A SPARSE SMOOTHING APPROACH FOR GAUSSIAN MIXTURE MODEL BASED ACOUSTIC-TO-ARTICULATORY INVERSION

Prasad Sudhakar[1], Laurent Jacques[1], Prasanta Kumar Ghosh[2]

[1]ICTEAM/ELEN, Universite catholique de Louvain, Belgium
[2]Department of Electrical Engineering, Indian Institute of Science (IISc), Bangalore, India.
{prasad.sudhakar, laurent.jacques}@uclouvain.be, prasantg@ee.iisc.ernet.in

ABSTRACT

It is well-known that the performance of the Gaussian Mixture Model (GMM) based Acoustic-to-Articulatory Inversion (AAI) improves by either incorporating smoothness constraint directly in the inversion criterion or smoothing (low-pass filtering) estimated articulator trajectories in a post-processing step, where smoothing is performed independently of the inversion. As the low-pass filtering is independent of inversion, the smoothed articulator trajectory samples no longer remain optimal as per the inversion criterion. In this work, we propose a sparse smoothing technique which constrains the smoothed articulator trajectory to be different from the estimated trajectory only at a sparse subset of samples while simultaneously achieving the required degree of smoothness. Inversion experiments on the articulatory database show that the sparse smoothing achieves an AAI performance similar to that using low-pass filtering but in sparse smoothing $\sim$15% (on average) of the samples in the smoothed articulator trajectory remain identical to those in the estimated articulator trajectory thereby preserve their AAI optimality as opposed to 0% in low-pass filtering.

Index Terms— acoustic-to-articulatory inversion, smoothing, Gaussian mixture model, sparsity, chambolle-pock, ℓ_1 minimization

1. INTRODUCTION

Acoustic-to-articulatory inversion is the task of estimating articulatory representation from an acoustic representation. A number of acoustic as well as articulatory representations could be used for this task. In this work we use Mel Frequency Cepstral Coefficients (MFCCs) as the acoustic representation. Similarly, for articulatory representation we use articulatory movement data from Electromagnetic Articulography (EMA). Depending on the representations in the acoustic and articulatory domains, the acoustic-articulatory map can be learnt in a number of ways – statistical models such as Gaussian mixture model [1], mixture density network (MDN) [2], trajectory hidden Markov model (HMM) [3] and generalized smoothness criterion (GSC) [4], codebook approach [5] and Neural Network approaches [6]; a comprehensive summary of different mapping techniques can be found in [7]. In this work we adopt the statistical approach for modeling the map between acoustic and articulatory spaces. In particular, we use the Gaussian mixture model (GMM) for modeling the statistical mapping [1].

PS is supported by the DETROIT project (WIST3), no. 1017073, Walloon Region, Belgium. LJ is supported by the Belgian FRS-FNRS fund.

Work supported by Department of Science and Technology (DST), Govt. of India.

In GMM based AAI, the articulatory representation is estimated from the acoustic representation separately in each analysis frame using minimum mean squared error (MMSE) criterion [1]. Without any continuity constraint across frames, the articulator trajectories estimated using GMM based AAI turn out to be rough and jagged in nature unlike a realistic articulator trajectories which are slowly varying and low-pass in nature [4]. It has been shown that a smooth articulatory estimate could be obtained by incorporating smoothness constraint directly in the inversion criterion or by smoothing estimated articulator trajectories using low-pass filter in a post-processing step [1, 8]. Both the integrated and separate smoothing approaches result in statistically similar AAI performances [8]. In this work we focus on smoothing as a post-processing step.

It is important to note that the samples of the smoothed articulator trajectory obtained by low-pass filtering, in general, differ from the corresponding samples of the estimated articulator trajectory from GMM based AAI. Thus, the smoothed articulatory representations no longer remain optimal for AAI criterion. The articulator trajectories obtained from AAI are often used subsequently for speech recognition [9, 10, 11, 12] and speech synthesis [13, 14]. One could consider an AAI criterion in a task-specific manner such that the target task gets maximal benefit from the estimated articulatory features obtained by AAI; in such cases performing smoothing could be detrimental for the task if the smoothing does not preserve optimality for task-specific AAI criterion in the smoothed representation. In such scenarios, it is preferable to use a smoothing technique which also preserves the optimality for AAI criterion. In this work, we propose a smoothing approach for GMM based AAI which achieves the required degree of smoothness while at the same time preserves MMSE optimality for as many samples of the smoothed trajectory as possible. We refer to the proposed smoothing by *sparse smoothing* because only a sparse subset of samples in the estimated trajectory is altered to obtain the required smoothness as opposed to all samples in the case of low-pass filtering in typical smoothing.

