# Robust Articulatory Speech Synthesis using Deep Neural Networks for BCI Applications

*Florent Bocquelet[1,2,3,4], Thomas Hueber[4], Laurent Girin[4], Pierre Badin[4], Blaise Yvert[1,2,3]*

[1] Inserm, Clinatec, U1167, Grenoble France
[2] Univ. Grenoble, Clinatec, U1167, Grenoble, France
[3] CEA, LETI, Clinatec, Grenoble, France
[4] GIPSA-Lab, UMR5216/CNRS/INP/UJF/U.Stendhal, Grenoble, France

florent.bocquelet@cea.fr

## Abstract

Brain-Computer Interfaces (BCIs) usually propose typing strategies to restore communication for paralyzed and aphasic people. A more natural way would be to use speech BCI directly controlling a speech synthesizer. Toward this goal, a prerequisite is the development a synthesizer that should i) produce intelligible speech, ii) run in real time, iii) depend on as few parameters as possible, and iv) be robust to error fluctuations on the control parameters. In this context, we describe here an articulatory-to-acoustic mapping approach based on deep neural network (DNN) trained on electromagnetic articulography (EMA) data recorded synchronously with produced speech sounds. On this corpus, the DNN-based model provided a speech synthesis quality (as assessed by automatic speech recognition and behavioral testing) comparable to a state-of-the-art Gaussian mixture model (GMM), yet showing higher robustness when noise was added to the EMA coordinates. Moreover, to envision BCI applications, this robustness was also assessed when the space covered by the 12 original articulatory parameters was reduced to 7 parameters using deep auto-encoders (DAE). Given that this method can be implemented in real time, DNN-based articulatory speech synthesis seems a good candidate for speech BCI applications.

**Index Terms**: articulatory speech synthesis, brain computer interface (BCI), deep neural networks, deep auto-encoder, EMA, noise robustness, dimensionality reduction

## 1. Introduction

In the past decades, Brain-Computer Interfaces (BCIs) have been studied to restore capabilities to people with severe paralysis, such as locked-in syndrome or tetraplegia. Several BCI studies succeeded in controlling the movement of effectors, such as robotic arms or computer mouse, both in animals and humans [1]–[4]. In the case of aphasia, current BCI approaches can provide ways to communicate, mostly through a typing process [5]. However, speech is our most natural way of communication. Restoring communication using BCIs could be thus applied to control a parametric speech synthesizer in real-time [6]. In such case, speech synthesis should be intelligible and performed in real-time. Moreover, BCI paradigms generally consider a restricted number of degrees of freedom, typically less than 7 [1]–[4]. Thus a speech synthesizer for BCI application should be controlled by a limited number of parameters. Finally, speech synthesis should be robust to input parameter fluctuations (i.e., uncontrolled fluctuations of brain activity during BCI).

Speech synthesis can be achieved in several ways, one consisting in predicting the acoustic speech signal from the position of vocal tract articulators (articulatory-to-acoustic mapping) [7]. Articulatory parameters vary more slowly than speech acoustic parameters and potentially rely on a low-dimension space [8], which are potential assets for BCI applications. Moreover, this strategy is compatible with decoding cortical signals from the somatotopically organized speech motor cortex [9]. The articulatory-to-acoustic mapping is useful in applications such as speech coding [10], speech synthesis [11], silent speech interfaces [12] and speech modification [13]. Regarding BCI applications, a previous study reported an electronic circuit implementation of the Maeda model of the vocal tract [14]. Here, we rather adopt a mathematical approach modeling the transformation of electro-magnetic articulography (EMA) recordings [15] into acoustic recordings, constructed from a French acoustic-articulatory database. A state-of-the-art method for articulatory-to-acoustic statistical mapping uses a Gaussian mixture model (GMM) to infer MEL-cepstral coefficients from 14 EMA signals [13]. EMA data has also been combined with electro-palatograph and laryngograph measurements (almost 200 parameters) to directly predict the spectrum of the audio signal using a single-layer neural network [16]. However, to date, there has been no evaluation of the performance of articulatory models in the case of noisy input signals, or when considering a reduction of the dimension of the space spanned by the articulatory parameters.

In the present work, we introduce a data-driven articulatory synthesizer based on Deep Neural Networks (DNN), which exhibits good robustness to noise and parameter reduction with respect to state-of-the-art methods (for instance a trajectory GMM), and is compatible with real-time implementation for future speech BCI applications.

