A generalized smoothness criterion for acoustic-to-articulatory inversion

Prasanta Kumar Ghosh[a] and Shrikanth Narayanan
Department of Electrical Engineering, Signal Analysis and Interpretation Laboratory, University of Southern California, Los Angeles, California 90089

(Received 1 September 2009; revised 17 March 2010; accepted 29 May 2010)

The many-to-one mapping from representations in the speech articulatory space to acoustic space renders the associated acoustic-to-articulatory inverse mapping non-unique. Among various techniques, imposing smoothness constraints on the articulator trajectories is one of the common approaches to handle the non-uniqueness in the acoustic-to-articulatory inversion problem. This is because, articulators typically move smoothly during speech production. A standard smoothness constraint is to minimize the energy of the difference of the articulatory position sequence so that the articulator trajectory is smooth and low-pass in nature. Such a fixed definition of smoothness is not always realistic or adequate for all articulators because different articulators have different degrees of smoothness. In this paper, an optimization formulation is proposed for the inversion problem, which includes a generalized smoothness criterion. Under such generalized smoothness settings, the smoothness parameter can be chosen depending on the specific articulator in a data-driven fashion. In addition, this formulation allows estimation of articulatory positions recursively over time without any loss in performance. Experiments with the MOCHA TIMIT database show that the estimated articulator trajectories obtained using such a generalized smoothness criterion have lower RMS error and higher correlation with the actual measured trajectories compared to those obtained using a fixed smoothness constraint.

© *2010 Acoustical Society of America.* [DOI: 10.1121/1.3455847]

PACS number(s): 43.72.Ar [MAH] Pages: 2162–2172

I. INTRODUCTION

Acoustic-to-articulatory inversion refers to the mapping of speech signal or model representations from the acoustic space to the articulatory space. The acoustic space is typically defined by one of several popular spectro-temporal features or model parameters derived from the acoustic speech signal. Similarly, the articulatory space can be represented in a variety of ways including through (1) stylized models such as the Maeda's model[1,2] or the lossless tube model[3] of the vocal tract, (2) linguistic rule based models[4–6] or (3) direct physiological data based representations of articulatory information.[7] In this work, we consider the physiological data based representation of the articulatory space, where articulatory data (e.g., position of the lips, jaw, tongue, velum etc.) during speech production are obtained directly from the talkers by means of a specialized instrument such as an electromagnetic articulograph (EMA), ultrasound, or magnetic resonance imaging. Hence, in this paper, by acoustic-to-articulatory inversion we refer to the problem of estimating the articulatory positions (physiological data) from a given acoustic speech signal.

Acoustic-to-articulatory inversion has received a great deal of attention from researchers over the last several decades, notably motivated by potential applications to speech technology development. All acoustic-to-articulatory inversion solutions are supervised, i.e., they require some knowledge about the possible articulatory positions for a given acoustic signal from some training data. Such solutions often provide complementary information to acoustics and, thus, can help improve the performance of current automatic speech recognition systems, especially in cases such as with noisy, spontaneous, or pathological speech.[8–11] In addition, articulatory gesture representations are considered to have a parsimonious description of the underlying dynamics for producing acoustic speech signal[4–6] and hence deriving these gestures from the speech signal or from the estimated articulatory positions or tract variables[12] can provide insight into linguistic phonology.

It is widely known that the difficulty in the acoustic-to-articulatory mapping lies in its ill-posed nature. It has been shown that multiple distinct articulatory configurations can result in the same or very similar acoustic effects. An empirical investigation of such non-uniqueness in acoustic-to-articulatory mapping can be found in Ref. 13. Atal *et al.*[14] also showed that an infinite number of articulatory configurations can generate three identical formant frequencies. The problem is highly non-linear, too; two somewhat similar articulatory states may give rise to totally different acoustic signals.[15] One of the reasons for this non-unique mapping may come from the limitation of modeling or parametric representation of both articulatory and acoustic spaces. For example, the non-uniqueness in mapping arises using only formant based acoustic representation, but additional knowledge about bandwidth in the acoustic representation reduces the non-uniqueness. Nonetheless, non-uniqueness in inverse mapping poses a serious problem in the estimation of articu-

[a] Author to whom correspondence should be addressed. Electronic mail: prasantg@usc.edu

latory parameters from acoustic ones and, hence, motivates investigation for a better solution to the inversion problem.

A common approach to address this ill-posed problem is to use regularization[16] or dynamic constraints while estimating the inverse mapping.[17–21] Sorokin *et al.*[17] chose a regularizing term that prevents inverse solutions from deviating too much from the neutral position of articulators. Schroeter and Sondhi[18] presented a method based on dynamic programming (DP) to search articulatory codebooks with a penalty factor for large “articulatory efforts,” that is, fast changes in the vocal tract so that the estimated articulator trajectories are smoothly evolving. They used LPC derived cepstral coefficients as the acoustic feature and introduced a lifter in the computation of the acoustic distance and dynamic cost in making a transition from one vocal tract shape to another. Toda *et al.*[21] used a Gaussian mixture model (GMM) to perform the inversion mapping but formulated it as a statistical trajectory model by augmenting observations (mel cepstral coefficients) with first and second derivatives features. Richmond[22] proposed a trajectory model which is based on a mixture density network for estimating maximum likelihood trajectories which respects constraints between the static and derived dynamic features. Similar methods using dynamical constraints have been proposed based on Kalman filtering and smoothing.[23–25] Dusan *et al.*[26] extended previous studies of estimating articulator trajectories by Kalman filtering by implementing phonological constraints by modeling different articulatory-acoustic sub-functions, each corresponding to a phonological coproduction model.

The essence of the regularization or smoothness constraints lies in the physical movement of the articulators. The trajectory of the articulators during speech production is in general smooth and slowly varying. Demanding smooth changes in the articulators can reduce the non-uniqueness in the inversion problem.[18] For example, Toda *et al.*[21] reported that with lowpass filtering of the solution of the GMM based mapping, they achieved lower RMS error. Similarly, Richmond[27] performed lowpass filtering as a postprocessing step. It was shown that low pass filtering of the MLP output by articulator specific cut-off frequencies indeed moderately improved the result, i.e., the RMS error decreased and the correlation score improved. Richmond *et al.*[28] discussed the usefulness of low-pass filtering on the articulator trajectory as a smoothness constraint in the optimization. For example, in Ref. 19, one such constraint was used as a part of the DP search through the output of their network, which constrained the articulator trajectories to be as smooth as possible. Also in Refs. 29 and 30, the articulator trajectories are constrained such that articulators move as slowly as possible.

The smoothing of a signal can be interpreted as linear time-invariant (LTI) filtering, in which the high frequency components of the signal are suppressed and low frequency components are preserved so that the signal becomes smooth. For example, authors in Refs. 18, 31, and 32 minimize the DP cost function, which contains $(A_t - A_{t-1})^2$, where A_t is the articulator variable at time frame t. By minimizing $(A_t - A_{t-1})^2$ over the entire time, the energy of the difference of the articulator variable is minimized. $\Sigma_t (A_t - A_{t-1})^2$ can be interpreted as the energy of the output of a discrete-time LTI filter with impulse response $h = [1 \ \ -1]$ where the input is A_t. $h = [1 \ \ -1]$ is a high pass filter, whose 3 dB cut-off frequency is $F_s/4$, where F_s is the sampling frequency. By minimizing the energy of the output of this filter, the high frequency component in the articulator trajectory is suppressed. However, a particular high-pass filter with fixed cut-off frequency may not be optimal for different articulators. A more systematic approach would be to design appropriate high pass filters for individual articulators and include them in the optimization. However, note that an arbitrary high pass filter might have large finite or an infinite impulse response. The complexity of DP increases exponentially with the length of the filter K and hence, it becomes computationally expensive even for an FIR filter with $K > 2$. When the smoothness constraints in the cost function involves an IIR filter, the cost function cannot be solved using DP at all.

In this paper, we derive a formulation where any arbitrary high pass filter can be used in the inversion problem for smoothing articulator trajectories. The cut-off frequency of the filter can be adaptively tuned in such a generalized smoothness setting and, hence, this formulation can provide a more realistic articulator trajectory compared to that obtained by a filter with fixed cut-off frequency. The formulation is similar to the codebook search approach but under a general smoothness criterion. Another key advantage of this formulation is that the solution of the articulator trajectory need not be computed all at once; rather, a recursive solution can be derived without any degradation in performance.

The paper is organized as follows: Section II discusses the data set and the required pre-processing on the articulatory data. The frequency domain analysis of the articulatory data is described in Section III. This is done to obtain insight into the nature of the smoothness of the articulatory data, which in turn is used to design the filters used in the formulation discussed in Section IV. The recursive solution to the problem is discussed in Section V. In Section VI various acoustic features are analyzed to obtain the best representative feature for this inversion problem. Experiments and results are discussed in Section VII followed by conclusions in section VIII.