We formulate the sparse smoothing as an optimization problem which reduces the high frequency components in the estimated articulator trajectory by changing a minimal set of its samples. The formulation is aimed at replacing the traditional 'smoothing through convolution' by 'smoothing through sparse addition' approach. Reduction of high frequency components results in a smooth and slowly varying articulator trajectory. We present a computational method for solving the sparse smoothing problem. Experiments on articulatory dataset demonstrate that the sparse smoothing achieves an inversion performance similar to that obtained by smoothing using low-pass filtering. However, the key characteristic of sparse smoothing is that it preserves the MMSE optimality for $\sim$15% of the samples as opposed to 0% for the low-pass filtering. Thus the

978-1-4799-2893-4/14/$31.00 ©2014 IEEE

proposed sparse smoothing yields realistic smooth articulator trajectories while simultaneously maintaining optimality of AAI criterion for as many frames as possible.

2. DATASET AND PRE-PROCESSING

In this paper, we use the Multichannel Articulatory (MOCHA) database [15] that contains speech and corresponding ElectroMagnetic Articulography (EMA) data from one male and one female talker of British English. The EMA data consist of dynamic positions of the EMA sensors in the midsagittal plane of the talker. Seven sensors are placed on upper lip (UL), lower lip (LL), lower incisor (LI), tongue tip (TT), tongue body (TB), tongue dorsum (TD), and velum (VEL)). Thus, we use 14 dimensional raw EMA features for representing articulatory space (i.e., X and Y co-ordinates of seven EMA sensors), namely ULx, LLx, LIx, TTx, TBx, TDx, VELx, ULy, LLy, LIy, TTy, TBy, TDy, VELy. The articulatory position data have high frequency noise resulting from the EMA measurement error. Also the mean position of the articulators changes between utterances; hence, the position data needs pre-processing before its use in analysis. Following the preprocessing steps outlined in [4], we obtain parallel acoustic and articulatory data at a frame rate of 100 observations per second. Acoustic feature MFCCs are computed using 20 msec frame length with 10 msec shift [16]. Each MFCC feature vector is 13 dimensional, where 13-th coefficient represents the log energy of the short-time frame. The first and second derivatives of MFCCs are computed and appended to the MFCC feature vector constructing a 39 dimensional acoustic feature vector.

3. SPARSE SMOOTHING FORMULATION

Let $\boldsymbol{w}_n$ denote the acoustic feature vector at the n-th frame and the corresponding articulatory feature vector be denoted by $\boldsymbol{x}_n = [x_n^1, \cdots, x_n^J]^T$ where x_n^j is the j-th articulatory feature and there are a total of J articulatory features; T is the transpose operator. In AAI, the articulatory features are estimated from the given acoustic feature sequence $\boldsymbol{w}_n,\ 1 \leq n \leq N$ of a sentence of length N frames. We use the GMM for parameterizing the statistical mapping between the acoustic and articulatory features. Estimated articulatory feature vector $\hat{\boldsymbol{x}}_n$ is obtained using minimum mean squared error (MMSE) criterion [1]. In a typical smoothing using low-pass filtering, the smoothed j-th articulatory feature trajectory $\hat{x}_n^{j,\mathrm{s}}$ is obtained by convolving the estimated articulator trajectory with a low-pass filter h_n^j with cut-off frequency f_c^j specific to the j-th articulator:

$$\hat{x}_n^{j,\mathrm{s}} = \hat{x}_n^j \star h_n^j \tag{1}$$

where $\star$ denotes the convolution operation. To avoid any phase distortion due to the low pass filtering on the estimated trajectories, the filtering process is performed twice ("zero-phase filtering") - the trajectory is initially filtered and then time-reversed and filtered again and time-reversed once more finally [17].

