ACOUSTIC-TO-ARTICULATORY INVERSION USING AN EPISODIC MEMORY

S. Demange, S. Ouni
LORIA, UMR 7503
BP329, 54506 Vandoeuvre-lès-Nancy, France

ABSTRACT
This paper presents a new acoustic-to-articulatory inversion method based on an episodic memory, which is an interesting model for two reasons. First, it does not rely on any assumptions about the mapping function but rather it relies on real synchronized acoustic and articulatory data streams. Second, the memory structurally embeds the naturalness of the articulatory dynamics. In addition, we introduce the concept of generative episodic memory, which enables the production of unseen articulatory trajectories according to the acoustic signals to be inverted. The proposed memory is evaluated on the MOCHA corpus. The results show its effectiveness and are very encouraging since they are comparable to those of recently proposed methods.

Index Terms—Episodic memory, acoustic-to-articulatory inversion, electromagnetic articulography (EMA)

1. INTRODUCTION
Acoustic-to-articulatory speech inversion aims at recovering the movements of articulators from an acoustic speech signal. This problem, of special interest in the community of speech processing, is challenging due to the non-linearity and non-uniqueness of the mapping [1]. Several studies on speech inversion relied on articulatory synthesis models. The inversion was performed by first using an articulatory synthesizer to create a codebook of allowed articulatory positions and their corresponding acoustics. The inversion was performed by looking up the input signal in the codebook combined with dynamic programming to retrieve the articulatory trajectories [2].

Recently, databases of synchronized acoustic and articulatory data streams using electromagnetic articulography have been possible. The availability of these corpora allowed the application of machine learning algorithms to perform the acoustic to articulatory regression. Moreover the regressions are more realistic as they rely on true acoustic-articulatory observations. Works have been done using support vector machines [3], Gaussian mixture models [4], hidden Markov models [5, 6] and artificial neural networks [7, 8] for example. However, statistical regression approaches (model driven) imply to make assumptions on the acoustic-to-articulatory mapping function, which is known to be highly non-linear and thus difficult to model. In contrary, memory-based methods (data driven) [9, 10] do not make any assumption on the mapping function but rather rely on a collection of synchronized acoustic-articulatory data.

We propose in this paper a new approach based on an episodic memory. While a codebook is often considered as a collection of independent acoustic-articulatory samples, an episodic memory accounts for the order the samples occur. An episodic memory is made of a collection of episodes, which are the realizations of predefined units (here phones). Thus an episodic memory is more suitable than a codebook for modeling the dynamics of the articulatory movements, which is of great interest for the inversion.

Estimating the unknown articulatory trajectories from a particular acoustic signal consists in finding the sequence of episodes, which acoustically best explains the input acoustic signal. We refer to such a memory as a concatenative memory (C-Mem). Actually a C-Mem lacks generalization capabilities as it contains only several examples of a given unit and fails to invert an acoustic signal, which is not similar to the ones it contains. However, if we look within each episode we can find local similarities between them. We propose to take advantage of these local similarities to build a generative episodic memory (G-Mem) by creating inter-episodes transitions.

In the following, we present how to build the G-Mem and how the inversion is performed. Next the evaluation of the proposed method is presented. Finally, comparisons to other works on the same data set as well as insights to improve the proposed method are given.

2. EPISODIC MEMORY-BASED INVERSION METHOD
The main advantage of an episodic memory is that it keeps track of the order of the observations and thus the dynamics of each episode. In order to preserve this property, the inter-episodes transitions have to be defined carefully. Indeed, they're expected to provide the memory with generalization capabilities but must not allow the memory to produce unrealistic trajectories. To this end, we define the transitions according to the concept of articulatory target interval (ATI).

2.1. Articulatory target interval
Let X be an articulatory trajectory of a particular phone expressed as a sequence of K observations: \( X = (x_1, x_2, \ldots, x_K) \). We define each observation \( x_{i+1} \) as the natural articulatory target of \( x_i \) as it has been observed following \( x_i \). In fact, \( x_{i+1} \) is a particular articulatory configuration but we can suppose it could have been slightly different. Indeed, starting from \( x_i \) at time \( t \), the articulators could have reached a different target at time \( t + 1 \) close to \( x_{i+1} \). Thus \( x_{i+1} \) is just a particular one. Then, for each \( x_i \) we define an articulatory target interval \( ATI_{x_i} \) as the interval \([x_{i+1} - \delta, x_{i+1} + \delta]\), where \( \delta \) is a positive value.