## 2. Methods

### 2.1. The PB2007 acoustic-articulatory database

Articulatory data were recorded synchronously with audio signals using the Carstens 2D EMA system (AG200). Six coils were glued on the tongue tip, blade, and dorsum, as well as on the upper lip, the lower lip and the jaw. Sequences of articulatory features were low-pass filtered at 20 Hz and down-sampled from 200 Hz to 100 Hz. The recorded database consisted of two repetitions of the 16 French vowels, two repetitions of 224 Vowel-Consonant-Vowels sequences (VCVs), two repetitions of 109 pairs of Consonant-Vowel-Consonants (CVCs) real French words, and 117 sentences (total 20 minutes without silences). The speech signal was down-sampled to 16 kHz and parameterized by 20 mel-cepstrum coefficients using SPTK *mcep* tools [17] (25ms frame length, 10ms frame shift). In order to take into account

the dynamic constraints on acoustic parameters, we concatenated 3 consecutive acoustic frames in one single feature vector for the DNN-based mapping, and we used one frame and its derivative for the GMM-based mapping. Principal component decomposition (PCA) keeping the full variance was applied to the contextualized articulatory data to obtain linearly uncorrelated features.

### 2.2. Articulatory-to-acoustic mapping with a trajectory Gaussian mixture model (GMM)

We considered the trajectory GMM approach proposed by Toda et al. [13] as a reference for articulatory-to-acoustic mapping. In the training stage, articulatory-acoustic relationships are modeled by a GMM. In the mapping stage, the estimated acoustic sequences is defined by $\hat{\boldsymbol{y}} = \arg\max_{\boldsymbol{y}} P(\boldsymbol{Y}|\boldsymbol{x},\boldsymbol{\theta})$, with $\boldsymbol{Y} = [\boldsymbol{y}\ \Delta\boldsymbol{y}]$, $\boldsymbol{\theta}$ the set of parameters of the GMM, **x** the sequence of articulatory features, **y** the sequence of acoustic features, and **Δy** its derivative, which can be solved in closed form. In our implementation, the suboptimum sequence of mixture components indices $\hat{\mathbf{m}}$ defined as $\hat{\mathbf{m}} = [\hat{m}_1, \ldots, \hat{m}_T]$, with $\hat{m} = \arg\max_m P(m|\boldsymbol{x},\boldsymbol{\theta})$, is determined using Viterbi algorithm. We refer the reader to [13] for further theoretical aspects.

### 2.3. Articulatory-to-acoustic mapping with a Deep Neural Network (DNN)

This section briefly describes DNN-based mapping. A deep neural network is a multi-layer discriminative model, made up of units organized in layers. The bottom layer is a visible input layer $\mathbf{h^0}$, while the next $L$ layers are hidden layers $\mathbf{h^1}$, $\mathbf{h^2}$, ..., $\mathbf{h^L}$, and the last one is the output layer $\mathbf{h^{L+1}}$. Here we consider a fully-connected feed-forward DNN: Each unit of a layer is connected to each unit of the next layer, and there are no connections between units belonging to the same layer. Connections weights are learned during the training phase, generally using the back-propagation algorithm. In the following, $w_{i,j}^{(l)}$ denotes the weight of a connection from the neuron $i$ of layer $l$-$1$ to the neuron $j$ of layer $l$. Compared to simple neural networks, DNNs have more than one or two hidden layers.

Each unit $i$ of a layer, except the input layer, has an activation function $\sigma$ and a bias $b_i$. The output of a unit is the application of the activation function of this unit to its weighted inputs. The output of the unit $i$ of the layer $\mathbf{h^l}$ is given by the following equation:

$$o_i^{(l)} = \sigma\left(\sum_{j=1}^{n_{l-1}} w_{j,i}^{(l)} . o_j^{(l-1)}\right) \quad (1)$$

where $n_{l-1}$ is the number of units in the layer $\mathbf{h^{l-1}}$.

When a deep neural network is used for a regression problem, its input units take the value of the input data, and the network is trained such that the outputs of the output layer units fit the output data. More details can be found in [18].