It is important to note that, after low-pass filtering, the n-th sample of the smoothed trajectory $\hat{x}_n^{j,\mathrm{s}}$ is in general different from that of the MMSE criterion based estimated $\hat{x}_n^j$. Thus, although $\hat{x}_n^j$ is optimal in MMSE sense, $\hat{x}_n^{j,\mathrm{s}}$ is not. In sparse smoothing, we would like to obtain a smoothed $\hat{x}_n^{j,\mathrm{ss}}$ from $\hat{x}_n^j$ while preserving MMSE optimality for as many samples as possible. In other words, the difference sequence between the smooth and estimated trajectories

$$d_n^j = \hat{x}_n^{j,\mathrm{ss}} - \hat{x}_n^j \tag{2}$$

should have as few non-zero entries as possible, i.e., d_n^j should be a sparse sequence, subjected to a constraint that $\hat{x}_n^{j,\mathrm{ss}}$ is smooth to a required degree. For sparse smoothing, we propose the following optimization problem (informal notation) for computing a sparse d_n^j

$$\min_{d_n^j} \ \|d_n^j\|_1 \text{ subject to } \|g_n^j \star (\hat{x}_n^j + d_n^j)\|_2 \leq \epsilon. \tag{3}$$

The objective is the ℓ_1-norm of the difference sequence d_n^j, whose minimization ensures that d_n^j is as sparse as possible. The constraint term expresses that the energy of $\hat{x}_n^{j,\mathrm{ss}} = \hat{x}_n^j + d_n^j$, filtered with a suitably chosen high-pass filter g_n^j, with cut-off frequency f_c^j (specific to j-th articulator), is bounded by ϵ. This constraint ensures that after sparse smoothing, $\hat{x}_n^{j,\mathrm{ss}}$ is smooth to the required degree.

Let us denote $\hat{\boldsymbol{x}}^j = [\hat{x}_1^j, \cdots, \hat{x}_N^j]^T$ and $\boldsymbol{d}^j = [d_1^j, \cdots, d_N^j]^T$ respectively. Due to the associativity of convolution and distributivity of time-reversal, it can be seen that zero-phase filtering with a N length g_n^j is equivalent to a multiplication by the $N \times N$ convolution matrix $\mathbf{G}^j$, constructed using the autocorrelation sequence of g_n^j: $[\mathbf{G}^j]_{kl} = \sum_n g_{n-k}^j g_{n-l}^j$. Thus the optimization problem can be written in matrix-vector notation as follows:

$$\min_{\boldsymbol{d}^j} \ \|\boldsymbol{d}^j\|_1 \text{ subject to } \|\mathbf{G}^j(\hat{\boldsymbol{x}}^j + \boldsymbol{d}^j)\|_2 \leq \epsilon.$$

Letting $\boldsymbol{y}^j = -\mathbf{G}^j \hat{\boldsymbol{x}}^j$, the problem can be rewritten as

$$\min_{\boldsymbol{d}^j} \ \|\boldsymbol{d}^j\|_1 \text{ subject to } \|\mathbf{G}^j(\boldsymbol{d}^j) - \boldsymbol{y}^j\|_2 \leq \epsilon. \tag{4}$$

which is in the standard Basis Pursuit DeNoising (BPDN) [18] form and related to LASSO [19]. A method for solving Eq. (4) is outlined in the next section.

4. SOLUTION TO SPARSE SMOOTHING

To solve the optimization problem Eq. (4), we use a primal-dual method called the Chambolle-Pock (CP) algorithm [20]. Though a plenty of toolboxes such as SPGL1, CVX, etc. are available for solving ℓ_1 minimization problems, the CP algorithm has the unique advantage of being flexible, where in additional constraints can be included very easily into the objective. Also, CP algorithm relies on the proximal operator of the functions which are easy to evaluate.

