An episodic memory-based solution for the acoustic-to-articulatory inversion problem

Sébastien Demange, Slim Ouni

An episodic memory-based solution for the acoustic-to-articulatory inversion problem

Sébastien Demange[1] and Slim Ouni[1]

[1] *Université de Lorraine,*

LORIA,

UMR 7503,

Vandœuvre-lès-Nancy,

F-54506,

France

(Dated: May 22, 2013)

Abstract

This paper presents an acoustic-to-articulatory inversion method based on an episodic memory. An episodic memory 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 inherently represents the real articulatory dynamics as observed. It is argued that the computational models of episodic memory, as they are usually designed, cannot provide a satisfying solution for the acoustic-to-articulatory inversion problem due to the insufficient quantity of training data. Therefore, an episodic memory is proposed, called generative episodic memory (G-Mem), which is able to produce articulatory trajectories that do not belong to the set of episodes the memory is based on. The generative episodic memory is evaluated using two Electromagnetic Articulography (EMA) corpora: one for English and one for French. Comparisons with a codebook-based method and with a classical episodic memory (which is termed concatenative episodic memory) are presented in order to evaluate the proposed generative episodic memory in terms of both its modeling of articulatory dynamics and its generalization capabilities. The results show the effectiveness of the method where an overall root-mean-square error of 1.65 mm and a correlation of .71 are obtained for the G-Mem method. They are comparable to those of methods recently proposed.

PACS numbers: 43.70.Bk, 43.70.Jt, 43.70.Aj

I. INTRODUCTION

Recovering the vocal tract shape from speech acoustics could allow a number of breakthroughs in automatic speech processing. For instance, the location of critical articulators [?] could be exploited to better characterize a given phoneme, which may improve speech synthesis[?] and recognition[? ? ? ?] . Indeed, articulatory features vary much more slowly than speech acoustic features, and thus they should be more robust than acoustic parameterizations, especially in noisy environments[?] .

Recently, databases of synchronized acoustic and articulatory data streams, using electromagnetic articulography (EMA) for instance, have become available. These corpora enable machine learning algorithms to perform acoustic-to-articulatory regression. The main techniques use support vector machines[?] , Gaussian mixture models[?] , hidden Markov models (HMM)[? ?] , or artificial neural networks[? ?] . These methods try to learn the acoustic-to-articulatory mapping function, which is known to be highly non-linear, non-unique (i.e., different vocal tract shapes producing the same acoustics) and thus difficult to model[? ? ? ? ?] . As an alternative, codebook-based approaches [? ? ? ? ?] make very few assumptions about the mapping function, but rather rely on a collection of pairs of acoustic-articulatory data.

We believe that the main difficulty of inversion is the lack of a good representation of the dynamics. The non-uniqueness problem[?] will very likely vanish if the dynamics are fully integrated within the inversion methods. In fact, Qin & Carreira-Perpiñán[?] , and to some extent Neiberg *et al.*[?] , argued that natural human speech is produced with a unique vocal tract configuration and there are few cases of non-uniqueness. Phonetic context naturally imposes constraints on the vocal tract related to coarticulation. Effective modeling of articulatory dynamics seems essential to solving the inversion problem. These dynamics can be modeled with HMMs[? ?] and neural networks[?] ; they can also be inferred from time derivative features[?] . For the codebook-based methods the dynamics are not modeled at all. Instead, continuity constraints are used during inversion. However, despite these constraints, the recovered articulatory trajectories show many discontinuities and need to be smoothed

with signal processing techniques.

In this paper, we present a new data-driven approach based on the concept of episodic memory[?] . An episodic memory can be considered to be a codebook that includes a temporal dimension. While a codebook models the relationship between two static observations, an episodic memory can model the relationship between two sequences of observations. The main advantage of episodic memory is that it keeps track of the order of the observations and thus preserves the acoustic and articulatory dynamics of each episode.

An episodic memory can deal with the non-linearity of the mapping function by using the one-to-one correspondence between the synchronized acoustic and articulatory observations. It can deal with the non-uniqueness of the mapping by exploiting the articulatory dynamics encoded through the time ordering of both the episodes and the observations they are comprised of. In addition, the need to smooth inferred articulator trajectories is greatly reduced compared to current codebook-based inversion methods.

In the following sections, we first present our motivations for developing a generative as opposed to a concatenative episodic memory. Then, we present our inversion method based on the generative memory model. Finally, inversion results are presented and compared to other methods that have been reported previously.

II. EPISODIC MODELING FOR SPEECH INVERSION

In speech processing, the episodes, which are the units of an episodic memory, are individual acoustic realizations of predefined linguistic units (e.g. phones, diphones, syllables or words). A similar approach has been used in speech recognition[?] , referred to as template-based or exemplar-based speech recognition, and in unit selection speech synthesis[? ?] . The memories used for these two problems are concatenative (C-Mem) because the episodes are indivisible and the recognized or synthesized sentences are always comprised of a concatenation of episodes.

We apply an episodic memory model to the acoustic-to-articulatory mapping problem,

whereby episodes comprise the synchronized acoustic and articulatory feature sequences for each linguistic unit. Then, for inversion, the memory could exploit precisely the acoustic-articulatory relationship, as well as real acoustic and articulatory dynamics.

Speech inversion differs from speech synthesis and recognition because the mapping is between two continuous spaces, while for speech recognition and synthesis, the target or the source space, respectively, is discrete. This is an important issue for an episodic memory model because the memories need to contain many episodes of each linguistic unit in order to achieve adequate coverage of the variability present in speech. Usually, several hours of acoustic speech are required to reach acceptable performance when episodic memory is used for synthesis or recognition. Furthermore, keeping in mind that the articulators can compensate for one another during speech (a phenomenon which introduces non-uniqueness to the solution), an episodic memory intended for speech inversion should be even larger to ensure good coverage of both the acoustic and articulatory variability. This requirement cannot be fulfilled, as almost all articulatory corpora currently available contain less than an hour of speech, usually much less. The biggest corpus, the EMA part of the *mngu0* dataset[?] , made available very recently and containing 1263 sentences (1 hour 27 minutes of speech) is still probably too small. We think that this lack of data is the main reason why episodic modeling has not yet been used for this problem.

To address this problem, we propose to provide the memory with a mechanism to simulate many more episodes than the ones it contains. We define an episode as synchronized acoustic and articulatory realizations of a linguistic unit (LU). Let us consider two episodes X and Y of a given linguistic unit. X and Y are almost identical, differing only at the beginning and end, due to coarticulation effects. A C-Mem which does not contain any episode of LU whose left and right contexts are those of X and Y, respectively, will invariably fail to invert acoustic realizations of LU in this context. However, the memory could perform better if it were allowed to go through the first part of X, then to switch to Y at any time during the central, nearly identical part, and finally to go through the final part of Y.

Even though all episodes of a given linguistic unit are not identical, they can exhibit

local articulatory similarities. Therefore, we propose to allow the memory to switch between any two episodes X and Y of the same linguistic unit during the inversion at times when observations of X and Y are similar. Care will be taken to produce realistic articulatory dynamics by considering similarities with regard to temporal alignments and with regard to the positions of the articulators. As the proposed memory will be able to generate episodes other than the ones in the database, we will refer to this memory as a generative memory (G-Mem).