II. DATA SET AND PRE-PROCESSING

The Multichannel Articulatory (MOCHA) database[7] is used for the analysis and experiments of this paper. The MOCHA database consists of acoustic and corresponding articulatory ElectroMagnetic Articulography (EMA) data from two speakers—one male (with a Northern English accent) and one female speaker (with a Southern English accent). The acoustic and articulatory data were collected while each speaker read a set of 460 phonetically-diverse British English TIMIT sentences. The articulatory data consist of X and Y coordinates of nine receiver sensor coils attached to nine points along the midsagittal plane, namely the lower incisor or jaw (li_x, li_y), upper lip (ul_x, ul_y), lower lip (ll_x, ll_y), tongue tip (tt_x, tt_y), tongue body (tb_x, tb_y), tongue dorsum (td_x, td_y), velum (v_x, v_y), upper incisor (ui_x, ui_y) and bridge of the nose (bn_x, bn_y). The last two are used as reference coils. Thus, the first seven coils

TABLE I. Number of frames of articulatory data available for training, development, and test set.

	No. articulator frames		
Speaker	Training set	Dev set	Test set
Male	85 673	8 866	14 553
Female	98 666	10 298	16 454

provide 14 channels of articulatory position information. The position of each coil was recorded at 500 Hz with 16 bit precision. The corresponding speech was collected at 16 KHz sampling rate.

Although the position data of seven articulators in the MOCHA database have been already processed to compensate for head movement, the data in this raw form is still not suitable for analysis or modeling.[27] The position data have high frequency noise resulting from EMA measurement error, while the articulatory movements are predominantly low pass in nature (we will see in the next section that 99% of the energy is contained below ~21 Hz for all the articulators). Hence the articulatory data of each channel is low pass filtered with a cut-off frequency of 35 Hz. Since articulatory data is low-pass due to the nature of the physical movement of articulators, the choice of 35 Hz is sufficient to keep the articulatory position information unaltered. To avoid any phase distortion due to the low pass filtering on the articulatory data, the filtering process is performed twice ("zero-phase filtering")—the data is initially filtered and then reversed and filtered again and reversed once more finally. After filtering, the articulatory data is downsampled by a factor of 5 so that the frame rate is 100 per second. Since the low pass cut-off frequency was 35 Hz, no aliasing occurs due to downsampling.

Each utterance of both speakers has silence in the initial portion and toward the end of the utterance. Since during non-speech portions the articulators can assume any position, considering data from these regions can increase the variability in the inverse mapping. Hence, the silence portions were manually selected and the corresponding articulatory data were omitted. Of the 460 utterances available from each speaker, data from 368 utterances (80%) are used for training, 37 utterances (8%) as the development set (dev set), and the remaining 55 utterances (12%) as the test set. In summary, for the two speakers, the number of frames of available articulatory data are shown in Table I.

The mean position for each articulator changes from utterance to utterance.[27] A few reasons for this variation of mean articulatory position have been stated in Ref. 27, namely change in temperature and shift in the location of the EMA helmet and transmitter coil relative to the subject's head. This means that even after low-pass filtering and downsampling, the articulatory data are still not directly ready for the modeling purpose. To make the data ready for such use, we first subtract the mean articulator location from the articulatory position for every utterance in a way similar to Ref. 27. Finally, we add the mean articulatory position, averaged over all utterances. These pre-processed articulator trajectories are used for further analysis and experiments.

III. EMPIRICAL FREQUENCY ANALYSIS OF ARTICULATORY DATA

The articulators in the human speech production system move to create distinct vocal tract shapes to generate different acoustic signals. The articulators, i.e., tongue, lips, jaw, velum, are in general slow moving and thus the articulatory data are low-pass in nature.[33] The purpose of analyzing the spectrum of the articulatory data is to understand the nature of the articulatory movement and quantify the effective maximum frequency content of such slowly varying signals. This in turn would inform us about the smoothness of the articulatory movement for designing appropriate smoothing criteria for different articulatory data.

The frequency domain analysis is performed separately on the articulator trajectories of each utterance in the training set. There are 14 different articulator trajectories for every utterance. Let $\{x[n];1\leq n\leq N\}$ denote any one of these 14 trajectories for a particular utterance. We compute the samples of its spectrum $S[k]$, $k=0,\ldots,N_F-1$ using discrete Fourier transform (DFT) with a DFT order $N_F=2^{14}=16384$ as follows:

$$S[k]=\left|\sum_{n=1}^{N}x_0[n]\exp^{-j(2\pi/N_F)kn}\right|^2, \qquad (1)$$

where $x_0[n]=x[n]-\frac{1}{N}\Sigma_{n=1}^{N}x[n]$ is the dc removed articulator trajectory. $S[k]$ of all 14 articulator trajectories are found to be low-pass, as expected. Since the sampling frequency of $x[n]$ is 100 Hz, the frequency resolution of the spectrum is $100/N_F=0.0061$ Hz. The total energy of $x[n]$ is $\Sigma_{n=1}^{N}x^2[n]=\frac{1}{N_F}\Sigma_{k=1}^{N_F}S[k]$ (by Parseval's theorem). We would like to calculate the frequency below which a certain percentage (say α%) of the total energy is contained. This is performed by finding N_c such that $(S[0]+2\Sigma_{k=1}^{N_c}S[k])/\Sigma_{k=1}^{N_F}S[k]=\alpha/100$. The corresponding frequency is $f_c=N_c100/N_F$ Hz. The mean f_c [along with standard deviation (SD)] averaged over all utterances for $\alpha=90$, 95 and 99 is tabulated in Table II for all 14 articulators of both speakers.

From Table II, it can be seen that the mean f_c of a particular articulator is similar for both speakers except for ul_x, ll_x, li_x. For a particular speaker, not all articulators have the same mean f_c for all α. For example, for $\alpha=90$, the mean f_c varies from 3.33 Hz (ul_y) to 4.52 Hz (v_x) in the case of the male speaker. For $\alpha=99$, this variation is even more. The same is true for the data of the female speaker.

It is well-known that the articulatory movements are for the most part slow and smooth.[33] However, not all the articulators have equal degrees of smoothness as demonstrated by the aforementioned empirical frequency analysis. These results will be invoked in selecting parameter values for smoothness constraints for the different articulators in implementing the inversion problem.

IV. GENERALIZED SMOOTHNESS CRITERION FOR THE INVERSION PROBLEM

Let $\{\mathbf{z}_i;1\leq i\leq T\}$ represent the acoustic feature vectors in the training set. Also let x_i denote the corresponding position value of any one of the 14 articulator channels. Now

TABLE II. The mean f_c (standard deviation in bracket) of articulatory data of two speakers in the MOCHA-TIMIT database.

	Mean f_c (SD) (in Hz)					
	Male			Female		
Articulator	$\alpha=90$	$\alpha=95$	$\alpha=99$	$\alpha=90$	$\alpha=95$	$\alpha=99$
ul_x	4.03 (1.77)	6.11 (2.10)	11.71 (3.11)	2.67 (0.71)	3.62 (0.96)	7.67 (3.31)
ll_x	4.02 (0.75)	5.07 (0.92)	9.63 (3.27)	2.88 (0.72)	3.89 (0.91)	8.20 (3.10)
li_x	4.15 (1.38)	5.81 (1.79)	11.00 (3.07)	2.69 (0.67)	3.66 (0.92)	7.77 (3.37)
tt_x	3.75 (0.66)	4.71 (0.79)	9.13 (3.67)	3.36 (0.63)	4.29 (0.74)	7.60 (2.51)
tb_x	3.64 (0.68)	4.60 (0.77)	8.64 (3.05)	3.27 (0.66)	4.14 (0.73)	7.15 (2.17)
td_x	3.56 (0.72)	4.53 (0.83)	8.12 (2.05)	3.43 (0.76)	4.43 (0.83)	7.81 (3.06)
v_x	4.52 (1.35)	6.93 (2.24)	21.68 (12.70)	3.94 (1.31)	5.97 (2.72)	20.63 (15.52)
ul_y	3.33 (0.67)	4.35 (0.85)	8.99 (4.40)	3.10 (0.67)	4.00 (0.78)	7.69 (3.14)
ll_y	4.40 (0.57)	5.23 (0.60)	9.27 (3.26)	4.11 (0.55)	4.92 (0.61)	7.74 (1.92)
li_y	3.37 (0.64)	4.23 (0.75)	8.26 (3.60)	3.49 (0.57)	4.37 (0.67)	7.75 (2.98)
tt_y	4.13 (0.71)	5.07 (0.77)	8.54 (2.33)	4.30 (0.71)	5.35 (0.76)	8.64 (1.65)
tb_y	3.60 (0.61)	4.43 (0.63)	7.44 (1.97)	3.38 (0.57)	4.19 (0.63)	7.06 (2.33)
td_y	3.71 (0.59)	4.52 (0.59)	7.55 (2.53)	3.46 (0.59)	4.38 (0.68)	8.57 (3.05)
v_y	3.88 (0.98)	5.62 (1.63)	15.53 (9.34)	3.80 (1.10)	5.34 (2.12)	15.74 (12.32)

suppose, for the inversion problem, a (test) speech utterance is given and the acoustic feature vectors computed for this utterance are denoted by $\{\mathbf{u}_n; 1 \le n \le N\}$. The goal is to find out the corresponding position values of each articulator channel denoted by $\{x[n]; 1 \le n \le N\}$ from the $\{\mathbf{u}_n; 1 \le n \le N\}$.