4.1. Chambolle-Pock algorithm

Let $\boldsymbol{K} : \mathbb{R}^N \to \mathbb{R}^M$ be a continuous linear operator with norm $\|\boldsymbol{K}\| < \infty$. Let $F : \mathbb{R}^M \to [0, +\infty]$ and $G : \mathbb{R}^N \to [0, +\infty]$ be two proper, convex, lower-semicontinuous functions. The CP algorithm, defined in Algorithm 1, is used to solve saddle-point problems obtained from the primal problem Eq. (5):

$$\min_{\boldsymbol{u} \in \mathbb{R}^N} F(\boldsymbol{K}\boldsymbol{u}) + G(\boldsymbol{u}). \tag{5}$$

Algorithm 1: Chambolle-Pock

- **Input:** Choose $\tau, \sigma > 0$, $(\boldsymbol{u}^0, \boldsymbol{v}^0) \in \mathbb{R}^N \times \mathbb{R}^M$, $\theta \in [0, 1]$ and $\bar{\boldsymbol{u}}^0 = \boldsymbol{u}^0$
- **Iterate:** For $n \geq 0$, until stopping criterion

$$\begin{aligned} \boldsymbol{v}^{n+1} &= \mathrm{prox}_{\sigma F^*}(\boldsymbol{v}^n + \sigma \boldsymbol{K} \bar{\boldsymbol{u}}^n) \\ \boldsymbol{u}^{n+1} &= \mathrm{prox}_{\tau G}(\boldsymbol{u}^n - \tau \boldsymbol{K}^* \boldsymbol{v}^{n+1}) \\ \bar{\boldsymbol{u}}^{n+1} &= \boldsymbol{u}^{n+1} + \theta(\boldsymbol{u}^{n+1} - \boldsymbol{u}^n) \end{aligned} \tag{6}$$

- **Output:** $(\boldsymbol{u}^n, \boldsymbol{v}^n)$

F^* and G^* are the convex conjugates of F and G respectively and $\boldsymbol{K}^*$ is the adjoint of $\boldsymbol{K}$. The *proximal* operator $\text{prox}_{\gamma f}$ of a proper, convex, lower semicontinuous function f, with a parameter γ, is defined by

$$\text{prox}_{\gamma f}(\boldsymbol{u}) := \underset{\boldsymbol{z}\in\mathbb{R}^N}{\arg\min} \frac{1}{2\gamma}\|\boldsymbol{u}-\boldsymbol{z}\|_2^2 + f(\boldsymbol{z}). \tag{7}$$

The closed form of Eq. (7) is easy to derive for several functions commonly used in signal processing [21]. Further, the proximal operator of the conjugate function f^* is easily computed using the celebrated Moreau's identity [20]:

$$\boldsymbol{z} = \text{prox}_{\gamma f}(\boldsymbol{z}) + \gamma \text{prox}_{\frac{1}{\gamma}f^*}\left(\frac{1}{\gamma}\boldsymbol{z}\right). \tag{8}$$

The closed forms of Eq. (7) for the specific functions in Eq. (4) is discussed in Sec. 4.3. To ensure the convergence of the algorithm, the parameters τ and σ have to be chosen such that $\tau\sigma\|\boldsymbol{K}\|^2 < 1$. Details on convergence can be found in [20].

4.2. Unconstrained version of sparse smoothing problem

In order to express the sparse smoothing problem Eq. (4) in the unconstrained form Eq. (5), we consider the indicator functions of the convex sets defined by the constraints. The convex indicator function $\imath_{\mathcal{C}}(\cdot)$ on a convex set $\mathcal{C}$ is defined by

$$\imath_{\mathcal{C}}(\boldsymbol{z}) = \begin{cases} 0 & \text{if } \boldsymbol{z}\in\mathcal{C}, \\ +\infty & \text{otherwise}. \end{cases} \tag{9}$$

Let $\mathcal{B}^j = \left\{\boldsymbol{z}\in\mathbb{R}^N \mid \|\boldsymbol{z}-\boldsymbol{y}^j\|_2 \le \epsilon\right\}$ be a convex norm ball. By including the convex indicator function of the set $\mathcal{B}^j$ into the objective function the constrained problem is turned into the following unconstrained problem, which can be solved using CP algorithm:

$$\boldsymbol{d}_*^j := \underset{\boldsymbol{d}^j\in\mathbb{R}^N}{\arg\min}\left(\|\boldsymbol{d}^j\|_1 + \imath_{\mathcal{B}^j}(\boldsymbol{d}^j)\right). \tag{10}$$

The action of the matrix $\mathbf{G}^j$ on $\boldsymbol{d}^j$ in the indicator function $\imath_{\mathcal{B}^j}(\cdot)$ is implicit in the CP iterations. The sparsely smoothed j-th articulatory feature trajectory is then given by $\boldsymbol{x}^{j,\text{ss}} = \hat{\boldsymbol{x}}^{j,\text{ss}} + \boldsymbol{d}_*^j$.