III. GENERATIVE MEMORY BASED INVERSION

As before, we define an episode as synchronized acoustic and articulatory realizations of a particular linguistic unit, in this case, phonemes. The phoneme identity will be referred to as the class of the episode. In the following, we consider two given episodes X and Y. X is a particular realization of a given phoneme expressed as a sequence of K articulatory-acoustic observations $X = (x_1, \ldots, x_K)$. Each observation $x_i = (x_i^{art}, x_i^{ac})$, where $i \in [1..N]$, is a synchronized pair composed of an articulatory observation x_i^{art} and its corresponding acoustic observation x_i^{ac}. The scalar articulatory observation x_i^{art} can be a given articulator description or even the x- or y-coordinate of a sensor glued onto an articulator as used in our work. Similarly, Y is another realization of the same phoneme expressed as a sequence of N observations: $Y = (y_1, \ldots, y_N)$, where $y_i = (y_i^{art}, y_i^{ac})$.

A. Local articulatory similarity

Dynamic Time Warping (DTW)[?] is a general algorithm to find the shortest distance $D(X, Y)$ between two episodes X and Y, which may vary in length. The episodes are warped non-linearly in order to minimize the effects of their temporal variability. Any given mapping leads to a particular alignment path $\Phi = (\Phi_1, \ldots, \Phi_M)$, where M is the number of alignments. $\Phi_i = (\Phi_{x,i}, \Phi_{y,i})$ is the i^{th} observation pairing along Φ with $\Phi_{x,i}$ and $\Phi_{y,i}$ as the indices in X and Y of the aligned observations. The distance between X and Y given Φ is

the sum of the local distances $d(.,.)$ between the aligned observations along Φ. The choice of this local distance $d(.,.)$ depends on the nature of the observations used to perform the mapping. Here, the articulatory observations are used because we are focusing on producing realistic articulatory trajectories. In our work, the articulatory observations are the x- and y-coordinates of sensors glued onto articulators in the midsagittal plane. Thus, Euclidean distance is used.

$D(X,Y)$ is the shortest distance over all Φ:

$$D(X,Y) = \arg\min_{\Phi} \sum_{i=1}^{M} d(x_{\Phi_{x,i}}, y_{\Phi_{y,i}}) \tag{1}$$

Many variations of the algorithm have been proposed[?] in order to prevent degenerate paths from occurring. In this work, we applied Itakura's constraints[?] . These constraints make the DTW asymmetric, such that each observation in sequence X is aligned with exactly one observation in Y. The mappings of many episodes onto X result in many alignment paths of the same length (equal to the length of X). Therefore, the distances from these episodes to X can be fairly compared and ranked. The other Itakura constraints impose bounds on the temporal deformation, in order to preserve a certain temporal consistency of the aligned observations.

Let X^{art} be an articulatory trajectory of an episode X expressed as a sequence of K articulatory positions $(x_1^{art}, x_2^{art}, \ldots, x_K^{art})$. We define each articulatory observation x_{i+1}^{art} as the natural articulatory target (local target) of x_i^{art} since it has been observed to follow x_i^{art}. In fact, x_{i+1}^{art} is a specific articulatory position, but we can suppose it could have been slightly different. Indeed, starting from x_i^{art} at time i, the articulators could have reached a different position at time $i+1$ close to x_{i+1}^{art} with no significant consequences to the acoustics. Then, for each x_i we define an articulatory target interval ATI_{x_i} as:

$$ATI_{x_i} = [x_{i+1}^{art} - \delta, x_{i+1}^{art} + \delta] \tag{2}$$

where δ is a given positive value.

We consider any articulatory position y_j^{art} to be similar to any articulatory position x_i^{art}, if y_j is aligned with x_i when mapping Y onto X, and if y_j^{art} belongs to the articulatory target interval of x_{i-1}^{art}.

B. Building the generative episodic memory

We model a G-Mem as an oriented graph $\mathcal{G}_{G-Mem}$. The nodes are the pairs of synchronized acoustic and articulatory observations comprising the episodes. The oriented edges indicate the allowed transitions between the articulatory positions the memory can follow during the inversion process. They are created according to Algorithm 1.

Require: Let Γ be the set of all episodes

for all $X = (x_1, \ldots, x_K)$ and $Y = (y_1, \ldots, y_N)$ in Γ,

$class(Y) = class(X)$ **do**

if $D(X, Y) \leq \Delta * length(X)$ **then**

for all $i = 1$ to $K - 1$ **do**

if $y^{art}_{\Phi_{y,i+1}} \in [x^{art}_{i+1} - \delta, x^{art}_{i+1} + \delta]$ **then**

Add an edge from x_i to $y_{\Phi_{y,i+1}}$

end if

end for

end if

end for

Algorithm. 1 Building the graph $\mathcal{G}_{G-Mem}$

For any pair of episodes X and Y of the same class we create an oriented edge from a given x_i to a given y_j if y_j is similar to x_{i+1} from an articulatory point of view (i.e, y_j^{art} is

similar to x_{i+1}^{art}) as defined previously:

$$\Phi_{y,i+1} = j \tag{3}$$

$$y_j^{art} \in ATI_{x_i} = [x_{i+1}^{art} - \delta, x_{i+1}^{art} + \delta] \tag{4}$$

In addition we impose that the asymmetric distance $D(X, Y)$ falls below a given threshold (proportional to the length of X):

$$D(X, Y) \leq \Delta * length(X) \tag{5}$$

The goal is to prohibit the memory from switching between episodes, which are globally very different (from an articulatory point of view) because it could lead the memory to produce unrealistic articulatory trajectories. As an example, consider the movements of the tongue tip, which might rise or fall during the production of a given phoneme. Although these trajectories are very different, it is likely that the tongue tip can reach similar positions midway through the fall and rise. Combining the fall and rise could possibly lead to a degenerate trajectory.

At the episode boundaries the memory is only subject to the articulatory continuity requirement expressed by equation (4). Let $Z = (z_1, z_2, \ldots, z_P)$ be the episode, which was observed after X. Then, an edge from x_K to the first observation w_1 of any episode W of any class is created if $w_1^{art} \in ATI_{x_K} = [z_1^{art} - \delta, z_1^{art} + \delta]$. If the episode X is the last of a record, its natural articulatory target is unknown and equation (4) cannot be satisfied; thus no edge to any other episode is possible. Note that a C-Mem only accounts for these transitions between episode boundaries. So, a C-Mem can be seen as a particular case of G-Mem for which Δ is set to zero.

C. An example

Figure 1 shows an example of a G-Mem built from a set of two episodes X and Y of the same class. The top two graphs illustrate the DTW mapping of Y onto X, and of X onto Y. Lines are drawn between aligned observations of X and Y. A thick line indicates the

aligned observations are within δ of each other, i.e., they are similar from an articulatory point of view, while a thin line indicates that the articulatory positions are very different. The bottom graph shows the resulting episodic memory with all possible transitions from X to Y and from Y to X. The resulting transitions within the memory are represented by the plain arrows from any x_i to x_{i+1} and from any y_j to y_{j+1} resulting from the mappings of X and Y onto themselves (not shown in the figure). The dotted arrows from any x_i to y_j result from the mapping of Y onto X while those from any y_j to x_i result from the mapping of X onto Y.

While a C-Mem would have contained only two articulatory trajectories X^{art} and Y^{art}, the resulting G-Mem can simulate six additional trajectories combining X and Y. Thus, the G-Mem is able to produce the following 8 articulatory trajectories according to a given input acoustic signal to be inverted:

1. $x_1^{art} \rightarrow x_2^{art} \rightarrow x_3^{art} \rightarrow x_4^{art}$ (X^{art})
2. $x_1^{art} \rightarrow x_2^{art} \rightarrow x_3^{art} \rightarrow y_5^{art}$
3. $x_1^{art} \rightarrow y_3^{art} \rightarrow y_4^{art} \rightarrow y_5^{art}$
4. $x_1^{art} \rightarrow y_3^{art} \rightarrow y_4^{art} \rightarrow x_4^{art}$
5. $y_1^{art} \rightarrow y_2^{art} \rightarrow y_3^{art} \rightarrow y_4^{art} \rightarrow y_5^{art}$ (Y^{art})
6. $y_1^{art} \rightarrow y_2^{art} \rightarrow y_3^{art} \rightarrow y_4^{art} \rightarrow x_4^{art}$
7. $y_1^{art} \rightarrow x_2^{art} \rightarrow x_3^{art} \rightarrow x_4^{art}$
8. $y_1^{art} \rightarrow x_2^{art} \rightarrow x_3^{art} \rightarrow y_5^{art}$

As explained in section III.B, note that when using the G-Mem, all the inverted trajectories start from the beginning of an episode and finish at the end of an episode.

