space using deep belief nets. In: Proceedings of INTERSPEECH, pp. 369–372.

Narayanan, S., Bresch, E., Ghosh, P.K., Goldstein, L., Katsamanis, A., Kim, Y., Lammert, A.C., Proctor, M.I., Ramanarayanan, V., Zhu, Y., 2011. A multimodal real-time MRI articulatory corpus for speech research. In: Proceedings of INTERSPEECH, pp. 837–840.

Özbek, I.Y., Hasegawa-Johnson, M., Demirekler, M., 2009. Formant trajectories for acoustic-to-articulatory inversion. In: Proceedings of INTERSPEECH, pp. 2807–2810.

Prabhavalkar, R., Fosler-Lussier, E., Livescu, K., 2011. A factored conditional random field model for articulatory feature forced transcription. In: Proceedings of ASRU, pp. 77–82.

Qin, C., Carreira-Perpinán, M.A., 2007. An empirical investigation of the nonuniqueness in the acoustic-to-articulatory mapping. In: Proceedings of INTERSPEECH, pp. 2300–2303.

Richmond, K., King, S., Taylor, P., 2003. Modelling the uncertainty in recovering articulation from acoustics. Comput. Speech Lang. 17, 153–172.

Rudzicz, F., 2010. Adaptive kernel canonical correlation analysis for estimation of task dynamics from acoustics. In: Proceedings of ICASSP, pp. 4198–4201.

Rumelhart, D., Hinton, G., Williams, R., 1986. Learning representations by back-propagating errors. Nature 323, 533–536.

Salakhutdinov, R., Hinton, G.E., 2009. Deep Boltzmann machines. In: Proceedings of AISTATS, Florida, USA, pp. 448–455.

Toda, T., Black, A.W., Tokuda, K., 2004. Mapping from articulatory movements to vocal tract spectrum with Gaussian mixture model for articulatory speech synthesis. In: Proceedings of ISCA, p. SSW5.

Toda, T., Black, A.W., Tokuda, K., 2007. Voice conversion based on maximum-likelihood estimation of spectral parameter trajectory. IEEE Trans. Audio Speech Lang. Process. 15, 2222–2235.

Toda, T., Black, A.W., Tokuda, K., 2008. Statistical mapping between articulatory movements and acoustic spectrum using a Gaussian mixture model. Speech Commun. 50, 215–227.

Toda, T., Muramatsu, T., Banno, H., 2012. Implementation of computationally efficient real-time voice conversion. Proceedings of INTERSPEECH, 94–97.

Toth, A.R., Black, A.W., 2005. Cross-speaker articulatory position data for phonetic feature prediction. Proceedings of INTERSPEECH, 2973–2976.

Uria, B., Murray, I., Renals, S., Richmond, K., 2012. Deep architectures for articulatory inversion. In: Proceedings of INTERSPEECH, pp. 867–870.

Westbury, J.R., 1994. X-ray Microbeam Speech Production Database User's Handbook Version 1.0. Waisman Center on Mental Retardation & Human Development, University of Wisconsin, Madison, WI.

Wrench, A.A., 2000. A multichannel articulatory database and its application for automatic speech recognition. In: Prodeedings of 5th Seminar of Speech Production, pp. 305–308.

Xu, Y., Du, J., Dai, L.-R., Lee, C.-H., 2014. An experimental study on speech enhancement based on deep neural networks. Signal Process. Lett. IEEE 21, 65–68.

You, Z., Wang, X., Xu, B., 2013. Investigation of deep Boltzmann machines for phone recognition. In: Proceedings of ICASSP, pp. 7600–7603.

Zen, H., Senior, A., Schuster, M., 2013. Statistical parametric speech synthesis using deep neural networks. In: Proceedings of ICASSP, pp. 7962–7966.

Zhang, Y., Salakhutdinov, R., Chang, H.-A., Glass, J., 2012. Resource configurable spoken query detection using deep Boltzmann machines. In: Proceedings of ICASSP, pp. 5161–5164.