Mapping between the Speech Signal and Articulatory Trajectories

Asterios Toutios, Konstantinos Margaritis

Abstract— **The main target of this paper is a comparison of parametric representations for the task of mapping the speech signal onto Electromagnetic Articulography trajectories. The principal method used for the mapping is Support Vector Regression, contrasting previous works that applied Neural Networks to the same speech inversion problem.**

I. Introduction

The acoustic-to-articulatory mapping [1], [2], also termed acoustic-to-articulatory inversion of speech or speech inversion, is a special speech processing related problem that has attracted the attention of several researchers for many years now. It refers to the estimation of articulatory (speech production related) information using solely the speech signal as an input source. A succesful solution could find numerous applications, such as helping individuals with speech and hearing disorders by providing visual feedback, very low bit-rate speech coding and the possibility of improved automatic speech recognition.

In the past, the articulatory features used in such a context were mostly inferred by the corresponding acoustic data using vocal-tract models, synthesis models, or linguisting rules. But recent technologies have made it possible to record actual articulator movements in parallel with speech acoustics in a minimally invasive way. This "real" human data is arguably preferable to older techniques, where additional complications may be imposed by intrinsic flaws of the models themselves.

One of the forementioned technologies is the Electromagnetic Midsagittal Articulography (EMMA) or Electromagnetic Articulography (EMA). Roughly speaking, for the acquisition of EMA data, sensor coils are attached to the human subject, on specific places on the lips, the teeth, the jaw, and the soft palate (velum). Then the human subject wears a special helmet which produces an alternating magnetic field that records the position of the coils at end points of small fixed-size time intervals. The outcomes are trajectories that illustrate the movement of the coils. Usually, there are two trajectories for each coil, one for the movement in the front-back direction of the head, and one for the movement in the top-bottom direction.

For our general work setup, we follow in a large extend Richmond's work [1], [3], who proposed a quite succesful mapping of the speech signal onto EMA data, using Neural Networks. We study an alternative –Machine Learning– approach using Support Vector Regression, a more recent and very promising method. We use the same dataset as Richmond, namely the fsew0 speaker data from the MOCHA dabase.

In this paper we compare the performance of three different parametric represenattions of the speech signal for our task, namely MFCCs, PLPs, and a combination of them.

The authors are with the Parallel and Distributed Processing Laboratory, Department of Applied Informatics, University of Macedonia, 156 Egnatia Str., P.O. Box 1591, 54006, Thessaloniki, Greece. E-mail: {*toutios*, *kmarg*}*@uom.gr*.

II. The MOCHA Database

The MOCHA (Multi-Channel Articulatory) [4] database is evolving in a purpose built studio at the Edinburgh Speech Production Facility at Queen Margaret University College.

During speech, four data streams are recorded concurrently straight to a computer: the acoustic waveform, sampled at 16kHz with 16 bit precision, together with laryngograph, electropalatograph and electromagnetic articulograph data. The articulatory channels include EMA sensors directly attached to the upper and lower lips, lower incisor (jaw), tongue tip (5-10mm from the tip), tongue blade (approximately 2-3cm posterior to the tongue tip sensor), tongue dorsum (approximately 2-3cm posterior from the tongue blade sensor), and soft palate. Two channels for every sensor are recorded at 500Hz: the positioning on the x-axis (front-back direction) and on the y-axis (top-bottom direction).

The speakers are recorded reading a set of 460 British TIMIT sentences. These short sentences are designed to provide phonetically diverse material and capture with good coverage the connected speech processes in English. All waveforms are labelled at the phonemic level.

The final release of the MOCHA database will feature up to 40 speakers with a variety of regional accents. At the time of writing this paper three speakers are available. For the experiments described herein, the acoustic waveform and EMA data, as well as the phonemic labels for the speaker fsew0, a female speaker with a Soutern English accent, are used.