D. Recovering the articulatory trajectories

As the nodes of the oriented graph $\mathcal{G}_{G-Mem}$ are bimodal (composed of an acoustic and an articulatory observation), each path within the graph corresponds to a particular articulatory trajectory and also to the acoustics that would have been produced by the articulatory trajectory. Thus, acoustic-to-articulatory inversion is performed by searching the path within $\mathcal{G}_{G-Mem}$ that best matches the acoustic speech signal to be inverted. The estimated articulatory trajectory is extracted from the articulatory observations of the visited nodes along this path.

All search paths can start only at nodes that represent the first observation of an episode. During inversion a breadth-first search is performed, applying the Viterbi algorithm. That is, at each step, the K-best paths obtained at the previous step are propagated through the $\mathcal{G}_{G-Mem}$ along the oriented edges defining allowed articulatory movements. The K-best propagated paths are kept while the others are discarded and the process is repeated up to the end of the speech signal.

The winning path is the one with the best acoustic score selected from all paths ending at a node that corresponds to the final observation of an episode. The score of each path is expressed as the sum of acoustic distances between the speech frames and the acoustic observations of the visited nodes along the path, computed on a predefined acoustic window $\mathcal{W}$.

IV. DATA AND INVERSION EXPERIMENTS

A. Corpora

All the experiments presented in this work were carried out on the following two corpora of synchronized acoustic speech signal and articulatory trajectories.

a. MOCHA We used the EMA corpus of MOCHA[?] . Two speakers of British English, one female (*fsew*) and one male (*msak*), were recorded reading 460 short phonetically balanced

British-TIMIT sentences. The audio is provided as waveforms sampled at 16 kHz, and each EMA receiver position is given as 2D coordinates in the mid-sagittal plane. We used seven sensors located at the lower incisors (li), upper lip (ul), lower lip (ll), tongue tip (tt), tongue body (tb), tongue dorsum (td), and velum (vl). The phonetic segmentation provided with the audio stream was used.

b. mdem We have recorded a French EMA corpus using an articulograph (AG500, Carstens Medizinelektronik). A male French speaker (*mdem*) was recorded uttering 400 phonetically balanced sentences. The audio is provided as waveforms sampled at 16 kHz, and the EMA data consist of 2D coordinates in the mid-sagittal plane. We used 6 sensors fixed in the mid-sagittal plane on the lower lip (ll), upper lip (ul), tongue tip (tt), tongue body (tb), tongue dorsum (td), and tongue back dorsum (tbd). The phonetic segmentation was obtained from a word-level transcription of the sentences, a dictionary containing several pronunciation variants for each word and a set of French monophone HMMs trained on several hours of speech and adapted to *mdem*'s voice. The segmentation was obtained by force aligning the phone HMMs onto the acoustics given the sentence word transcription and the pronunciation dictionary.

Each corpus was split into training, development and test sets. For MOCHA, care was taken that the selected utterances for each set corresponded exactly to the ones used by Richmond[?] , as this split was also used in Toutios & Margaritis[?] and Zhang & Renals[?] . Information about the different sets are given in table I. Note that the durations only account for usable speech (without the start and end silences). Figure 2 shows the distributions of the articulatory samples for each speaker and coil.

B. Feature extraction

The silences occurring at the beginning and end of each recording were first discarded, as the articulators can move unpredictably during such intervals. A Linear Predictive Analy-

sis[?] was performed on the speech signal using the HTK toolkit[?] . 12 cepliftered MF-PLPs[?] and the logarithmic energy of the signal comprised the acoustic feature vector extracted from every 25 ms speech frame shifted by 10 ms.

The articulatory data were first down-sampled from 500 to 100 Hz (for MOCHA) and from 200 Hz to 100 Hz (for *mdem*) to match the acoustic frame-shift. Then, all trajectories were low-pass filtered, in order to remove the recording noise, using a cut-off frequency of 20 Hz. A final data adjustment was performed in order to take into account the observed drift in the global trajectories throughout the recording sessions (see Richmond[?] for details). According to Richmond[?] , these variations can reflect speaker compensation/adaptation to the presence of the coils within the mouth, or may be due to measurement errors. We should note that the true underlying cause of these apparent long-term inconsistencies has not been properly and deeply investigated. In our study, we applied the same procedure proposed by Richmond[?] to remove these very long-term variations by subtracting a low-pass filtered version of the trajectory means from the EMA data.

C. Quality measurements

The quality of the recovered articulatory trajectories was evaluated using root-mean-square error (RMSE) that quantifies the difference between the measured x_i^{art} and estimated $f(x_i^{ac})$ articulatory coordinates:

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(f(x_i^{ac}) - x_i^{art})^2} \tag{6}$$

as well as Pearson's correlation between the values in the two trajectories. This is obtained by dividing the covariance of the two trajectories by the product of their standard deviations.

$$Cor = \frac{\sum_{i=1}^{N}(f(x_i^{ac}) - \overline{f(x^{ac})}).(x_i^{art} - \overline{x^{art}})}{\sqrt{\sum_{i=1}^{N}(f(x_i^{ac}) - \overline{f(x^{ac})})^2 . \sum_{i=1}^{N}(x_i^{art} - \overline{x^{art}})^2}} \tag{7}$$

D. Experiment design

We have implemented three different inversion methods: a codebook-based approach, as described in Suzuki *et al.*[?] , and two memory-based approaches (C-Mem and G-Mem) as described above. We have chosen to compare the memory-based approaches to this codebook-based approach, as they differ only in the manner of inferring the dynamics of the recovered articulatory trajectories. While the memories can model and use observed trajectories, the codebook only relies on continuity constraints.

In fact, this codebook method consists in looking up a set Γ_i of the N best entries in the codebook for each given acoustic signal sample Y_i. The best entries are the ones, which minimize the acoustic distance to Y_i. In fact, this distance is the average acoustic distance computed over a window of a predefined length $\mathcal{W}$. The articulatory trajectory is obtained by looking for the path through Γ_i that minimizes the weighted sum of the acoustic distances and squared distances between subsequent articulatory parameters. The reader may refer to Suzuki *et al.*[?] for details of the implementation.

We consider the recovery of the articulatory trajectories for each coil, and along both the x and y axes, as independent inversion problems. Thus, the experiments presented here consist of fourteen (for *fsew* and *msak*) and twelve (for *mdem*) distinct inversion problems. For each inversion problem, a dedicated codebook, C-Mem and G-Mem are built and optimized.

The parameters of the codebook are the length of the spectral window, and the weight of the articulatory constraints with regard to the acoustic distances. During the inversion, the 1000 best codebook entries are considered at any given time. For G-Mem, the parameters to be set are δ, the half ATI length, and Δ, the maximum allowed articulatory distance between two episodes X and Y for allowing the memory to switch from one to the other. Note that the parameter Δ is equal to zero for a C-Mem, so that each episode can only be combined with itself. An acoustic window was used to compute the acoustic distances. The length of this window is a parameter to be optimized. Euclidean distance is used for both acoustic and articulatory distance calculation.