2.2. Building the generative episodic memory
2.2.1. Inter-episode transitions
Let \( Y = (y_1, y_2, \ldots, y_N) \) be a second articulatory trajectory of the same class as \( X \) (i.e. \( X \) and \( Y \) are two different realizations of the same phone). We define \( \phi = (\Phi_1, \ldots, \Phi_M) \) as the alignment path corresponding to the shortest distance \( D(X, Y) \) between
X and Y obtained by the well-known dynamic time warping algorithm (DTW). Each \( \Phi_i \) is a pair of indexes of the elements of X and Y, which are aligned together: \( \Phi_i = (\Phi_x,i, \Phi_y,i) \). For example, \( \Phi_3 = (4,5) \) indicates that the third element of the path is an alignment between \( x_4 \) the fourth observation of X and \( y_5 \) the fifth observation of Y. For our problem, we extended the DTW algorithm with the Itakura constraints [11] to impose temporal constraints on the alignment paths. Once the DTW distance between X and Y is computed, a transition in the memory from any \( x_i \) to \( y_j \) is created if \( y_j \) matches the two following conditions:

\[
\Phi_{y,i+1} = j \quad (1)
\]
\[
y_j \in ATI_x_i \quad (2)
\]

Equation 1 requires \( y_j \) to be aligned with \( x_{i+1} \) when mapping Y onto X. In other words, it has to be aligned with the following observation of \( x_i \). This condition ensures that the transitions are consistent with the temporality of the episodes. Equation 2 states that \( y_j \) has to belong to the \( ATI_x_i \). It locally ensures the physical validity and naturalness of the transition since \( y_j \) is close to \( x_{i+1} \), which is the natural articulatory target of \( x_i \). Note that the articulatory trajectories of two episodes of the same phone can be significantly different due to the co-articulation effects as their phonetic contexts differ. Combining two episodes, which match only on a very small segment but which drastically differ outside, could lead to an unrealistic trajectory. To avoid this undesired effect, transitions from X to Y are created only if Y is similar to X:

\[
D(X,Y) \leq \Delta \quad (3)
\]

where \( \Delta \) is a positive value. Algorithm 1 resumes the procedure for creating the inter-episode transitions. One can remark that the G-Mem is still conservative as all the original articulatory trajectories of the episodes it is made of are preserved since:

\[
D(X,X) = 0 \leq \Delta \quad \Phi_{x,i+1} = i + 1 \quad (5)
\]
\[
x_{i+1} \in ATI_x_i \quad (6)
\]

2.2.2. Between-episode transitions

The transitions at episode boundaries are only subject to the articulatory continuity requirement expressed by the equation 2. Let \( Z \) be the episode observed following X. Then, a transition from \( x_K \) (the last observation of X) to the first observation \( u_1 \) of any episode \( W \) of any class is created if \( u_1 \in ATI_x_K = [z_1 - \delta, z_1 + \delta] \). If the episode X is the last of a record its natural articulatory target is unknown and equation 2 cannot be verified, thus no transition to any other episode is possible. Note that a C-Mem only accounts for these between-episodes transitions. So, it can be seen as a particular G-Mem for which \( \Delta \) of equation 3 is set to zero.

2.3. Recovering the articulatory trajectories

In practice the generative memory is modeled as an oriented graph \( G_{G-Mem} \). The nodes are synchronized acoustic and articulatory observations and the edges are the allowed transitions between the articulatory configurations. The transitions are created according to the procedure described above in the articulatory space using Euclidean distance. Each path within the graph corresponds to an acoustic sequence and its corresponding articulatory trajectory.

Recovering the unknown articulatory trajectory of a given acoustic signal X is performed by searching the path within \( G_{G-Mem} \) that acoustically best explains X. All paths can start only at nodes representing the first observation of the episodes and can end only at nodes representing the last observation of the episodes. During the inversion a breadth first search is performed applying the Viterbi algorithm. At each step, only the K best paths are propagated. The score of each path is the sum of the local distances between its acoustic components and the samples of X. The local acoustic distances are computed over a window using the Euclidean distance. The complexity of the inversion is \( O(T.K.N) \) where \( T \) is the number of samples of X, \( K \) is the search beam width and \( N \) is the average branching factor of each node.