We need to minimize the high frequency components in $x[n]$ to ensure that the estimated articulatory position is smooth and slowly varying. Hence, the smoothness requirement is equivalent to minimizing the energy of the output of a high pass filter with input $\{x[n]; 1 \le n \le N\}$. Also suppose, based on the knowledge of the frequency content of the articulator trajectory, the high pass filter h is given. h can be an FIR or IIR filter. For an FIR filter the impulse response $h[n]$ is specified and for an IIR filter the rational transfer function $H(z)$, the $\mathcal{Z}$ transform of $h[n]$, is specified. Let $y[n]$ denote the output of h with input $\{x[n]; 1 \le n \le N\}$, i.e.,

$$y[n] = \sum_{k=1}^{N} x[k]h[n-k]. \tag{2}$$

Let L possible values of the articulatory position at the n^{th} frame of the test speech utterance be denoted by $\{\eta_n^l; 1 \le l \le L\}$. These are obtained using a training set $\{(\mathbf{z}_i, x_i); 1 \le i \le T\}$ and $\mathbf{u}_n$. Let p_n^l denote the probability that η_n^l is the value of the articulatory position at the n^{th} frame given that $\mathbf{u}_n$ is the acoustic feature. L can be, in general, equal to T. Then the inversion problem can be stated as follows:

$$\begin{aligned}\{x^\star[n]; 1 \le n \le N\} &= \arg\min_{\{x[n]\}} J(x[1], \ldots, x[N]) \\ &\triangleq \arg\min_{\{x[n]\}} \\ &\times \left\{ \sum_n (y[n])^2 + C \sum_n \sum_l (x[n] - \eta_n^l)^2 p_n^l \right\},\end{aligned} \tag{3}$$

where J denotes the cost function to be minimized and $y[n]$ is given in Eq. (2).

The first term $\Sigma_n (y[n])^2$ in the cost function is the energy of the output of the filter h. The second term $\Sigma_n \Sigma_l (x[n] - \eta_n^l)^2 p_n^l$ denotes the weighted cost of how different $x[n]$ is from η_n^l, $1 \le l \le L$, where the weights are p_n^l (η_n^l and p_n^l are determined from the training set). For example, if $p_n^l = 1$ for $l=1$ and $p_n^l = 0$ for $l > 1$, this means $x[n]$ has to be as close as η_n^1. In other words, if it turns out that the probability of the articulatory position being η_n^1 is very high based on the training set, the solution $x^\star[n]$ has to be as close as η_n^1. More generally, the probability of $x[n]$ being equal to η_n^l is p_n^l, $1 \le l \le L$. $C(>0)$ is the trade off parameter between these two terms. For minimization, we set

$$\frac{\partial J}{\partial x[m]} = 0, \quad m = 1, \ldots, N,$$

$$\Rightarrow 2\left\{ \sum_n \left(\sum_k x[k]h[n-k] \right) h[n-m] + C \sum_l (x[m] - \eta_m^l) p_m^l \right\} = 0, \quad m = 1, \ldots, N,$$

$$\Rightarrow \sum_k x[k] \left(\sum_n h[n-k]h[n-m] \right) + \left(C \sum_l p_m^l \right) x[m] = C \sum_l \eta_m^l p_m^l, \quad m = 1, \ldots, N,$$

$$\Rightarrow \sum_{k=1}^{N} x[k] R_h[m-k] + \left(C \sum_l p_m^l \right) x[m] = C \sum_l \eta_m^l p_m^l, \quad m = 1, \ldots, N,$$

where $R_h[m-k] \triangleq \Sigma_n h[n-k]h[n-m]$, the autocorrelation sequence of $h[n]$. The above set of N equations can be written in matrix vector form as follows:

$$\begin{pmatrix} R_h[0]+C\sum_l p_1^l & R_h[1] & \cdots & R_h[N-1] \\ R_h[-1] & R_h[0]+C\sum_l p_2^l & \cdots & R_h[N-2] \\ . & . & . & . \\ . & . & . & . \\ R_h[-(N-1)] & R_h[-(N-2)] & \cdots & R_h[0]+C\sum_l p_N^l \end{pmatrix} \begin{pmatrix} x[1] \\ x[2] \\ . \\ . \\ x[N] \end{pmatrix} = \begin{pmatrix} C\sum_l \eta_1^l p_1^l \\ C\sum_l \eta_2^l p_2^l \\ . \\ . \\ C\sum_l \eta_N^l p_N^l \end{pmatrix}. \qquad (4)$$

Assuming p_n^l are normalized such that $\Sigma_l p_n^l = 1\,\forall n$, (it does not alter the solution, since any constant can be absorbed in C) we can rewrite Eq. (4),

$$(\mathbf{R}+C\mathbf{I})\mathbf{x} = \mathbf{d}, \qquad (5)$$

where $\mathbf{R}=\{R_{ij}\}=\{R(j-i)\}=\{R|j-i|\}$ (since the autocorrelation is symmetric), $\mathbf{I}$ is $N\times N$ identity matrix, $\mathbf{x}=[x[1],\cdots,x[N]]^{\mathrm{T}}$, and $\mathbf{d}=[C\Sigma_l\eta_1^l p_1^l,\cdots,C\Sigma_l\eta_N^l p_N^l]^{\mathrm{T}}$. $[\cdot]^{\mathrm{T}}$ denotes transpose operation.

Note that if $C=0$, the solution is $x^{\star}[n]=0$, i.e., when there is no information about η_n^l and p_n^l or we do not consider any information from the training data, the solution is zero. This is because the only way by which we can minimize the energy of $y[n]$ is by feeding a zero signal at the input of the filter h. On the other hand, if $h=0$, i.e., no filter is provided or no smoothing criterion is imposed, then $x^{\star}[n]=\Sigma_l\eta_n^l p_n^l$,, i.e., it is the convex combination of the possible values of the articulatory positions learned from the training data. If $p_n^1=1$ and $p_n^l=0$, for $l>1$, the solution is $x^{\star}[n]=\eta_n^1$, the only possible value of the articulatory position. Thus, in general, the second term of the objective function [Eq. (3)] constrains the solution to be in the convex hull of η_n^l, $1\le l\le L$. It is easy to show that the second term, in turn, ensures that the acoustic feature vector corresponding to $x^{\star}[n]$ is also in the covex hull of the acoustic feature vectors corresponding to η_n^l, $1\le l\le L$ under the assumption of local linearity on the non-linear mapping between acoustic and articulatory space. Thus, the acoustic proximity between the estimated and the possible articulatory configurations is indirectly considered in our proposed optimization framework, although we do not directly consider an acoustic proximity term in the objective function unlike that in dynamic programming formulation.[18]

If both C and h are nonzero, then the solution of Eq. (5) can be found as follows:

$$\mathbf{x}^{\star} = (\mathbf{R}+C\mathbf{I})^{-1}\mathbf{d}. \qquad (6)$$

Since R is an autocorrelation matrix and hence symmetric toeplitz and since $C>0$, $(\mathbf{R}+C\mathbf{I})$ is always invertible and hence the solution of $\mathbf{x}$ always exists.

Before concluding this section, we describe the strategy to determine η_n^l and p_n^l, $l=1,\ldots,L$ from the training set.