Two types of experiment have been carried out: with and without phonetic constraints. Note that the G-Mem used phonetic segmentation only for building the memory, and the segmentation is not needed for inversion. The purpose of adding experiments with phonetic constraints is to ascertain what extra information is added for use of the phonetic knowledge to the different inversion methods. The reference phonetic segmentation is obtained by force aligning HMM acoustic models onto the speech signal according to the transcription of the uttered words. In the phonetically constrained mode, the estimated articulatory position at time t has to come from a codebook entry or an episode of the same phoneme as the one indicated by the segmentation at time t.

For the three systems, the parameters have been jointly optimized through a grid search optimizing RMSE on the development set. Since the articulators move in different ranges, different sets of parameters were obtained for all inversion problems. Globally, the length of the acoustic window $\mathcal{W}$ is approximately 150 ms for the codebook and 90 ms for the memories. δ ranges from a few hundredths of a millimeter (for the velum) to at most one millimeter, and Δ ranges from a tenth of a millimeter to two millimeters for different articulators. Finally, the memory search beam width is set to 10 000 for the memories and 1000 for the codebook.

V. INVERSION RESULTS AND ANALYSIS

A. Results

Figure 3 shows the results of the three inversion methods, with (grey) and without (black) phonetic constraints. The bars show the overall RMSE (means over the coils and x- and y-coordinates) in millimeters for each corpus; the RMSE values are indicated above the bars. The respective Pearson's correlations are given below the bars. The figure shows that the memory-based approaches always outperform the codebook-based method. This suggests the articulatory dynamics are modeled more effectively in the memory-based systems than by continuity constraints used by the codebook for speech inversion. It also illustrates how

much an episodic memory can benefit from the generalization capability of the G-Mem, as it always outperforms the C-Mem. The best Pearson's correlation scores were obtained using a G-Mem. Without any phonetic knowledge, an overall RMSE of 1.65 mm and a correlation of .714 were obtained on MOCHA with the proposed G-Mem, while an RMSE of 1.81 mm and 1.88 mm, and a correlation of .668 and .641 were obtained with the C-Mem and codebook, respectively. Using the phonetic segmentation of the test recordings, the RMSE decreased to 1.50 mm and the correlation increased to .757 for the G-Mem. The relative improvements due to the phonetic segmentation of the acoustic signal was roughly the same across the three inversion methods and across the three speakers.

Figure 4 shows the tongue tip movements (thick curves) along the vertical (up/down) axis recovered by each of the three approaches from a two second long speech signal corresponding to the French sentence "juste quelques extrémités de branches gelées" (*only a few frozen branch tips*). For each of the three graphs, the reference trajectory is provided as the thin curve and the estimation errors are emphasized as filled areas between the estimated and reference trajectories. Though optimized, the dynamic articulatory constraints of the codebook-based approach do not prevent discontinuities, as the recovered articulatory trajectory is very jerky compared to the smooth and continuously varying reference trajectory. The results obtained by both the C-Mem and G-Mem are visibly better. One might have expected smoother trajectories using the C-Mem, as the result is expressed as a concatenation of natural articulatory episodes. Through deeper analysis of the decoding paths within the C-Mem, we have noticed that most of the time the C-Mem does not contain episodes, which acoustically match the test signal well enough. In order to counteract this lack of good episodes, the C-Mem tends to select many short episodes. Indeed, the sum of the acoustic distances of short episodes, which locally match the test signal well, is usually smaller than the acoustic distance of a longer episode with partial acoustic mismatches. Finally, the G-Mem succeeds in estimating the articulatory movements accurately. The combination of episodes significantly reduces the estimation error. Furthermore, the articulatory dynamics embodied in the G-Mem transition graph contribute to the naturalness of the resulting

articulatory movements.

B. Statistical significance

We have conducted a statistical analysis in order to quantify the confidence in our results and also in the improvements. We have applied a bootstrap method proposed in Bisani & Ney[?] that was originally designed for speech recognition performance evaluation, but which is equally applicable here. This method relies on a bootstrap replication by creating N pseudo data sets X_i from the original data set X with replacements.

A pseudo data set can thus contain several or no samples of the original data set. The statistics $Stat_i$ are then calculated over each pseudo data set X_i. For a large N the distribution of the statistics is approximately Gaussian and thus, the true statistic lies with 99% confidence within the interval $\overline{Stat_i} \pm 3\sigma$, where $\overline{Stat_i}$ and σ are the mean and standard deviation of the statistics distribution, respectively. See Bisani & Ney[?] for more details on this statistical analysis.

We have computed the 99% confidence intervals of the RMSE, generating 100 000 pseudo data sets for each experiment. These intervals are reported in figure 3 as error bars on each bar. These confidence intervals only represent the range of performance we would obtain when applying one of the three methods on a new data set of one of the three speakers.

In addition, table II gives the probabilities of improvement, as well as the 99% confidence intervals of the expected improvement of each method over the others, for the three corpora. Each cell contains an integer, which is the probability that the method corresponding to the row outperforms the method corresponding to the column, as well as the 99% confidence interval of the expected improvements, expressed in percent. These numbers confirm the performance boost of the memory-based approaches over the codebook-based method and the superiority of the G-Mem over the C-Mem.

C. Smoothness and naturalness of articulatory trajectories

We argued in the introduction that the episodic memories are well-suited models for articulatory inversion, as they are able to preserve the dynamics of the episodes. We offer here an analysis that gives more insight on this point. Many proposed solutions to the inversion problem include low-pass filtering the recovered articulatory trajectories. This filtering aims to remove rapid changes (usually of low amplitude) that come from errors during the acoustic-to-articulatory mapping, and that can be considered noise. Indeed, all articulators have their proper velocity and can move more or less quickly, but they all move continuously. Comparing the recovered trajectories with their smoothed version is therefore one possible way of assessing how well the articulatory dynamics have been approximated.

For each coil, in both directions, we have determined the best cut-off frequency minimizing the RMSE on the test set. Figure 5 summarizes the best case scenario smoothing results. As in Figure 3, the bars represent the RMSE, and Pearson's correlations are provided below the bars. The numbers in square brackets are the relative percentage improvements over the non-smoothed trajectories.

The most obvious effect is that the codebook-based method significantly benefits from this filtering. Its RMSE improves by approximately 10%, while the improvements for the memory-based approaches do not exceed 3%. The same trend can be observed for the Pearson's correlations. This indicates that the memory-based approaches really do take advantage of the observed dynamics of the episodes. Applying articulatory continuity constraints during the inversion does not yield the same benefits. Note that similar observations have been reported in Toda *et al.*[?] . Indeed, the authors proposed an MLE-based mapping that accounts for correlation between frames. They reported significant improvements over their baseline GMM-based mapping, but also showed that the low-pass filtering effect became negligible. A relative improvement of 9.36% for RMSE was obtained over the baseline GMM-based mapping with a low-pass filter, but only .72% over the MLE-based mapping.

D. Computational resource requirement

Let N_s be the number of samples contained in the training set, K_s be the search beam width, and B be the average branching factor of the G-Mem, i.e. the average number of transitions allowed from any sample.

At any time t, the codebook method needs to compute the acoustic distance between the current test sample and all the N_s training samples. Then, the K_s best training samples are selected and the dynamic articulatory constraints are computed between each of them and each of the K_s best hypotheses computed at time $t-1$. The complexity $\mathcal{O}_{codebook}$ is then:

$$\mathcal{O}_{codebook} = N_s + (K_s \times K_s) \approx K_s^2 \tag{8}$$

Using a beam width of 1000 leads to one million dynamic articulatory constraints computed for each test sample.

