Modeling the articulatory space using a hypercube codebook for acoustic-to-articulatory inversion

Slim Ouni

Perceptual Science Laboratory

University of California - Santa Cruz, CA 95064 USA[a)]

Yves Laprie

LORIA - UMR 7503

BP 239 - 54506 Vandœuvre-lès-Nancy Cedex France[b)]

Received:

Running title: Modeling the articulatory space for acoustic-to-articulatory inversion

Short title: Acoustic-to-articulatory inversion using hypercubes

[a)] Electronic address: slim@fuzzy.ucsc.edu

[b)] Electronic address: laprie@loria.fr

ABSTRACT

The acoustic to articulatory inversion is a difficult problem mainly because of the non-linearity between the articulatory and acoustic spaces and the non-uniqueness of this relationship. To resolve this problem, we have developed an inversion method that provides a complete description of the possible solutions without excessive constraints and which retrieves realistic temporal dynamics of the vocal tract shapes. We present an adaptive sampling algorithm to ensure that the acoustical resolution is almost independent of the region under consideration in the articulatory space. This leads to a codebook that is organized in the form of a hierarchy of hypercubes, and ensures that, within each hypercube, the articulatory-to-acoustic mapping can be approximated by means of a linear transform. The inversion procedure retrieves articulatory vectors corresponding to acoustic entries from the hypercube codebook. A non-linear smoothing algorithm together with a regularization technique is then used to recover the best articulatory trajectory. The inversion ensures that inverse articulatory parameters generate original formant trajectories with high precision and a realistic sequence of the vocal tract shapes.

PACS numbers: 43.70.Bk, 43.70.Aj

I. INTRODUCTION

Estimating vocal tract shape from speech signal has received considerable attention because it offers new perspectives for speech processing. Indeed, recovering the vocal tract shape would enable knowing how a speech signal has been articulated. This potential knowledge could give rise to a number of breakthroughs in automatic speech processing. For speech coding, this would allow spectral parameters to be replaced by a small number of articulatory parameters[39] that vary slowly with time. In the case of automatic speech recognition the location of critical articulators could be exploited[36] to discard some acoustic hypotheses. For language acquisition and second language learning this could offer articulatory feedbacks. Lastly, in the domain of phonetics, inversion would enable knowing how sounds were articulated without requiring medical imaging or other measurement techniques.

Most of the acoustic-to-articulatory methods rest on an analysis-by-synthesis approach. Indeed, among the variety of acoustic signals the ear is exposed to, speech is one of the few for which a sufficiently good numerical simulation (including the deformations of the vocal tract geometry together with the resolution of the acoustical equations) is available. One of the essential issues is to evaluate the precision required by this numerical model to guarantee that sufficiently accurate and relevant information is recovered so that it can be interpreted from a phonetic point of view. The precision issue concerns both the geometric measures giving the vocal tract shape and the dynamic commands that control the vocal tract shape over time.

An inversion method as neutral as possible with respect to the articulatory behavior of the vocal tract should be devised. That is to say the inversion method should not provide particular solutions and omit other solutions. We will then study how modeling errors and with external constraints provided by either phonetic, physiological knowledge, X-ray data or the tracking of visible articulators influence results.

Using an analysis-by-synthesis approach means that the articulatory-to-acoustic mapping is used directly or indirectly in the inversion. Generally, the mapping is used indirectly, either explicitly, in the form of a table giving acoustic parameters (in general formants frequencies) for well chosen articulatory points (see Larar et al.[22] for instance), or implicitly, in the form of neural networks[40]. An articulatory synthesizer built on an articulatory model is generally used to generate a table (also called codebook) or to provide the training data for the neural network. The quality of the table construction strongly influences inverse solutions recovered since these trajectories use

vectors of articulatory parameters of the table.

In our work, we want to develop an inversion method that easily enables the evaluation of constraints that can be added to reduce the under determination of the problem, independdently of the inversion algorithm itself. The evaluation comprises both the acoustical distance between re-synthesized and measured acoustic parameters together with the realism of the temporal dynamics of the vocal tract shapes recovered. This thus requires that a complete description of the possible inverse solutions is potentially easily available. For these reasons, we have developed an adaptive sampling algorithm to ensure that the acoustic resolution is almost independent of the region under consideration in the articulatory space. The adaptive sampling leads to a codebook that is organized in the form of a hierarchy of hypercubes, and ensures that, within each hypercube, the articulatory-to-acoustic mapping can be approximated by means of a linear transform. During the inversion, all the articulatory points that produce measured formants have to be found. Then, the best articulatory trajectories, including one of these points at each time of the utterance to be inverted have to be constructed. This amounts to finding the best paths given the articulatory points recovered at the first step.

In this paper, we present the difficulties of the inversion and how the problems are usually resolved. Then, we present our inversion method in detail: we start by presenting the hypercube codebook generation method, the inversion using this codebook and the variational method to retrieve the temporal dynamics of the vocal tract shapes along time.

II. THE INVERSION PROBLEM

The articulatory-to-acoustic mapping can be defined as

$$A(\mathbf{x}) = \mathbf{b} \tag{1}$$

where $\mathbf{x}$ is an articulatory vector which gives the vocal tract shape and $\mathbf{b}$ is an acoustic vector, here the first three formants. $A : X \mapsto B$ is a non-linear and many-to-one mapping from X, the articulatory domain, into B the acoustic domain. The inverse mapping consists of retrieving $\mathbf{x}$ from $\mathbf{b}$. $\mathbf{b}$ is evaluated from the speech signal, and therefore is only an approximation of the real formants. Indeed, formants are obtained by a formant tracking algorithm (see Laprie and Berger[20] for instance) that searches for the best interpretation of the spectral peaks in terms of formants.

In addition, the articulatory synthesizer represented by A is based on a $2D$ articulatory model, which is an approximation of the mid-sagittal section of the vocal tract, and the reconstruction of

the $3D$ vocal tract is made by an approximate transformation [5,18,32]. It should be also noted that, the articulatory model does not fit exactly the geometry of the particular speaker to be analyzed. Existing adaptation methods are based either on the determination of factor scales applied to the global length of the vocal tract or to mouth and pharynx sizes[16,28]. More precise adaptation methods (for instance in Mathieu and Laprie[25] that adjusts the motionless contour of the vocal tract) can be applied only when images of the vocal tract are available. This means that it is nearly impossible to obtain an articulatory model that exactly represents speaker's vocal tract since this kind of adaptation only concerns the static and not the dynamic characteristics of the articulatory model.

Moreover, the measurement space, i.e. frequencies of the first three formants, is fairly under dimensioned compared to the object space since the number of articulatory parameters is greater than that of acoustic parameters. Therefore, there is an infinite number of solutions for one 3-tuple of formant frequencies as shown by Atal et al.[1].

These reasons explain why there does not exist any direct inversion method and why optimization methods are often used to tackle this problem as they enable the exploration of the solution space. The optimization based methods act on articulatory parameters or area functions to minimize an acoustic or spectral distance between generated and measured acoustic parameters. Generally, one considers that a solution, or at least a local optimum, is found when the gradient of the cost function vanishes. Usually, optimization methods rely on some iterative scheme that requires the knowledge of initial solutions. In the case of acoustic-to-articulatory inversion, the initial solutions are obtained by searching a codebook, or by means of a artificial neural network trained on selected articulatory and acoustic data.

As the solution space is potentially vast and solutions possibly not realistic from a phonetic point of view, constraints can be incorporated to focus the exploration in articulatory regions of interest. Schoentgen and Ciocea[38] introduced a local constraint based on either kinematic or potential energy to select one solution at each step of the inversion. More generally, a common solution used to address ill posed problems is to add a regularizing term. Sorokin et al.[43] chose a regularizing term that prevent inverse solutions to deviate too much from the neutral position of articulators. They particularly studied how to set-up the compromise between the discrepancy term, i.e. the acoustical distance with respect to original formants, and the regularizing term according to the error on formants measured and articulatory model. However, despite its interest with respect to the reduction of under determination this regularizing term may prevent correct vocal

tract shapes to be retrieved simply because they present high values for some articulatory parameters. Shapes close to those expected for /i/, /y/, /a/ or /u/ could thus be penalized. We therefore advocate for a dynamic regularizing term (see section VII A) that involves the evolution of articulatory parameters along time. The expected advantage is that this give rise to more natural and efficient constraints imposed onto the regularity of articulatory parameters, and not directly onto the values of these parameters. Furthermore, it provides and efficient manner to jointly improve the acoustical proximity with original acoustical data and the regularity of articulatory trajectories.

Neural network methods have also become very popular to address the inverse problem[2,19,31,33,41,42] because they propose an efficient way of exploring the solution space. However, it should be noted that these methods rely on an implicit sampling of the articulatory space for the training stage. Therefore, their main advantage lies instead in their ability to represent articulatory knowledge in a compact form rather than in their coverage of the articulatory space. The closer the training examples are to the solution the more accurate the result. For all the inverse methods, and particularly for neural based methods, because the sampling is implicit, the corpus of data used for training should be representative of the non-linearities of the articulatory-to-acoustic mapping. For these reasons, special attention must be paid to generating codebooks, or more generally, sampling of the articulatory space.

Once allowable inverse solutions have been found at every time of the utterance some minimum path algorithm must be applied to recover trajectories for every articulatory parameter. This search can be carried out by using dynamic programming[35,39] or neural networks[34] that were trained to identify dynamic patterns of the articulators. In many cases, the trajectories found can be optimized after choosing the startup solutions by means of genetic algorithms combined with a dynamic articulatory model [26] or more generally a gradient method[44].

III. WHICH ARTICULATORY MODEL FOR INVERSION?

The choice of the vocal tract representation is crucial since it determines the number of parameters to recover. As the acoustic data are generally the frequencies of the first three formants, an as-concise-as-possible description must be adopted to reduce the indeterminacy of the problem. However, the phonetic exploitation of articulatory models goes against models that describe the vocal tract with a very small number of parameters, e.g. some area function models, even if they enable an excellent frequency precision as that used by Schoentgen and Ciocea[38].

For this reason we accepted Maeda's model that approximates the sagittal slice of the vocal tract instead of an area function model[7,12–14,47] whose faithfulness with respect to the human vocal tract cannot be guaranteed. This model, as others, rely on the processing of vocal tract images (either X-ray images for Maeda[23] and Gabioud[15] or MRI for Badin[3] and Engwall[10]). Unlike purely geometric models, articulatory parameters correspond to deformations of the vocal tract produced by true speakers. Consequently, these models cover well the domain of vocal tract shapes that a human speaker can produce with a relatively small number of parameters - between 7 and 9. Note that the third dimension (section areas) must be approximated from the knowledge of the sagittal slice which is only 2D information which would not be the case with true 3D models (those of Badin and Engwall[3,10] for instance). However, we accepted Maeda's model because it derives from a sufficiently large number of sagittal slices, and consequently provides a good coverage of possible articulatory configurations, which is not the case for true 3D models that exploit only a small number of MRI images. Another strong point of Maeda's model is the possibility to adapt it to a new speaker easily by modifying pharynx and mouth sizes.

Maeda's model was constructed by applying a factor analysis method derived from principal component analysis to vocal tract contours[23,24]. These contours were extracted by hand from X-ray images of vowels and projected onto a semi-polar coordinate system that enables a 1D parameterization of contours. These measures were then centered and normalized before deformation modes were extracted by the factor analysis in the form of linear components. Each of the seven parameters of Maeda's articulatory model is allowed to vary over a range of $\pm 3\sigma$ (where σ is the standard deviation of that parameter). For convenience each parameter is normalized by dividing it by its standard deviation. Thus the normalized parameters all vary between -3 and 3. Seven factors (see Fig. 1) are used to describe vocal tract deformations because they cover more than 98% of the total variance. This means that the inverse transformation has to be applied to get the vocal tract shape from articulatory parameters. This inverse transformation involves the multiplication by the standard deviation of each articulatory parameter, the summation of the linear components, then the multiplication by the standard deviation of each geometrical measure of the sagittal slice and finally the addition of the average value of these geometrical measures.

IV. METHODS FOR GENERATING ARTICULATORY CODEBOOKS

A codebook is a collection of a vast number of vocal tract shapes given by articulatory or area function parameters indexed by their acoustic parameters. The acoustic parameters, generally the first three formants, are obtained by using an articulatory synthesizer. The articulatory space should be spanned so that the codebook represents all of the possible geometric configurations of the vocal tract.

We experimented three existing methods to generate a codebook. The first method is random sampling[6,22]. In this method, the codebook is generated by sampling articulatory parameters randomly. The inconvenience is that it does not respect the non-linearities of the articulatory-to-acoustic mapping. Therefore, the codebook does not reliably represent the actual density of the articulatory space. The second method for generating codebooks is the root-shape interpolation[22] . This method consists of sampling the articulatory space in a non-uniform manner by sampling the most probable regions, i.e. those corresponding to the most often observed vocal tract shapes. To do this, two root-shapes are chosen among predefined shapes corresponding to vowels. The intermediate shapes produced by moving from one root shape to another linearly in the articulatory space are then added to the codebook. Sorokin and Trushkin[44] used the same approach that allowed them to drastically reduce the size of the codebook to only 1900 nodes in the one-dimensional space of minimal cross-sectional area although their articulatory model comprise 17 parameters. The expected advantage of this method is that only realistic vocal tract shapes are taken into account. Preliminary experiments we carried out by using a similar approach applied to Maeda's model shown that this method suffers from two important weaknesses. The first is that there exists a possibly vast number of root shapes for each vowel or consonant. This results from the compensatory properties of the vocal tract geometry that probably are essential in the speech production process. In fact, in most, if not all cases, there is not a one-to-one relationship between an uttered sound and a particular vocal tract configuration. Therefore, it is difficult to guarantee that root shapes accepted to derive the articulatory sampling are the most appropriate ones. It means that realistic vocal tract shapes may be "missed" by this sampling method. A second inconvenience is that moving from one shape to another by varying articulatory parameters linearly does not give rise to linear transitions of formant frequencies. This explains why parts of the acoustic space are sparsely covered. In Fig. 2, the acoustic space (F1-F2, F1-F3 and F2-F3 planes) is plotted for random sampling and root-shape codebooks. It appears clearly that the

acoustic space produced with the root-shape method does not cover the whole possible acoustic space.

The third method for generating a codebook is the regular sampling of the articulatory space. The obvious weakness lies in the huge number of shapes generated even when the discretization is relatively rough. For instance, let us consider Maeda's model[23] that describes the vocal tract with seven parameters between -3σ and 3σ (σ is the standard deviation of the articulatory parameters). Using only 10 steps to describe each articulatory parameter leads to about 8.000.000 shapes after unrealistic shapes have been eliminated. And to obtain a fine regular sampling of the seven parameters with a relatively rough sampling step equal to $1/3\sigma$ would lead to $19^7 \approx 900$ million vocal tract shapes, which becomes unrealistic from the point of view of both construction time and the codebook size required. Linear or polynomial interpolation could be used to reduce the size of a regularly sampling method[4,9]. However, this would require further processing to evaluate the precision of this interpolation.

The examination of the acoustic spreading of these three codebooks (see Fig.2) together with preliminary inversion experiments have shown that, they do not present accurate coverage of both the articulatory space and the acoustic space.

V. HYPERCUBE CODEBOOK

A. Introduction

The difficulty of generating codebooks lies in the fact that the relations between articulator positions and acoustics are non-linear[8,11,45,46]. In fact, there are articulatory regions where a small variation in articulatory parameters produces a large variation of acoustic parameters. And conversely, there are some regions where a large variation in articulatory parameters does not produce any significant acoustic changes.

Our approach aims at densely discretizing the articulatory space only in the regions where the mapping is highly non-linear. For this purpose we use a hypercube structure to organize the codebook. In the next paragraphs, we describe how the codebook is generated.

B. Articulatory hypercubes

A hypercube of order N (N-hypercube) is a generalization in the N-dimensional space of a square in 2-dimensional space and cube in the 3-dimensional space. An N-hypercube is an N-dimensional convex polytope (N-polytope). An N-hypercube H_c is defined by its origin vertex $\mathbf{U_0} \in I\!R^N$ (i.e. the vertex with the lowest coordinates) and the length $\ell \in I\!R$ of one edge. We denote this hypercube by $H_c(\mathbf{U_0}, \ell)$:

$$H_c(\mathbf{U_0}, \ell) = \prod_{j=1}^{N} [u_0^j, \ell] \tag{2}$$

where $\prod$ is the Cartesian product, ℓ the hypercube edge length and $u_0^j \in I\!R$ is the j^{th} component of $\mathbf{U_0}$. We represent a hypercube by its vertices. Let $\mathbf{V_i}$ be one of these vertices. The j^{th} component v_i^j of $\mathbf{V_i}$ is calculated as follows:

$$v_i^j = u_0^j + \varphi_{ij}\ell \tag{3}$$

Where φ_{ij} is the j^{th} digit of the number i written in binary form including leading zeroes (see Fig. 3). As we can note in (3), the hypercube is defined simply in terms of the origin coordinates and the edge length.

C. The hypercube generation method

Regardless of the articulatory or area function model used, the parameters vary within a limited range. As mentioned above, the articulatory parameters of Maeda's model vary between -3 and 3. Therefore the codebook is inscribed within a root hypercube denoted by $H_c^1(\mathbf{U_0}, \ell)$. Sampling the articulatory space amounts to finding reference points that limit linear regions. However, as the articulatory-to-acoustic mapping is not represented in a closed form some heuristic exploration and linearity evaluation have to be designed. Charpentier faced the same problem to sample parameters of the area function proposed by Ishizaka et al.[14] and proposed choosing these points by calculating the curvature of formant trajectories along articulatory trajectories obtained by varying one parameter at a time[8]. Reference points were chosen at regularly spaced intervals along the curvature. This solution cannot be used in the case of Maeda's model because there are more parameters and, above all, articulatory parameters do not control almost independent regions of the vocal tract as in the case of an area function model. The four jaw and tongue parameters,

for instance, control the same region in the vocal tract and there are two levels of potential non-linearities (from articulatory parameters to the area function, and from the area function to the acoustic parameters).