4.3. Proximal operators of $\|\boldsymbol{u}\|_1$ and $\imath_{\mathcal{B}}(\boldsymbol{u})$

The proximal operator for the ℓ_1-norm is the simple componentwise soft-thresholding operator, defined for a scalar u as:

$$\text{prox}_{\gamma\|\cdot\|_1}(u) := \begin{cases} 0 & \text{if } |u|\le\gamma, \\ (|u|-\gamma)\text{sgn}(u) & \text{otherwise}. \end{cases} \tag{11}$$

The proximal operator of the function $\imath_{\mathcal{B}^j}(\boldsymbol{u})$ is the following projection function onto convex set $\mathcal{B}$:

$$\text{prox}_{\imath_{\mathcal{B}^j}}(\boldsymbol{u}) := \boldsymbol{y}^j + (\boldsymbol{u}-\boldsymbol{y}^j)\min\left(1, \epsilon/\|\boldsymbol{u}-\boldsymbol{y}^j\|_2\right). \tag{12}$$

5. EXPERIMENTAL EVALUATION

We experimentally evaluate the effectiveness of the proposed sparse smoothing formulation by performing AAI followed by sparse smoothing on the MOCHA corpus. We compare the inversion performance obtained by sparse smoothing with that using low-pass filtering. As our goal is to post-process the MMSE estimates, we did not compare the sparse smoothing with other inversion criteria [1, 4], which include smoothness directly in the inversion criterion. The experimental details and results are described below.

5.1. Experimental setup

We perform AAI followed by sparse smoothing separately for the male and the female subjects in the MOCHA corpus and compare the smoothed trajectories obtained by sparse smoothing with the ones obtained by low-pass filtering. The experiments are performed in a 5-fold cross validation setup, where 80% of the data is used for training, 5% is used as development set and remaining 15% is used as test set. Following the work by Toda et al. [1], we have used 64 mixture GMM for learning the acoustic-articulatory map using the training data. The development set is used to optimize the hyperparameters of the sparse smoothing optimization, i.e., f_c^j, σ, and ϵ. The set of values $\{3,\ 4,\ 5, \cdots,\ 24\}$Hz are chosen for f_c^j, {e-5, e-4, e-3, 0.01, 0.1} for σ and $\{0.01,\ 0.1,\ 1,\ 10,\ 50,\ 100,\ 500,\ 1000\}$ for ϵ. The combination of hyperparameters which yields the best performance on the development set is finally chosen and used for sparse smoothing on the test set. Articulator specific high-pass filter g_n^j is chosen as a 5-order Chebyshev type-II IIR filter (using `cheby2()` in Matlab) with stop-band ripple 40 dB down compared to the pass-band ripple [22].

We report the inversion performance as an average performance over all sentences in the testsets of all folds. Root mean squared error (RMSE) between the original and the estimated articulator trajectories of each sentence is used as a performance measure. The RMSE reflects the average closeness between the original and the estimated articulatory features. However, a minimum RMSE does not always mean the trajectories are similar; for example, the estimated one can be very jagged although it might be close to the actual one. Thus, as an additional performance measure, we compute Pearson correlation coefficient (PCC) [23] between the original and estimated feature trajectories of each sentence. Average RMSE and PCC over all sentences are used to measure the quality of inversion.

In addition to comparing the inversion performances by using sparse smoothing and low-pass filtering, we evaluate the effectiveness of the sparse smoothing by computing the degree of sparsity of the difference sequence d_n^j between the sparsely smoothed trajectory and the GMM based estimated trajectory (Eq. (2)). This is done by counting the % number of zeros (NZ) in d_n^j for every utterance. NZ samples of the smoothed trajectory will be optimal in MMSE sense; higher the NZ, more articulatory features in the sparsely smoothed trajectory would be optimal as per the AAI criterion.