$\mathbf{u}_n$ denotes the acoustic feature vector at the n^{th} frame of the test speech utterance. $\{(\mathbf{z}_i,\mathbf{x}_i);1\le i\le T\}$ is the pair of acoustic feature and articulatory position vector in the training set. Let $\delta_{n,i}=\|\mathbf{u}_n-\mathbf{z}_i\|$, $1\le i\le T$. At each frame n, $\delta_{n,i}$, $1\le i\le T$ are computed and sorted in an ascending order. The articulatory position vectors $\mathbf{x}_i$ in the training set corresponding to the top L sorted $\delta_{n,i}$ are denoted by $\{\eta_n^l;1\le l\le L\}$. That means $\{\eta_n^l;1\le l\le L\}$ are the L articulatory position vectors in the training set, the corresponding acoustic features of which are closest to $\mathbf{u}_n$. Let the top L sorted $\delta_{n,i}$ be denoted by $\{\delta_l;1\le l\le L\}$. Then p_n^l are computed as $p_n^l=\delta_l^{-1}/\Sigma_l\delta_l^{-1}$. This ensures that $\Sigma_l p_n^l=1$. p_n^l computed in this way implies that if the test acoustic feature vector $\mathbf{u}_n$ is closer to the training acoustic feature vector $\mathbf{z}_{l_1}$ compared to some other $\mathbf{z}_{l_2}$, then $\mathbf{x}_{l_1}$ is more likely to be the articulatory position than $\mathbf{x}_{l_2}$ at the n^{th} frame of the test utterance.

As an alternative to normalized sorted distance, we considered the Parzen window based density estimation for determining p_n^l. In this approach, a probability density function is estimated on the entire training space (joint space of $\mathbf{z}_i$ and $\mathbf{x}_i$) using the sum of Gaussian windows at each data point. The probability density values at $(\mathbf{z}_i,\mathbf{x}_i)$ corresponding to top L sorted $\delta_{n,i}$ were considered as p_n^l. However, this approach did not result in a better estimate of the articulatory positions. This could be due to the fact that the Parzen window based pdf estimation is efficient only when large number of data samples are available, particularly if the related space is high dimensional. Also, the relation between $\mathbf{z}_i$ and $\mathbf{x}_i$ is non-linear and hence the probability in the joint space might not be a good measure of p_n^l.

V. RECURSIVE SOLUTION TO THE INVERSION PROBLEM

The goal of the recursion in the inversion problem is to estimate the articulatory position at the $(N+1)^{\mathrm{th}}$ frame using the acoustic feature at $(N+1)^{\mathrm{th}}$ frame and the estimated articulatory positions up to the N^{th} frame, i.e., $x[1],\cdots,x[N]$.

Let $\mathbf{x}_N=[x[1]\cdots x[N]]^{\mathrm{T}}$ and let $\mathbf{R}_N$ be the $N\times N$ autocorrelation matrix of the filter h and we have the solution [using Eq. (6)]

$$\mathbf{x}_N = (\mathbf{R}_N+C\mathbf{I})^{-1}\mathbf{d}_N, \qquad (7)$$

where $\mathbf{d}_N$ is $N\times 1$ vector. Suppose we get $\mathbf{d}_{N+1}=\binom{\mathbf{d}_N}{d_{N+1}}$, we need to solve $\mathbf{x}_{N+1}(=(\mathbf{R}_{N+1}+C\mathbf{I})^{-1}\mathbf{d}_{N+1})$ using $\mathbf{x}_N$.

Let $\mathbf{A}_N = \mathbf{R}_N + C\mathbf{I}$. $\mathbf{A}_{N+1}$ can be partitioned as follows:

$$\mathbf{A}_{N+1} = \left(\begin{array}{ccc|c} & & & R_h[N] \\ & \mathbf{A}_N & & \vdots \\ & & & R_h[1] \\ \hline R_h[N] & \cdots & R_h[1] & R_h[0]+C \end{array}\right)$$

$$= \begin{pmatrix} \mathbf{A}_N & \mathbf{J}\mathbf{r}_N \\ \mathbf{r}_N^{\mathrm{T}} & R_h[0]+C \end{pmatrix}, \qquad (8)$$

where

$$\mathbf{r}_N = \begin{pmatrix} R_h[1] \\ \vdots \\ R_h[N] \end{pmatrix} \text{ and } \mathbf{J} = \begin{pmatrix} 0 & \cdots & 1 \\ \vdots & 1 & \vdots \\ 1 & \cdots & 0 \end{pmatrix}.$$

Using matrix partitioning,[34]

$$\mathbf{A}_{N+1}^{-1} = \begin{pmatrix} \mathbf{A}_N^{-1} & \mathbf{0} \\ \mathbf{0}^{\mathrm{T}} & 0 \end{pmatrix} + \frac{1}{P_N}\begin{pmatrix} \mathbf{b}_N \\ 1 \end{pmatrix}\begin{pmatrix} \mathbf{b}_N^{\mathrm{T}} & 1 \end{pmatrix}, \qquad (9)$$

where, $\mathbf{b}_N = -\mathbf{A}_N^{-1}\mathbf{J}\mathbf{r}_N$ and $P_N = R_h[0] + C + \mathbf{r}_N^{\mathrm{T}}\mathbf{J}\mathbf{b}_N$. So

$$\begin{aligned} \mathbf{x}_{N+1} &= \mathbf{A}_{N+1}^{-1}\mathbf{d}_{N+1} = \mathbf{A}_{N+1}^{-1}\begin{pmatrix} \mathbf{d}_N \\ d_{N+1} \end{pmatrix} \\ &= \left\{\begin{pmatrix} \mathbf{A}_N^{-1} & \mathbf{0} \\ \mathbf{0}^{\mathrm{T}} & 0 \end{pmatrix} + \frac{1}{P_N}\begin{pmatrix} \mathbf{b}_N \\ 1 \end{pmatrix}\begin{pmatrix} \mathbf{b}_N^{\mathrm{T}} & 1 \end{pmatrix}\right\}\begin{pmatrix} \mathbf{d}_N \\ d_{N+1} \end{pmatrix} \\ &= \begin{pmatrix} \mathbf{x}_N \\ 0 \end{pmatrix} + \begin{pmatrix} \mathbf{b}_N \\ 1 \end{pmatrix}\alpha_N, \end{aligned} \qquad (10)$$

where $\alpha_N = (-\mathbf{x}_N^{\mathrm{T}}\mathbf{J}\mathbf{r}_N + d_{N+1})/P_N$. So if $\mathbf{b}_N$ is known we can compute $\mathbf{x}_{N+1}$ from $\mathbf{x}_N$ without any matrix inversion. Thus we need to derive a recursion for $\mathbf{b}_N$.

Let us define

$$\begin{aligned} \mathbf{a}_N &\triangleq \mathbf{J}\mathbf{b}_N \quad (\Rightarrow \mathbf{b}_N = \mathbf{J}\mathbf{a}_N) \\ &= -\mathbf{J}\mathbf{A}_N^{-1}\mathbf{J}\mathbf{r}_N = -\mathbf{A}_N^{-1}\mathbf{r}_N. \end{aligned} \qquad (11)$$

Thus we need to compute $\mathbf{a}_{N+1}(=-\mathbf{A}_{N+1}^{-1}\mathbf{r}_{N+1})$ from $\mathbf{a}_N$.

$$\begin{aligned} \mathbf{a}_{N+1} &= -\mathbf{A}_{N+1}^{-1}\mathbf{r}_{N+1} = -\mathbf{A}_{N+1}^{-1}\begin{pmatrix} \mathbf{r}_N \\ r_{N+1} \end{pmatrix} \\ &= -\left\{\begin{pmatrix} \mathbf{A}_N^{-1} & \mathbf{0} \\ \mathbf{0}^{\mathrm{T}} & 0 \end{pmatrix} + \frac{1}{P_N}\begin{pmatrix} \mathbf{b}_N \\ 1 \end{pmatrix}\begin{pmatrix} \mathbf{b}_N^{\mathrm{T}} & 1 \end{pmatrix}\right\}\begin{pmatrix} \mathbf{r}_N \\ r_{N+1} \end{pmatrix} \\ &= \begin{pmatrix} \mathbf{a}_N \\ 0 \end{pmatrix} + \begin{pmatrix} \mathbf{b}_N \\ 1 \end{pmatrix}\gamma_N, \end{aligned} \qquad (12)$$

where $\gamma_N = -(\mathbf{a}_N^{\mathrm{T}}\mathbf{J}\mathbf{r}_N + r_{N+1})/P_N$. Thus if we know $\mathbf{a}_N$ (or $\mathbf{b}_N = \mathbf{J}\mathbf{a}_N$), we can compute $\mathbf{a}_{N+1}$ without matrix inversion. Thus, if we know $\mathbf{x}_N$, we can compute $\mathbf{x}_{N+1}$ using $\mathbf{x}_N$ using Eq. (10) and (12). No explicit matrix inversion is required in each step. The steps in the recursive solution of Eq. (3) are given below:

Step 1 (Initialization):

$n=1$, estimate η_1^l and p_1^l, $l=1,\cdots,L$ from
$\mathbf{u}_1 . d_1 = C\Sigma_l \eta_1^l p_1^l$.
$\mathbf{x}_1 = x[1] = d_1/(R_h[0]+C)$
$\mathbf{r}_1 = R_h[1]$
$\mathbf{b}_1 = \mathbf{r}_1/(R_h[0]+C)$ and $\mathbf{a}_1 = \mathbf{b}_1$
$P_1 = R_h[0] + C + \mathbf{r}_1^{\mathrm{T}}\mathbf{J}\mathbf{b}_1$
$n=2$.