Therefore we devised a heuristic linearity test and evaluate its figure of merit by measuring the deviation between formants obtained by synthesis and those obtained by interpolation from codebook points (see section V D). One of the issues is the choice of articulatory points where the deviation has to be calculated. Points can be chosen in each hypercube randomly, regularly with respect to each of the articulatory parameter or distributed according to another geometric strategy. This choice is important because most of the time spent for the codebook construction will be dedicated to evaluating linearity. Indeed, using only three, resp. four, steps to sample each articulatory parameter gives resp. $3^7 = 2187$ and 16384 tests. As the later solution would have led to an excessive construction time we accepted three regularly spaced samples for each articulatory parameter. In addition to hypercube vertices that are not considered, this corresponds to middle points of segments formed by any two vertices (Fig. 3). For each segment the middle point interpolation takes into account only the two vertices and no other vertex of the hypercube. This means the linearity was assessed more than once for the midpoints, depending how the two vertices are placed with respect to each other: one time for two contiguous vertices and 2^6 times for two vertices on the main diagonal, which correspond to the hypercube center. In all this gives $2^7 \times (2^7 - 1) = 8128$ linearity tests.

The test for linearity is carried out as follows: acoustic values, i.e. the first three formant frequencies, are linearly interpolated at the middle point between two vertices from the acoustic values calculated at these vertices and the result is compared against that directly given by the articulatory synthesizer, i.e.:

$$abs(\ \frac{F_a^i + F_b^i}{2} - f(\frac{p_a + p_b}{2})\ ) \leq \Delta\epsilon^i \qquad 1 \leq i \leq 3$$

where i is the formant number, $p_a,\ p_b$ are the two vertices, f represents the articulatory-to-acoustic mapping (the synthesizer), $\mathbf{F_a}$ and $\mathbf{F_b}$ the vector of the first three formants at the articulatory points p_a and p_b ($f(p_a) = \mathbf{F_a}$ and $f(p_b) = \mathbf{F_b}$), and $\Delta\epsilon^i$ the predefined linearity threshold for formant i. The test succeeds if the three inequalities hold, and then, the articulatory-to-acoustic mapping is considered to be linear in the hypercube. Otherwise this hypercube is split into 2^7 equal sub-hypercubes and the linearity test is repeated for every new hypercube. This procedure is repeated recursively until the hypercube edge becomes smaller than a predefined value or no non-linearity

higher than the predefined threshold exists anymore. $\Delta\epsilon^i$ can be set experimentally for the first three formants. An articulatory region represented by a hypercube is considered linear (i.e. the articulatory-to-acoustic mapping is linear), if the 8128 tests succeed, otherwise, this region is considered non-linear.

As the allowable articulatory space, i.e. the space where articulatory parameters yield an open vocal tract, does not fit exactly in the hypercube, there are vertices for which acoustic parameters cannot be calculated because they correspond to a vocal tract shape with a complete constriction. These vertices thus belong to forbidden regions (the term used by Atal et al.[1]). When forbidden vertices are found in a hypercube, the hypercube is decomposed in order to obtain hypercubes where all the vertices are allowable. Boundaries of forbidden regions are thus well defined (Fig. 3). Nevertheless, the risk is to create a huge number of small hypercubes to get a very precise boundary whereas this articulatory region is probably of little interest because it is not often reached by a human speaker. We accepted therefore not too small a hypercube lowest edge size below which the decomposition stops.

The result of these successive recursive decompositions is a hierarchical structure composed of hypercubes of different sizes; the bigger the hypercube, the more linear the articulatory-to-acoustic mapping within this articulatory region. We save in the codebook only the origin of the hypercube (in the articulatory space), the length of one edge and the acoustic values of the vertices. The advantage of the hierarchical structure is to accelerate the search procedure in the codebook.

D. Experimental evaluation of the hypercube codebook

We generated a first hypercube codebook ($CB1$), using the linearity test with threshold values as follows: $\Delta\epsilon$ = 50Hz for $F1$, 75Hz for $F2$ and 100Hz for $F3$. This hypercube hierarchy is composed of 390000 hypercubes. Then, we generated another codebook ($CB2$) using the linearity test threshold $\Delta\epsilon$ = 0.3 Bark. The number of the hypercubes is 128000. The average time spent for each linearity test, i.e. the calculation and comparison of 8128 middle points, thus represents a non negligible time. It turns out that large size hypercubes at initial stages of the hypercube construction are eliminated because several vertices are outside the allowable articulatory space.

To evaluate the quality of sampling we used the codebook to calculate acoustic values by interpolating them from codebook entries. The interpolation was applied with respect to the hypercube

center $\mathbf{P_0}$:

$$\mathbf{F_x} = \mathbf{F_0} + \mathbf{J_F}(\mathbf{P_0}).(\mathbf{P_x} - \mathbf{P_0}) \quad (4)$$

where $\mathbf{P_x}$ is the articulatory vector we calculate its acoustic image $\mathbf{F_x}$ for and $\mathbf{J_F}(\mathbf{P_0})$ is the Jacobian matrix of $\mathbf{F}$ calculated at $\mathbf{P_0}$ by taking first differences. $\mathbf{P_0}$ can be chosen as the center or the nearest vertex of the hypercube $\mathbf{P_x}$ belongs to. We randomly chose 4000 articulatory vectors, 1850 of them representing valid area functions, then used the codebook $CB1$ to interpolate the acoustic values corresponding to the valid articulatory vectors. The mean error, i.e. the difference between formant frequencies calculated by the articulatory synthesizer and those interpolated from the codebook, does not exceed 10Hz for $F1$ and $F2$, and 20Hz for $F3$. Compared to the margin of error accepted for the codebook test linearity (50Hz for $F1$, 75Hz for $F2$ and 100Hz for $F3$), it is clear that we have a good acoustic precision. For the second codebook $CB2$, that we used in the inversion experiments reported in section VIII, we evaluated the acoustic precision on a much larger number of articulatory points. We randomly chose 1,000,000 articulatory vectors, 641,846 of them representing valid area functions. As shown in Table I the accuracy is very good since the overall mean error is less than 8Hz. The precision obtained is better than that imposed during the codebook construction because, unlike the interpolation using the Jacobian matrix, the linearity test involves two vertices only to predict the unknown formant values. As it can be noticed there is no significant precision difference between formants despite the Bark scale that was used for linearity tests and should lead to a lower precision for F2 and F3. This is probably due to a certain redundancy between the linearity test applied to the three formants with different precisions. The most rigorous test is that for F1. Since the magnitude of formant frequency variations is roughly the same for the three formants the precision imposed on F1 gives the overall precision.

Tests reported above confirm the expected properties of our codebook. Its main characteristic is that it offers a quasi-uniform acoustic resolution because of the adaptive sampling. Therefore, the complete acoustic behavior of the articulatory model is accurately represented by this codebook.

VI. THE INVERSION METHOD EXPLORING THE SOLUTION SPACE

Our inversion method exploits the codebook by recovering the possible articulatory vectors for each acoustic entry of the signal to be inverted, i.e. the first three formants extracted at each time frame of the utterance by automatic formant tracking[20]. The second stage, i.e. the recovery of articulatory trajectories is described in § VII. For each acoustic entry, all the hypercubes whose

acoustic image contains the acoustic entry are considered. The acoustic image is overestimated as the rectangular parallelepiped defined by minimal and maximal values of F1, F2 and F3. As the inclusion in the considered hypercube is checked for each inverse solution (see §VI B) this weaker assumption does not introduce any solution outside the hypercube.

A. The inversion method

The hypercube codebook is used to retrieve the articulatory parameters corresponding to the acoustic entry. All the hypercubes whose acoustic image contains the acoustic entry are examined. From now on, we only consider one hypercube to describe the inversion process. Let $\mathbf{F}$ be the acoustic vector (represented by the first three formants) to be inverted. Let H_c be the hypercube which contains articulatory vectors giving the acoustic vector $\mathbf{F}$. Let $\mathbf{P}$ be an articulatory vector (represented by the seven parameters of Maeda's articulatory model) that we are looking for. Using the Jacobian calculated at a particular point $\mathbf{P_0}$ in the hypercube (the center for instance) we approximate $\mathbf{F}$ by:

$$\mathbf{F} = \mathbf{F_0} + \mathbf{J_F}(\mathbf{P_0}).(\mathbf{P} - \mathbf{P_0}) \tag{5}$$

where $\mathbf{J_F}(\mathbf{P_0})$ is the Jacobian matrix of $\mathbf{F}$ calculated at $\mathbf{P_0}$ and $\mathbf{F_0}$ is the acoustic vector corresponding to $\mathbf{P_0}$.

Thus, to perform the inversion, we have to solve the following equation:

$$\mathbf{F} - \mathbf{F_0} = \mathbf{J_F}(\mathbf{P_0}).(\mathbf{P} - \mathbf{P_0}) \tag{6}$$

The matrix form is:

$$\begin{bmatrix} F^1 - F_0^1 \\ F^2 - F_0^2 \\ F^3 - F_0^3 \end{bmatrix} = \begin{bmatrix} \frac{\partial F^1}{\partial \alpha_1} \frac{\partial F^1}{\partial \alpha_2} \cdots \frac{\partial F^1}{\partial \alpha_7} \\ \frac{\partial F^2}{\partial \alpha_1} \frac{\partial F^2}{\partial \alpha_2} \cdots \frac{\partial F^2}{\partial \alpha_7} \\ \frac{\partial F^3}{\partial \alpha_1} \frac{\partial F^3}{\partial \alpha_2} \cdots \frac{\partial F^3}{\partial \alpha_7} \end{bmatrix} \begin{bmatrix} P^1 - P_0^1 \\ P^2 - P_0^2 \\ \vdots \\ P^7 - P_0^7 \end{bmatrix} \tag{7}$$

where F^i, F_0^i are the components of $\mathbf{F}$ and $\mathbf{F_0}$, i.e. the i^{th} formant and P^i, P_0^i the components of $\mathbf{P}$ and $\mathbf{P_0}$. We chose the center of the hypercube as $\mathbf{P_0}$ because this guarantees that the underlying assumption of linearity is approximately verified everywhere in the hypercube with respect to this point. Equation (7) has the form:

$$A \cdot \mathbf{x} = \mathbf{b} \tag{8}$$

where A is the $(M \times N)$ Jacobian matrix, $\mathbf{b}$ and $\mathbf{x}$ are the acoustic and articulatory vectors. When M is less than N, A is singular and the $N - M$ dimensional space where vectors are transformed into zero is the null space. The general solution of Eq. (8) is given by a particular solution plus any vector from the null space. This means that adding a linear combination of the base vectors of the null space does not change formants. The SVD (*singular value decomposition*) method as described in Golub and Van Loan[17] gives one particular solution set, i.e. the one with the smallest norm $|x|^2$. Besides, SVD constructs an orthonormal base of the null space. The particular solution together with the base of the null space completely describes the solution space. In our case, as $M = 3$ (3 formants) and $N = 7$ (7 articulatory parameters), the null space dimension is generally 4. To retrieve all the solutions for a given articulatory precision, the null space must be determined and sampled.

B. Sampling the null space

Let P_{svd} be the particular solution given by the SVD method. A general solution is:

$$\mathbf{P_s} = \mathbf{P_{svd}} + \sum_{j=1}^{4} \beta_j \mathbf{v_j} \tag{9}$$

where $\{\mathbf{v_j}\}_{j=1..4}$ is an orthonormal base of the null space and $\beta_{j=1..4}$ the coordinates in this space. Furthermore, this solution must belong to the hypercube where the linearity assumption holds. Therefore this solution is acceptable if:

$$\mathbf{P_s} \in H_c \tag{10}$$

Let α^i_{inf} and α^i_{sup} define the maximum and minimum values of the i-th articulatory parameter in H_c (i.e., H_c is the Cartesian product $H_c = \prod_{i=1}^{7}[\alpha^i_{inf}, \alpha^i_{sup}]$). Then we have:

$$\alpha^i_{inf} \leq P^i_{svd} + \sum_{j=1}^{4} \beta_j v^i_j \leq \alpha^i_{sup} \qquad i = 1..7 \tag{11}$$

where v^i_j is the projection of the j^{th} basis vector of the null space onto the i^{th} articulatory parameter. The matrix form of Ineq.(11) is:

$$\begin{bmatrix} \alpha^1_{inf} \\ \alpha^2_{inf} \\ \vdots \\ \alpha^7_{inf} \end{bmatrix} \leq \begin{bmatrix} P^1_{svd} \\ P^2_{svd} \\ \vdots \\ P^7_{svd} \end{bmatrix} + \begin{bmatrix} v^1_1 & v^1_2 & v^1_3 & v^1_4 \\ v^2_1 & v^2_2 & v^2_3 & v^2_4 \\ \vdots & \vdots & \vdots & \vdots \\ v^7_1 & v^7_2 & v^7_3 & v^7_4 \end{bmatrix} \cdot \begin{bmatrix} \beta_1 \\ \beta_2 \\ \beta_3 \\ \beta_4 \end{bmatrix} \leq \begin{bmatrix} \alpha^1_{sup} \\ \alpha^2_{sup} \\ \vdots \\ \alpha^7_{sup} \end{bmatrix} \tag{12}$$

This system defines a 4-polytope, i.e. a bounded intersection of 4 half-spaces. To completely define this 4-polytope, we need to find all the extreme points of this domain, since the polytope solution is the convex hull of these points and determine the space contained in the polytope. As far as we know, this problem, which is simple in dimension 2 (i.e. finding the intersection of a square with a line) has not received any close form solution in the general case yet. This explains why we have developed this two step algorithm to approximate the intersection. In the first step the smallest 4-dimensional hypercube which contains the polytope is determined by linear programming. The second step consists in sampling this 4-dimensional hypercube and keeping samples that belong to H_c. The 4-dimensional hypercube is defined by its vertices which are given by the minimum and maximum values of β_i which satisfy inequalities of (12). The values of the β_i can be found by resolving the following eight linear programs:

(1) four linear programs defined by the inequalities of (12) to **maximize** $\beta_i \quad (i = 1..4)$, the objective function being the **maximization** of β_i.

(2) four linear programs defined by the inequalities of (12) to **minimize** $\beta_i \quad (i = 1..4)$, the objective function being the **minimization** of β_i.

By finding all the β_i, we can easily calculate the vertices of the 4-hypercube by replacing the β_i in Eq. (9). Then, this 4-dimensional domain is sampled and solutions that do not belong to the hypercube H_c, i.e. Eq. (10) are eliminated (see Fig.5).

For a given 3-tuple of formants and a hypercube whose image contains this 3-tuple of formants the number of inverse solutions directly depends on the sampling step of the null space. The smaller the sampling step, the smoother the trajectories recovered. We accepted 3 steps for each of the 4 dimensions of the null space which keeps the potential number of points at a reasonable value of $3^4 = 81$ while guaranteeing a sufficient smoothness of articulatory trajectories. Together with the hierarchical representation of the articulatory space, this null space exploration method provides a quasi-complete description of the inversion solution set.

We evaluated the acoustic precision of the inversion for 489 random 3-tuples of formants $F1$, $F2$ and $F3$ synthesized with the articulatory synthesizer. This yielded approximatively 4 million solutions. After resynthesis with the articulatory synthesizer, the acoustic values were compared with the original data. As shown in Table II, the accuracy is very good since the overall mean error is less than 11Hz. It should be noted that this error is appreciably smaller than the frequency threshold of the linearity test used to decompose hypercubes. This indicates that even if the linearity test is potentially incomplete it is relatively strict. It is important to note that during the

whole inversion process, as presented above, we did not use the articulatory synthesizer and all the inversion solutions were obtained by interpolation and sampling.

As shown above, this inversion method provides a quasi-exhaustive description of all the possible vocal tract shapes that give a 3-tuple of formants, and with a very little error on formant frequencies. This method thus enables the investigation of articulatory constraints that can be added to guide inversion and to recover realistic articulatory trajectories. The key point is that this inversion method enables a clear separation between the representation of the articulatory space and the incorporation of constraints or knowledge to guide inversion. Moreover, this inversion method provides potential tools to investigate articulatory variability of speech production and compensatory effects a speaker can exploit.

Fig. 6 and Fig. 7 give inversion results of two speech sequences [au] and [ui] in the articulatory space for one articulatory parameter only (jaw parameter) for sake of clarity. For each sequence, the left graph presents the solution without sampling the null space, i.e. particular solutions given by the SVD method, and the right graph presents the solutions obtained by applying SVD and sampling the null space. Clearly, the solutions obtained after sampling the null space more finely cover the articulatory space.

VII. RECOVERING ARTICULATORY TRAJECTORIES

Recovering articulatory trajectories consists of choosing at each time an articulatory vector among those obtained by the inversion presented above. This amounts to finding an ”articulatory path” expressing the temporal sequence of the vocal tract shapes during the utterance to be inverted. The resulting articulatory trajectory should vary “slowly” (variations of articulatory parameters are small during an average pitch period, i.e. approximately 10 ms) and generates spectra as close as possible to those of the original speech. This corresponds to the satisfaction of two criteria: proximity to acoustic data and smoothness of articulatory trajectories. In this section, we present the overall inversion algorithm that combines these two criteria and works as follows:
(1) The first step of the inversion consists of recovering all of the inverse articulatory solutions at each point in time of the utterance to be processed by exploring the codebook.
(2) In the second step a non-linear smoothing algorithm described below finds smooth articulatory trajectories from the knowledge of the sets of inverse points recovered at each point in time.
(3) The third step consists of regularizing the trajectories built by using the non-linear smoothing

algorithm. This regularization is achieved through a variational method.