DNN training is usually a complex task since large initial weights typically lead to poor local minima, while small initial weights lead to small gradients making the training infeasible with many hidden layers [19]. We trained our network using the usual back-propagation algorithm. However, we chose to add the different layers successively. During the first step, the network was only composed by the input layer $\mathbf{h^0}$, the first hidden layer $\mathbf{h^1}$, and the output layer $\mathbf{h^{L+1}}$. This initial network was randomly initialized then fine-tuned using back-propagation. Then the next layer was added so that the new network was now composed by the input layer $\mathbf{h^0}$, the first two hidden layers $\mathbf{h^1}$ and $\mathbf{h^2}$, and the output layer $\mathbf{h^{L+1}}$. The weights from the input layer $\mathbf{h^0}$ to the fist hidden layer $\mathbf{h^1}$ were those obtained at the previous step and the other weights were randomly initialized. Back-propagation was then applied to this network for fine-tuning. This process was repeated until all the hidden layers were added. The input and output data were z-scored before being fed to the network.

At each step the weights where randomly initialized using a Gaussian distribution with a 0.0001 standard deviation. The error criterion was the mean squared error (MSE) between predicted and expected values. The minimization of the error was done with the conjugate gradient method using a 3 lines search, on successive batches: at each epoch, the training data samples were randomly shuffled then divided into 100 blocs. Dividing into batches allows more efficient computation than when using single samples [20]. Non-linear units used the logistic sigmoid function as activation function. This training method led to good and fast convergence while classic back-propagation could not converge with more than 2 layers.

### 2.4. Artificial degradation of articulatory data

#### *2.4.1. Noisy data*

Because the use of a synthesizer for BCI application will imply noisy inputs reflecting uncontrolled fluctuations of brain activity, we tested the robustness of the articulatory-to-acoustic mapping by adding artificial noise to the test input articulatory data (no noise was added during the training step). We added white noise low-pass filtered below 20 Hz (as were the original EMA data) and re-centered. We tested different signal to noise ratio (SNR) values as defined by the ratio of the peak-to-peak amplitude of each articulatory signal by the standard deviation of the filtered noise. The noise amplitude was adjusted across EMA signals so that all had identical SNR. For the subjective evaluation (listening tests, see below), we only considered one SNR value (SNR = 10.0).

#### *2.4.2. Dimensionality reduction*

In practice, accurate real-time BCI control of effectors can only be expected with a few degrees of freedom, typically less than 7. Hence, we tested to which extend it is possible to reduce the number of articulatory parameters, starting from the 12 parameters of our EMA database, while preserving acceptable speech synthesis quality. We compared two main dimension reduction methods: the principal component analysis (PCA) and deep auto-encoders (DAE). Deep auto-encoders are deep neural networks trained to reproduce their input as their output [19]. Their architecture is symmetric with a "bottle-neck" linear middle layer containing fewer units than the input layer thus forcing the network to learn a dimensionality reduction of the input data. Such a network can be spliced into two sub-networks: the encoding network, which reduces its input data, and the decoding network, which allows recovering the data from the reduced one. For more details, we refer the reader to [19]. DAE training was done using the dimensionality reduction toolbox [21], [22]. We tested the performance of DNN- and GMM-based speech synthesis with all possible reduced dimensions, from 1 to 12, by feeding the originally trained models with reduced-then-recovered articulatory parameters. For the subjective

evaluation, we only considered 7 reduced parameters, which was the number of articulatory parameters retained in [7].

### 2.5. Model evaluation

#### *2.5.1. Generalities*

The acoustic-articulatory database was randomly shuffled and then divided into 5 partitions of equal size. A 5-fold cross-validation was employed for evaluation of each model: one partition was used for testing and the remaining 4 for training, and this was repeated 5 times to test all the partitions. The folds used to train the DNN and the GMM were identical. The reduction models (PCA, DAE) were computed for each fold, using only the training data. This allowed obtaining mean and standard deviation (SD) for each evaluation. Significant differences between results were assessed using the Quade test with Conover correction [23], using recognition accuracy by phone for the objective evaluation (35 scores per condition) and recognition accuracy by participants for the subjective evaluation (11 scores per condition).