3. EVALUATION

3.1. Data

All the experiments presented in this work have been carried out on the MOCHA corpus [12]. Two speakers, a female (fsew) and a male (msak) British English speakers, were recorded while reading 460 short phonetically balanced British-TIMIT sentences. We used in this work the acoustic and the EMA data streams. The acoustic was provided as waveforms sampled at 16 kHz and the EMA data consist in 2D data with coordinates expressed in the mid-sagital plane sampled at 200Hz. The silences occurring at the beginning and the end of the records were discarded as the articulators may move unpredictably. Nine sensors were used; located at the bridge of the nose (bn), upper incisors (ui), lower incisors (li), upper lip (ul), lower lip (ll), tongue tip (tt), tongue body (tb), tongue dorsum (td) and velum. The two first were used to normalize the trajectories of the last seven with regard to the head movements. Each corpus was split into training, development and test sets. Care is taken, so that the utterances selected for each set correspond exactly to the ones used by Richmond [7] and which is commonly adopted in many works. Information about the different sets is given in table 1. Note that the durations only account for usable speech. Figure 1 shows the distributions of the articulatory samples for each speaker and coil as well as their standard deviations along both the horizontal and vertical directions.

3.2. Features extraction

A Linear Predictive Analysis is performed on the speech signal using the HTK toolkit.12 cepliftered MF-PLPs plus the logarithmic energy of the signal comprise the acoustic feature vector extracted from every 25 ms speech frame shifted by 10 ms.

The articulatory data were first down sampled to 100 Hz to match the acoustic frame rate. Then, all trajectories were low-pass
Three inversion methods have been implemented and compared. The first one is a codebook-based approach as presented in [9]. This method is based on a search of an articulatory-acoustic codebook, which is built from synchronized articulatory and acoustic data stream. The codebook search employs dynamic constraints both on acoustic and articulatory behaviors. The following two methods are the implementations of the episodic memories: C-Mem and G-Mem.

These methods share the same paradigm as they are all data driven. They differ in the way the dynamics of the recovered trajectories is retrieved and their capability of generalizing over the data. The codebook approach applies dynamic constraints during the search step. Meanwhile the articulatory dynamics is already integrated within the episodic memories. The C-Mem can only express the recovered trajectories as a concatenation of the episodes but cannot generalize over the data. On the contrary, the G-Mem is able to produce new trajectories according to the acoustic speech signal to be inverted.

Three inversion methods have been implemented and compared. The first one is a codebook-based approach as presented in [9]. This method is based on a search of an articulatory-acoustic codebook, which is built from synchronized articulatory and acoustic data stream. The codebook search employs dynamic constraints both on acoustic and articulatory behaviors. The following two methods are the implementations of the episodic memories: C-Mem and G-Mem.

These methods share the same paradigm as they are all data driven. They differ in the way the dynamics of the recovered trajectories is retrieved and their capability of generalizing over the data. The codebook approach applies dynamic constraints during the search step. Meanwhile the articulatory dynamics is already integrated within the episodic memories. The C-Mem can only express the recovered trajectories as a concatenation of the episodes but cannot generalize over the data. On the contrary, the G-Mem is able to produce new trajectories according to the acoustic speech signal to be inverted.

Three inversion methods have been implemented and compared. The first one is a codebook-based approach as presented in [9]. This method is based on a search of an articulatory-acoustic codebook, which is built from synchronized articulatory and acoustic data stream. The codebook search employs dynamic constraints both on acoustic and articulatory behaviors. The following two methods are the implementations of the episodic memories: C-Mem and G-Mem.

These methods share the same paradigm as they are all data driven. They differ in the way the dynamics of the recovered trajectories is retrieved and their capability of generalizing over the data. The codebook approach applies dynamic constraints during the search step. Meanwhile the articulatory dynamics is already integrated within the episodic memories. The C-Mem can only express the recovered trajectories as a concatenation of the episodes but cannot generalize over the data. On the contrary, the G-Mem is able to produce new trajectories according to the acoustic speech signal to be inverted.