The non-linear smoothing algorithm used in the second step is derived from a non-linear smoothing algorithm initially proposed by Ney[29] for post-processing results of F0 determination. Let $s(i)$ be the set of inverse points retrieved at time frame i, and $S = (s(i))$, $1 \leq i \leq N$ the sequence of these sets over the utterance to be inverted. The construction of a trajectory gives rise to a double selection (see Fig. 8):

(i) the choice of time frames at which the trajectory is defined, i.e. the choice of a subsequence of S defined by a function j: $\overline{S} = (s(j(0)) \dots s(j(k)) \dots s(j(K)))$ where $K < N$ (N is the number of time frames) and j is a monotonic function: $0 \leq j(k) < j(k+1) \leq N$.

(ii) the choice of one inverse point in each of the sets selected $s(j(0)) \dots s(j(k)) \dots s(j(K))$. The point chosen out of the set $s(j(k))$ is denoted $\alpha(j(k))$ ($\alpha(j(k)) \in I\!R^7$) and the articulatory trajectory is therefore $\overline{A} = (\alpha(j(0)) \dots \alpha(j(k)) \dots \alpha(j(K)))$.

Let $f_j(t)$ be the j^{th} formant frequency extracted from speech a time frame t, and $F_j(\alpha(j(k)))$ that computed by the acoustical simulation for the inverse point $\alpha(j(k))$. The cost of choosing $\alpha(j(k))$ after $\alpha(j(k-1))$ incorporates the acoustical distance together with the articulatory distance:

$$C(\alpha(j(k)), \alpha(j(k-1))) = \sum_{j=1}^{3} (f_j(t) - F_j(\alpha(j(k))))^2 + \lambda \sum_{i=1}^{7} m_i(\alpha_i(j(k)) - \alpha_i(j(k-1)))^2 \quad (13)$$

where λ is the weight of the articulatory distance with respect to the acoustical distance. Based on this local cost, the overall cost function to be minimized is $D = \sum_{j=1}^{K} (C(\alpha(j(k)), \alpha(j(k-1))) - B)$ where B is a positive bonus (as proposed by Ney) that prevents the minimization of returning an empty trajectory. This bonus has been set to a constant value but it could render the probability that this inverse point can be articulated by the target subject. The minimization is solved by dynamic programming and returns the best articulatory trajectory.

The local smoothness depends on the quality of the inverse solutions, and particularly the step used to sample the null space. Furthermore, as mentioned above, trajectories may be incomplete. For these reasons, the best solution provided by the non-linear smoothing algorithm is regularized through a variational method.

A. Variational regularization method

Any inversion method must lead to slowly changing parameters which generate spectra as close as possible to those of the original speech. This corresponds to satisfying two criteria: proximity to acoustic data and smoothness of articulatory trajectories. Generally, existing methods cannot allow for the two criteria at the same level, or at least must favor one criterion to the detriment of the other. Indeed, methods using dynamic programming often impose constraints upon the articulatory parameters dynamics. Then, a local optimization improves the acoustic proximity with the input signal at each time of the utterance analyzed. In contrast, our regularizing method combines both local and global aspects. This method utilizes the well known theory of variational calculus[37] which gives rise to an iterative process. This process starts with an initial solution (obtained by the non-linear smoothing algorithm) and generates a sequence of articulatory trajectories which optimizes a cost function that combines acoustic distance and changing rate of articulatory parameters.

There are two major advantages of this method compared to many other existing methods. Firstly, it involves the continuous nature of articulatory trajectories and the global acoustic and articulatory consistency without further optimization. Secondly, it incorporates the acoustic behavior of the articulatory model by means of sensitivity functions of formants, with respect to articulatory parameters.

The seven parameters of the articulatory model are time functions $\alpha(\mathbf{t}) = (\alpha_1(t) \dots \alpha_i(t) \dots \alpha_7(t))$, $t \in [t_i, t_f]$. Formant trajectories extracted from speech $f_j(t)$, $1 \leq j \leq 3$ are the input data. Those generated by the acoustic simulation are $F_j(\alpha(t))$ ($1 \leq j \leq 3$). A cost function for evaluating acoustic-to-articulatory mapping incorporates two components:

(1) $\sum_{j=1}^{3}(f_j(t) - F_j(\alpha(t)))^2$ which expresses the proximity between observed acoustic data, i.e. formants trajectories $f_j(t)$, and those generated by the articulatory model $F_j(\alpha(t))$.

(2) $\sum_{i=1}^{7} m_i \alpha_i'(t)^2$ which expresses the changing rate of articulatory parameters. In order to penalize large articulatory efforts and prevent the vocal tract from reaching positions too far from equilibrium, a potential energy term $\sum_{i=1}^{7} k_i \alpha_i^2(t)$ is added.

The cost function to be minimized has the following form

$$\begin{aligned} I = & \int_{t_i}^{t_f} \sum_{j=1}^{3}(f_j(t) - F_j(\alpha(t)))^2 dt \\ & + \lambda \int_{t_i}^{t_f} \sum_{i=1}^{7} m_i \alpha_i'(t)^2 dt + \beta \int_{t_i}^{t_f} \sum_{i=1}^{7} k_i \alpha_i^2(t) dt \end{aligned} \tag{14}$$

where t_i and t_f define the time interval over which the inversion is carried out, λ and β express the compromise between the changing rate of articulatory parameters, their distance from equilibrium and the acoustic distance. m_i is the pseudo mass of the ith articulator, and k_i is its spring constant. Equation (14) can be written as

$$I = \int_{t_i}^{t_f} \Phi(\alpha(\mathbf{t}), \alpha'(\mathbf{t}), t)dt$$

Variational calculus[37] can be used to minimize I. Euler-Lagrange equations express the vanishing of the derivative of I with respect to each of the α_i. These equations are a necessary condition to ensure a minimum of I and can be written

$$\begin{cases} \frac{\partial \Phi}{\partial \alpha_1} - \frac{d}{dt}\frac{\partial \Phi}{\partial \alpha_1'} = 0 \\ \dots \\ \frac{\partial \Phi}{\partial \alpha_7} - \frac{d}{dt}\frac{\partial \Phi}{\partial \alpha_7'} = 0 \end{cases} \tag{15}$$

Considering the definition of Φ, each of the Euler-Lagrange equations becomes:

$$\begin{aligned} \textstyle\sum_{j=1}^{3}(f_j(t) - F_j(\alpha(t))\frac{\partial F_j}{\partial \alpha_i} + \beta k_i \alpha_i(t) - \lambda m_i \alpha_i''(t) = 0 \\ i = 1 \dots 7 \end{aligned} \tag{16}$$

where $\alpha_i''(t)$ is the second time derivative of $\alpha_i(t)$. From now on we only consider one of the equations of the system (15) for sake of clarity. We can define an iterative process $\alpha_i^\tau(t)$ such that

$$\lim_{\tau \to \infty} \alpha_i^\tau(t) = \alpha_i(t)$$

(where $\alpha_i^{\tau=0}(t)$ is the startup solution) using the associated evolution equation

$$\gamma \frac{\partial \alpha_i^\tau}{\partial \tau} + \beta k_i \alpha_i^\tau - \lambda m_i \alpha_i^{\tau\prime\prime} = -\sum_{j=1}^{3}(f_j(t) - F_j(\alpha^\tau(t)))\frac{\partial F_j}{\partial \alpha_i^\tau} \tag{17}$$

where $\frac{\partial \alpha_i^\tau}{\partial \tau}$ represents the evolution of parameter α_i during the iteration process and γ a parameter for controlling the evolution rate. A solution to the static equation (16) is found when the term $\gamma \frac{\partial \alpha_i^\tau}{\partial \tau}$ vanishes. For sake of convenience we set m and k to 1. Let $\boldsymbol{\alpha^\tau} = (\alpha_{i,0}^\tau, \dots \alpha_{i,k}^\tau, \dots \alpha_{i,N}^\tau)$ denote the discrete representation of $\alpha_i(t)$, $\alpha_{i,k}^\tau$ represents the value of α_i^τ at discrete time $t = t_i + k\frac{t_f - t_i}{N}$ in the iteration τ. Since solving (17) for α_i is independent of other articulatory trajectories, $\alpha_{i,k}^\tau$ is noted α_k^τ for sake of clarity. Let $(f_0, \dots f_k, \dots f_N)$ denote the observed formant trajectory and $(F_0, \dots F_k, \dots F_N)$ the formant trajectory generated by the acoustic simulation. Finite difference

approximation of the derivative $\alpha''(t)$ leads to the following equation

$$\begin{aligned}\gamma(\alpha_k^\tau - \alpha_k^{\tau-1}) + \beta\alpha_k^\tau - \lambda(\alpha_{k+1}^\tau - 2\alpha_k^\tau + \alpha_{k-1}^\tau) \\ = -\sum_{j=1}^{3}(f_{j,k} - F_{j,k})\left.\frac{\partial F_j}{\partial \alpha}\right|_{\alpha_{1,k}^\tau \dots \alpha_{7,k}^\tau}\end{aligned} \tag{18}$$

where τ represents the iteration under process and k the discrete time. The derivative term $\left.\frac{\partial F_j}{\partial \alpha}\right|_{\alpha_{1,k}^\tau \dots \alpha_{7,k}^\tau}$ is calculated for the parameter α_i at point $(\alpha_{1,k}^\tau \dots \alpha_{7,k}^\tau)$ and incorporates the behavior of the acoustic modeling with respect to the evolution of articulatory parameters. Boundary conditions are needed to ensure that (18) has a unique solution. Since we do not impose any constraint on the positions of the extremities of $\alpha(t)$

$$\alpha''(0) = \alpha''(N) = 0$$

are the boundary conditions. Let B be an $(N+1) \times (N+1)$ matrix

$$B = \begin{bmatrix} \gamma+\beta+\lambda & -\lambda & 0 & \cdots & 0 \\ -\lambda & \gamma+\beta+2\lambda & -\lambda & \cdots & 0 \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & \dots & -\lambda & \gamma+\beta+2\lambda & -\lambda \\ 0 & \cdots & 0 & -\lambda & \gamma+\beta+\lambda \end{bmatrix}$$

$$\boldsymbol{\alpha^\tau} = (\alpha_0^\tau, \dots \alpha_k^\tau, \dots \alpha_N^\tau)^T$$

$$\boldsymbol{c^\tau} = \begin{bmatrix} \gamma\alpha_0^{\tau-1} - \sum_{j=1}^{3}(f_{j,0} - F_{j,0})\frac{\partial F_j}{\partial \alpha} \\ \gamma\alpha_1^{\tau-1} - \sum_{j=1}^{3}(f_{j,1} - F_{j,1})\frac{\partial F_j}{\partial \alpha} \\ \cdots \\ \gamma\alpha_N^{\tau-1} - \sum_{j=1}^{3}(f_{j,N} - F_{j,N})\frac{\partial F_j}{\partial \alpha} \end{bmatrix}$$

Equation (18) can be put in matrix form

$$B\boldsymbol{\alpha^\tau} = \boldsymbol{c^\tau}$$

Solving (15) may be carried out as an iterative process. At each iteration $\boldsymbol{\alpha^\tau}$ is calculated for each of the seven articulatory parameters α_i. In order to ensure that a minimal solution of (14) is reached, one needs to choose a good startup solution that provided by the non-linear smoothing method. The startup solution is then iteratively transformed so that (14) is minimized.

VIII. EXPERIMENTS

Mouth and pharynx sizes of Maeda's model can be adjusted to take into account the morphology of the target speaker. We used the method proposed in Galván-Rdz[16] and also in Naito et al.[28]

to adapt the articulatory model to our subject. Two vocal tract scale factors were sampled in a reasonable domain to allow at most $\pm$ 20% size variations. For each sample of this grid the first three formants of five extreme vowels /i e a o u/ were calculated from their reference articulatory parameters. The points given by the 5×3 formant frequencies build the surface of acoustical points that can be reached by deforming Maeda's model for these five extreme vowels. The point measured (three formant frequencies for the five vowels) for the target speaker from speech is thus projected orthogonally onto this surface. The orthogonal projection minimizes the distance between the surface and formants realized by the subject. The two scale factors corresponding to this point give the best adaptation of the articulatory model. The hypercube codebook was built for these scale factors. Therefore it cannot be used without further adaptation for another speaker.

The evaluation of an acoustic-to-articulatory inversion procedure comprises two aspects. The first is the acoustical faithfulness and ensures that inverted articulatory parameters are able to reproduce a speech signal as closely as possible to the original. Here the closeness is evaluated by measuring the distance between original and synthetic formants. It should be noted that the average distance is lower than 15 Hz and thus very good. The second aspect is that of the articulatory faithfulness, which requires that the synthetic vocal tract shape, i.e. the output of the articulatory model using inverted articulatory parameters, is compared to vocal tract images of the speaker uttering the same speech segment. This is thus tightly connected to the avaibility of articulatory databases that associate the description of the vocal tract shape together with the speech signal. Despite their potential interest there are almost no dynamic data available to perform this evaluation because either they do not provide the whole vocal tract (for instance cineradiographic databases recorded in the Seventies and recovered by Munhall et al.),[27] furthermore with a poor image and sound quality, or describe the vocal tract for a very small number of points in a limited region of the vocal tract (for instance the microbeam database)[48]. Therefore, the evaluation consists of a qualitative analysis of results in terms of the evolution of the place of articulation and the main phonetic characteristics. These two characteristics enable the goodness of realism to be evaluated easily and, more importantly, independently of speaker's variablity.

To evaluate our inversion method, we inverted several vowel-vowel and vowel-vowel-vowel sequences. The evaluation criteria used in these experiments are the smoothness and slow variation in time of the articulatory trajectories. This is the general behavior of the vocal tract of a real speaker. We also examined the animation of the vocal tract frame by frame to see whether there are any unnatural movement of any articulator. More advanced evaluation technique might

be consider as discussed in section IX. The trajectories of the first three formants were extracted from spectrograms of the utterances produced by the subject. The first step of the inversion (i.e. the recovery of articulatory points that produce the 3-tuple of formants extracted from speech) gave between 500 and 8000 solutions for each 3-tuple depending on the number of the steps used to sample the null space. For these experiments, the number of steps was set to 3 for each of the four dimensions, and therefore the number of samples was less than $81 = 3^4$ (see section VI B). The non-linear smoothing algorithm was then applied to get regular and realistic articulatory trajectories. Finally, the variational regularization was applied to the best articulatory trajectories so that the trajectories are simultaneously smooth and produce formant trajectories close to those extracted from speech.

Fig. 9 and Fig. 10 present inversion results for the sequence [iui]. In Fig 9, the original spectrogram of the utterance together with original and resynthesized formants are presented. As we can clearly see, all the solutions present a good acoustic proximity to the original formants. In Fig. 10, we present the result in the articulatory space for each of the seven parameters of Maeda's model. Each graph shows the trajectory obtained by the non-linear smoothing and the same trajectory optimized by applying the variational regularization (the smoothest trajectories are those obtained by the variational method). The obtained trajectories are smooth and vary slowly in time, which is the behavior of the vocal tract of a real speaker. In Fig. 11, we present for the same sequence [iui] the temporal dynamics of the vocal tract shapes (the mid-sagittal section), frame by frame. This "animation" clearly shows that the vocal tract goes from one shape to the other smoothly and does not present any unrealistic transition.

We carried out a large number of vowel-vowel and vowel-consonant-vowel inversion experiments[30]. The results are quite similar as those presented for the sequence [iui]. In Fig. 12, we present the inversion result of the transition [ua] (only final inversion results are displayed). Here again, all the articulatory parameters vary smoothly while guaranteeing a very good proximity to original data. Furthermore, inverse solutions recovered preserve main phonetic cues (main constriction position and vocal tract opening) as it can be seen in Fig. 13.

As presented above the inversion method is the baseline system that we will use for further inversion studies. One first stake will be the recovery of finer phonetic and articulatory cues which are very important for some phonemes. In Laprie and Ouni[21] we studied the solution space for the /yi/ transition because the main articulatory difference between /y/ and /i/ is the protrusion which is very strong for the French /y/. All the solutions recovered provide a very good fitting between

original and resynthesized formants and the main constriction and opening are correct. However, the best solution, in the sense of the criterion used for optimization in Ney's algorithm, does not present strong protrusion (see Fig. 14). In order to explore the solution space, we added a simple constraint on the lip protrusion (supplemented by a secondary constraint on jaw position). This constraint is implemented in the form of a strong bonus, attached to the first point of the inversion. Fig. 15 shows that this simple constraint enables the recovery of a more conform protrusion. We thus will investigate how constraints can be derived from phonetic knowledge.

IX. CONCLUDING REMARKS

Most of the existing acoustic-to-articulatory inversion methods introduce biases in the solution obtained because they exploit codebooks that do not cover the whole articulatory space. Consequently, there exist articulatory trajectories not found by the inversion although they are quite relevant from an articulatory point of view. On the contrary, one of the advantages of our method is that it ensures that all the possible inversion solutions can be explored, given an articulatory model and the frequency precision set for the formants being recovered, and does not implicitly favor any particular articulatory solution. To the best of our knowledge, this is the only inversion method based on an articulatory model that may guarantee that all the trajectories allowed by the model are explored. Furthermore, the regularization applied to startup solutions allows a global optimization over whole trajectories to be applied and not an optimization that processes points independently from each other. Experiments carried out validate our approach in terms of acoustic precision with respect to original data and smoothness of trajectory recovered.