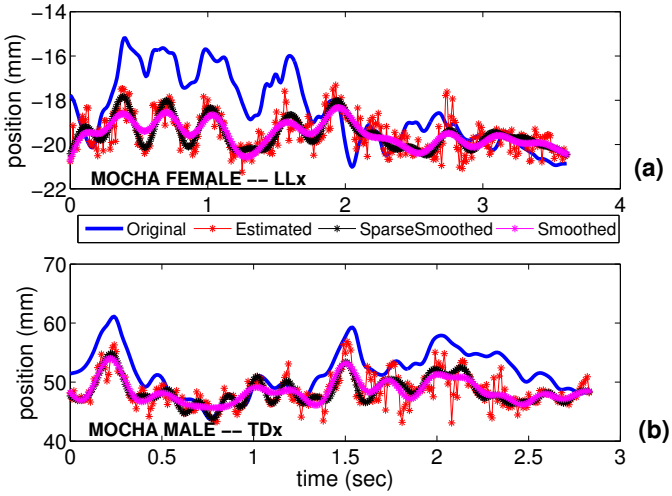
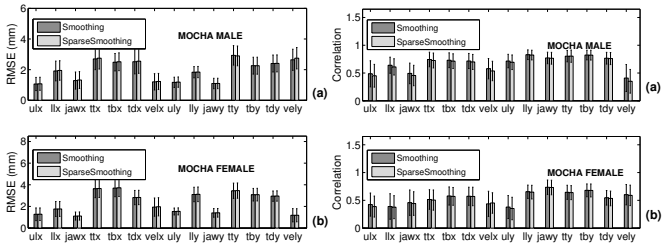


Fig. 1. *Illustrative examples of the sparse smoothing and smoothing using low-pass filter for (a) female subject's LLx and (b) male subject's TDx trajectories.*



I. RMSE of AAI II. PCC of AAI

Fig. 2. *Comparison of low-pass filtering (smoothing) and sparse smoothing - error bars indicate average inversion performance with* $\pm$ *one standard deviation.*

5.2. Results and discussion

The hyperparameters f_c^j, σ, and ϵ are optimized on the development set for each articulator separately. The optimized high-pass filter cut-off frequencies for different articulators range from 3Hz to 5Hz. This suggests that the frequency contents of the estimated trajectories above 3-5Hz are attenuated in the sparse smoothing optimization resulting in smoothed articulator trajectories.

If the value of ϵ is too large, then the solution $\boldsymbol{d}^j$ tends to be a zero vector, which means that no smoothing occurs. When ϵ is too small, then $\boldsymbol{d}^j$ tends closer to $-\hat{\boldsymbol{x}}^j$, thereby pushing the smoothed articulator trajectories towards 0 (the maximally smooth trajectory). The optimized values of ϵ balances these two factors to result in the smoothed articulator trajectories with required degree of smoothness. We observe that the optimized values of ϵ are typically higher for VELx and LLy compared to other articulators.

Fig. 1 illustrates the smoothed trajectories using sparse smoothing and low-pass filtering for LLx and TDx trajectories of female and male subjects respectively. Fig. 1 also shows the respective original and estimated trajectories using GMM based AAI. It is clear from the figure that the sparsely smoothed trajectory is similar to the low-pass filtered trajectory except that they are different at few places, confirming that the sparsely smoothed trajectory preserves as many samples of the estimated trajectory as possible. In Fig. 2, we compare the inversion performances of AAI using smoothing (low-pass filtering) and the proposed sparse smoothing using RMSE and PCC for each of 14 articulators of both the male and female subjects. The barplots demonstrate the RMSE (and PCC) values averaged over all sentences in test set. Errorbars indicate $\pm$ 1 standard deviation around the average measure. It is clear from Fig. 2 that in terms of RMSE and PCC, the inversion performance of the two methods are not significantly different. This is also true for both subjects of the MOCHA database. Thus, sparse smoothing, formulated as an optimization problem unlike low-pass filtering, preserves the inversion performance obtained using low-pass filtering.