At any time t, the G-Mem needs to propagate all of the K_s best hypotheses computed at time $t-1$. Setting the average branching factor to B, the complexity $\mathcal{O}_{G-Mem}$ is:

$$\mathcal{O}_{G-Mem} = K_s \times B \tag{9}$$

The average branching factor B is about 250 for the G-Mem. Using a beam width of $K_s = 10\,000$ leads to 2 500 000 paths to be investigated. Actually, these paths end in a subset of all the N_s acoustic/articulatory frames contained in the memory. Thus, the maximum number of acoustic distances to be computed each time is N_s. Unlike the codebook approach, propagating the path here does not require any distance computation.

So, the two different beam widths used for the codebook-based and G-Mem-based methods roughly lead to the same computational resource requirements.

Figure 6 shows the evolution of the running time for both the codebook-based and G-Mem-based methods on the *msak* test set as a function of the search beam width. The plot is logarithmically scaled along both axes. As stated in equations (8) and (9), we can verify that the complexity is exponential for the codebook and linear for the G-Mem with respect

to the search beam width.

Figures 7 shows the evolution of the overall RMSE for both the codebook based and G-Mem based methods on the test set of *msak* as a function of the search beam width. Varying the beam width from 1000 to 8000 does not lead to a significant RMSE improvement. Therefore, a beam width of 1000 appears to be a good compromise between performance and execution time.

VI. DISCUSSION

We have proposed an episodic memory solution for acoustic-to-articulatory inversion. This model has some similarities with a codebook; in particular, it relies on a one-to-one correspondence between articulatory and acoustic observations. However, unlike a codebook, an episodic memory accounts for the temporal dimension and is thereby able to preserve the articulatory dynamics of the episodes. We have also proposed an algorithm, which allows the memory to combine different episodes during the inversion to simulate many other episodes than the ones in the database. Through the presented experiments we have shown that the trajectories produced by the memory-based models are better than those produced by the codebook.

In addition, the estimation errors using a G-Mem are very encouraging compared with the state of the art. Hiroya & Honda[?] reported RMSEs of 1.50 and 1.73 mm with and without phonetic segmentations, respectively, using a HMM-based production model. However, we cannot directly compare our results with theirs, as they used a Japanese database. On MOCHA, Toda *et al.*[?] used Gaussian mixture models to map the acoustic space onto the articulatory space. They reduced the RMSE from 1.58 to 1.40 mm by applying a maximum likelihood estimation (MLE) of the dynamic features. Zhang & Renals[?] obtained an RMSE of 1.71 mm using a trajectory HMM. They included velocity features in their acoustic front end and performed speech recognition prior to inversion to provide their system with a phonetic segmentation. Even without phonetic segmentation, the G-Mem performs slightly

better. Moubayed & Ananthakrishnan[?] 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 MLE of the dynamic features to improve the results. An RMSE of 1.52 mm was reported using this method. Finally, Richmond[?] reported an RMSE of 1.40 mm using trajectory mixture density neural networks. We share with Zhang & Renals[?] , Richmond[?] and Moubayed & Ananthakrishnan[?] the same train, dev and test sets and all reported RMS errors range from 1.40 to 1.73 mm on this corpus. We can claim that the proposed G-Mem, with an RMSE of 1.65 mm, performs as well as the machine learning based approaches.

The proposed G-Mem can be improved in the following direction. A local linear regression as proposed by Moubayed & Ananthakrishnan[?] could further improve our results, as the G-Mem can produce unseen trajectories but it is unable to precisely map an acoustic frame onto the articulatory space if this acoustic frame does not belong to the memory.

Many studies have shown significant improvements using the reference phonetic segmentation of the acoustic signal to be inverted. This suggests that the inversion takes advantage of phonetic knowledge: if the phonetic content of the speech signal to be inverted is available, the articulatory movements can be recovered more precisely. As for the HMM-based method[?] , a first phone recognition pass could be performed with well-trained acoustic HMMs in order to provide the G-Mem with an accurate phonetic segmentation. 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 search paths and decrease the RMSE to reach results similar to the ones we have obtained using the reference phonetic segmentation.

Finally, we define the episodes as the acoustic and articulatory realizations of particular phonemes. That is, we implicitly hypothesize that the articulatory trajectories can be segmented into nonoverlapping elementary units that are phones, and that this segmentation is the same for all studied articulators. However, Ananthakrishnan & Engwall[?] recently

proposed an automatic method for segmenting the articulatory movements into articulatory gestures. This method accounts for the notion of critical articulators. That is, the production of a particular phoneme depends mainly on the movements of few articulators (the critical articulators), which have to reach precise articulatory target positions. The other articulators can move more freely and thus can anticipate the production of the next phoneme to be uttered. Then, the movements of our articulators overlap with each other and contribute to coarticulate all the uttered sounds. We believe that such an articulatory based segmentation might lead to better inversion results than those presented in our study, which used a purely acoustic segmentation.

VII. CONCLUSION

In this paper, we have shown that the concept of episodic memory can help to enhance the articulatory-to-acoustic mapping. It has not been our aim to provide an inversion method that beats the state of the art. Instead, we have focused on the added value of this approach to address the inversion problem. Both C-Mem and G-Mem integrate the articulatory dynamics in the memory structure, which is essential to fully resolve the inversion problem. Indeed the articulatory dynamics reflect the articulatory strategy, which is chosen by a speaker to utter a sequence of phones. Regarding the physical constraints of each articulator as well as the phonetic content of the sentence, the speaker determines the best manner, among many, to articulate and coarticulate each sound. Thus, the articulatory dynamics help to resolve the non-uniqueness of the acoustic-to-articulatory inversion problem. The dynamics are actually those naturally produced by the speaker and not the result of an *a posteriori* computational smoothing process. Moreover, as the size of the corpus is very limited and does not include all possible articulatory transitions, the G-Mem was provided with a generalization mechanism. The G-Mem is able to combine different observed articulatory strategies according to the speech signal to be inverted. Then, even if a particular phone sequence has never been observed, the memory can predict how the

speaker would utter this phone sequence based on experienced and stored episodes.

Acknowledgment

The authors are thankful to Yves Laprie for providing the articulograph machine facility.

TABLE I. Overview of the corpora.

Corpora	Sets	Durations	Sentences	Phones
fsew	train	16 min 35 sec	368	11179
	dev	1 min 57 sec	46	1324
	test	2 min 5 sec	46	1457
msak	train	13 min 59 sec	368	11179
	dev	1 min 41 sec	46	1324
	test	1 min 45 sec	46	1457
mdem	train	8 min 24 sec	319	6355
	dev	1 min 2 sec	40	817
	test	1 min 3 sec	40	814

TABLE II. Probabilities of improvement and 99% confidence intervals of expected improvements (both expressed in percentage). Each cell contains the probability that the method corresponding to the row outperforms the method corresponding to the column, as well as the 99% confidence interval of the expected improvements.