In some sense the main characteristic of our inversion method is its "neutrality" with respect to the articulatory trajectories recovered. However, early language acquisition leads human speakers to prefer some articulatory strategies. These preferences can be linked to a particular speaker, but the existence of phonetically invariant features argue for deeper reasons stemming from biomechanical and acoustic efficiency. The neutrality of our inversion method will enable us to evaluate several strategies for guiding the inversion process. The first strategy is the incorporation of constraints stemming either from phonetic knowledge (for instance, the fact that lips must be protruded for rounded vowels like /y/ and /u/ in French) or from facial information extracted by computer vision when a speaker's face is visible. The second strategy is to incorporate preferences into the articulatory codebook through a learning stage that can exploit EMG, X-Ray or MRI data, or, on

the other hand, phonetic knowledge on articulatory features.

Other future work will concern the study of the precision required to adapt the articulatory model. Indeed, the acoustic space covered by the model depends on its geometric dimensions. Therefore, the model must be adapted before inversion. The adaptation, in our case that of Galván-Rdz[16], often necessitates the knowledge of the articulatory configurations for several vowels. This prior knowledge is only approximate because of speaker variability and compensatory properties of the articulatory model. If the adaptation mismatch is too great there is a risk that the inversion may fail or, the inversion may improperly exploit compensatory properties of the model to compensate for the adaptation mismatch. Therefore, we will investigate the acoustic precision required to guarantee the relevancy of the articulatory information recovered from speech together with the precision required for the model adaptation.

REFERENCES

[1] 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. *Journal of Acoustical Society of America*, 63(5):1535–1555, May 1978.

[2] B.S. Atal and O. Rioul. Neural networks for estimating articulatory positions from speech. *J. Acoust. Soc. Amer.*, 86(Supp. 1, S67):123–131, 1989.

[3] P. Badin, L. Pouchoy, G. Bailly, M. Raybaudi, C. Segebarth, JF. Lebas, M. Tiede, E. Vatikiotis-Bateson, and Y. Tohkura. Un modèle articulatoire tridimensionnel du conduit vocal basé sur des données irm. In *Actes XXIImes Journées d'Etude sur la Parole*, Martigny, Switserland, 1998. JEP.

[4] G. Bailly, C. Abry, R. Laboissièrre, P. Perrier, and J.-L. Schwartz. Inversion and speech recognition. In J. Vandewalle, R. Boite, M. Mooner, and A. Osterlinck, editors, *Signal processing VI: Theories and Applications*, volume 1, pages 159–164, Brussels, Belgium, August 1992. Elsevier.

[5] D. Beautemps, P. Badin, and R. Laboissière. Deriving vocal tract area functions from midsagittal profiles and formant frequencies: a new model for vowels and fricative consonants based on experimental data. *Speech Communication*, 16:27–47, 1995.

[6] L.-J. Boë, P. Perrier, and G. Bailly. The geometric vocal tract variables controlled for vowel production: proposals for constraining acoustic-to-articulatory inversion. *Journal of Phonetics*,

20:27–38, 1992.

[7] R. Carré and M. Mrayati. Articulatory-acoustic-phonetic relations and modeling, regions and modes. In W.J. Hardcastle and A. Marchal, editors, *Speech production and speech modelling*, pages 211–240. Kluwer Academic Publisher, Amsterdam, 1990.

[8] F. Charpentier. Determination of the vocal tract shape from the formants by analysis of the articulatory-to-acoustic non-linearities. *Speech Communication*, 3:291–308, 1984.

[9] S. Dusan and L. Deng. Recovering vocal tract shapes from MFCC parameters. In *Proceedings International Conference on Spoken Language Processing*, volume 2, Sydney (Australia), December 1998. ICSLP.

[10] O. Engwall. Modeling of the vocal tract in three dimensions. In *Eurospeech*, pages 113–116, Budapest, 1999. ISCA.

[11] G. Fant. *Acoustic Theory of Speech Production*. Mouton & Co., The Hague, 1960.

[12] G. Fant. Analytical constraints on the composition of speech spectra. In *Acoustic Theory of Speech Production, Second Printing*, pages 48–62. Mouton & Co., The Hague, 1970.

[13] G. Fant. Swedish vowels and a new three-parameter model. Technical report, TMH-QPSR 1, 2001.

[14] J.L. Flanagan, K. Ishizaka, and K.L. Shipley. Signal models for low bit-rate coding of speech. *J. Acoust. Soc. Amer.*, 68(3):780–791, 1980.

[15] B. Gabioud. Articulatory models in speech synthesis. In E. Keller, editor, *Fundamentals of Speech Synthesis and Speech Recognition*, chapter 10. John Wiley & Sons, West Sussex, England, 1994.

[16] A. Galván-Rdz. *Etudes dans le cadre de l'inversion acoustico-articulatoire: Amélioration d'un modèle articulatoire, normalisation du locuteur et récupération du lieu de constriction des plosives*. PhD thesis, Institut de la Communication Parlée, 1997.

[17] G.H. Golub and C.F. Van Loan. *Matrix Computations*. The John Hopkins University Press, Baltimore, 1989.

[18] J. M. Heinz and K. N. Stevens. On the relations between lateral cineradiographs, area functions and acoustic spectra of speech. In *Proceedings of the 5th International Congress on Acoustics*, page A44., Liège, 1965. ICA.

[19] R. Laboissièrre and A. Galvàn. Inferring the commands of an articulatory model from acoustical specifications of stop/vowel sequences. In *Proceedings of the Fourth International Congress of Phonetic Sciences*, volume 1, pages 358–361, Stockholm, August 1995. ICPhS.

[20] Y. Laprie and M.-O. Berger. Cooperation of regularization and speech heuristics to control automatic formant tracking. *Speech Communication*, 19(4):255–270, October 1996.

[21] Y. Laprie and S. Ouni. Introduction of constraints in an acoustic-to-articulatory inversion. In *7th International Conference on Spoken Language Processing*, Denver, USA, Sep 2002. ICSLP.

[22] J. N. Larar, J. Schroeter, and M. M. Sondhi. Vector quantization of the articulatory space. *IEEE Trans. Acoust., Speech, Signal Processing*, ASSP-36(12):1812–1818, December 1988.

[23] S. Maeda. Un modèle articulatoire de la langue avec des composantes linéaires. In *Actes 10èmes Journées d'Etude sur la Parole*, pages 152–162, Grenoble, May 1979. JEP.

[24] S. Maeda. Compensatory articulation during speech: Evidence from the analysis and synthesis of vocal-tract shapes using an articulatory model. In W.J. Hardcastle and A. Marchal, editors, *Speech production and speech modelling*, pages 131–149. Kluwer Academic Publisher, Amsterdam, 1990.

[25] B. Mathieu and Y. Laprie. Adaptation of Maeda's model for acoustic to articulatory inversion. In *Proceedings of the 5th European Conference on Speech Communication and Technology*, volume 4, pages 2015–2018, Rhodes, Greece, 1997. Eurospeech.

[26] R.S. McGowan. Recovering articulatory movement from formant frequency trajectories using task dynamics and a genetic algoritm: preliminary model tests. *Speech Communication*, 14:19–48, 1994.

[27] K. G. Munhall, E. Vatikiotis-Bateson, and Y. Tokhura. X-ray film database for speech research. *J. Acoust. Soc. Am.*, 98(2):1222–1224, 1995.

[28] M. Naito, L. Deng, and Y. Sagisaka. Model-based speaker normalization methods for speech recognition. In *Proceedings of the 6th European Conference on Speech Communication and Technology*, Budapest, September, 1999. Eurospeech.

[29] H. Ney. A dynamic programmation algorithm for nonlinear smoothing. *Signal Processing*, 5(2):163–173, March 1983.

[30] S. Ouni. *Modélisation de l'espace articulatoire par un codebook hypercubique pour l'inversion acoustico-articualtoire*. PhD thesis, Université Henri poincaré, 2001.

[31] G. Papcun, J. Hochberg, T.R. Thomas, F. Laroche, J.Zacks, and S. Levy. Inferring articulation and recognizing gestures from acoustics with a neural network trained on x-ray microbeam data. *J. Acoust. Soc. Amer.*, 92(2):688–700, 1992.

[32] P. Perrier, L.-J. Boë, and R. Sock. Vocal tract area function estimation from midsagittal dimensions with ct scans and a vocal tract cast: modeling the transition with two sets of coefficients.

J. Speech and Hearing Research, 35:53–67, 1992.

[33] M.G. Rahim and C.C. Goodyear. Estimation of vocal tract filter parameters using a neural net. *Speech Communication*, 9:49–55, 1990.

[34] M.G. Rahim, C.C. Goodyear, W.B. Kleijn, J. Schroeter, and M.M. Sondhi. On the use of neural networks for in articulatory speech synthesis. *J. Acoust. Soc. Amer.*, 93(2):1109–1121, 1993.

[35] H. B. Richards, J. S. Bridle, M. J. Hunt, and J. S. Mason. Dynamic constraint weighting in the context of articulatory parameter estimation. In *Proceedings of the 5th European Conference on Speech Communication and Technology*, volume 5, pages 2535–2538, Rhodes, Greece, 1997. Eurospeech.

[36] R.C. Rose, J. Schroeter, and M.M. Sondhi. An investigation of the potential role of speech production models in automatic speech recognition. In *Proceedings of International Conference on Spoken Language Processing*, volume 2, pages 575–578, Yokohama, Japan, 1994. ICSLP.

[37] R.S. Schechter. *The variational Method in Engineering*. McGraw-Hill, New York, 1967.

[38] J. Schoentgen and S. Ciocea. Kinematic formant-to-area mapping. *Speech Communication*, 21:227–244, 1997.

[39] J. Schroeter and M. M. Sondhi. Speech coding based on physiological models of speech production. In S. Furui and M. M. Sondhi, editors, *Advances in Speech Signal Processing*, pages 231–267. Dekker, New York, 1992.

[40] K. Shirai and T. Kobayashi. Estimating articulatory motion from speech wave. *Speech Communication*, 5(2):159–170, 1986.

[41] K. Shirai and T. Kobayashi. Estimating articulatory motion using neural networks. *J. Phonetics*, 19:379–385, 1991.

[42] A. Soquet, M. Saerens, and P. Jospa. Acoustic-articulatory inversion based on a neural controller of a vocal tract model: further results. In O. Simula T. Kohonen, K. Mokisara and J. Kangas, editors, *Artificial Neural Networks*, pages 371–376. North Holland: Elsevier, The Netherlands, 1991.

[43] V.N. Sorokin, A.S. Leonov, and A.V. Trushkin. Estimation of stability and accuracy of inverse problem solution for the vocal tract. *Speech Communication*, 30:55–74, 2000.

[44] V.N. Sorokin and A.V. Trushkin. Articulatory-to-acoustic mapping for inverse problem. *Speech Communication*, 19:105–118, 1996.

[45] K.N. Stevens. *Human communication: A unified view*, pages 51–66. McGraw Hill, New York, 1972.

[46] K.N. Stevens. On the quantal nature of speech. *J. Phonetics*, 27:3–45, 1989.

[47] K.N. Stevens and A.S. House. Development of a quantitative description of vowel articulation. *J. Acoust. Soc. Amer.*, 27:484–493, 1955.

[48] J. R. Westbury. X-ray microbeam speech production database user's handbook version 1.0. Technical report, Waisman Center on Mental Retardation & Human Development, University of Wisconsin, Madison, 1994.

TABLE I. Acoustic precision of the interpolation (the precision is measured by comparing formant values interpolated from codebook points with those calculated by the articulatory synthesizer directly)

	mean error	standard deviation
F1	6.47Hz	6.93Hz
F2	7.90Hz	9.96Hz
F3	6.92Hz	9.43Hz

TABLE II. Acoustic precision of the inversion

	$\triangle F1$	$\triangle F2$	$\triangle F3$
Mean error	8.39Hz	10.86Hz	10.45Hz
Standard deviation	10.03Hz	12.11Hz	12.53Hz

FIGURE CAPTIONS

FIG. 1. Parameters of Maeda's articulatory model: P1 (jaw position, vertical movement) P2 (tongue dorsum position that can move roughly horizontally from the front to the back of the mouth cavity) P3 (tongue dorsum shape, i.e. rounded or unrounded) P4 (apex position ; this parameter only deforms the apex part of the tongue by moving it up or down) P5 (lip height) P6 (lip protrusion) P7 (larynx height)

FIG. 2. Comparison of the first three formants of the root-shape and the random codebooks. We do not present the regular sampling codebook as it has almost the same covering space as the random codebook. The regions in light gray (resp. dark gray) represent the acoustic space of the root-shape (resp. random sampling) codebook. The random sampling codebook covers a space larger than that covered by the root-shape codebook.

FIG. 3. For sake of clarity we represent a 3D hypercube. Note that the edge length is ℓ and U_0 is the origin of the hypercube. V_i $(i = 0..7)$ are the vertices of the hypercubes. The linearity test is performed on the segments $[V_i, V_j]$ where $i \neq j$. If the test fails the hypercube is split into 8 sub-hypercubes (8 is the number of the vertices in 3D). These sub-hypercubes are represented with dashed lines. The upper table gives the values of the parameter φ_{ij} for the 8 vertices indexed from 0 to 7.

FIG. 4. A $2D$ partial representation of the hypercube codebook. For sake of clarity, we only present jaw and tongue (α_1, α_2). We clearly see that there are different regions more or less linear (i.e. the corresponding hypercubes are more or less big). Shaded regions are the forbidden

FIG. 5. The 4-dimensional hypercube (for illustration, represented here by the square) is the smallest hypercube containing the 4-polytope (represented by the polygon). It is defined by the vertices A, B, C, D. The 4-dimensional hypercube is discretized (the points represent the possible solutions) and the solutions that do not verify Eq. (10) are eliminated (the points lying outside the polygon).

FIG. 6. Representation of the inversion solutions for the utterance [au] in the articulatory space

(jaw parameter). The horizontal axis represents the time (in milliseconds)and the vertical axis represents the variation of one parameter expressed in standard deviations. The left graph presents all the solutions obtained by SVD without sampling the null space. The right graph presents solutions obtained by sampling the null space.

FIG. 7. Representation of the inversion solutions for the utterance [ui] in the articulatory space (jaw parameter). The horizontal axis represents the time (in milliseconds) and the vertical axis represents the variation of one parameter expressed in standard deviation. The left graph presents all the solutions obtained by SVD without sampling the null space. The second presents solutions obtained by sampling the null space.

FIG. 8. Double selection achieved by the non smoothing algorithm: time frames and articulatory candidates. For clarity sake articulatory candidates are 1-dimensional points. The articulatory candidates are given for each time frame (each vertical dotted line). The best trajectory is the solid line and contains some gaps (time frames 3, 6, 7 and 13) because the incorporation of outliers would decrease the quality of the whole trajectory.

FIG. 9. Inversion result for the sequence [iui]. The horizontal axis represents the time (in milliseconds)and the vertical axis represents formants (in Hertz). From top down: (a) spectrogram, (b) original formants trajectories and all the formants solutions resynthesized from articulatory points retrieved from the hypercube codebook, and finally, (c) formants trajectories resynthesized from results of the nonlinear smoothing before and after variational regularization (smooth trajectories).

FIG. 10. Inversion results for the sequence [iui]. The first graph presents the formant trajectories and each of the other graphs shows the trajectory of one articulatory parameter. The horizontal axis represents the time (in milliseconds) and the vertical axis represents formants (in Hertz). In each graph the trajectory obtained by non-linear smoothing and that obtained by using the variational regularization method are plotted (the smoothest trajectories are those obtained by the variational regularization).

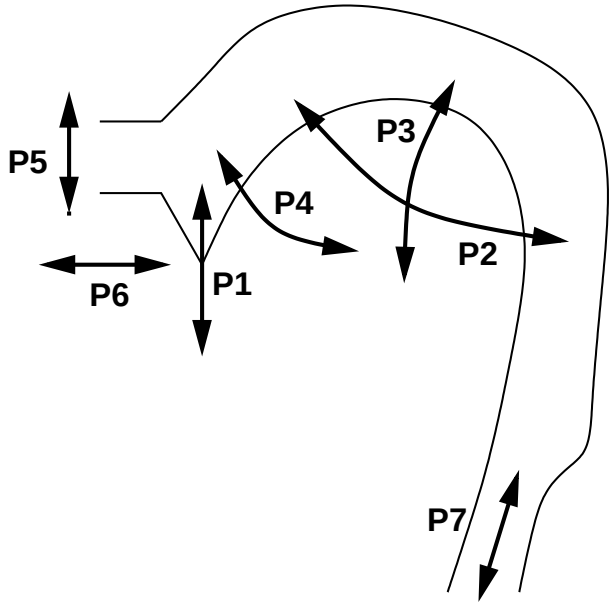
FIG. 11. Temporal dynamics of the vocal tract shapes for the sequence[iui].

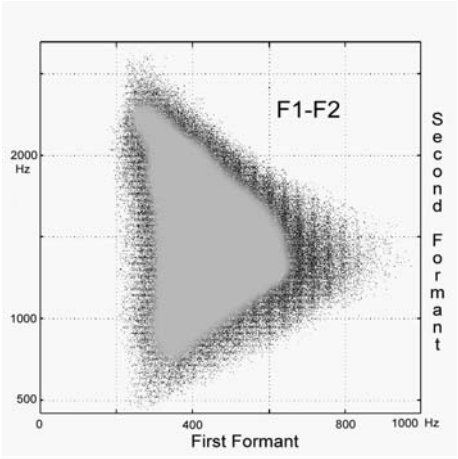
FIG. 12. Inversion results for the transition [ua]. The first graph presents the formant trajectories and each of the other graphs shows the trajectory of one articulatory parameter.