#### *2.5.2. Objective evaluation using automatic speech recognition based on Hidden Markov Models (HMM)*

An objective evaluation was performed using an HMM-based phonetic decoder trained on the spectral data of the reference speaker using a standard training procedure of context-dependent triphone tied-state HMM [24]. The recognition accuracy (defined as $\boldsymbol{Acc\%} = 100 \cdot (N-D-S-I)/N$, where $N$ is the total number of phones in the test set, $S$, $D$ and $I$ are respectively the number of substitutions, deletions and insertions) was used as a measurement of the quality of the synthetic spectral trajectories at the phonetic level. This approach was preferred to the typical calculation of the mel-cepstral distortion between original and synthetic spectral trajectories (as used in [13]) since the obtained results were better correlated with the human perception of the synthetic speech.

#### *2.5.3. Subjective evaluation using behavioral testing*

Eleven subjects participated to an intelligibility test. All participants were French native speakers with no hearing impairment. The presented stimuli consisted of 10 French vowels /a/, /i/, /u/, /o/, /œ/, /e/, /y/, /ã/, /ɛ̃/, /õ/, and 30 vowel-consonant-vowel (VCV) pseudo words made of the 10 consonants /p/, /t/, /k/, /f/, /s/ /ʃ/, /m/, /n/, /r/, /l/, in /a/, /i/, /u/ contexts. The seven following synthesis conditions were tested: analysis-synthesis, GMM based synthesis with and without noise, DNN-based synthesis with and without noise, and DNN-based synthesis with and without reduced parameters. The mel-cepstrum coefficients obtained by each mapping were converted to audible sounds using the MLSA filter [25]. Excitation signal was designed with constant pitch for the vowels and null pitch for the VCVs. In total, each participant had to identify 360 sounds that were played in random order. Participants were seated in quiet environment and instructed that they would be listening to isolated vowels or VCV sequences. For each utterance, they had to pick the corresponding vowel in the case of an isolated vowel, or the middle consonant in the case of a VCV sequence. They were told that some of the sounds were noisy and difficult to identify, and thus to not evaluate the sound quality but only its intelligibility. If the sound was not intelligible at all, they could report it explicitly via a specific option. Stimuli were presented at identical sound levels, and subjects could replay them as many times as necessary. No performance feedback was provided during the test. The recognition accuracy was defined as $\boldsymbol{Acc\%}=R/N$ with R the number of correct answers for the N presented sounds of the test.

## 3. Results

### 3.1. Speech synthesis in absence of noise using DNN

#### *3.1.1. Influence of DNN hyper-parameters*

Objective evaluations were conducted for various DNN architectures, with different numbers of layers (1 to 4) and numbers of units per hidden layer (20, 50, or 100) identical across hidden layers. As shown in Figure 1, adding more units for a given layer increased recognition accuracy, while adding more layers first led to an increase before a stabilization or small degradation in accuracy. Overall, a good compromise was to use a DNN with 3 hidden layers of 100 units each, ensuring an accuracy of 71.13 ± 2.75%.

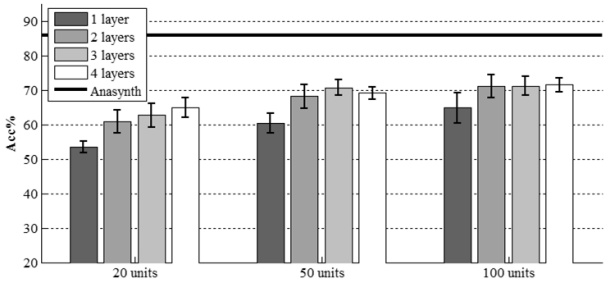


Figure 1: *Phone recognition accuracy according to the number of layers and the number of units per layer using DNN-based mapping (mean±SD). The thick line represents the recognition accuracy on original audio.*

#### *3.1.2. Comparison to GMM*

We also conducted objective evaluations of speech synthesis for various numbers of mixture components in the GMM, from 16 to 256. The recognition accuracy increased with increasing number of components, stabilizing after 128. Fitting 256 components with full covariance matrices however often led to ill-conditioned covariance matrices. Thus, we thereafter chose 128 components (ensuring an accuracy of 75.68 ± 1.27%), consistent with the results of [13].