III. Support Vector Regression

The ϵ-SVR algorithm [5] is a generalization of the better known Support Vector Classification algorithm [6] to the regression case. Given n training vectors $\mathbf{x_i}$ and a vector $y \in R^n$ such that $y_i \in R$, we want to find an estimate for the function $y = f(\mathbf{x})$ which is optimal from a Structural Risk Minimization viewpoint. According to ϵ-SVR, this estimate is:

$$f(\mathbf{x}) = \sum_{\mathbf{i=1}}^{\mathbf{n}} (\mathbf{a_i^*} - \mathbf{a_i}) \mathbf{k}(\mathbf{x_i}, \mathbf{x}) + \mathbf{b}, \qquad (1)$$

where b is a bias term and $k(\mathbf{x_i x_j})$ is a special function called the *kernel*. The coefficients a_i and a_i^* are the solution of the quadratic problem

$$\begin{aligned} &\text{maximize} \\ &W(\mathbf{a}, \mathbf{a}^*) = -\epsilon \sum_{\mathbf{i=1}}^{\mathbf{n}} (\mathbf{a_i^*} + \mathbf{a_i}) + \sum_{\mathbf{i=1}}^{\mathbf{n}} (\mathbf{a_i^*} - \mathbf{a_i}) \mathbf{y_i} \\ &-\frac{1}{2} \sum_{i,j=1}^{n} (a_i^* - a_i)(a_j^* - a_j) k(\mathbf{x_i x_j}) \\ &\text{subject to} \\ &0 \leq a_i, a_i^* \leq C, i = 1, \ldots, n, \\ &\sum_{i=1}^{n} (a_i^* - a_i) = 0, \end{aligned} \tag{2}$$

where $C > 0$ and $\epsilon \geq 0$ are parameters chosen by the user. The "penalty parameter" C may be as high as infinity, while usual values for ϵ are 0.1 or 0.01.

The kernel function serves to convert the data into a higher-dimensional space in order to account for non-linearities in the estimate function. A commonly used kernel is the Radial Basis Function (RBF) kernel:

$$k(\mathbf{x}, \mathbf{y}) = \exp(-\gamma \parallel \mathbf{x} - \mathbf{y} \parallel^{\mathbf{2}}), \tag{3}$$

where the γ parameter is selected by the user.

IV. Data Processing

The MOCHA database includes 460 utterances of the fsew0 speaker. In order to render these data into input-output pairs suitable for function estimation, we process them as follows.

First, based on the label files we omit silent parts from the beginning and end of the utterances. During silent stretches the articulators can possibly take any configuration, something that could pose serious difficulties to our task.

Next, using HTK [7], we split the signal into overlapping frames with a duration of 16ms (256 points) and a shift of 5ms. For each frame we calculate the log-energy of the signal, the 12 first Mel Frequency Cepstral Coefficients [8] (with 30 filterbanks) and the 12 first Perceptual Linear Predictive Coefficients [9].

We account for three diffferent representations for our experimental setup. In the first one (MFCC case) the representation for every acoustic frame consists of the 12 MFCCs plus the log-energy, in the second one (PLP case) it consists of the 12 PLPs plus the log-energy and in the third one (PLP+MFCC case) it consists of the MFCCs plus the PLPs plus the log-energy.

In order to account for the dynamic properties of the speech signal and cope with the temporal extent of our problem, we just use the commonplace in the speech processing field *spatial metaphor for time*. That is, we construct input vectors spanning over a large number of acoustic frames. Based on some previous small-scale experiments

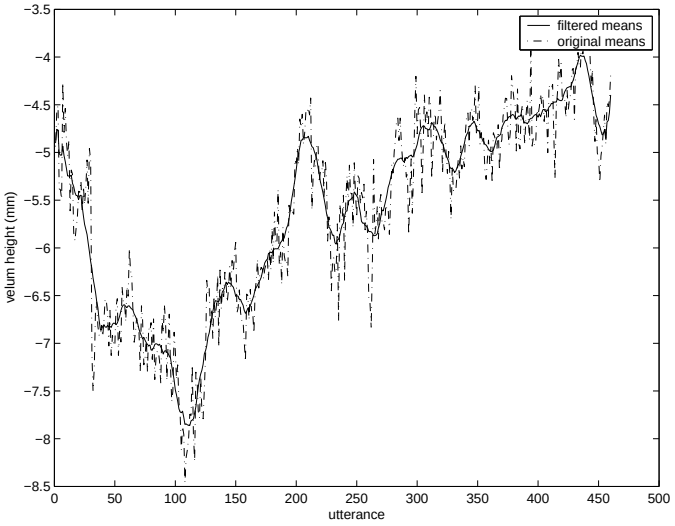

Fig. 1. Mean values of the "velum height" (v_y) channel across the utterances in the recording session. The dashed line shows the real means and the solid line their filtered version which is actually used for normalization.

of ours, we construct input vectors consisting of the representations of 17 frames: the frame in question, plus the 8 previous ones, plus the 8 next ones.