Step 2 (Recursion):

Estimate η_n^l and p_n^l, $l=1,\cdots,L$ from $\mathbf{u}_n$. $d_n = C\Sigma_l \eta_n^l p_n^l$
$\gamma_{n-1} = -(\mathbf{a}_{n-1}^{\mathrm{T}}\mathbf{J}\mathbf{r}_{n-1} + R_h[n])/P_{n-1}$
$\alpha_{n-1} = (-\mathbf{x}_{n-1}^{\mathrm{T}}\mathbf{J}\mathbf{r}_{n-1} + d_n)/P_{n-1}$ and $\mathbf{x}_n = \begin{pmatrix} \mathbf{x}_{n-1} \\ 0 \end{pmatrix} + \begin{pmatrix} \mathbf{b}_{n-1} \\ 1 \end{pmatrix}\alpha_{n-1}$
$\mathbf{r}_n = \begin{pmatrix} \mathbf{r}_{n-1} \\ R_h[n] \end{pmatrix}$
$\mathbf{a}_n = \begin{pmatrix} \mathbf{a}_{n-1} \\ 0 \end{pmatrix} + \begin{pmatrix} \mathbf{b}_{n-1} \\ 1 \end{pmatrix}\gamma_{n-1}$
$\mathbf{b}_n = \mathbf{J}\mathbf{a}_n$ and $P_n = R_h[0] + C + \mathbf{r}_n^{\mathrm{T}}\mathbf{J}\mathbf{b}_n$.

Step 3:

Increment n to $n+1$ and go to Step 2.

VI. SELECTION OF ACOUSTIC FEATURES FOR THE INVERSION PROBLEM

Appropriate acoustic feature selection is crucial for the inversion problem because, in every analysis frame, the acoustic feature is used to determine possible articulatory positions from the training set. In turn, from these possible positions the smoothness criterion estimates the best position so that the articulator trajectory is as smooth as possible for a given h. The possible articulatory positions at every test frame are chosen such that the corresponding acoustic vectors in the training set are in the neighborhood of the acoustic vector of the test frame (as discussed in Section IV). The more the correlation or dependency between the acoustic feature and the corresponding articulatory position, the more accurate are the possible articulatory positions. Therefore, quantifying the dependency between the acoustic feature and the articulatory position is essential to compare different acoustic features and select the best one for the inversion problem.

We compute the statistical dependency between the acoustic feature and the articulatory position by mutual information (MI). Let Z denote the acoustic feature vector and X be the vector whose elements are the position values of all articulators at every frame. Since there are 7 articulators each with x and y coordinates in our experimental data, X has 14 dimensions; the dimension of Z depends on the chosen acoustic feature. For acoustic features, we consider mel frequency cepstral coefficients (MFCCs), linear prediction coefficients (LPCs), cepstral representation of LPC (LPCC), and variants of LPC, i.e., line spectral frequency (LSF), reflection coefficient (RC), log area ratio (LAR). Each of these features were computed every 10 ms to match the rate of

TABLE III. Mutual information between various acoustic features and articulatory position.

	$I(Q(Z),Q(X))$	
Z	Male	Female
MFCC	**1.8179**	**1.8594**
LPC	1.3394	1.3931
LPCC	1.3339	1.4936
LSF	1.7025	1.6080
RC	1.6148	1.5309
LAR	1.6921	1.5834

articulatory position data. Speech signal is pre-emphasized and windowed using 20 ms hamming window before computing frame-based features. For MFCC, Z is a 13 dimensional vector. LPCs were computed using an order of 12; thus, Z for LPC and LPCC is 13 dimensional, but for LSF, RC, and LAR Z is 12 dimensional. In this paper, we only consider static features; no dynamic features have been used.

Since the probability density functions of Z and X are not directly known, we consider MI estimation by quantization of the space of Z and X from the training data set with a finite number of quantization bins, then estimating the joint distribution of Z and X in the newly quantized finite alphabet space using standard maximum likelihood criterion—frequency counts;[35] and finally applying the discrete version of the MI.[36] More precisely, let us denote the pair of acoustic feature and articulatory position vectors in the training set by $\{(\mathbf{z}_i,\mathbf{x}_i);i=1,\cdots,T\}$, where $\mathbf{z}_i$ and $\mathbf{x}_i$ take values in $\mathcal{R}^{K_1}$ and $\mathcal{R}^{K_2}$. The quantizations of these spaces are denoted by $Q(Z)$: $\mathcal{R}^{K_1}\rightarrow\mathcal{A}_z$ and $Q(X)$: $\mathcal{R}^{K_2}\rightarrow\mathcal{A}_x$, where $|\mathcal{A}_z|<\infty$ and $|\mathcal{A}_x|<\infty$. Then the MI is given by:

$$I(Q(Z),Q(X)) = \sum_{q_z\in\mathcal{A}_z,q_x\in\mathcal{A}_x} P(Q(Z)=q_z,Q(X)=q_x) \cdot \log\frac{P(Q(Z)=q_z,Q(X)=q_x)}{P(Q(Z)=q_z)P(Q(X)=q_x)}. \tag{13}$$

It is well known that $I(Q(Z),Q(X))\leq I(Z,X)$, because quantization reduces the level of dependency between random variables. On the other hand, increasing the resolution of $Q(\cdot)$, implies that $I(Q(Z),Q(X))$ converges to $I(Z,X)$ as the number of bins tends to infinity.[37] However, this result assumes that we know the joint distribution, which implies having an infinite amount of training data and a consistent learning approach. Consequently, for the finite training data scenario there is a tradeoff between how precisely we want to estimate $I(Q(Z),Q(X))$, versus how close we want to be to the analytical upper bound $I(Z,X)$. We decided to have a resolution of $Q(\cdot)$ that guarantees good estimation of the joint distribution, and consequently a precise lower bound estimation for $I(Z,X)$. K-means vector quantization was used to characterize the quantization mapping.[36,35]

For each acoustic feature vector and the articulatory position vector, K-means vector quantization with 512 prototypes was used, i.e., $|\mathcal{A}_z|=|\mathcal{A}_x|=512$. Table III shows the mutual information between various acoustic features and articulatory positions for both the male and female speaker. It can be observed that the mutual information between MFCC and articulatory position is maximum among all other acoustic features for the data of both speakers. LSF has the second highest MI with articulatory position, and the least MI occurs for LPC. It should be noted that change in the number of prototypes in K-means does not alter the relative value of MI for different acoustic features. For example, we computed MI using $|\mathcal{A}_z|=|\mathcal{A}_x|=64$, 128, 256, 1024 and we found MFCC to have maximum MI with articulatory position in all cases. This is consistent for both speakers. It is interesting to note that Qin *et al.*[38] also achieved maximum correlation between original and estimated articulator trajectories by using MFCC features. Based on this observation, we use MFCC as the acoustic feature for all of the following experiments.

VII. EXPERIMENTS AND RESULTS

The acoustic-to-articulatory inversion experiments are performed separately for the male and female speaker data. The accuracy of inverse mapping is evaluated separately on the test set for both speakers in terms of both root mean squared (RMS) error and correlation between actual articulatory position in the test set $x_r[n]$ and the position estimated by inverse mapping $x^\star[n]$. The RMS error $\mathcal{E}$ reflects the average closeness between $x_r[n]$ and $x^\star[n]$. The correlation ρ indicates how similar the actual and estimated articulator trajectories are. A minimum $\mathcal{E}$ does not always mean that the trajectories are similar since the estimated one can be very jagged although it might be close to the actual one. Jagged trajectories are physically less likely during speech production since articulators cannot move in such a way in real life. Such jagged trajectories can be identified by poor ρ values. We use Pearson correlation ρ between the actual and estimated trajectory for each utterance, where

$$\rho = \frac{N\sum_n x_r[n]x^\star[n] - \sum_n x_r[n]\sum_n x^\star[n]}{\sqrt{N\sum_n (x_r[n])^2 - \left(\sum_n x_r[n]\right)^2}\sqrt{N\sum_n (x^\star[n])^2 - \left(\sum_n x^\star[n]\right)^2}}. \tag{14}$$

The development set is used to tune the cut-off frequency γ_c of filter h and the trade-off parameter C. For our experiment we considered $L=200$. Increasing L further did not improve the result.