Although the inversion performances of typical low-pass filter based smoothing and sparse smoothing are similar, sparse smoothing, by its formulation, has a key property, unlike low-pass filtering, that it preserves the optimality criterion of AAI at as many frames as possible. To examine this, we report the degree of sparsity of the difference signal d_n^j by reporting NZ for different articulators in Fig. 3 for both subjects in the MOCHA database. For the male subject the NZ ranges from 6.5% (for VELy) to 23.5% (for LLy) with a mean of 14.6% across all articulators. Similarly for the female subject the NZ ranges from 9% (for JAWx) to 27.7% (for VELx) with a mean of 14.9% across all articulators. Thus, on average $\sim$15% of the samples in the sparsely smoothed articulator trajectories are identical to the corresponding samples in the GMM based estimated trajectories, which are optimal in MMSE sense. It is important to note that in the case of low-pass filter based smoothing, no samples of the smoothed trajectory are identical to those of the estimated trajectory; thus, NZ for low-pass filter based smoothing is 0% for all articulators. Therefore, sparse smoothing leads to a smoothed trajectory which has similar AAI performance as that of the low-pass filtering, but has a significantly higher percentage of samples ($\sim$15%) with AAI optimality as opposed to that (0%) for low-pass filtering.

The advantages of retaining MMSE optimal estimates after smoothing remain to be evaluated in the context of an application which uses AAI. However, sparse smoothing formulation trades this requirement with the extent of smoothing through a single tuning parameter, without a need to redesign the smoothing filter every time.

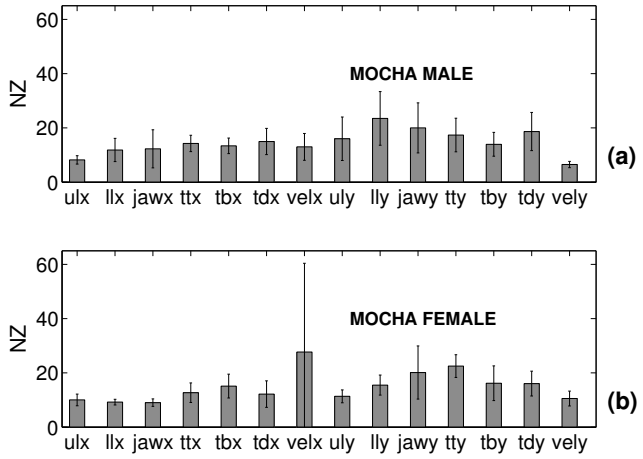


Fig. 3. *The number of zeros (NZ) [in percent] (average* $\pm$ *one standard deviation) in the difference signal between the estimated and smoothed articulator trajectories for all 14 articulators in the case (a) male and (b) female subjects in MOCHA corpus.*

6. CONCLUSIONS

We propose a sparse smoothing formulation which is used as a post-processing step to the estimated articulator trajectories from GMM based AAI to obtain smooth articulator trajectories. The key property of the proposed sparse smoothing is that it constrains the smoothed trajectory samples to be identical to the samples of the estimated trajectory at as many frames as possible, thereby preserving the optimality of AAI criterion for these samples unlike in a typical smoothing using low-pass filtering.

The proposed formulation sparsely smooths each articulator trajectory separately. Thus a subset of the smoothed articulatory feature vector could be optimal as per AAI criterion in certain frames while the rest of the articulatory features at those frames may not remain optimal. An alternate sparse smoothing optimization could be formulated where all the articulatory features can be jointly (e.g., total variation) considered such that following sparse smoothing all articulatory features at a frame remain optimal as per the AAI criterion. It is worth noting that the proposed sparse smoothing, being a post-processing step, is applicable beyond GMM based AAI. This could be used as the post processing step to the articulatory estimates from the AAI using any other inversion criterion so that the respective AAI criterion optimality is preserved for as many samples in the smoothed trajectory as possible. Phonetic constraints could also be incorporated in the sparse smoothing framework. Further, we could also envisage to integrate the sparse smoothing step in the AA inversion step itself.

7. REFERENCES

[1] T. Toda, A. Black, and K. Tokuda, "Statistical mapping between articulatory movements and acoustic spectrum using a gaussian mixture model," *Speech Communication*, vol. 50, pp. 215–217, 2008.

[2] K. Richmond, "A trajectory mixture density network for the acoustic-articulatory inversion mapping," *Proc. ICSLP, Pittsburgh,USA*, pp. 577–580, September 2006.

[3] L.Zhang, "Acoustic-articulatory modeling with the trajectory hmm," *IEEE Signal Processing Letters*, vol. 15, pp. 245–248, 2008.