The steps taken to process the EMA data are similar to those described by Richmond. First, the EMA data are resampled to match the frameshift of the acoustic coefficients (5ms). At the same time, they are smoothed, using a moving average window of 40ms so that recording noise is eliminated (after all, it is known that EMA trajectories vary relatively slowly with time).

The mean values of the EMA trajectories calculated for every utterance vary considerably during the recording process. There are two kinds of variation: rapid changes, due to the phonemic content of each utterance, and slowly moving trends, mainly due to the fact that the subject's articulation adapts in certain ways during the recording session. It is beneficial to remove from the EMA data the second type of variation, while keeping the first. Thus, we calculate the means, low-pass filter them, and subtract those filtered means from the EMA data. (See Figure 1 for an explanation).

Finally, we scale the EMA data by four times their standard deviation (across the whole corpus), so that they roughly lie in the interval $(-1, 1)$, something crucial for SVR training.

So, we end up with training examples, each consisting of a 221-dimensional (for the MFCC case and the PLP case) or 425-dimensional (for the MFCC+PLP case) real-valued vector as input and a 14-dimensional real-valued vector as output. We split our data into two big halves: the even-numbered utterances constitute what we call an "extended training set", and the odd-numbered ones an "extended test set". Each one contains more than 100.000 examples.

Due to training time considerations we don't use the whole "extended training set' for training. Instead we take 1 out of 20 input-output pairs, arriving at a final training

set with about 6500 examples.

V. SVR Training and Results

The ϵ-SVR algorithm, as described, works for only one output. It does not work "as is" for multiple outputs. Thus, we have to split our problem into 14 distinct (and assumably independent) function estimation problems, considering each time a different EMA trajectory as output.

We employ the LibSVM software [10] and use the RBF kernel. For the choice of training hyperparameters C, γ and ϵ we adapt a combination of usual SVM heuristics (e.g. [11]) and a Cross-Validation grid-search ([12]), applied to each of the 14 regressors. We keep the ϵ fixed at 0.1 and choose the best pair of C and γ, out of a grid with $C \in (0.35, 0.5, 0.7, 11.41, 2, 2.83)$ and $\gamma \in (0.0016, 0.0023, 0.0032, 0.0045, 0.0064, 0.009, 0.0128)$, for the MFCC case and PLP case, or $\gamma \in (0.00083, 0.00118, 0.00166, 0.00235, 0.00332, 0.0047, 0.0066)$ for the MFCC+PLP case.

For evaluating the performance of our system of regressors we use two measures. The first one is the RMS error which is an indication of the overall "distance" between two trajectories. It is calculated as:

$$E_{RMS} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(o_i - t_i)^2} \tag{4}$$

where N is the number of input-output vector pairs, in the test set, o_i is the estimated value for the articulator channel output, and t_i is the real value. The values are rescaled back to the original domain measurement in millimeters.

The second measure is the correlation score, which is an indication of similarity of shape and synchrony of two trajectories. It is calculated by dividing their covariance by the product of their variances:

$$r = \frac{\sum_i (o_i - \bar{o})(t_i - \bar{t})}{\sqrt{\sum_i (o_i - \bar{o})^2 \sum_i (t_i - \bar{t})^2}} \tag{5}$$

where $\bar{o}$ and $\bar{t}$ are the mean channel value for the estimated and real articulator position respectively.

For testing our system we use 10 utterances spanning our whole "extended test set". Our overall results for this test set are presented in Table I where we also include the standard deviation of the EMA cannels (as an indication of the range of their values), and the results achieved in [3], for a comparison. Figure 2 shows the real and estimated trajectories for a single utterance using the MFCC+PLP approach. Figure 3 shows the real and estimated trajectories for a single EMA channel using all three approaches.