We considered an IIR high pass filter with cut-off frequency γ_c, and stop-band ripple 40 dB down compared to the pass-band ripple. A rational transfer function having order 5 for both numerator and denominator polynomials is constructed for the desired specification. The MATLAB function cheby2 is used for this purpose. We choose an IIR filter so that the roll-off of the high-pass filter is large and hence the filter becomes close to the articulator specific ideal high-pass filter. We chose γ_c and C from a set of values, which yield the best performance on the development set. From Section III, we observe that most of the energy of the spectrum of the articulator trajectories is below 9–10 Hz; hence, we consider the set of values for γ_c to be $\{\gamma_c\}=\{1.5+((k-1)/19)(7.5);k=1,\cdots,20\}$, i.e., the set of values is 20 equally spaced points between 1.5 Hz and 9 Hz. Similarly the set of values for C was chosen to be {0.001, 0.005, 0.01, 0.05, 0.1, 0.5, 1, 5, 10, 50, 100}. The values of C were chosen to have a wide range of orders. For every γ_c and C combination, Eq. (6) was solved recursively using Eq. (10) and (12) for each utterance of the development set. As a metric of performance of the inverse mapping, we measure $\mathcal{E}$ between the actual value of the articulatory positions and the estimated positions.

γ_c and C, for which the minimum value of the averaged $\mathcal{E}$ (averaged over all utterances of the dev set) was obtained, are shown in Table IV for each articulator and for both speakers. We can see that the velum has a slightly higher γ_c compared to other articulators to achieve the least $\mathcal{E}$. The values of the best C for different articulatory positions do not differ in its order much.

To estimate the position of a particular articulator from the acoustics in the test set, we use the corresponding γ_c and C optimized on the dev set. For each utterance in the test set, the articulatory positions from the acoustic signal are estimated by solving Eq. (6) recursively as outlined in Section V As a baseline, we estimated the articulatory positions using a fixed filter $h=[1\ \ -1]$ with $\gamma_c(=25\ \text{Hz}=F_s/4)$; C is optimized on the dev set. The purpose of choosing such a baseline is to investigate the change in performance when articulator specific γ_c are used compared to a fixed γ_c.

We also implemented the dynamic programming (DP) based inversion mapping with a cost function similar to that outlined in the work by Richards *et al.*[31] The cost function, which is minimized, is as follows:

$$D=\sum_{n=1}^{N} K\|\mathbf{u}_n-\mathbf{z}_n\|^2+\|\mathbf{x}_n-\mathbf{x}_{n-1}\|^2. \qquad (15)$$

At each frame n, the possible articulatory positions were η_n^l, $1\le l\le L$, through which the best path was found. $\mathbf{z}_n$ are chosen from the acoustic feature vectors in the training set corresponding to η_n^l, $1\le l\le L$. K was optimized on the dev set to achieve least average $\mathcal{E}$. The solution of the DP based inversion is low-pass filtered following the work by Toda *et al.*[39] The cut-off frequencies the low-pass filters for post-processing are chosen to be the ones given by Toda *et al.*[39]

The cost in dynamic programming D [Eq. (15)] is different from the cost function in our proposed approach [Eq. (3)]. Thus, they are not directly comparable in terms of their cost functions. The motivation for selecting DP followed by low-pass filtering as a part of our experiment is to analyze the quality of the estimated articulatory positions using the proposed generalized smoothness approach with respect to the positions obtained by the well-established DP approach with smoothing as a post-processing.

14 trajectories corresponding to 14 different articulatory positions are randomly picked from the test set, and their estimates using both the proposed approach and the DP approach are shown in Fig. 1 overlaid on the actual position. It can be seen that the estimated trajectories are smooth and, on average, they follow the actual trajectories. The closeness of the estimated trajectory to the actual one depends on the corresponding $\{\eta_n^l;1\le l\le L\}$ and $\{p_n^l;1\le l\le L\}$. The trajectories estimated using the DP approach are also very close to the actual one. For the examples chosen in Fig. 1, trajectories estimated by the proposed approach and DP appear similar. For clarity, we have not shown the trajectories estimated by our proposed approach with a fixed γ_c. We evaluate the performance of different approaches through error analysis over the entire test set.

For a comprehensive error analysis, we computed the $\mathcal{E}$ and ρ for all utterances in the test set. The mean $\mathcal{E}$ and ρ (with their SD) between the actual trajectories and the estimated trajectories by inverse mapping using generalized smoothness criterion (for both fixed γ_c and articulator specific γ_c) and the DP (followed by low-pass filtering) approach are tabulated in Tables V and VI for the female and male speaker, respectively. The tables also show the range of the position values for each articulator so that the quality of the inverse mapping can be understood from the mean $\mathcal{E}$.

From Tables V and VI, it can be observed that the averaged $\mathcal{E}$ values obtained by generalized smoothness criterion are of the order of 10% of the range of the corresponding

TABLE IV. Best choices of γ_c and C for all articulatory positions optimized on dev set.

	Best choices of γ_c (Hz) and C			
	Female speaker		Male speaker	
Articulator	γ_c	C	γ_c	C
ul_x	3.07	0.50	3.47	0.10
ll_x	4.26	0.10	4.26	0.10
li_x	3.47	0.10	4.65	0.10
tt_x	3.47	0.10	3.86	0.10
tb_x	3.86	0.05	3.86	0.10
td_x	3.86	0.10	3.07	0.50
v_x	5.05	0.50	5.05	0.50
ul_y	4.26	0.50	4.26	0.50
ll_y	4.65	0.10	5.84	0.10
li_y	3.07	1.00	5.05	0.50
tt_y	4.65	0.10	5.05	0.10
tb_y	3.07	0.50	4.65	0.10
td_y	3.07	0.50	4.26	0.10
v_y	6.23	0.50	5.44	0.50