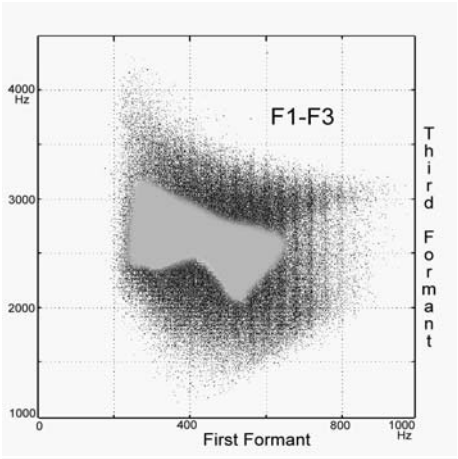
FIG. 13. Temporal dynamics of the vocal tract shapes for the transition [ua].

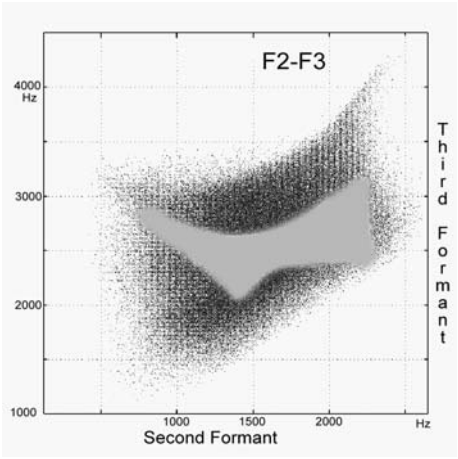
FIG. 14. Temporal evolution of three articulatory parameters (jaw, tongue position and protrusion) without any constraint imposed.

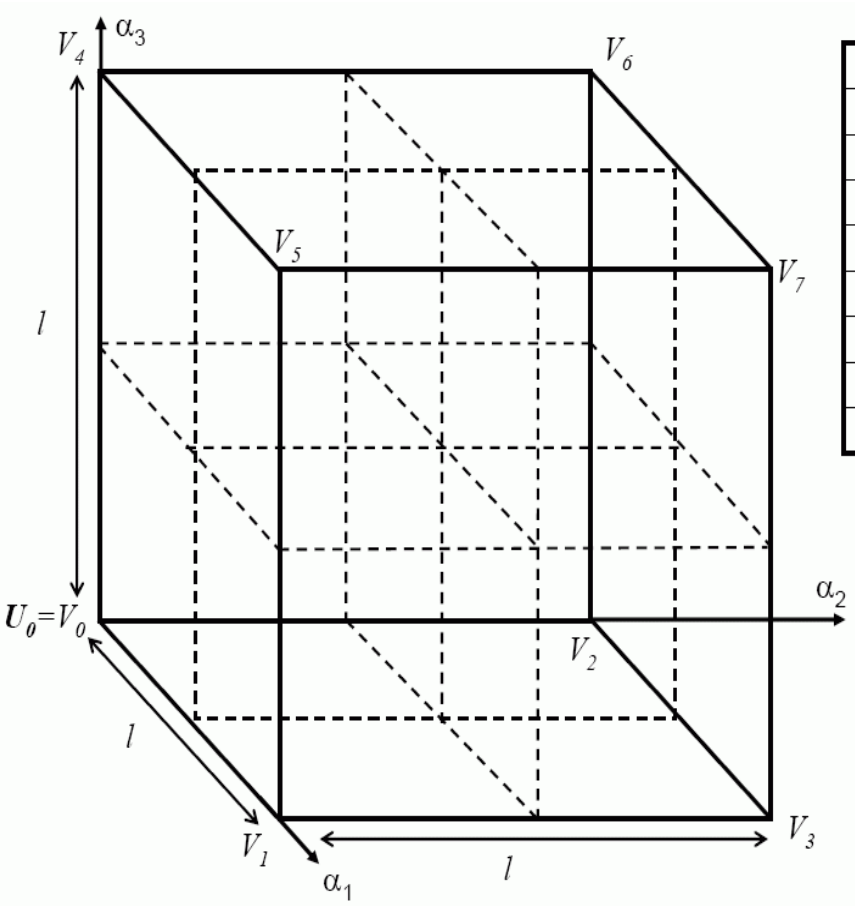
FIG. 15. Temporal evolution of three articulatory parameters (jaw, tongue position and protrusion) when imposing the protrusion to be near to 2.7 and the jaw position to 1.5 for the first point.


P5
P3
P4
P2
P6
P1
P7

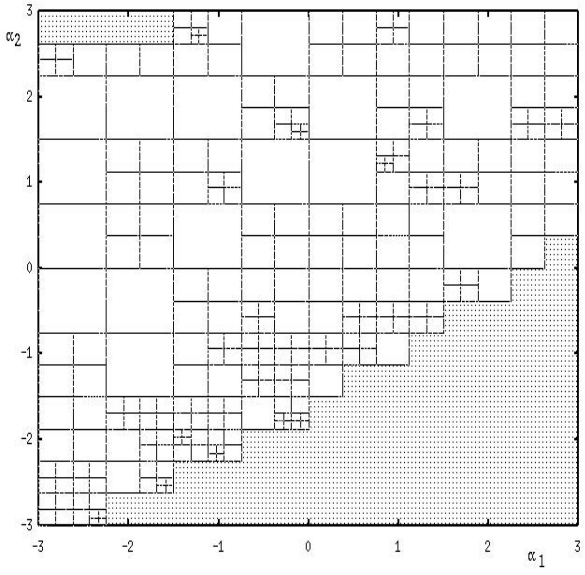

F1-F2
2000
Hz
1000
500
0
400
800
1000 Hz
First Formant
Second Formant

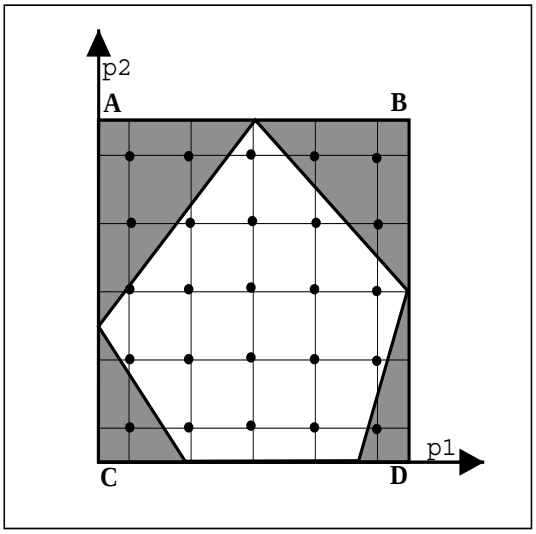

F1-F3
4000
Hz
3000
2000
1000
0
400
800
1000
Hz
First Formant
Third Formant

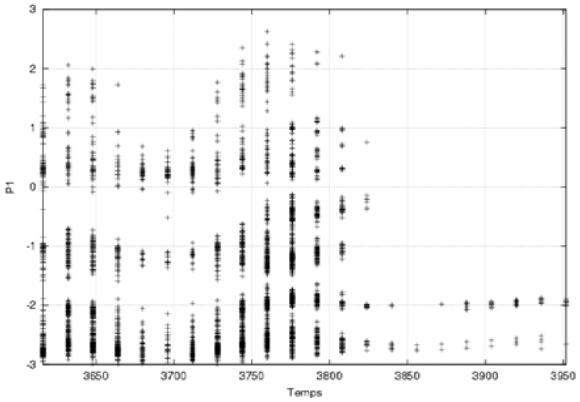

F2-F3
4000
Hz
3000
2000
1000
1000
1500
2000
Hz
Second Formant
Third Formant

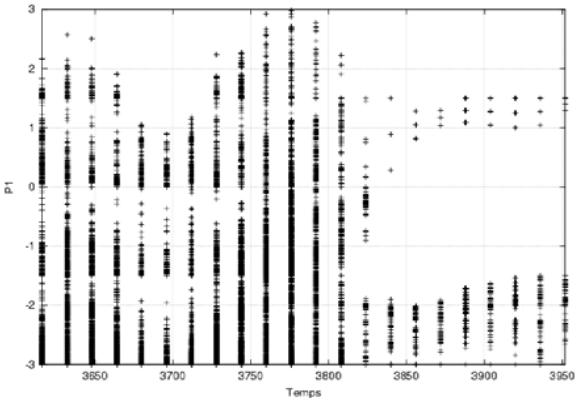


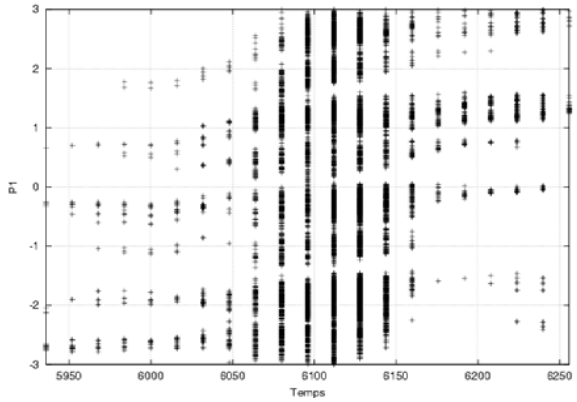
α_3	α_2	α_1	i
0	0	0	0
0	0	1	1
0	1	0	2
0	1	1	3
1	0	0	4
1	0	1	5
1	1	0	6
1	1	1	7

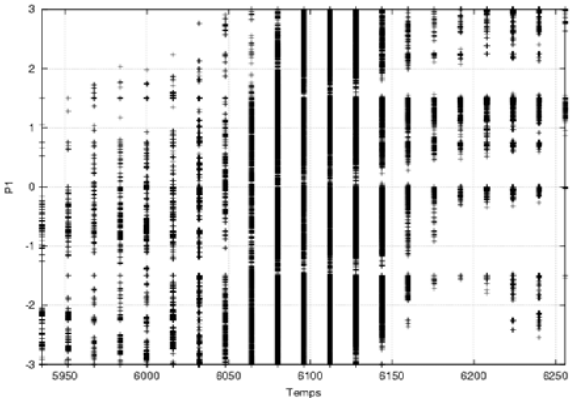

3
2
1
0
-1
-2
-3
α2
-3
-2
-1
0
1
2
3
α1

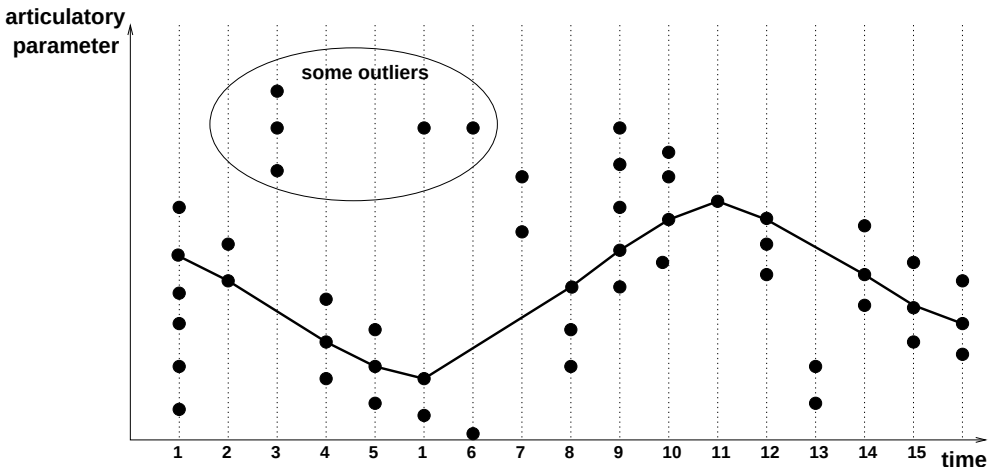

p2
A
B
C
D
p1

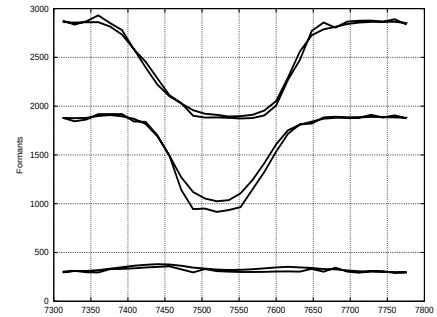

p1
Temps
3
2
1
0
-1
-2
-3
3650
3700
3750
3800
3850
3900
3950

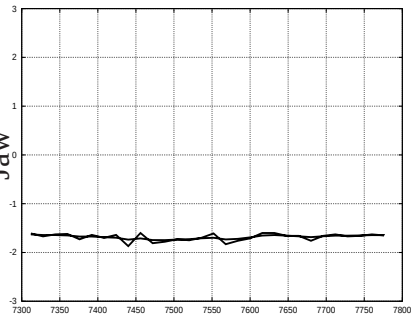

p1
Temps
3
2
1
0
-1
-2
-3
3650
3700
3750
3800
3850
3900
3950

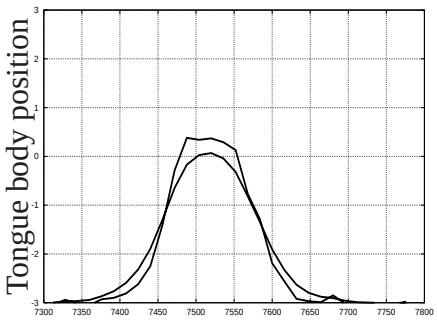

p1
Temps
3
2
1
0
-1
-2
-3
5950
6000
6050
6100
6150
6200
6250

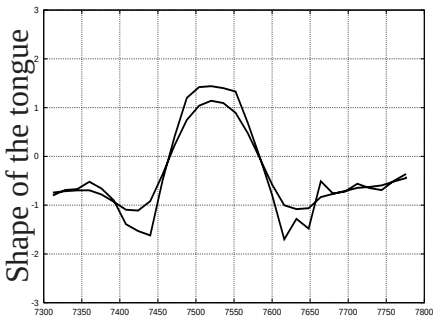

p1
Temps
3
2
1
0
-1
-2
-3
5950
6000
6050
6100
6150
6200
6250

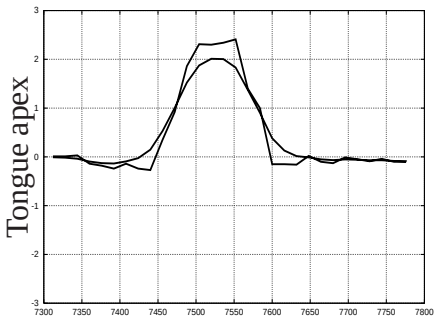

articulatory
parameter
some outliers
1
2
3
4
5
1
6
7
8
9
10
11
12
13
14
15
time

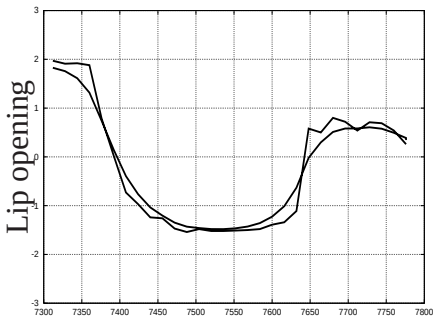

Formants
3000
2500
2000
1500
1000
500
0
7300 7350 7400 7450 7500 7550 7600 7650 7700 7750 7800

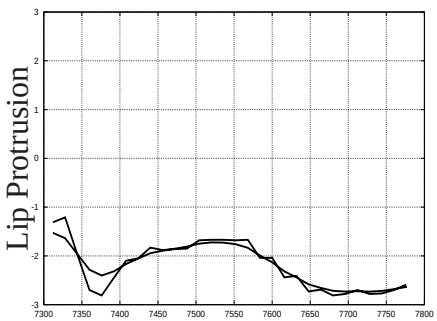

Jaw
3
2
1
0
-1
-2
-3
7300 7350 7400 7450 7500 7550 7600 7650 7700 7750 7800

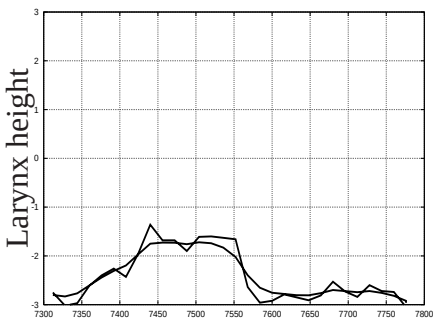

Tongue body position
3
2
1
0
-1
-2
-3
7300 7350 7400 7450 7500 7550 7600 7650 7700 7750 7800

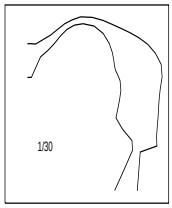

Shape of the tongue
3
2
1
0
-1
-2
-3
7300 7350 7400 7450 7500 7550 7600 7650 7700 7750 7800

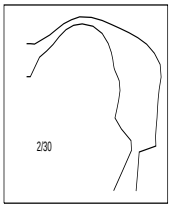

Tongue apex
3
2
1
0
-1
-2
-3
7300 7350 7400 7450 7500 7550 7600 7650 7700 7750 7800

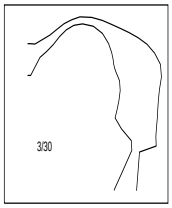

Lip opening
3
2
1
0
-1
-2
-3
7300 7350 7400 7450 7500 7550 7600 7650 7700 7750 7800

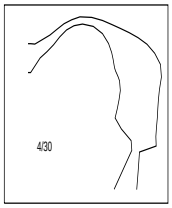

Lip Protrusion
3
2
1
0
-1
-2
-3
7300 7350 7400 7450 7500 7550 7600 7650 7700 7750 7800

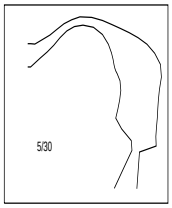

Larynx height
3
2
1
0
-1
-2
-3
7300 7350 7400 7450 7500 7550 7600 7650 7700 7750 7800