	Codebook	C-Mem	G-Mem
Codebook	-	fsew: 0 [-7;-3] msak: 0 [-7;-1] mdem: 0 [-7;-1]	fsew: 0 [-14;-10] msak: 0 [-15;-11] mdem: 0 [-16;-9]
C-Mem	fsew: 100 [3;7] msak: 100 [1;7] mdem: 100 [1;7]	-	fsew: 0 [-10;-6] msak: 0 [-12;-7] mdem: 0 [-10;-2]
G-Mem	fsew: 100 [10;14] msak: 100 [11;15] mdem: 100 [9;16]	fsew: 100 [6;10] msak: 100 [7;12] mdem: 100 [2;10]	-

List of Figures

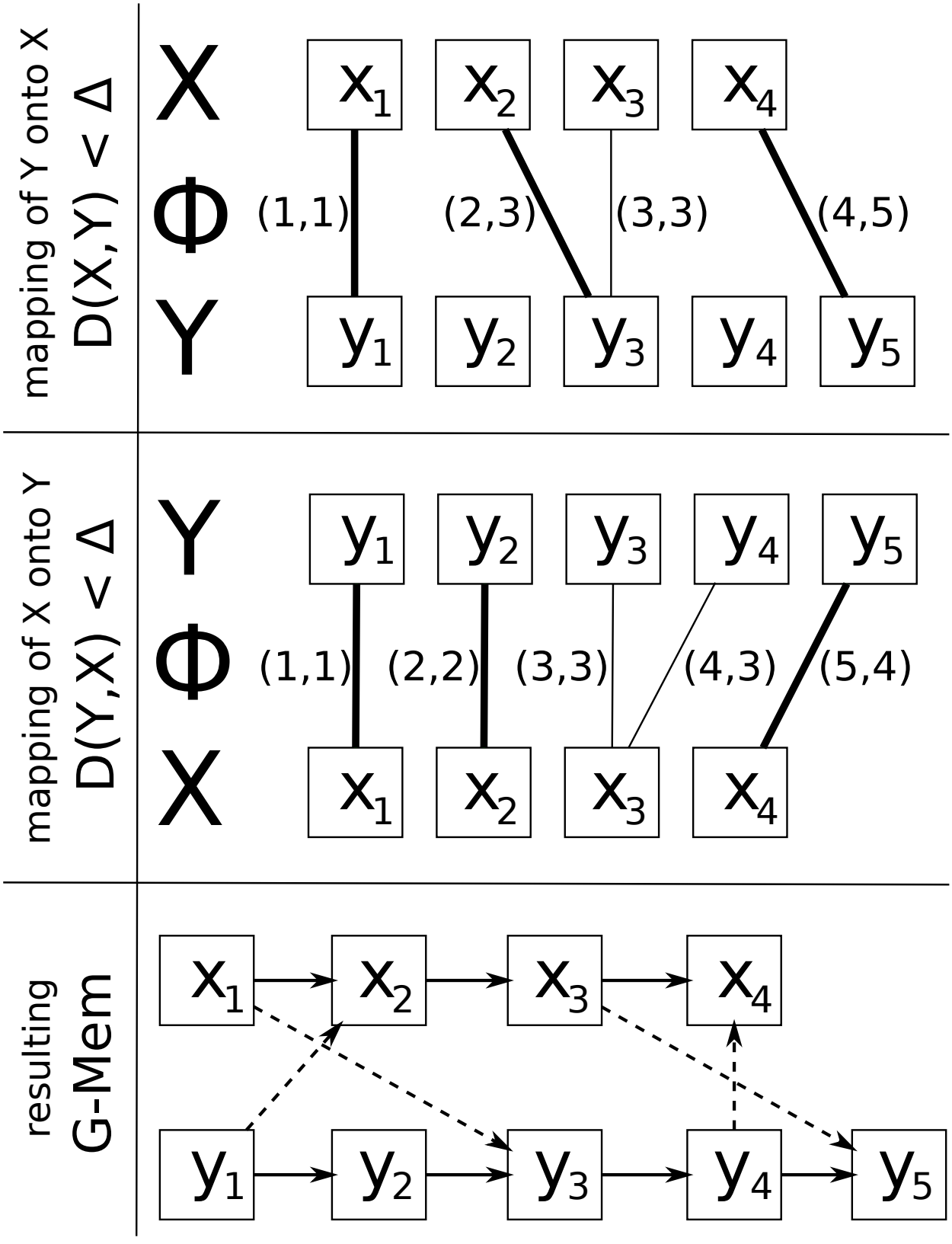
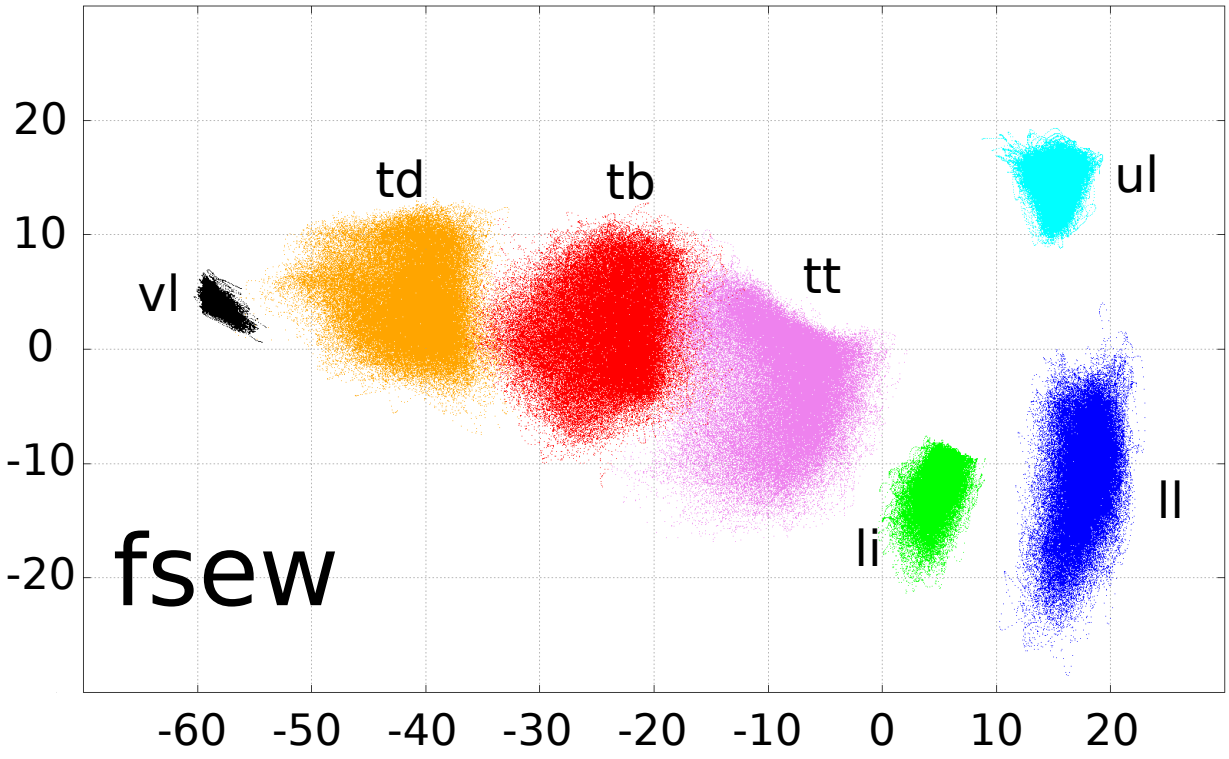
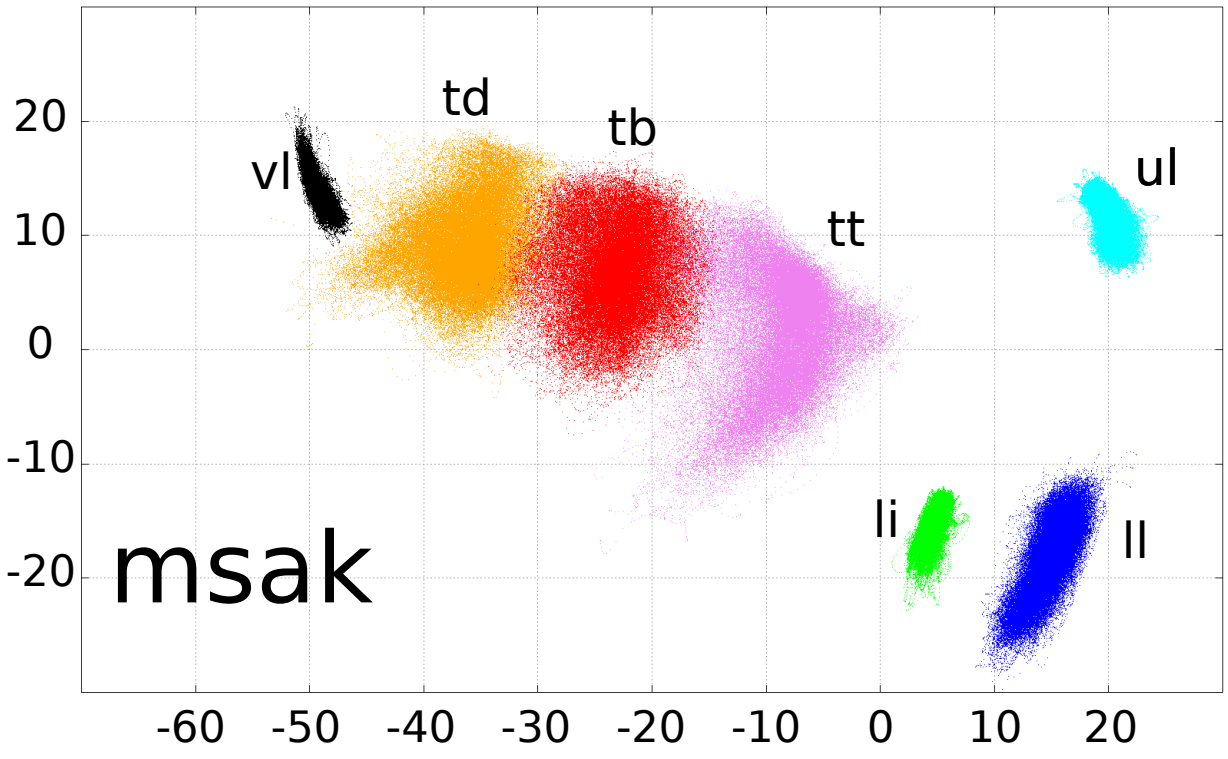