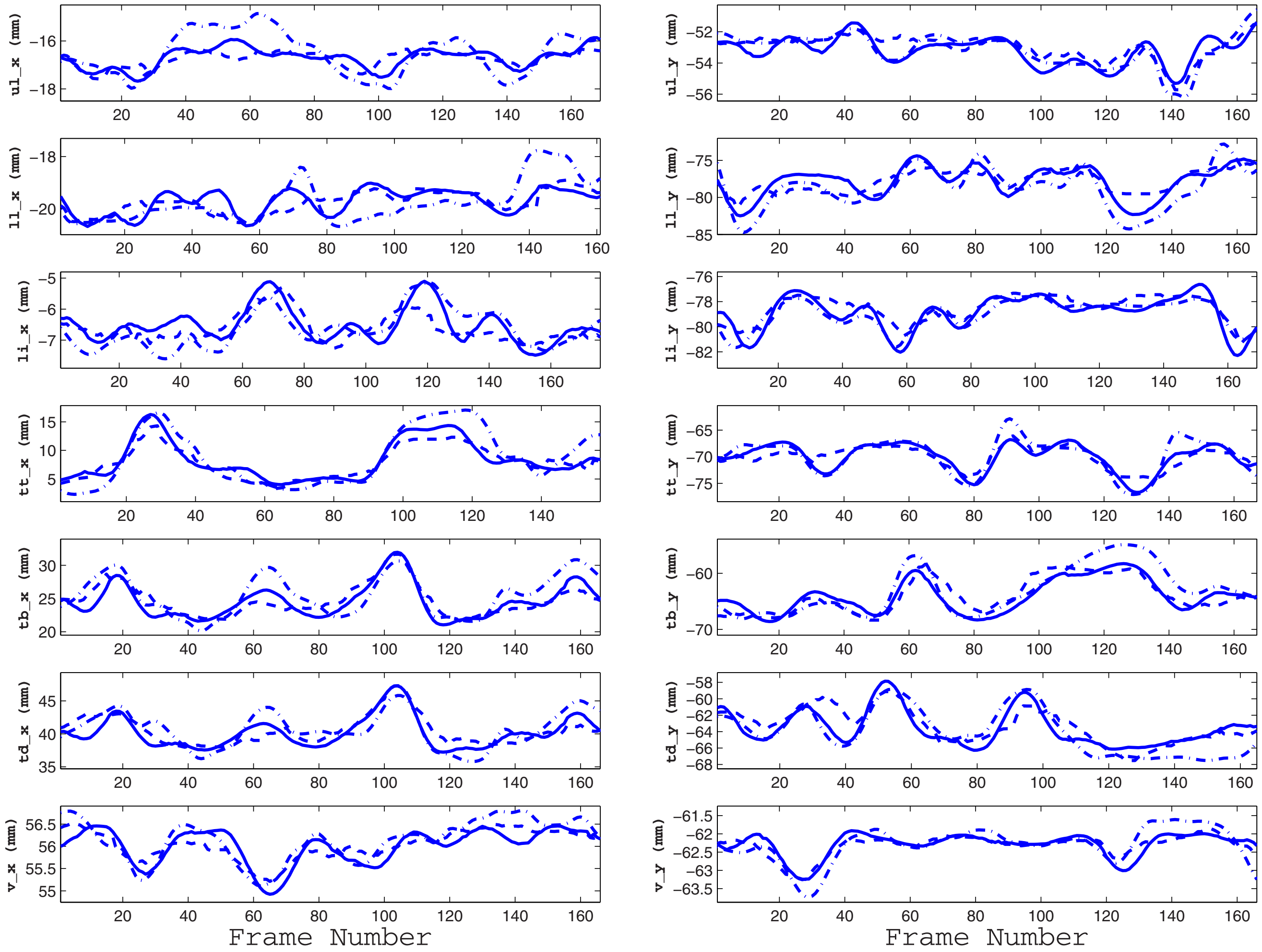

FIG. 1. (Color online) Illustrative example of inverse mapping: randomly chosen examples of the test articulator trajectory (dash-dotted) and the corresponding estimated trajectory for 14 articulatory positions using generalized smoothness criterion (solid line) and dynamic programming (DP) approach (dashed line).

TABLE V. Accuracy of inversion in terms of RMS error $\mathcal{E}$ and correlation ρ (Female speaker).

		Mean (SD) of $\mathcal{E}$ (mm) and ρ					
		Generalized smoothness					
	Range	Artic. specific γ_c		Fixed γ_c		DP low-pass filtered	
Articulators	(mm)	$\mathcal{E}$	ρ	$\mathcal{E}$	ρ	$\mathcal{E}$	ρ
ul_x	6.64	0.82(0.21)	0.58(0.15)	0.83(0.20)	0.52(0.15)	0.85(0.21)	0.51(0.22)
ll_x	10.71	1.27(0.34)	0.53(0.13)	1.30(0.33)	0.46(0.12)	1.38(0.35)	0.35(0.24)
li_x	7.22	0.75(0.18)	0.57(0.15)	0.77(0.17)	0.52(0.13)	0.84(0.20)	0.39(0.25)
tt_x	23.48	2.39(0.41)	0.76(0.10)	2.54(0.40)	0.70(0.10)	2.60(0.49)	0.69(0.16)
tb_x	26.19	2.24(0.41)	0.76(0.08)	2.35(0.40)	0.72(0.08)	2.41(0.49)	0.72(0.12)
td_x	24.40	1.95(0.39)	0.74(0.10)	2.04(0.39)	0.71(0.10)	2.15(0.45)	0.69(0.14)
v_x	5.31	0.33(0.08)	0.73(0.10)	0.34(0.08)	0.70(0.10)	0.34(0.09)	0.70(0.13)
ul_y	8.91	1.23(0.22)	0.58(0.18)	1.26(0.22)	0.54(0.18)	1.31(0.31)	0.49(0.25)
ll_y	29.23	2.78(0.61)	0.79(0.05)	2.87(0.60)	0.75(0.06)	3.27(0.66)	0.67(0.16)
li_y	13.26	1.23(0.28)	0.80(0.08)	1.27(0.27)	0.78(0.08)	1.41(0.31)	0.76(0.12)
tt_y	23.38	2.46(0.44)	0.78(0.08)	2.60(0.42)	0.75(0.08)	2.70(0.44)	0.75(0.11)
tb_y	20.85	2.38(0.47)	0.78(0.07)	2.49(0.45)	0.74(0.08)	2.64(0.52)	0.72(0.13)
td_y	18.61	2.38(0.47)	0.69(0.09)	2.45(0.44)	0.64(0.09)	2.64(0.53)	0.57(0.16)
v_y	4.61	0.36(0.10)	0.78(0.07)	0.37(0.10)	0.75(0.09)	0.40(0.11)	0.74(0.11)

TABLE VI. Accuracy of inversion in terms of RMS error $\mathcal{E}$ and correlation ρ (Male speaker).

		Mean (SD) of $\mathcal{E}$ (mm) and ρ					
		Generalized smoothness					
	Range	Artic. specific γ_c		Fixed γ_c		DP low-pass filtered	
Articulators	(mm)	$\mathcal{E}$	ρ	$\mathcal{E}$	ρ	$\mathcal{E}$	ρ
ul_x	7.48	0.76(0.19)	0.45(0.17)	0.78(0.19)	0.37(0.15)	0.80(0.19)	0.29(0.33)
ll_x	10.71	1.15(0.24)	0.70(0.12)	1.19(0.23)	0.64(0.11)	1.34(0.27)	0.51(0.20)
li_x	8.44	0.59(0.12)	0.63(0.13)	0.60(0.12)	0.58(0.12)	0.64(0.13)	0.53(0.21)
tt_x	25.55	2.41(0.64)	0.73(0.14)	2.54(0.63)	0.66(0.13)	2.79(0.66)	0.58(0.22)
tb_x	26.19	2.39(0.57)	0.69(0.13)	2.48(0.56)	0.62(0.12)	2.68(0.63)	0.51(0.22)
td_x	24.40	2.20(0.51)	0.67(0.14)	2.26(0.50)	0.62(0.13)	2.42(0.51)	0.57(0.22)
v_x	5.64	0.79(0.23)	0.60(0.17)	0.81(0.23)	0.55(0.16)	0.87(0.24)	0.40(0.30)
ul_y	9.74	1.20(0.21)	0.65(0.11)	1.23(0.20)	0.59(0.11)	1.34(0.28)	0.49(0.24)
ll_y	29.23	1.92(0.35)	0.81(0.08)	2.02(0.35)	0.76(0.08)	2.36(0.43)	0.67(0.14)
li_y	15.01	1.02(0.23)	0.73(0.08)	1.05(0.22)	0.70(0.08)	1.13(0.25)	0.68(0.15)
tt_y	25.23	3.08(0.70)	0.77(0.08)	3.23(0.68)	0.72(0.08)	3.50(0.63)	0.69(0.14)
tb_y	22.24	2.32(0.43)	0.78(0.08)	2.43(0.42)	0.73(0.09)	2.63(0.48)	0.71(0.14)
td_y	19.30	2.38(0.51)	0.71(0.11)	2.45(0.51)	0.66(0.11)	2.72(0.51)	0.59(0.20)
v_y	4.71	0.80(0.18)	0.56(0.15)	0.81(0.18)	0.51(0.14)	0.85(0.20)	0.46(0.22)

articulator. Consistent higher values of ρ in the case of articulator specific γ_c compared to fixed γ_c indicates that the estimated articulatory trajectories are more similar to the actual ones when they are smoothed in an articulator specific fashion. Similarly, lower values of $\mathcal{E}$ demonstrate that, on average, the generalized smoothness criterion indeed improves the inverse mapping accuracy compared to a fixed smoothing. The mean $\mathcal{E}$ and the mean ρ obtained by the DP (followed by low-pass filtering) approach also have a similar order for most of the articulators. Note that the solution of DP is optimal according to the DP cost function but once the solution is low-pass filtered it is no longer necessarily optimal and furthermore it is, in general, difficult to establish what cost function the low-pass filtered trajectory might be optimal to, if at all it is. In contrast, our proposed optimization results in an optimal solution as per the objective function [Eq. (3)] for any arbitrary filter. The use of higher order articulator specific smoothing filters in the DP cost function [Eq. (15)] can further improve the accuracy of the estimated articulatory positions but the complexity order increases exponentially with the length of the filter. DP has a complexity order of $L^K N$, where L is the number of possible articulatory positions in each frame, K is the length of the impulse response of the filter and N is the number of frames. Even for our experiment where we choose $L=200$, choice of an FIR filter h of length 5 makes the complexity order 3.2×10^{11} N. Hence, we have not reported any results of applying DP when a higher order smoothness filter is used in Eq. (15). In contrast, the order complexity of the proposed optimization scheme does not change with the filter type.

VIII. CONCLUSIONS

The generalized smoothness criterion proposed in this paper can be useful to estimate any smooth trajectory beyond the articulator trajectory. As long as the mapping between two spaces under consideration can be locally linearly approximated, the smoothness criterion will find the best possible smooth trajectory, using the knowledge about the possible solutions $\{\eta_n^l; 1 \leq l \leq L\}$. The flexibility in choosing the filter h for the smoothness criterion is advantageous since it provides a good way to analyze various degrees of smoothness requirement for the trajectory to be estimated. Note that, in the DP approach of articulatory inversion, an acoustic proximity term $\|\mathbf{u}_n - \mathbf{z}_n\|^2$ is directly considered in the optimization; this is indirectly performed in our proposed optimization by choosing the candidate articulators based on the acoustic proximity.