We compared GMM-based and DNN-based synthesis using both the objective (HMM phonetic decoding) and the subjective (listening) tests. Consistent results were obtained, as shown in Figure 2. In the objective test, GMM recognition accuracy reached 75.68% and the DNN, 71.13%. In the subjective test, the GMM recognition accuracy was 66.59% and 69.77% for the DNN. Both GMM and DNN recognition accuracies were below the recognition accuracy on original audio ($P < 10^{-4}$ for the objective evaluation and $P < 0.002/0.01$ for the subjective one for GMM/DNN), which was 85.77% for the objective evaluation, and 87.95% for the subjective one. The GMM performed slightly better than the DNN in the objective evaluation ($P < 10^{-3}$) while no significant difference was observed between both models in the subjective evaluation ($P > 0.3$). Importantly, this similar accuracy was obtained with the DNN with nearly ten times less adjustable parameters than with the GMM: 274,560 parameters for the

GMM with 128 components and full covariance matrices compared to 25,920 for the 3 layers DNN.

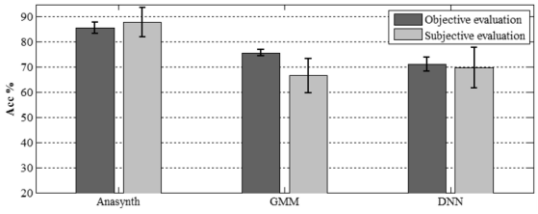


Figure 2: *Recognition accuracy (mean±SD) for objective and subjective evaluations of GMM-based and DNN-based mappings.*

### 3.2. Speech synthesis from reduced articulatory data

Let remind that both PCA and DAE were combined with the GMM-based and the DNN-based mappings to test successive dimension reduction from 12 to 1 articulatory parameter. For less than 10 reduced parameters, the use of DAE led to better results, both for the GMM- ($P < 10^{-4}$) and the DNN-based mappings ($P = 0.01$), while no significant difference was observed with 11 and 12 parameters (Figure 3). Using 7 or more DAE-reduced parameters allowed obtaining a recognition accuracy of above 60% both for the GMM- and the DNN-based mappings, while 9 or more parameters were needed to achieve the same accuracy when using PCA. Moreover, no significant difference was observed between GMM- and DNN-based mappings for less than 9 reduced parameters obtained by PCA ($P > 0.8$), while the GMM results were slightly better than the DNN for more than 3 reduced parameters obtained by DAE ($P = 0.01$).

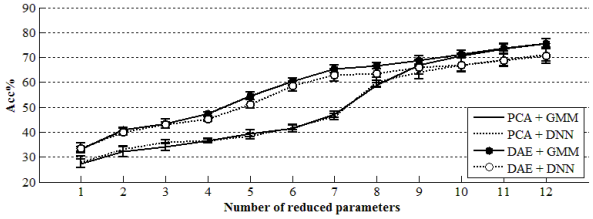


Figure 3: *Phone recognition accuracy (mean±SD) with reduced parameters obtained both by PCA and DAE, and with both GMM- and DNN-based mappings.*

### 3.3. Speech synthesis with noisy articulatory data

Both GMM- and DNN-based mappings were then objectively evaluated with noisy input data with different SNR (Figure 4).

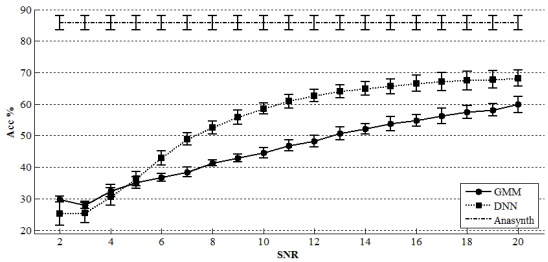


Figure 4: *Phone recognition accuracy (mean±SD) on noisy data as a function of SNR.*

With the GMM-based mapping, a recognition accuracy above 60% was reached with a SNR of more than 20, while the DNN-based mapping obtained more than 60% recognition accuracy with a SNR higher than 10. The DNN-based mapping generally obtained better recognition accuracy than the GMM-based mapping ($P < 10^{-4}$). A subjective test was then conducted for GMM- and DNN-based mapping of noisy EMA data (SNR=10, which corresponds to 44.54% and 58.59% of recognition accuracy for the GMM- and the DNN-based mapping respectively, in the objective test).

