Improving acoustic-to-articulatory inversion by using hypercube codebooks

Slim Ouni, Yves Laprie

IMPROVING ACOUSTIC-TO-ARTICULATORY INVERSION BY USING HYPERCUBE CODEBOOKS

Slim Ouni and Yves Laprie

LORIA
BP 239
54506 Vandœuvre-les-Nancy, FRANCE
ouni,laprie@loria.fr

ABSTRACT

This paper presents an articulatory codebook construction method which gives a good space coverage with a limited number of points. We represent the articulatory space by hypercubes. Therefore the articulatory space is decomposed into regions represented by few number of points. We describe our interpolation method to retrieve points from the hypercube and the inversion method based on the same interpolation method. The advantage of the codebook and the inversion method is its robustness with respect to articulatory-to-acoustic mapping non-linearity problems.

1. INTRODUCTION

Recovering articulatory parameters from the acoustic signal has received a considerable attention last years because of the perspectives of applying this technique in speech recognition, perception or speech coding. Most of the acoustic-to-articulatory inversion methods use articulatory models which approximate the vocal tract realistically with a limited number of parameters. A codebook is usually used to represent the mapping between articulatory and acoustic spaces. Numerous inversion methods use codebook lookup procedure combined with optimization procedure to perform the inversion. The result expected is an articulatory trajectory that varies slowly in time and corresponds to a realistic human behavior. The quality of the result is highly dependent on the initial solutions given by the codebook. Thus, it is important that the codebook gives a good coverage of the articulatory space.

Our method represents the codebook in the form of a hypercube structure. Our aim is to reduce time and space required by a systematic sampling of the articulatory space. As we will present later, the obtained codebook is still vast. However, we are sure that the articulatory sampling that we use does not influence the inversion process, unlike other exiting methods. As we already explained in [6], existing codebooks do not represent the articulatory space reliably: points are generally chosen randomly or according some criteria that do not allow an exhaustive representation of the articulatory space. Moreover existing methods are based on the choice of a limited number of points. Even if, in some cases, the number of points is huge, it is not sufficient to cover the whole articulatory space. In fact, all methods based on representing the articulatory space in a discrete manner require a huge number of points to sample the space correctly. For example, as we use the Maeda's articulatory model [5] which describes the vocal tract with seven parameters, if we roughly sample these parameters in the interval $[-3\sigma\ ,\ 3\sigma]$ (where σ is the standard deviation) with a sampling step of $\frac{1}{3}\sigma$, we would obtain $(19^7)\ = 900$ millions points.

The underlying difficulty in generating articulatory codebooks comes from the non-linear nature of the articulatory-to-acoustic mapping. In some articulatory regions a small perturbation of the articulatory parameters could give rise to large acoustic modifications [2]. Thus, a region presenting such a non-linearity could be omitted if the number of points is not sufficient to cover this region. Our idea is to provide an inexpensive but accurate sampling method. It must be accurate to retrieve all the possible solutions, so we can study the influence of the constraints added to the inversion process. In fact, most existing inversion methods exploit compensatory effects of the vocal tract excessively, which can distord iversion results. In this paper, we present our codebook construction method with a good articulatory space coverage and then we present the inversion method exploiting this codebook.

2. HYPERCUBE CODEBOOK GENERATION METHOD

2.1. The idea

As the non-linearity of the mapping is the real problem with the codebook construction, in our approach we discretize the articulatory space densely only in the regions where the mapping is highly non-linear. For this aim, we accepted a hypercube structure to organize the codebook. Thus the articulatory space is represented by a hierarchy of hypercubes. Each hypercube represents a region of the articulatory space in which the articulatory-to-acoustic mapping is linear. The details of the construction of the hypercubes are given in [6]. We just remind that a hypercube of order N is a region of

an N-dimensional space delimited by hyperplanes.

2.2. The method

We suppose that the whole articulatory space is contained in a hypercube. If the mapping is non-linear in this hypercube, we decompose it into sub-hypercubes. Once again, for each sub-hypercube, if the mapping is linear we keep the hypercube, otherwise, we decompose it again. This procedure is repeated recursively until it reachs a hypercube size beyond which, the obtained hypercube is sufficiently small to consider the mapping linear (for details see [6]).