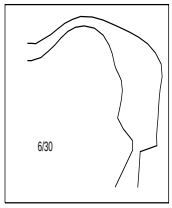

1/30

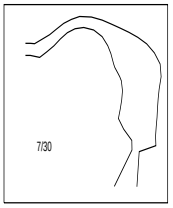

2/30

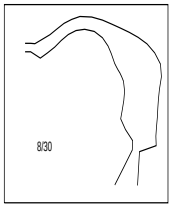

3/30

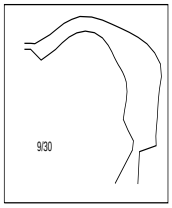

4/30

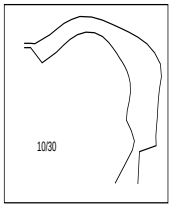

5/30

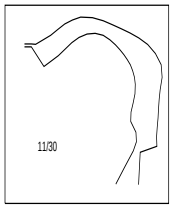

6/30

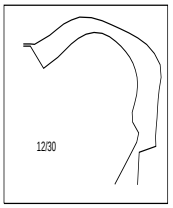

7/30

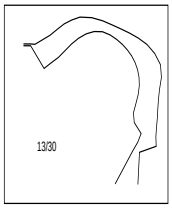

8/30

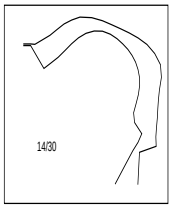

9/30

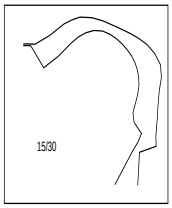

10/30

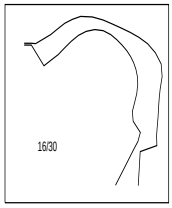

11/30

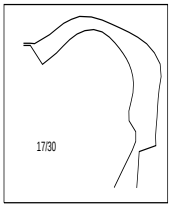

12/30

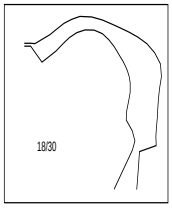

13/30

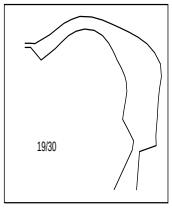

14/30

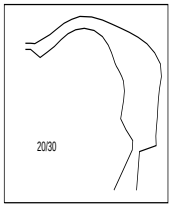

15/30

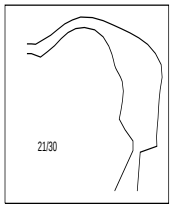

16/30

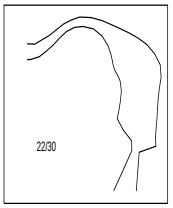

17/30

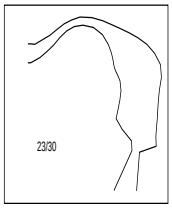

18/30

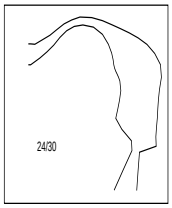

19/30

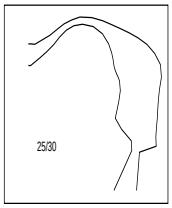

20/30

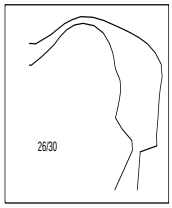

21/30

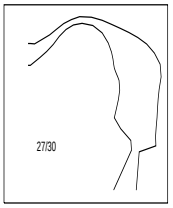

22/30

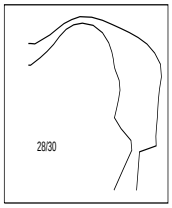

23/30

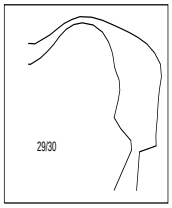

24/30

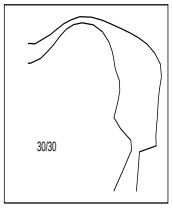

25/30

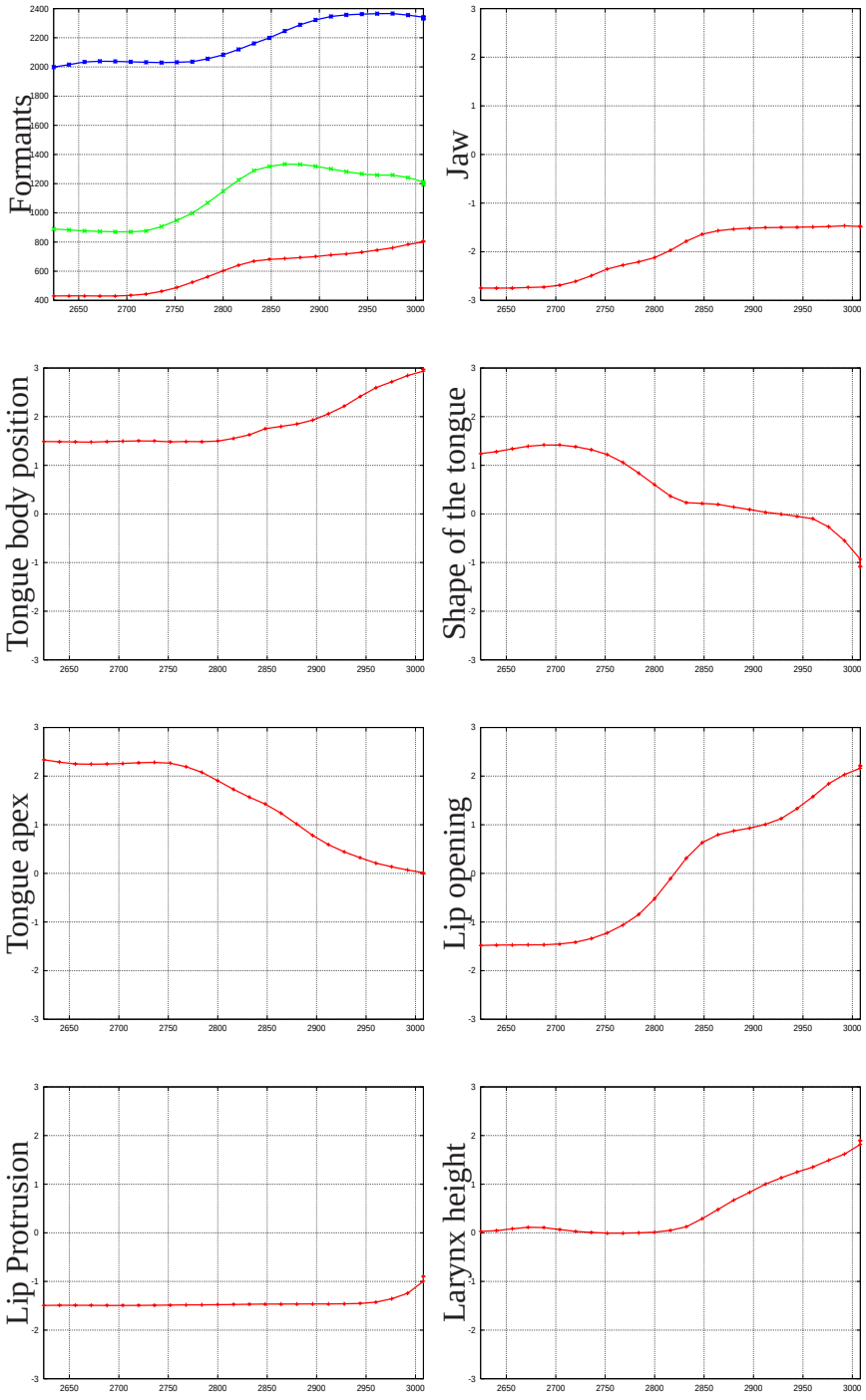

26/30

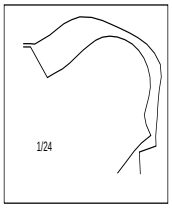

27/30

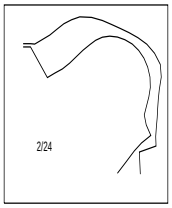

28/30

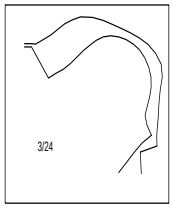

29/30

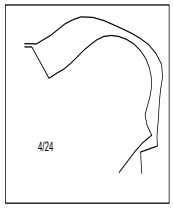

30/30

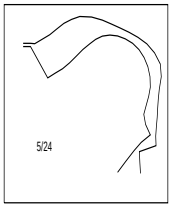

Formants
Jaw
Tongue body position
Shape of the tongue
Tongue apex
Lip opening
Lip Protrusion
Larynx height

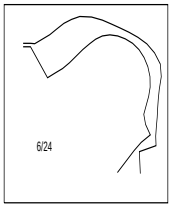

1/24

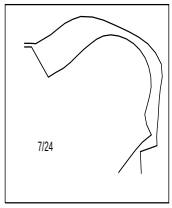

2/24

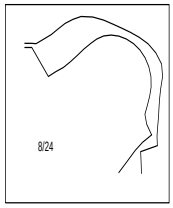

3/24

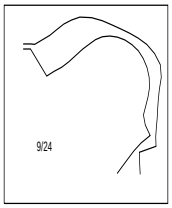

4/24

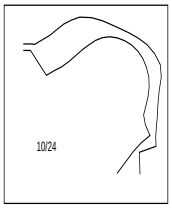

5/24

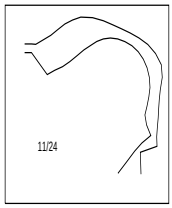

6/24

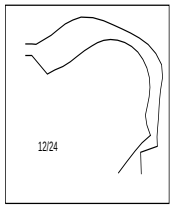

7/24

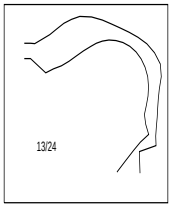

8/24

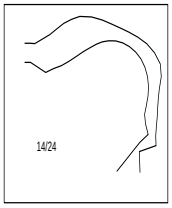

9/24

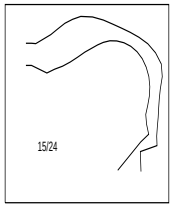

10/24

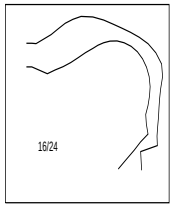

11/24

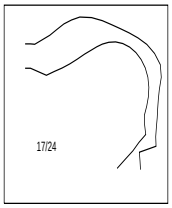

12/24

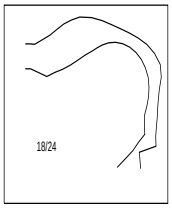

13/24

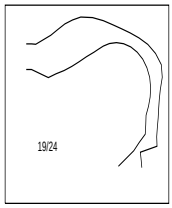

14/24

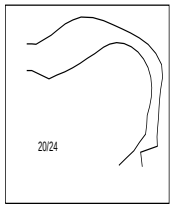

15/24

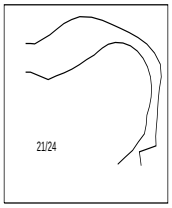

16/24

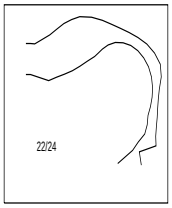

17/24

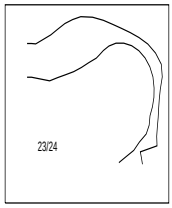

18/24

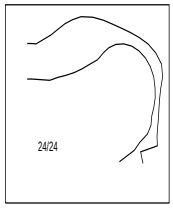

19/24

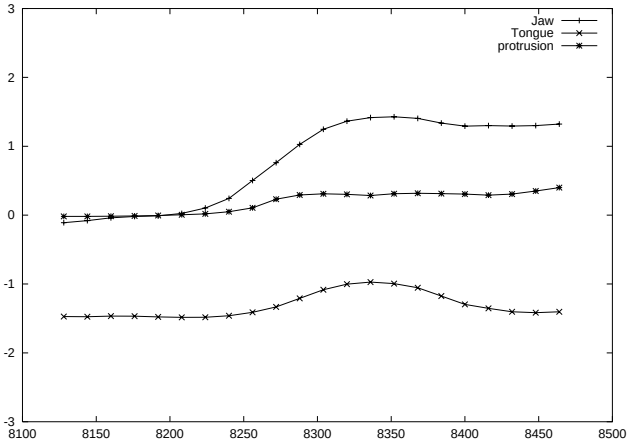

20/24


21/24

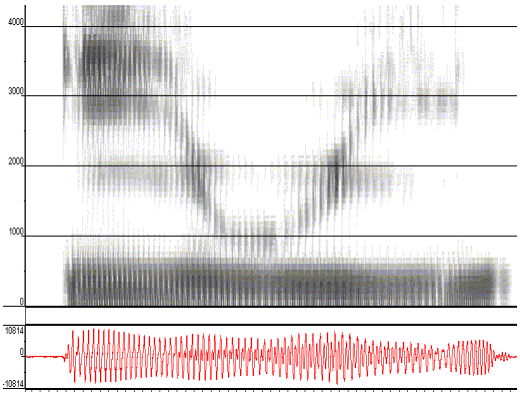

22/24


23/24

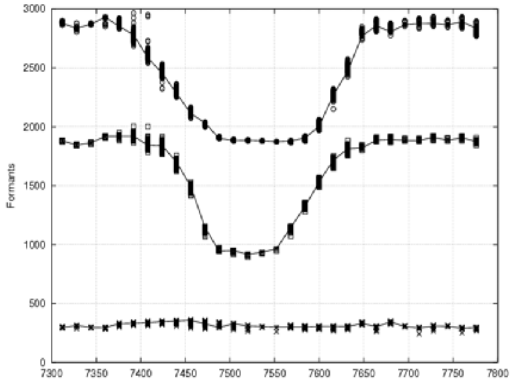

24/24


Jaw
Tongue
protrusion
3
2
1
0
-1
-2
-3
8100
8150
8200
8250
8300
8350
8400
8450
8500

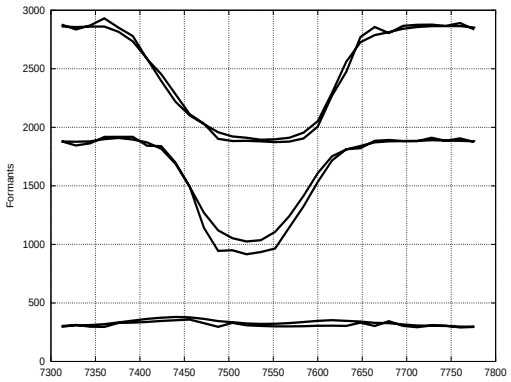
a)

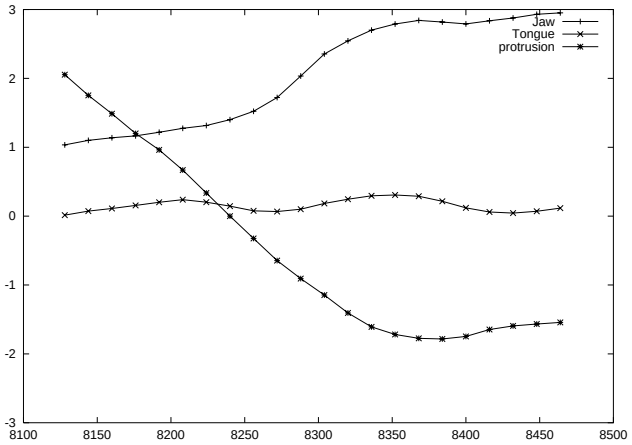

4000
3000
2000
1000
0
10814
0
-10814

b)


Formants
3000
2500
2000
1500
1000
500
0
7300
7350
7400
7450
7500
7550
7600
7650
7700
7750
7800

c)


Formants
3000
2500
2000
1500
1000
500
0
7300
7350
7400
7450
7500
7550
7600
7650
7700
7750
7800


Jaw
Tongue
protrusion
3
2
1
0
-1
-2
-3
8100
8150
8200
8250
8300
8350
8400
8450
8500

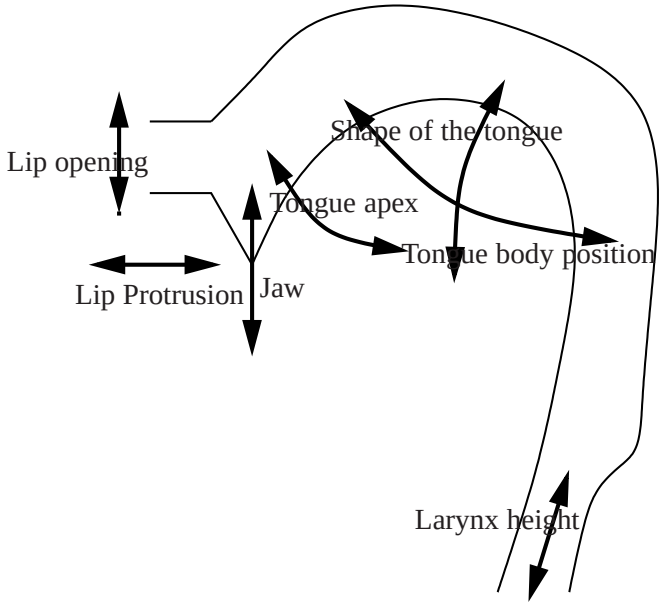


FIG. 1. Parameters of Maeda's articulatory model: P1 (jaw position, vertical movement) P2 (tongue dorsum position that can move roughly horizontally from the front to the back of the mouth cavity) P3 (tongue dorsum shape, i.e. rounded or unrounded) P4 (apex position ; this parameter only deforms the apex part of the tongue by moving it up or down) P5 (lip height) P6 (lip protrusion) P7 (larynx height)