This test also included DNN-based mapping of DAE-reduced data (7 parameters) with and without noise addition (Figure 5). The GMM-based mapping obtained a recognition accuracy of 32.27%, while the DNN-based mapping obtained 59.32% with no parameters reduction, and 53.86% when using 7 DAE-reduced parameters. Consistently with the objective evaluation, the DNN-based mapping was found to perform better than the GMM-based mapping in noisy condition ($P<10^{-4}$). Moreover, the DNN-based mapping with reduced and noisy parameters performed better than the GMM-based mapping with full and noisy parameters ($P = 0.01$). Finally, no significant difference in subjective accuracy of the DNN-based mapping with reduced parameters was observed between clean and noisy conditions ($P > 0.2$).

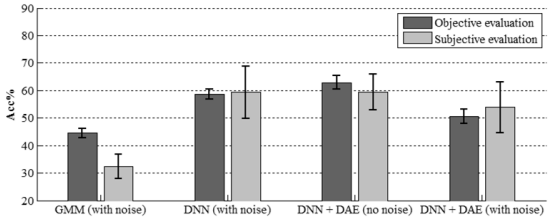


Figure 5: *Recognition accuracy (mean±SD) based on objective and subjective evaluations of GMM- and DNN-based mapping on noisy articulatory data (SNR = 10), and on reduced data (DAE with 7 reduced parameters) for the DNN-based mapping*

## 4. Conclusion

In this paper we have presented an articulatory-to-acoustic mapping method based on a deep neural network (DNN). We have proposed a training process to overcome the difficulties generally encountered when training deep neural networks, which allowed good convergence. The performance of this mapping method was then evaluated on clean and noisy articulatory data, with and without reducing the dimensionality of these input parameters. Results were compared to the state-of-the-art method which relies on a trajectory Gaussian mixture model (GMM). The two-models where objectively evaluated using a HMM-based speech recognition method, and subjectively evaluated with a listening test. Objective and subjective evaluations were consistent and pointed out that the DNN-based mapping was reaching a phone recognition accuracy of around 70% which is almost similar to the results obtained with the GMM-based mapping. It also showed that it was more robust to noise. We also studied the impact of reducing the dimension of the articulatory space on the speech synthesis quality, using either principal component analysis (PCA) or deep auto-encoders (DAEs). Results showed that DAEs were more appropriate than PCA, both for GMM- and DNN-based mappings. Finally, the DNN-based mapping has a very low computational cost once the network has been trained, and is thus compatible with real time applications such as BCI for speech rehabilitation.

# 5. References


[1] V. Gilja, P. Nuyujukian, C. a Chestek, J. P. Cunningham, B. M. Yu, J. M. Fan, M. M. Churchland, M. T. Kaufman, J. C. Kao, S. I. Ryu, and K. V Shenoy, "A high-performance neural prosthesis enabled by control algorithm design", *Nat. Neurosci.*, vol. 15, no. 12, pp. 1752–7, Dec. 2012.

[2] M. Velliste, S. Perel, M. C. Spalding, A. S. Whitford, and A. B. Schwartz, "Cortical control of a prosthetic arm for self-feeding", *Nature*, vol. 453, no. 7198, pp. 1098–101, Jun. 2008.

[3] J. L. Collinger, B. Wodlinger, J. E. Downey, W. Wang, E. C. Tyler-Kabara, D. J. Weber, A. J. C. McMorland, M. Velliste, M. L. Boninger, and A. B. Schwartz, "High-performance neuroprosthetic control by an individual with tetraplegia", *Lancet*, vol. 381, no. 9866, pp. 557–64, Feb. 2013.

[4] L. R. Hochberg, D. Bacher, B. Jarosiewicz, N. Y. Masse, J. D. Simeral, J. Vogel, S. Haddadin, J. Liu, S. S. Cash, P. van der Smagt, and J. P. Donoghue, "Reach and grasp by people with tetraplegia using a neurally controlled robotic arm", *Nature*, vol. 485, no. 7398, pp. 372–5, May 2012.

[5] J. S. Brumberg and F. H. Guenther, "Development of speech prostheses: current status and recent advances", *Expert Rev. Med. Devices*, vol. 7, no. 5, pp. 667–79, Sep. 2010.

[6] F. H. Guenther, J. S. Brumberg, E. J. Wright, A. Nieto-Castanon, J. a Tourville, M. Panko, R. Law, S. a Siebert, J. L. Bartels, D. S. Andreasen, P. Ehirim, H. Mao, and P. R. Kennedy, "A wireless brain-machine interface for real-time speech synthesis", *PLoS One*, vol. 4, no. 12, p. e8218, Jan. 2009.