VI. Conclusion

We applied three different parametric represenattions of the speech signal to the acousti-to-EMA mapping task. The example of Figure 3 indicates no considerable difference in the performance of the corresponding. Nevertheless, the numbers in Table I show a slightly better performance in general of the PLPs over the MFCCs. There

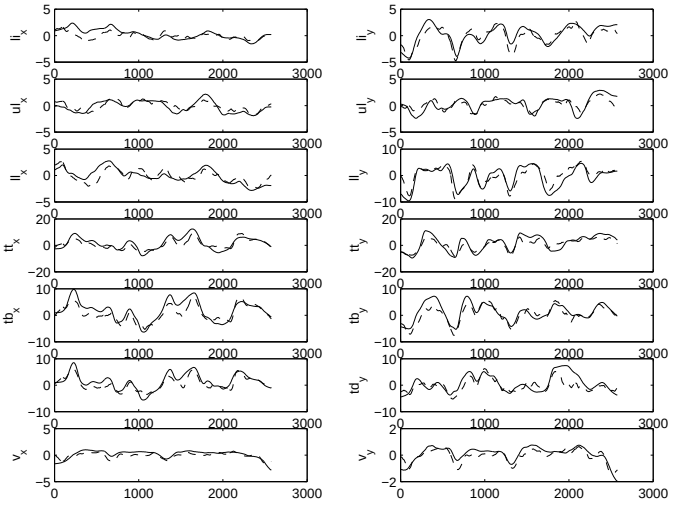

Fig. 2. Real (solid lines) and estimated by the MFCC+PLP method (dashed lines) articulatory trajectories of fsew0 uttering the phrase "How would you evaluate this algebraic expression?", plotted against time in milliseconds. From top to bottom: lower incisor (jaw), upper lip, lower lip, tongue tip, tongue dorsum, tongue blade and velum. The first column shows the projections of the articulators' movement on the x axis, while the second one those on the y axis.

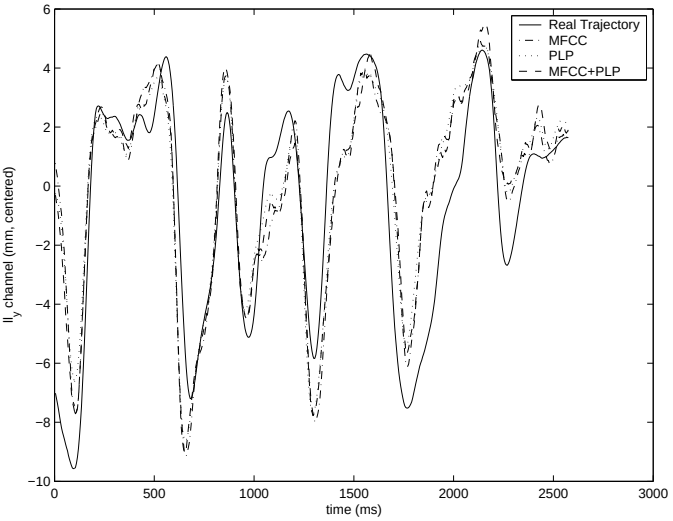

Fig. 3. Projection of the movement of the lower lip on the y axis, while fsew0 is uttering the phrase "How would you evaluate this algebraic expression?", and the corresponding estimated trajectories by the three methods presented in this paper

is further improvement when a combination of PLPs and MFCCs is used, but this comes with the expense of an increase in the dimensionality of the input space (and training time).

Our main broad goal is to improve upon the results presented in [1], [3] (see last column of Table I). This has not been clearly achieved so far. We may point out two reasons for this: the first one is that we use a context window that spans only about 80 ms of the duration of the speech signal. This is rather small compared to other related works (e.g. [13], [1]) that propose the use of context windows spanning over 200 ms. The second reason