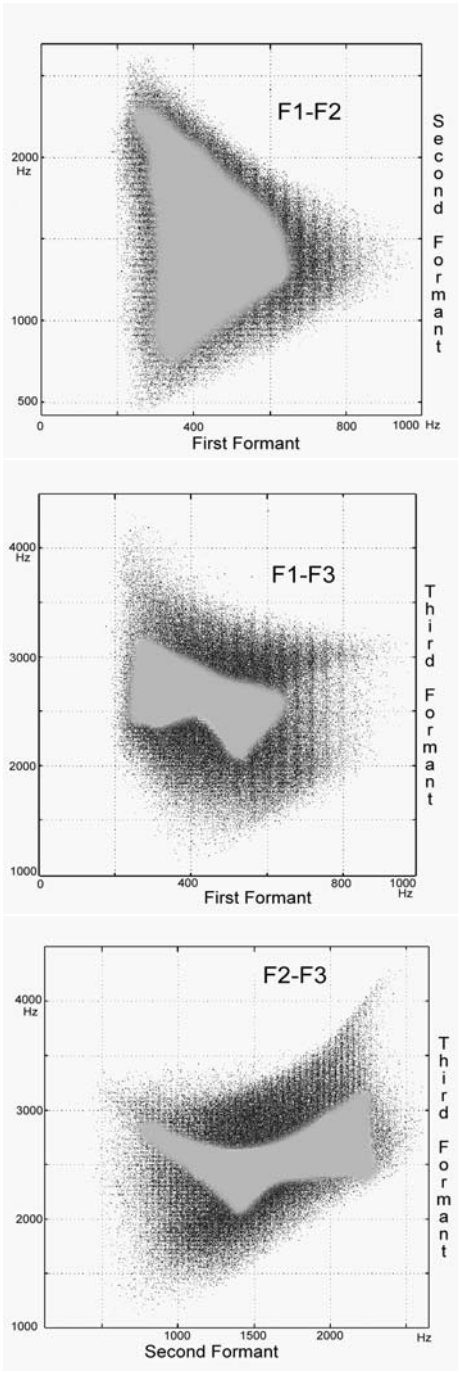
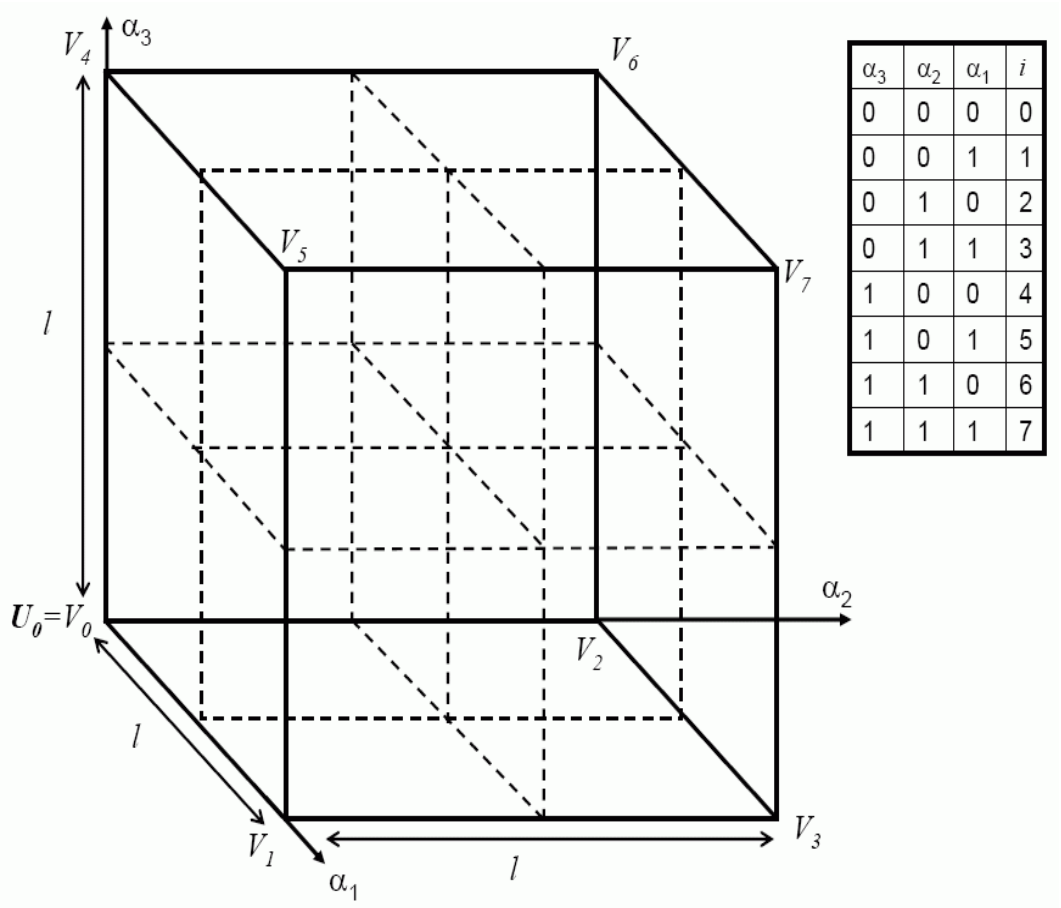


FIG. 2. Comparison of the first three formants of the root-shape and the random codebooks. We do not present the regular sampling codebook as it has almost the same covering space as the random codebook. The regions in light gray (resp. dark gray) represent the acoustic space of the root-shape (resp. random sampling) codebook. The random sampling codebook covers a space larger than that covered by the root-shape codebook.



α_3	α_2	α_1	i
0	0	0	0
0	0	1	1
0	1	0	2
0	1	1	3
1	0	0	4
1	0	1	5
1	1	0	6
1	1	1	7

FIG. 3. For sake of clarity we represent a 3D hypercube. Note that the edge length is ℓ and U_0 is the origin of the hypercube. V_i $(i = 0..7)$ are the vertices of the hypercubes. The linearity test is performed on the segments $[V_i, V_j]$ where $i \neq j$. If the test fails the hypercube is split into 8 sub-hypercubes (8 is the number of the vertices in 3D). These sub-hypercubes are represented with dashed lines. The upper table gives the values of the parameter φ_{ij} for the 8 vertices indexed from 0 to 7.

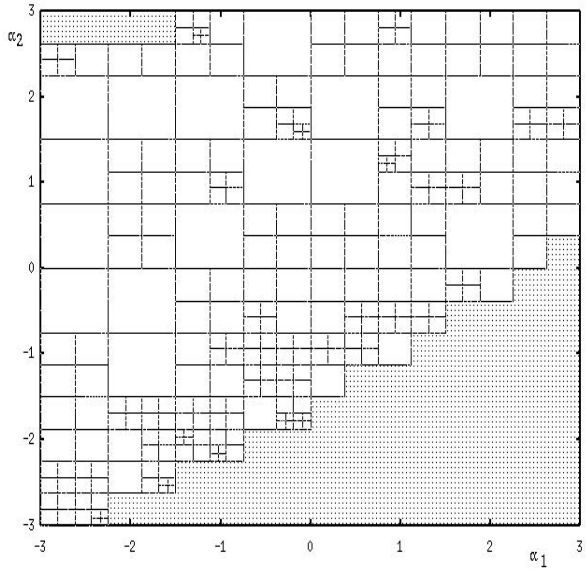


FIG. 4. A $2D$ partial representation of the hypercube codebook. For sake of clarity, we only present jaw and tongue (α_1, α_2). We clearly see that there are different regions more or less linear (i.e. the corresponding hypercubes are more or less big). Shaded regions are the forbidden.

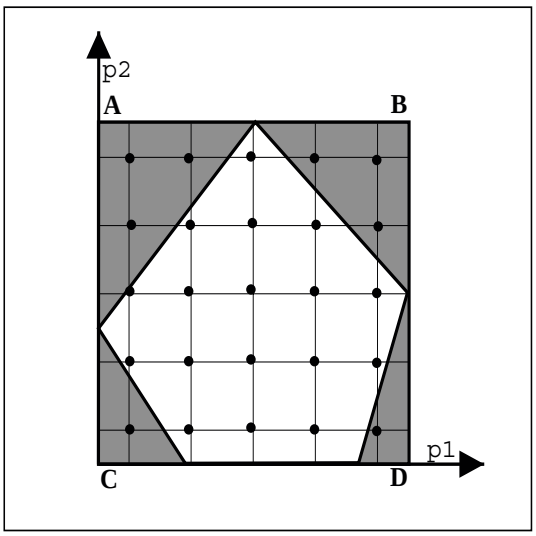


FIG. 5. The 4-dimensional hypercube (for illustration, represented here by the square) is the smallest hypercube containing the 4-polytope (represented by the polygon). It is defined by the vertices A, B, C, D. The 4-dimensional hypercube is discretized (the points represent the possible solutions) and the solutions that do not verify Eq. (10) are eliminated (the points lying outside the polygon).

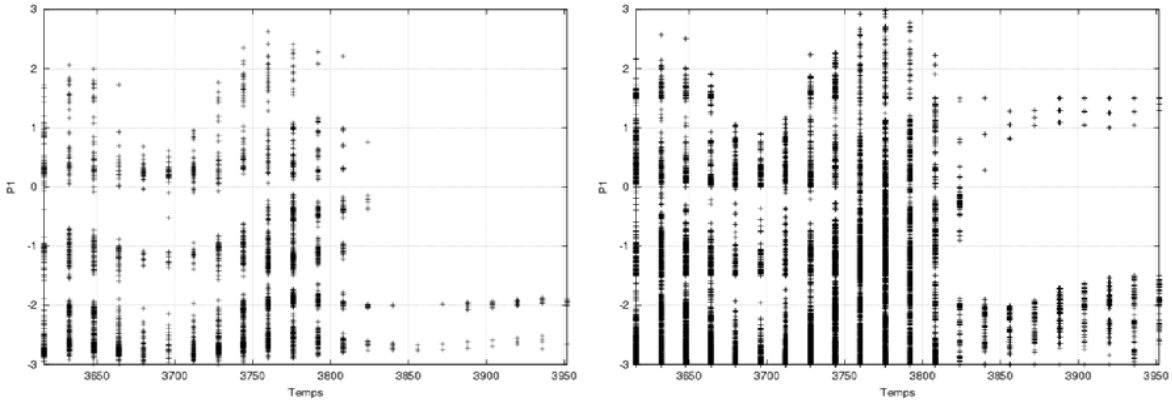


FIG. 6. Representation of the inversion solutions for the utterance [au] in the articulatory space (jaw parameter). The horizontal axis represents the time (in milliseconds)and the vertical axis represents the variation of one parameter expressed in standard deviations. The left graph presents all the solutions obtained by SVD without sampling the null space. The right graph presents solutions obtained by sampling the null space.

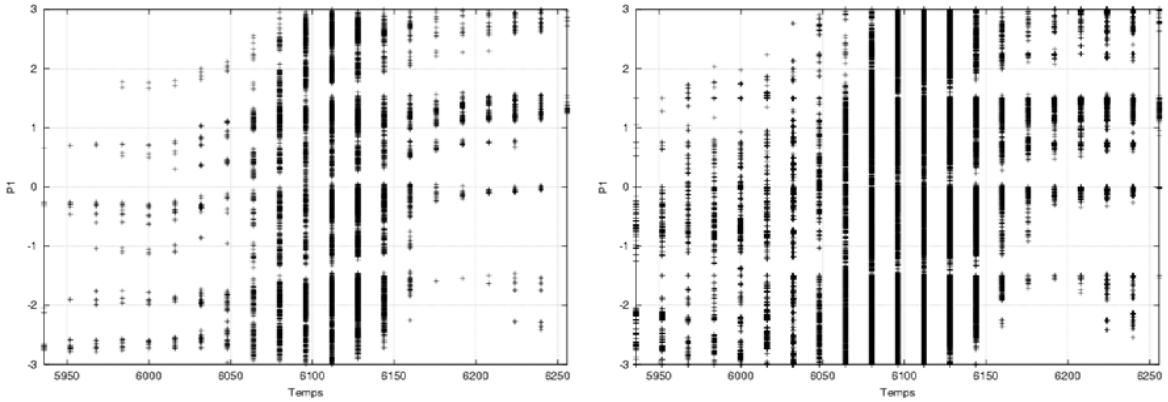


FIG. 7. Representation of the inversion solutions for the utterance [ui] in the articulatory space (jaw parameter). The horizontal axis represents the time (in milliseconds) and the vertical axis represents the variation of one parameter expressed in standard deviation. The left graph presents all the solutions obtained by SVD without sampling the null space. The second presents solutions obtained by sampling the null space.

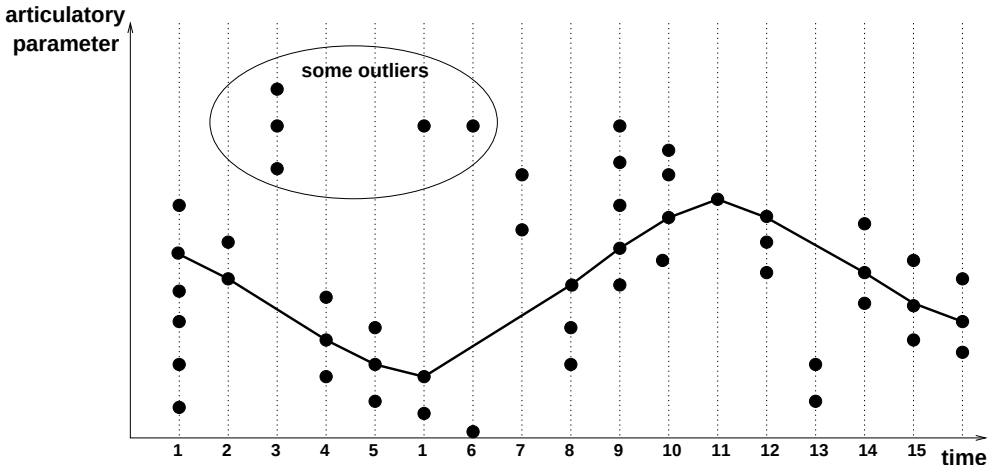


FIG. 8. Double selection achieved by the non smoothing algorithm: time frames and articulatory candidates. For clarity sake articulatory candidates are 1-dimensional points. The articulatory candidates are given for each time frame (each vertical dotted line). The best trajectory is the solid line and contains some gaps (time frames 3, 6, 7 and 13) because the incorporation of outliers would decrease the quality of the whole trajectory.

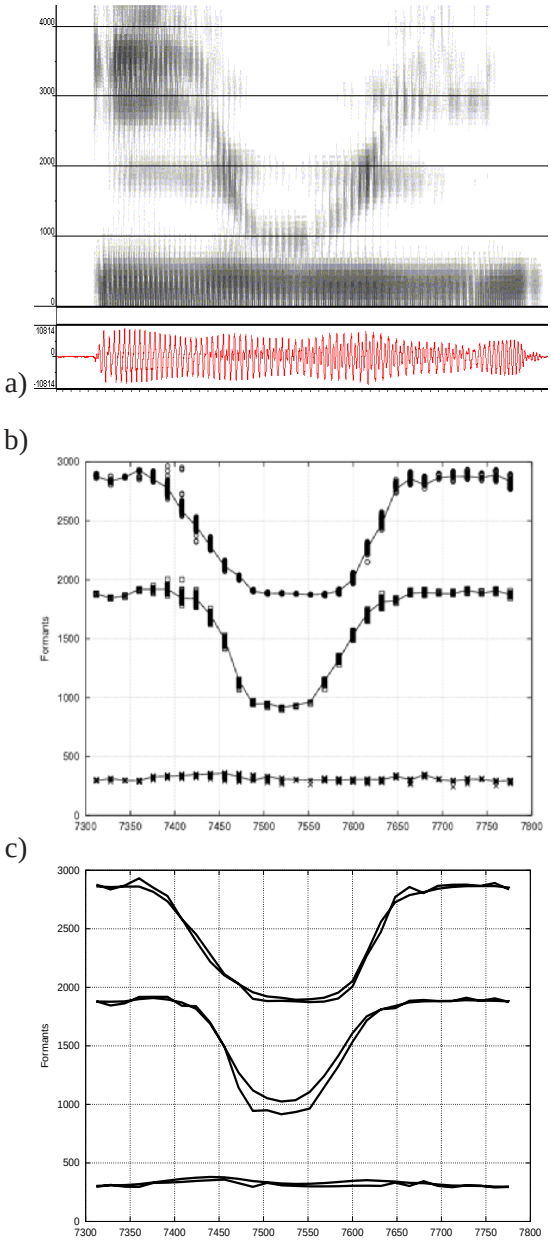


FIG. 9. Inversion result for the sequence [iui]. The horizontal axis represents the time (in milliseconds)and the vertical axis represents formants (in Hertz). From top down: (a) spectrogram, (b) original formants trajectories and all the formants solutions resynthesized from articulatory points retrieved from the hyper-cube codebook, and finally, (c) formants trajectories resynthesized from results of the nonlinear smoothing before and after variational regularization (smooth trajectories).