The quality of the recovered articulatory trajectories is evaluated with regard to the root-mean-square (RMS) error, which quantifies the difference between the measured $x_{i}^{Ac}$ and estimated $f(x_{i}^{Ac})$ articulatory coordinates where $x_{i}^{Ac}$ and $x_{i}^{Ar}$ are respectively the articulatory and acoustic components of the test records:

$$\text{error}_{RMS} = \sqrt{\frac{1}{N} \sum_{i=1}^{N} (f(x_{i}^{Ac}) - x_{i}^{Ar})^2}$$

and with regard to the Pearson’s correlation, which expresses the similarity and synchrony of two trajectories. It is obtained by dividing the covariance of the two trajectories by the product of their standard deviations (where overlines denote mean values over the test set):

$$\text{cov} = \frac{\sum_{i=1}^{N} (f(x_{i}^{Ac}) - \overline{f(x_{i}^{Ac})})(x_{i}^{Ar} - \overline{x_{i}^{Ar}})}{\sqrt{\sum_{i=1}^{N} (f(x_{i}^{Ac}) - \overline{f(x_{i}^{Ac})})^2 \cdot \sum_{i=1}^{N} (x_{i}^{Ar} - \overline{x_{i}^{Ar}})^2}}$$

### 3.3. Experiment design

The results are reported on figure 2. The histogram bars show the overall RMS errors (means over the coils and x- and y-coordinates) in millimeters for each corpus applying the Codebook, C-Mem and G-Mem based approaches. The respective Pearson’s correlations are given below the bars. The error bars represent the bootstrap-t 99% confidence intervals [13]. Ten thousands of pseudo test sets are generated by randomly selecting records from the original test

<table>
<thead>
<tr>
<th>Corpora</th>
<th>Sets</th>
<th>Durations</th>
<th>Sentences</th>
<th>Phones</th>
</tr>
</thead>
<tbody>
<tr>
<td>fsew</td>
<td>train</td>
<td>16 min 35 sec</td>
<td>368</td>
<td>11179</td>
</tr>
<tr>
<td></td>
<td>dev</td>
<td>1 min 57 sec</td>
<td>46</td>
<td>1324</td>
</tr>
<tr>
<td></td>
<td>test</td>
<td>2 min 5 sec</td>
<td>46</td>
<td>1457</td>
</tr>
<tr>
<td>msak</td>
<td>train</td>
<td>13 min 39 sec</td>
<td>368</td>
<td>11179</td>
</tr>
<tr>
<td></td>
<td>dev</td>
<td>1 min 41 sec</td>
<td>46</td>
<td>1324</td>
</tr>
<tr>
<td></td>
<td>test</td>
<td>1 min 45 sec</td>
<td>46</td>
<td>1457</td>
</tr>
</tbody>
</table>

Table 1. Synthetic description of the MOCHA corpus.

Fig. 2. Overall RMS errors in millimeters using the Codebook, C-Mem and G-Mem based methods with and without the phonemic segmentations. The error bars represent the bootstrap-t 99% confidence intervals. For each experiment the Pearson’s correlation is provided below histogram bars.

We have chosen to consider the recovery of the articulatory trajectories of each coil and along both the $x-$ and $y-$coordinate as independent inversion problems. The proposed episodic memory allows considering the coils all together during the inversion and would benefit from it as it would capture the correlations between the articulators. However such strategy would require much more training examples than those available in the corpora we used. This is why fourteen distinct inversion problems are solved each using its own set of parameters optimized on the development sets (the acoustic window lengths, the weights of the dynamic constraints for the codebook and $\delta$ and $\Delta$ for the memories). All presented results have been obtained on the test sets.

Two series of experiments have been done, with and without the phonemic segmentations of the test records obtained by forced aligning acoustic models onto the test sets.

The quality of the recovered articulatory trajectories is evaluated with regard to the root-mean-square (RMS) error, which quantifies the difference between the measured $x_{i}^{Ac}$ and estimated $f(x_{i}^{Ac})$ articulatory coordinates where $x_{i}^{Ac}$ and $x_{i}^{Ar}$ are respectively the articulatory and acoustic components of the test records:

$$\text{error}_{RMS} = \sqrt{\frac{1}{N} \sum_{i=1}^{N} (f(x_{i}^{Ac}) - x_{i}^{Ar})^2}$$

and with regard to the Pearson’s correlation, which expresses the similarity and synchrony of two trajectories. It is obtained by dividing the covariance of the two trajectories by the product of their standard deviations (where overlines denote mean values over the test set):

$$\text{cov} = \frac{\sum_{i=1}^{N} (f(x_{i}^{Ac}) - \overline{f(x_{i}^{Ac})})(x_{i}^{Ar} - \overline{x_{i}^{Ar}})}{\sqrt{\sum_{i=1}^{N} (f(x_{i}^{Ac}) - \overline{f(x_{i}^{Ac})})^2 \cdot \sum_{i=1}^{N} (x_{i}^{Ar} - \overline{x_{i}^{Ar}})^2}}$$