		MFCC		PLP		MFCC+PLP		Results from [3]	
EMA Channel	σ	E_{RMS}	r	E_{RMS}	r	E_{RMS}	r	E_{RMS}	r
Lower incisor x	1,156	1,065	0,540	1,072	0,536	1,054	0,551	0,89	0,56
Lower incisor y	2,068	1,211	0,805	1,207	0,807	1,182	0,815	1,19	0,80
Upper lip x	1,155	0,792	0,613	0,790	0,614	0,788	0,617	0,99	0,58
Upper lip y	1,626	1,166	0,691	1,158	0,697	1,150	0,698	1,16	0,72
Lower lip x	1,655	1,382	0,575	1,374	0,578	1,366	0,586	1,21	0,61
Lower lip y	4,088	2,617	0,807	2,617	0,806	2,577	0,812	2,73	0,75
Tongue tip x	3,974	2,409	0,766	2,385	0,770	2,356	0,779	2,43	0,79
Tongue tip y	4,715	2,508	0,828	2,468	0,834	2,446	0,837	2,56	0,84
Tongue body x	3,709	2,221	0,765	2,218	0,766	2,178	0,776	2,19	0,81
Tongue body y	4,104	2,232	0,833	2,217	0,836	2,186	0,841	2,14	0,83
Tongue dorsum x	3,306	1,943	0,749	1,939	0,751	1,902	0,762	2,04	0,79
Tongue dorsum y	3,388	2,359	0,762	2,355	0,762	2,323	0,771	2,31	0,71
Velum x	0,641	0,411	0,808	0,407	0,812	0,403	0,816	0,42	0,79
Velum y	0,612	0,402	0,802	0,401	0,803	0,391	0,814	0,41	0,77

TABLE I

Overall performances on the test set of the system of function estimators derived by the three approaches. Also presented: the standard deviations of the dataset and the results achieved in [3]. RMS values in millimeters.

of our shortcoming is the fact that we used for training only a small subset of the data available to us. A considerable improvment should be expected with the increase of the training set size, nevertheless the training time requirements are rather big.

Aknowledgement

The first author is funded by a dissertation scholarship through the Operational Programme for Education and Initial Vocational Training ("Heracletus" Project).

References

[1] Korin Richmond. *Estimating Articulatory Parameters from the Speech Signal*. PhD thesis, The Center for Speech Technology Research, Edinburgh, 2002.

[2] Miguel Á. Carreira-Perpiñán. *Continuous Latent Variable Models for Dimensionality Reduction and Sequential Data Reconstruction*. PhD thesis, University of Sheffield, UK, February 2001.

[3] Korin Richmond, Simon King, and Paul taylor. Modelling the uncertainty in recovering articulation from acoustics. *Computer Speech and Language*, 17:153–172, 2003.

[4] Alan A. Wrench and William J. Hardcastle. A multichannel articulatory database and its application for automatic speech recognition. In *5th Seminar on Speech Production: Models and Data*, pages 305–308, Kloster Seeon, Bavaria, 2000.

[5] Alex Smola and Bernhard Schölkhopf. A tutorial on support vector regression. *Statistics and Computing*, 14(3):199–222, August 2004.

[6] Vladimir Vapnik. *Statistical Learning Theory*. Wiley, New York, 1998.

[7] Steve Young, Gunnar Evermann, Mark Gales, Thomas Hain, Dan Kershaw, Gareth Moore, Julian Odell, Dave Ollason, Dan Povey, Valtcho Valtchev, and Phil Woodland. *The HTK Book (for HTK version 3.3)*. Cambridge University Engineering Department, 2005.

[8] Steven B. Davis and Paul Mermelstein. Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences. In Alexander Waibel and Kai-Fu Lee, editors, *Readings in speech recognition*, pages 65–74. Morgan Kaufmann Publishers Inc., 1990.

[9] Hynek Hermansky. Perceptual linear predictive (PLP) analysis of speech. *Journal of the Acoustical Society of America*, 87(4):1738–1752, April 1990.

[10] Chih-Chung Chang and Chih-Jen Lin. *LIBSVM: a library for support vector machines*, 2001. Software available at http://www.csie.ntu.edu.tw/~cjlin/libsvm.

[11] Jason Weston, Arthur Gretton, and Andre Elisseeff. SVM practical session (how to get good results without cheating). Machine Learning Summer School 2003, Tuebingen, Germany.

[12] Chih-Wei-Hsu, Chih-Chung Chang, and Chih-Jen Lin. A practical guide to support vector classification. Available at http://www.csie.ntu.edu.tw/ cjlin/papers/guide/guide.pdf, 2003.

[13] George Papcun, Judith Hochberg, Timothy R. Thomas, François Laroche, Jeff Zacks, and Simon Levy. Inferring articulation and recognizing gestures from acoustics with a neural network trained on x-ray microbeam data. *Journal of the Acoustical Society of America*, 92(2):688–700, August 1992.