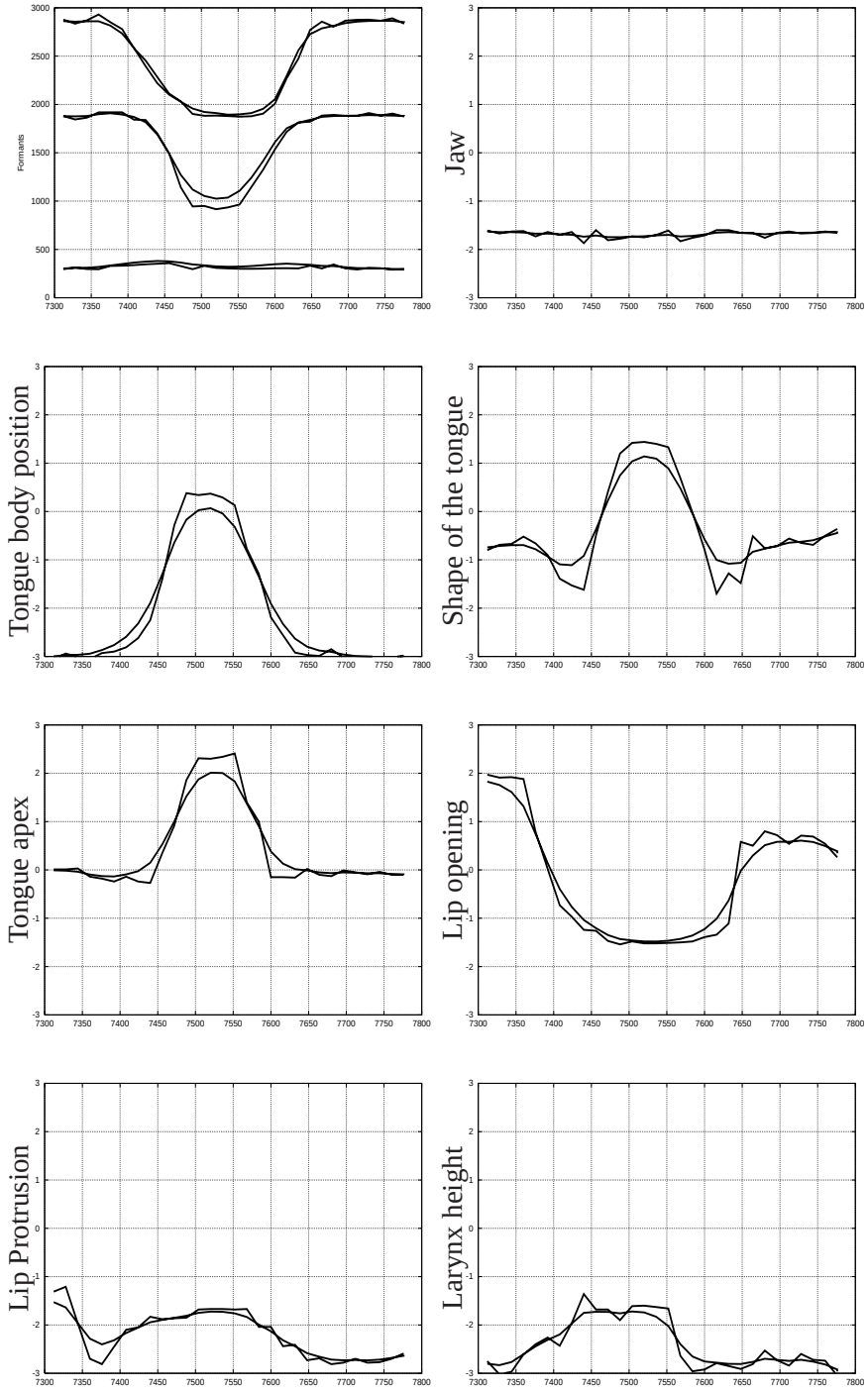


FIG. 10. Inversion results for the sequence [iui]. The first graph presents the formant trajectories and each of the other graphs shows the trajectory of one articulatory parameter. The horizontal axis represents the time (in milliseconds) and the vertical axis represents formants (in Hertz). In each graph the trajectory obtained by non-linear smoothing and that obtained by using the variational regularization method are plotted (the smoothest trajectories are those obtained by the variational regularization).

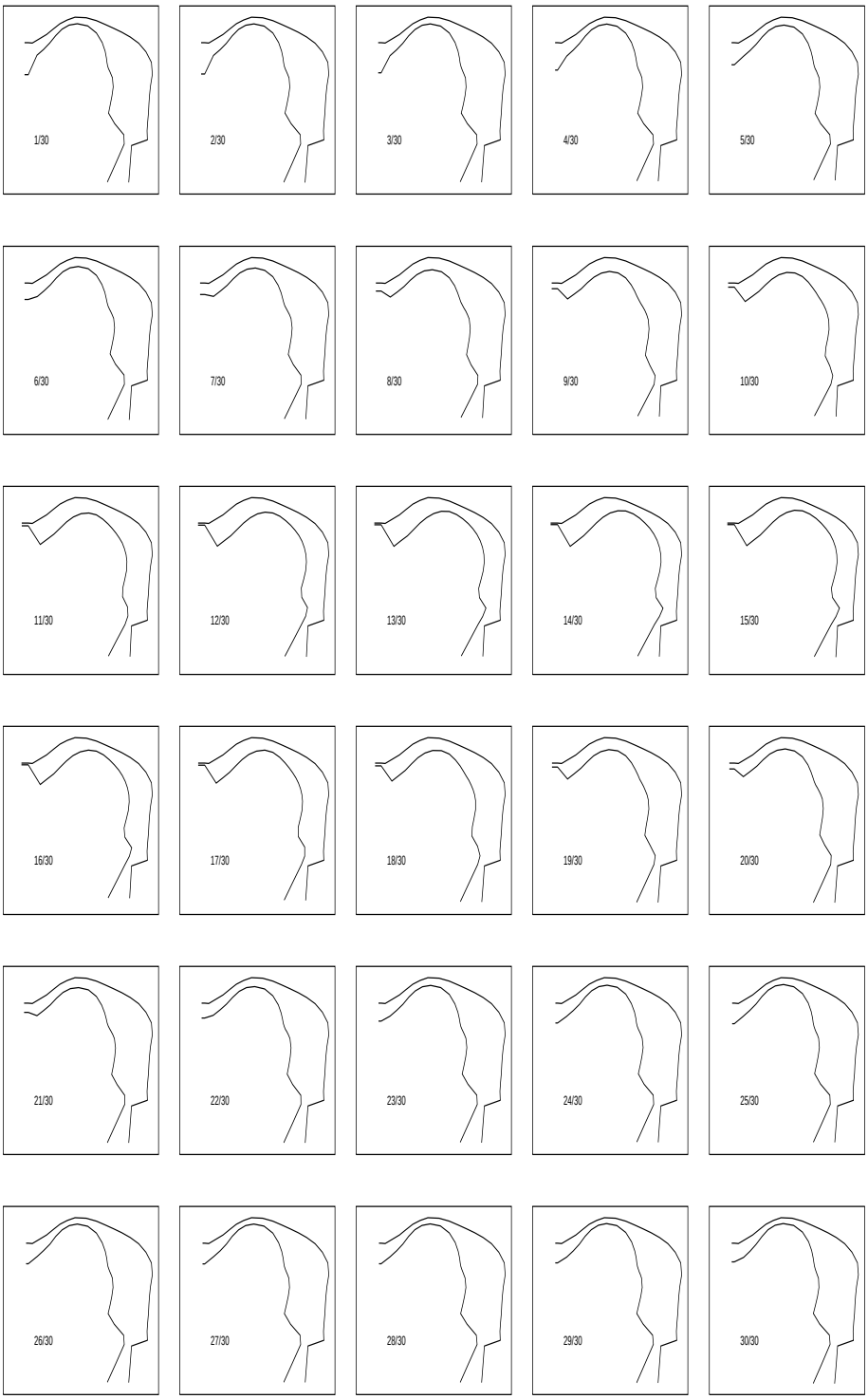


FIG. 11. Temporal dynamics of the vocal tract shapes for the sequence[iui].

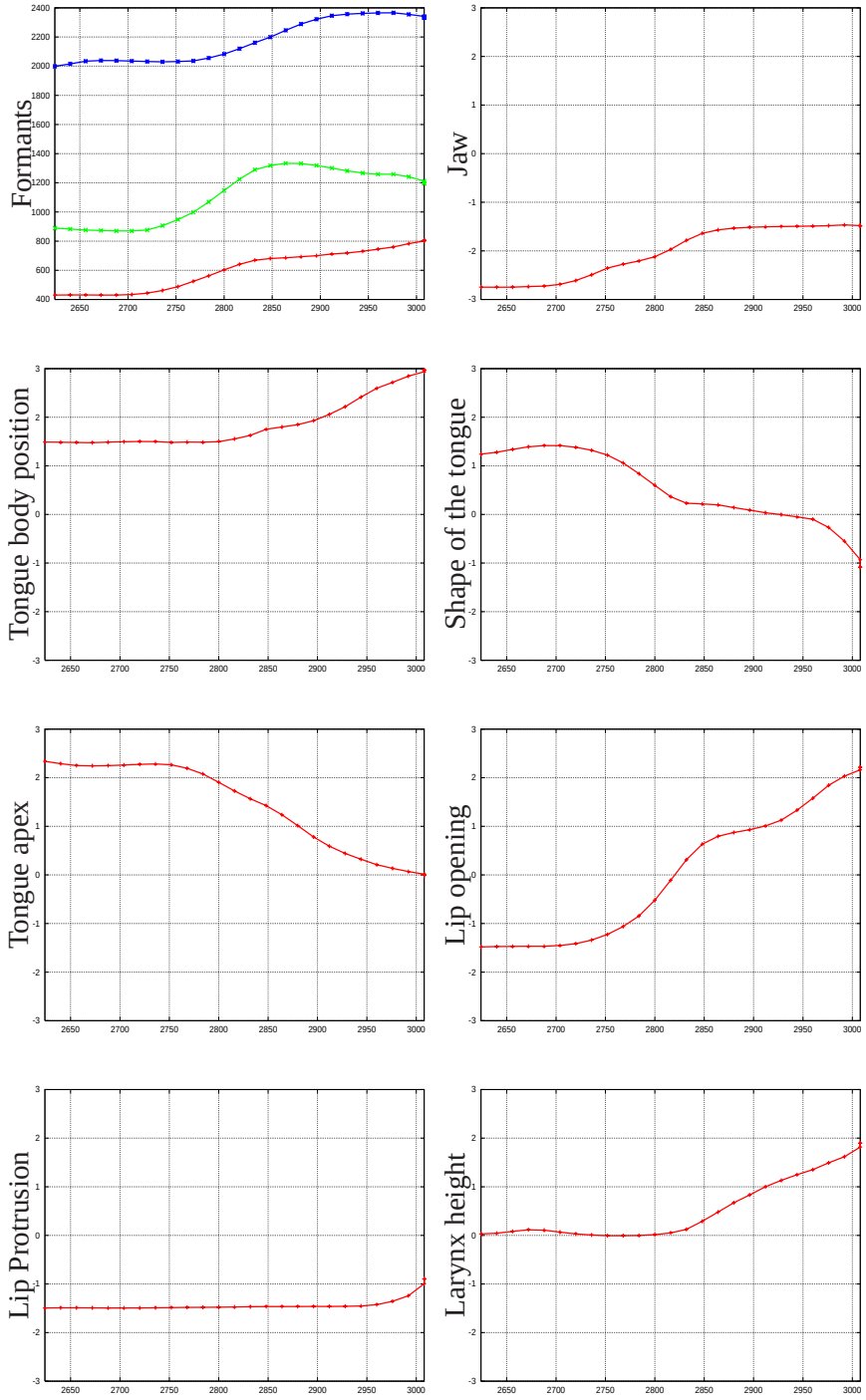


FIG. 12. Inversion results for the transition [ua]. The first graph presents the formant trajectories and each of the other graphs shows the trajectory of one articulatory parameter.

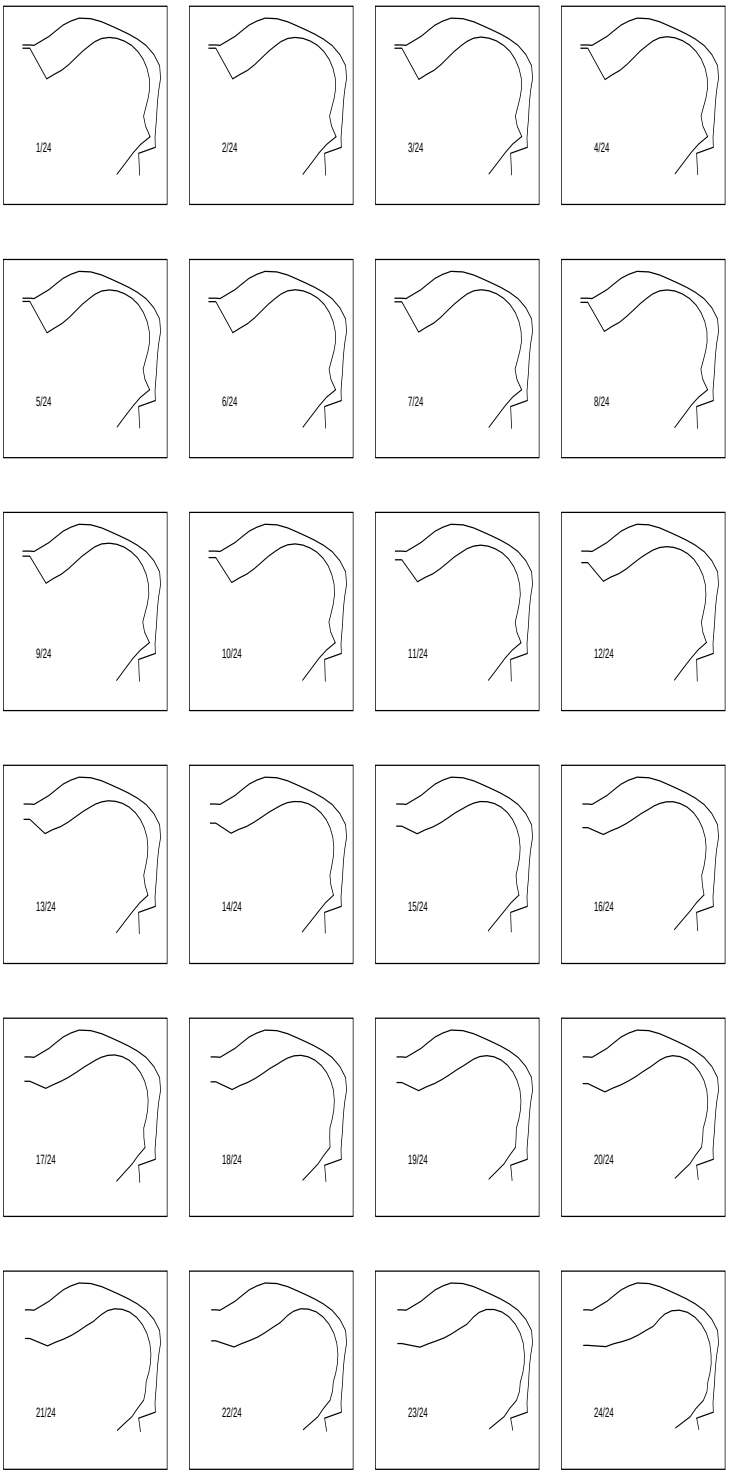


FIG. 13. Temporal dynamics of the vocal tract shapes for the transition [ua].

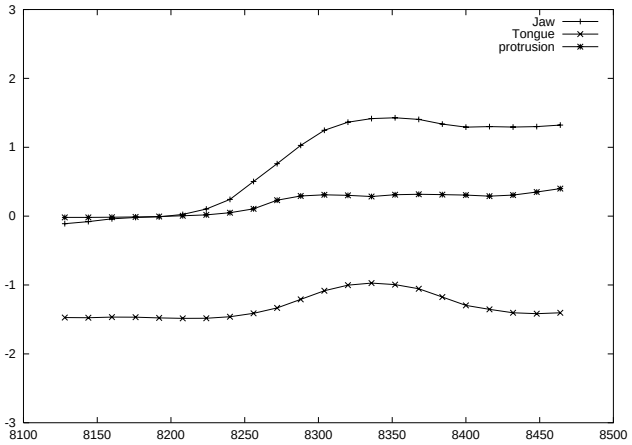


FIG. 14. *Temporal evolution of three articulatory parameters (jaw, tongue position and protrusion) without any constraint imposed.*

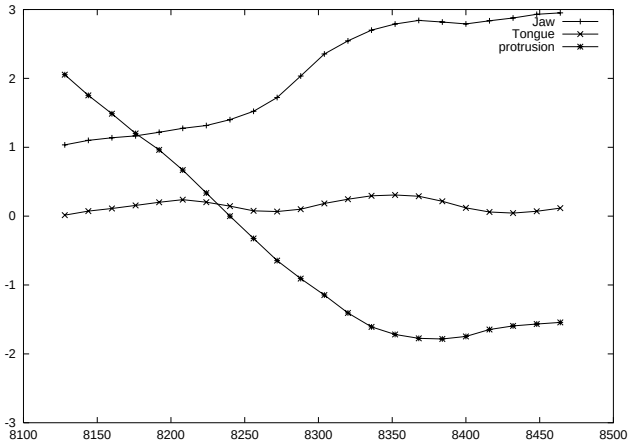


FIG. 15. *Temporal evolution of three articulatory parameters (jaw, tongue position and protrusion) when imposing the protrusion to be near to 2.7 and the jaw position to 1.5 for the first point.*

TABLE I. Acoustic precision of the interpolation (the precision is measured by comparing formant values interpolated from codebook points with those calculated by the articulatory synthesizer directly)

	mean error	standard deviation
F1	6.47Hz	6.93Hz
F2	7.90Hz	9.96Hz
F3	6.92Hz	9.43Hz

TABLE II. Acoustic precision of the inversion

	$\triangle F1$	$\triangle F2$	$\triangle F3$
Mean error	8.39Hz	10.86Hz	10.45Hz
Standard deviation	10.03Hz	12.11Hz	12.53Hz