[4] P. K. Ghosh and S. S. Narayanan, "A generalized smoothness criterion for acoustic-to-articulatory inversion," *J. Acoust. Soc. Am.*, vol. 128, no. 4, pp. 2162–2172, 2010.

[5] S. Ouni and Y. Laprie, "Modeling the articulatory space using a hypercube codebook for acoustic-to-articulatory inversion," *J. Acoust. Soc. Am.*, vol. 118, no. 1, pp. 444–460, 2005.

[6] S. King and P. Taylor, "Detection of phonological features in continuous speech using neural networks," *Computer. Speech Lang.*, vol. 14, pp. 333–345, 2000.

[7] A. Toutios and K. Margaritis, "A rough guide to the acoustic-to-articulatory inversion of speech," *Proceedings of the 6th Hellenic European Conference on Computer Mathematics and its Applications (HERCMA-2003)*, pp. 746–753, September 2003.

[8] P. K. Ghosh and S. S. Narayanan, "On smoothing articulatory trajectories obtained from gaussian mixture model based acoustic-to-articulatory inversion," *J. Acoust. Soc. Am. Express Letters (JASAEL)*, vol. 134, no. 2, pp. EL258–EL264, July 2013.

[9] A. A. Wrench and K. Richmond, "Continuous speech recognition using articulatory data," *Proc. ICSLP, Beijing, China*, pp. 145–148, 2000.

[10] J. Frankel, K. Richmond, S. King, and P. Taylor, "An automatic speech recognition system using neural networks and linear dynamic models to recover and model articulatory traces," *Proc. ICSLP, Beijing, China*, vol. 4, pp. 254–257, October 2000.

[11] G. Ramsay and L. Deng, "Maximum-likelihood estimation for articulatory speech recognition using a stochastic target mode," *Proc. EUROSPEECH*, pp. 1401–1404, 1995.

[12] T. Stephenson, H. Bourlard, S. Bengio, and A. C. Morris, "Automatic speech recognition using dynamic bayesian networks with both acoustic and articulatory variables," *Proc. ICSLP*, pp. 951–954, 2000.

[13] S. Maeda, "Compensatory articulation during speech: Evidence from the analysis and synthesis of vocal tract shapes using an articulatory model," *Speech production and speech modelling, edited by W. Hardcastle and A. Marchal (Kluwer Academic Publishers, Dordrecht, The Netherlands)*, pp. 131–149, 1990.

[14] M. M. Sondhi, "Articulatory modeling: a possible role in concatenative text-to-speech synthesis," *IEEE 2002 Workshop on Speech Synthesis, Santa Monica, USA*, pp. 73–78, September 2002.

[15] A. A. Wrench and H. J. William, "A multichannel articulatory database and its application for automatic speech recognition," *5th Seminar on Speech Production: Models and Data, Bavaria*, pp. 305–308, 2000.

[16] S. J. Young, "The HTK hidden markov model toolkit: Design and philosophy," *Entropic Cambridge Research Laboratory, Ltd*, vol. 2, pp. 2–44, 1994.

[17] Richard G. Lyons, *Understanding Digital Signal Processing*, Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1st edition, 1996.

[18] M. Fornasier, "Numerical methods for sparse recovery," *Theoretical Foundations and Numerical Methods for Sparse Recovery, Radon Series Comp. Appl. Math. deGruyter*, 2010.

[19] M. Schmidt, *Least squares optimization with L1-norm regularization*, Project Report, University of British Columbia, 2005.

[20] A. Chambolle and T. Pock, "A first-order primal-dual algorithm for convex problems with applications to imaging," *Journal of Mathematical Imaging and Vision*, vol. 40, no. 1, pp. 120–145, May 2011.

[21] P L Combettes and J-C Pesquet, "Proximal splitting methods in signal processing," *Fixed-Point Algorithms for Inverse Problems in Science and Engineering*, Jan. 2011.

[22] A. Antoniou, *Digital Filters: Analysis, Design & Applications*, McGraw-Hill Education (India) Pvt Limited, 2001.

[23] J. L. Rodgers and W. A. Nicewander, "Thirteen ways to look at the correlation coefficient," *The American Statistician*, vol. 42, no. 1, pp. 59–66, 1988.