The recursive version of the solution of the articulator trajectory estimate is a key feature of the formulation presented in this work. Recursive algorithms are very useful for online processing and suitable for speech applications that need an estimate of articulators on-the-fly.

We observed that the correlation between the original trajectory and the estimated trajectory using generalized smoothness criterion is better than that obtained with a fixed smoothing filter, indicating the effectiveness of using the articulator specific smoothing filter. It should be noted that for each frame of the test utterance, the DP (without any post-processing) approach selects the best possible articulator position from what were seen in the training set, while the proposed technique does not. Rather, it provides a real valued solution that best fits the smoothness criterion and data consistency. In this work, we analyzed the smoothness of articulators in a speaker specific manner; a study on smoothness over a large set of speakers can be performed to obtain a generic smoothness parameter for each articulator. We estimate each articulator in an independent fashion and do not use their correlation explicitly although the candidate positions of different articulators from training data have correlations between themselves. The correlation between different articulators can be utilized to appropriately extend the proposed optimization for estimating more realistic articulator trajectories.

ACKNOWLEDGMENTS

This work was supported by funds from NIH, NSF, and ONR.

[1] S. Maeda, "Un modele articulatoire de la langue avec des composantes lineaires (An articulatory model of the tongue with linear components)," Actes 10emes Journees d'Etude sur la Parole (Grenoble, France) (1979), pp. 152–162.

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

[3] J. L. Kelly and C. C. Lochbaum, "Speech synthesis," in Proceedings of the 4th International Congress on Acoustics, Copenhagen (1962), pp. 1–4.

[4] C. P. Browman and L. Goldstein, "Towards an articulatory phonology," in *Phonology Yearbook 2* edited by C. Ewen and J. Anderson, Cambridge University Press, Cambridge, 1986) pp. 219–252.

[5] C. P. Browman and L. Goldstein, "Articulatory gestures as phonological units," Phonology **6**, 201–251 (1989).

[6] C. P. Browman and L. Goldstein, "Gestural specification using dynamically-defined articulatory structures," J. Phonetics **18**, 299–320 (1990).

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

[8] I. Zlokarnik, "Adding articulatory features to acoustic features for automatic speech recognition," J. Acoust. Soc. Am. **97**, 3246 (1995).

[9] A. A. Wrench and K. Richmond, "Continuous speech recognition using articulatory data," in Proceedings of the ICSLP, Beijing, China, 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," in Proceedings of the ICSLP, Beijing, China (2000), Vol. **4**, pp. 254–257.

[11] K. Kirchhoff, "Robust speech recognition using articulatory information," Ph.D. thesis, University of Bielefeld, Bielefeld, Germany (1999).

[12] X. Zhuang, H. Nam, M. Hasegawa-Johnson, L. Goldstein, and E. Saltzman, "The entropy of the articulatory phonological code: Recognizing gestures from tract variables," in Proceedings of the Interspeech, Brisbane, Australia (2008), pp. 1489–1492.

[13] C. Qin and M. A. Carreira-Perpinan, "An empirical investigation of the nonuniqueness in the acoustic-to-articulatory mapping," in Proceedings of the Interspeech (2007).

[14] B. S. Atal, J. J. Chang, M. V. Mathews, and J. W. Tukey, "Inversion of articulatory-to-acoustic transformation in the vocal tract by a computer-sorting technique," J. Acoust. Soc. Am. **63**, 1535–1555 (1978).

[15] A. Toutios and K. Margaritis, "Acoustic-to-articulatory inversion of speech: A review," in Proceedings of the International 12th TAINN (2003).

[16] V. Morozov, *Regularization Methods for Ill-Posed Problem* (CRC, Boca Raton, FL, 1993).

[17] V. N. Sorokin, A. Leonov, and A. V. Trushkin, "Estimation of stability and accuracy of inverse problem solution for the vocal tract," Speech Commun. **30**, 55–74 (2000).

[18] J. Schroeter and M. M. Sondhi, "Dynamic programming search of articulatory code-books," in Proceedings of the ICASSP, Glasgow, UK (1989), Vol. **1**, pp. 588–591.

[19] M. G. Rahim, W. B. Kleijn, J. Schroeter, and C. C. Goodyear, "Acoustic-to-articulatory parameter mapping using an assembly of neural networks," in Proceedings of the ICASSP (1991), pp. 485–488.

[20] J. Hogden, A. Lofqvist, V. Gracco, I. Zlokarnik, P. Rubin, and E. Saltzman, "Accurate recovery of articulator positions from acoustics: New conclusions based on human data," J. Acoust. Soc. Am. **100**, 1819–1834 (1996).

[21] T. Toda, A. Black, and K. Tokuda, "Acoustic-to-articulatory inversion mapping with Gaussian mixture model," in Proceedings of the ICSLP, Jeju Island, Korea (2004), pp. 1129–1132.

[22] K. Richmond, "A trajectory mixture density network for the acoustic-articulatory inversion mapping," in Proceedings of the ICSLP, Pittsburgh (2006), pp. 577–580.

[23] K. Shirai and M. Honda, "Estimation of articulatory motion," *Dynamic Aspects of Speech Production* (Tokyo University Press, Tokyo, 1976), pp. 279–302.

[24] R. Wilhelms, P. Meyer, and H. W. Strube, "Estimation of articulatory trajectory by Kalman filter," in *Signal Processing III: Theories and Application*, edited by I. T. Young (Elsevier Science Ltd., 1986), pp. 477–480.

[25] G. Ramsay and L. Deng, "Maximum-likelihood estimation for articulatory speech recognition using a stochastic target mode," in Proceedings of the Eurospeech (1995), pp. 1401–1404.

[26] S. Dusan and L. Deng, "Acoustic-to-articulatory inversion using dynamic and phonological constraint," in The 5th Speech Production Seminar, Munich, Germany (2000), pp. 237–240.

[27] K. Richmond, "Estimating articulatory parameters from the acoustic speech signal," Ph.D. thesis, Edinburgh University, Edinburgh (2002).

[28] K. Richmond, "Mixture density networks, human articulatory data and acoustic-to-articulatory inversion of continuous speech," in Proceedings Workshop on Innovation in Speech Processing WISP (2001).

[29] S. Chennoukh, D. Sinder, G. Richard, and J. Flanagan, "Voice mimic system using an articulatory codebook for estimation of vocal tract shape," in Proceedings of the Eurospeech, Rhodes, Greece (1997), pp. 429–432.

[30] R. Kuc, F. Tutuer, and J. R. Vaisnys, "Determining vocal tract shape by applying dynamic constraints," in Proceedings of the ICASSP, Tampa, FL (1985), pp. 1101–1104.

[31] H. B. Richards, J. S. Mason, M. J. Hunt, and J. S. Bridle, "Deriving articulatory representations from speech with various excitation modes," in Proceedings of the ICSLP, Philadelphia, PA (1996), pp. 1233–1236.

[32] A. Lammert, D. P. W. Ellis, and P. Divenyi, "Data-driven articulatory inversion incorporating articulator priors," in ISCA Tutorial and Research Workshop on Statistical And Perceptual Audition, SAPA, Brisbane, Australia, (2008).

[33] E. Müller and G. MacLeod, "Perioral biomechanics and its relation to labial motor control," J. Acoust. Soc. Am. **71**, S33 (1982).

[34] D. G. Manolakis, V. K. Ingle, and S. M. Kogon, *Statistical and Adaptive Signal Processing: Spectral Estimation, Signal Modeling, Adaptive Filtering and Array Processing* (Artech House, Boston, 2005).

[35] R. O. Duda and P. E. Hart, *Pattern Classification and Scene Analysis* (Wiley, New York, 1983).

[36] T. M. Cover and J. A. Thomas, *Elements of Information Theory* (Wiley Interscience, New York, 1991).

[37] G. A. Darbellay and I. Vajda, "Estimation of the information by an adaptive partition of the observation space," IEEE Trans. Inf. Theory **45**, 1315–1321 (1999).

[38] C. Qin and M. A. Carreira-Perpinan, "A comparison of acoustic features for articulatory inversion," in Proceedings of the Interspeech (2007), pp. 2469–2472.

[39] T. Toda, A. Black, and K. Tokuda, "Statistical mapping between articulatory movements and acoustic spectrum using a Gaussian mixture model," Speech Commun. **50**, 215–227 (2008).