mapping of Y onto X
D(X,Y) < Δ
X
Φ
Y
x1
x2
x3
x4
y1
y2
y3
y4
y5
(1,1)
(2,3)
(3,3)
(4,5)
mapping of X onto Y
D(Y,X) < Δ
Y
Φ
X
y1
y2
y3
y4
y5
x1
x2
x3
x4
(1,1)
(2,2)
(3,3)
(4,3)
(5,4)
resulting
G-Mem
x1
x2
x3
x4
y1
y2
y3
y4
y5

FIG. 1.


fsew
vl
td
tb
tt
li
ul
ll
20
10
0
-10
-20
-60
-50
-40
-30
-20
-10
0
10
20


msak
vl
td
tb
tt
li
ul
ll
20
10
0
-10
-20
-60
-50
-40
-30
-20
-10
0
10
20

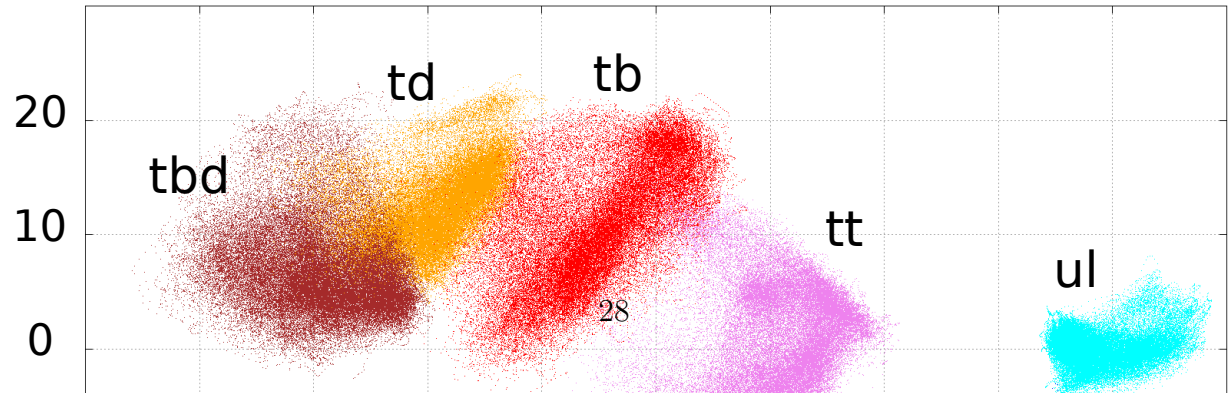

tbd
td
tb
tt
ul
20
10
0

FIG. 2.

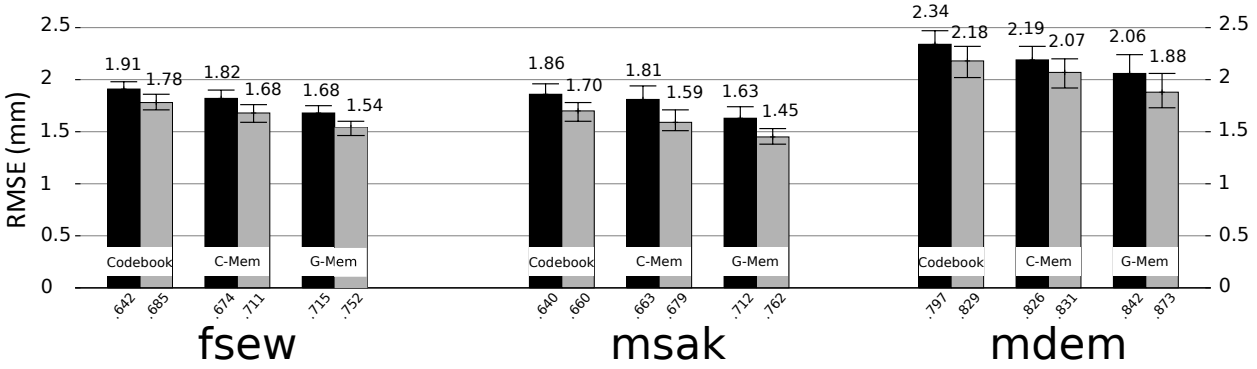


FIG. 3.

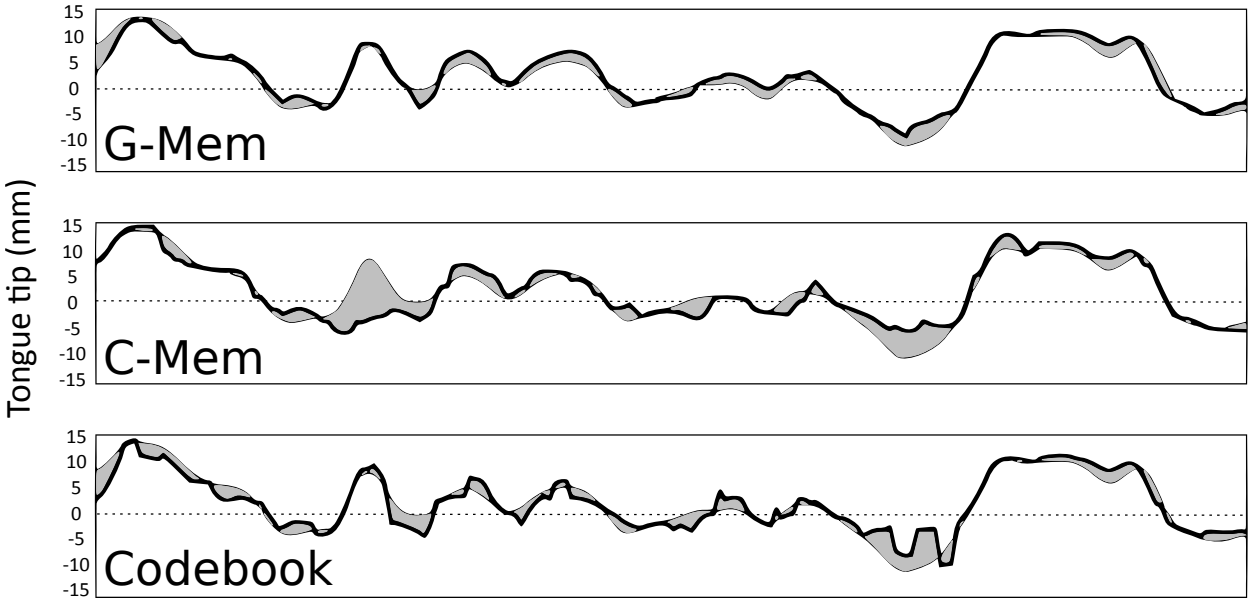


FIG. 4.