[7] S. Maeda, "Compensatoy Articulation During Speech: Evidence from the analysis and Synthesis of Vocal-Tract shapes using an articulatory model", *Speech Prod. Speech Model.*, pp. 131–149, 1990.

[8] D. Beautemps, P. Badin, and G. Bailly, "Linear degrees of freedom in speech production: Analysis of cineradio- and labio-film data and articulatory-acoustic modeling", *J. Acoust. Soc. Am.*, vol. 109, no. 5, pp. 2165–2180, May 2001.

[9] K. E. Bouchard, N. Mesgarani, K. Johnson, and E. F. Chang, "Functional organization of human sensorimotor cortex for speech articulation", *Nature*, vol. 495, no. 7441, pp. 327–32, Mar. 2013.

[10] S. C. C. Silva, "Articulatory Analysis using a Codebook for Articulatory based Low BIt-Rate Speech Coding", in *The 5th International Conference on Spoken Language Processing*, 1998.

[11] W. K. Lo and P. C. Ching, "Phone-based speech synthesis with neural network and articulatory control", *Proceeding Fourth Int. Conf. Spok. Lang. Process. ICSLP '96*, vol. 4, pp. 2227–2230.

[12] T. Hueber, E. Benaroya, B. Denby, and G. Chollet, "Statistical Mapping Between Articulatory and Acoustic Data for an Ultrasound-Based Silent Speech Interface", *Proc. Interspeech*, no. August, pp. 593–596, 2011.

[13] 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, Mar. 2008.

[14] K. H. Wee, L. Turicc, and R. Sarpeshkar, "An Articulatory Speech-Prosthesis System", *2010 Int. Conf. Body Sens. Networks*, pp. 133–138, Jun. 2010.

[15] K. Richmond, P. Hoole, S. King, and I. Forum, "Announcing the Electromagnetic Articulography (Day 1) Subset of the mngu0 Articulatory Corpus", *Proc. Interspeech*, no. Day 1, pp. 1–4.

[16] C. T. Kello and D. C. Plaut, "A neural network model of the articulatory-acoustic forward mapping trained on recordings of articulatory parameters", *J. Acoust. Soc. Am.*, vol. 116, no. 4, p. 2354, 2004.

[17] K. Tokuda, K. Oura, A. Tamamori, S. Sako, H. Zen, T. Nose, T. Takahashi, J. Yamagishi, and Y. Nankaku, "Speech Signal Processing Toolkit (SPTK)", *http://sp-tk.sourceforge.net/*. .

[18] H. Larochelle, Y. Bengio, and P. Lamblin, "Exploring Strategies for Training Deep Neural Networks", *J. Mach. Learn. Res.*, vol. 1, pp. 1–40, 2009.

[19] G. E. Hinton and R. R. Salakhutdinov, "Reducing the dimensionality of data with neural networks", *Science*, vol. 313, no. 5786, pp. 504–7, Jul. 2006.

[20] G. Hinton, "A Practical Guide to Training Restricted Boltzmann Machines A Practical Guide to Training Restricted Boltzmann Machines", *Comput. Sci.*, vol. 7700, pp. 599–619.

[21] L. van der Maaten, "Matlab Toolbox for Dimensionality Reduction", *http://homepage.tudelft.nl/19j49/Matlab_Toolbox_for_Dimensionality_Reduction.html*. .

[22] L. J. P. Van Der Maaten, E. O. Postma, and H. J. Van Den Herik, "Dimensionality Reduction : A Comparative Review", *J. Mach. Learn. Res.*, no. January, 2008.

[23] D. Quade, "Using Weighted Rankings in the Analysis of Complete Blocks with Additive Block Effects", *J. Am. Stat. Assoc.*, vol. 74, no. 367, pp. 680–683, Sep. 1979.

[24] M. Gales and S. Young, "The Application of Hidden Markov Models in Speech Recognition", *Found. Trends® Signal Process.*, vol. 1, no. 3, pp. 195–304, 2007.

[25] S. IMAI, "Cepstral analysis synthesis on the mel frequency scale", *Acoust. Speech, Signal Process. IEEE Int. Conf. ICASSP*, pp. 93–96, 1983.