3.4. Results

The results are reported on figure 2. The histogram bars show the overall RMS errors (means over the coils and x- and y-coordinates) in millimeters for each corpus applying the Codebook, C-Mem and G-Mem based approaches. The respective Pearson’s correlations are given below the bars. The error bars represent the bootstrap-t 99% confidence intervals [13]. Ten thousands of pseudo test sets are generated by randomly selecting records from the original test

![Fig. 1. The plots show on the left column the articulatory sample distributions for the two speakers. The right column shows their standard deviations. Each rectangle is centered at the mean position of a coil and the sides along both x- and y-coordinates are one standard deviation long. The coordinates are expressed in millimeters.](image-url)

![Fig. 2. Overall RMS errors in millimeters using the Codebook, C-Mem and G-Mem based methods with and without the phonemic segmentations. The error bars represent the bootstrap-t 99% confidence intervals. For each experiment the Pearson’s correlation is provided below histogram bars.](image-url)
set with replacement. An overall RMS error is computed for each pseudo test set. The bootstrap-\( t \) 99% confidence interval is defined as \([m - 3\sigma, m + 3\sigma] \) with \( m \) the mean over the pseudo test sets RMS errors and \( \sigma \) the associated standard deviation. As the confidence intervals sometimes overlap, the probabilities of improvement (\( poi \)) as defined in [13] have been computed. Let \( poi(S_1, S_2) \) be the number of pseudo test sets for which \( S_1 \) outperforms \( S_2 \) normalized by the total number of pseudo test sets. We found that \( poi(G-Mem,Codebook) = 100\% \) and \( poi(G-Mem,C-Mem) = 100\% \). Thus, the episodic memory-based approaches always outperform the codebook-based method. This shows that it is more beneficial to embed the trajectory dynamics within the memory structure than applying dynamic constraints during the search. The G-Mem always outperforms the C-Mem. This illustrates how much an episodic memory can benefit from a generalization capability. The best Pearson’s correlations are obtained using a G-Mem indicating that the G-Mem succeed in producing natural articulatory trajectories. Without any phonemic knowledge an average RMS error of 1.65 mm and a correlation of 0.714 are obtained with the proposed G-Mem. Using the phonemic segmentation of the test records the average RMS error decreases to 1.50 mm and the correlation rises up to 0.757. Thus, we can expect a RMS error lying in [1.50, 1.65] mm using phonemic segmentations resulting from a standard speech decoding for example.

4. DISCUSSION AND FUTURE WORK

The evaluation clearly shows that the G-Mem significantly outperforms a codebook-based approach with dynamic constraints as proposed in [9]. Hiroya et al. [5] reported RMS errors of 1.50 and 1.73 mm with and without phonemic segmentations using an HMM-based production model. However, we cannot fairly compare with as they used a Japanese database.

On MOCHA, Zhang et al. [6] obtained an RMS error of 1.71 mm using a trajectory HMM. They included velocity features into their acoustic front-end and have done a speech recognition prior the inversion to provide their system with a phonemic segmentation. Even without phonemic segmentation the G-Mem performs slightly better. Toda et al. [4] used Gaussian mixtures models to map the acoustic space onto the articulatory space. They decreased the RMS error from 1.58 to 1.40 mm applying a maximum likelihood estimation (MLE) of the dynamic features. Moubayed et al. [10] proposed a memory-based method. They used a linear regression on the local neighborhood of the codebook entries to map the acoustic input frames onto the articulatory space. They also used the MLE of the dynamic features to improve the trajectories. A RMS error of 1.52 mm was reported using this method. Finally, Richmond [8] reported an RMS error of 1.4 mm using trajectory multiple density neural networks.

For now, the proposed G-Mem does not use dynamic acoustic features, which seem to be beneficial. Moreover the Euclidean distance was used to compare the acoustic during the inversion and this distance is not robust for speech processing. The use of other acoustic distances, in particular local kernel distances, as well as the use of the first and second derivative acoustic features are currently investigated. In addition, a local linear regression as proposed by Moubayed et al. [10] could further improve our results as the G-Mem can produce unseen trajectories but fails to precisely map an acoustic frame onto the articulatory space if this acoustic frame does not belong to the memory. Moreover, a phonemic segmentation has been shown beneficial for the inversion but require a first pass over the test data to be generated. As an alternative, a language model could be used to rescore the paths within the memory during the inversion. A dictionary could further constrain the paths and decrease the RMS error to reach similar results we have obtained with the phonemic segmentation.

5. REFERENCES