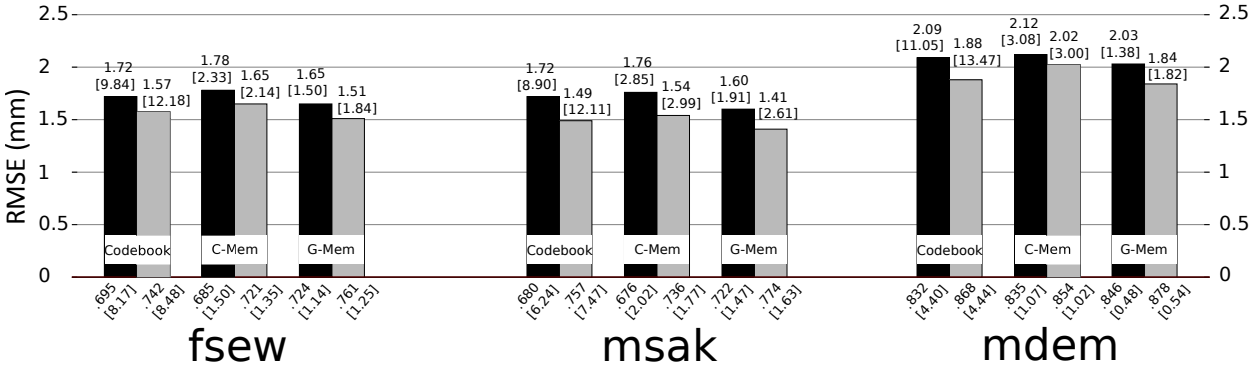


FIG. 5.

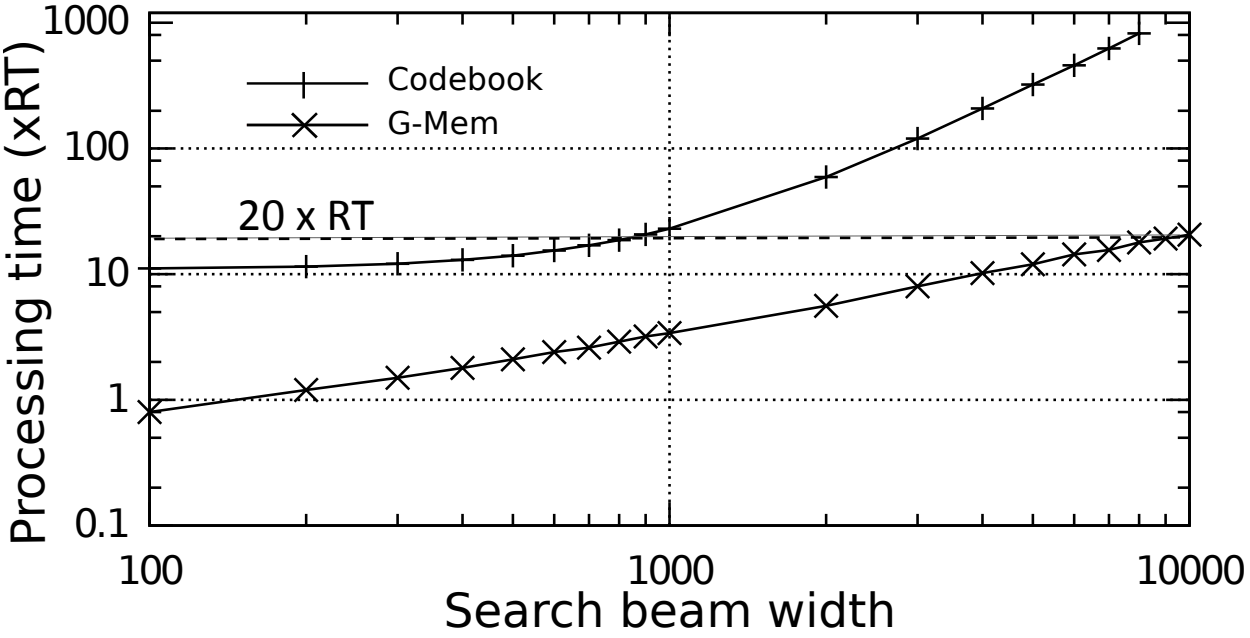


FIG. 6.

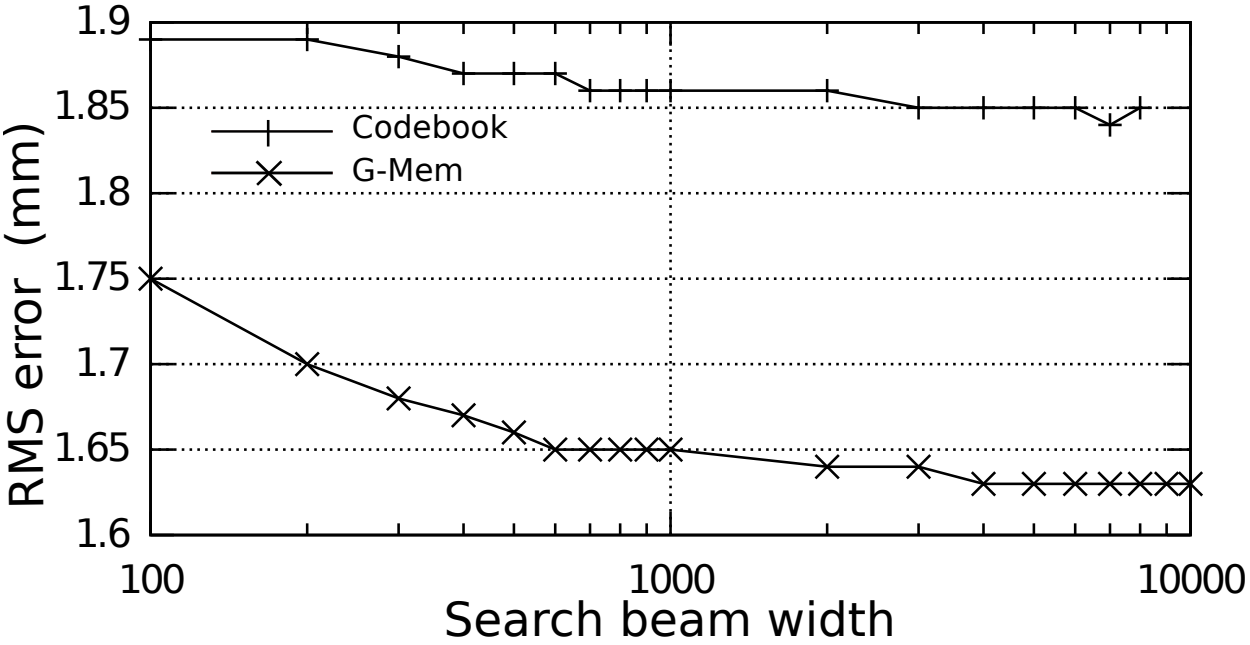


FIG. 7.