2.3. The linearity test

For all the segments between any pair of hypercube vertexes, we consider the middles of these segments and we linearly interpolate the corresponding acoustic values. Then, we compare them to the corresponding acoustic values calculated by the articulatory synthesizer. If the difference between the acoustic values synthesized and those interpolated is less than a predefined threshold $\Delta\epsilon$, the mapping in this hypercube is considered as linear. Note that the mapping is linear in a hypercube with a margin of error of $\Delta\epsilon$ in the acoustic domain. We assume that this linarity test is sufficient. However, one could use a more accurate test as that described in [2] which would be time consuming.

2.4. Description of the hypercube codebook

A hypercube is defined by its vertexes in the articulatory space. In the hypercube codebook, a hypercube is represented by its origin vertex, the size of one edge (the hypercube can be reconstructed from this information) and the acoustic values of the vertexes. Therefore the hypercube codebook is composed of hypercubes of different sizes. The bigger the hypercube, the more linear the mapping is. Hypercubes are organized in a hierarchical structure.

3. INTERPOLATION IN A HYPERCUBE

3.1. The interpolation method

The information needed could be retrieved form the hypercube vertexes. In fact, the knowledge of acoustic parameters at the hypercube vertexes, provides sufficient information to evaluate acoustic parameters at any articulatory point inside the hypercube. This is achieved by the interpolation procedure presentend below.

To make our interpolation method robust and accurate, it uses the gradient calculated at the nearest vertex to the articulatory point at which we want to evaluate acoustic values. The nearest vertex is chosen among the $128(=2^7)$ vertexes of the hypercube. We can formalize this as follows: let HC be a hypercube. The HC vertexes are represented by the pairs $(P_j, F_j)(j = 1..2^N$, N is the dimension of the articulatory space). P_j is the articulatory vector and F_j is the corresponding acoustic vector, i.e. formant values. Let P_x be the articulatory vector at which we want to evaluate F_x. To interpolate F_x, we use the gradient. Let P_0 be the nearest articulatory vertex to P_x, F_0 the corresponding acoustic vector to P_0. The interpolation by gradient at the nearest vertex P_0 is given by:

$$F_x = F_0 + (P_x - P_0).\nabla F \quad (1)$$

where ∇F is the gradient of F calculated at P_0.

3.2. Experiments

To evaluate the precision of our interpolation method, we generated an acoustic trajectory which is synthesized by our articulatory synthesizer from an articulatory trajectory. Besides this direct calculation, the same trajectory is interpolated from the hypercube codebook using our interpolation method. Then, we compare the proximity of the acoustic signal interpolated from the codebook to the synthesized one. Results obtained are quite good. In fact, the error margin chosen is rather large for the generated codebook (50Hz for the first formant, 75Hz for the second formant and 100Hz for the third one). We generated 37 articulatory trajectories. The mean error for the first two formants does not exceed 10Hz, and 20Hz for the third formant. This is a good approximation which attests the quality of our interpolation method.

4. USING THE HYPERCUBE CODEBOOK FOR THE INVERSION

4.1. The inversion method

To perform the inversion, the articulatory parameters corresponding to the acoustic entry (i.e. the first three formant frequencies and amplitudes of the current signal frame) are retrieved from the codebook. We search for all the hypercubes in the hypercube codebook which contain the acoustic entry. To avoid problems at the boundaries of the hypercube, the acoustic domain corresponding to the current articulatory hypercube is slightly extended. At this step we get back all the hypercubes which contain the acoustic entry but each hypercube contains only one acoustic entry because of the quasi-linearity hypothesis of the articulatory-to-acoustic mapping in a hypercube.

We now explain how the accurate inversion is carried out in each hypercube in order to recover the acoustic data extracted from the original speech signal. Let F be the acoustic vector (represented by the first three formants) to be inverted. HC is the hypercube which may contain an articulatory point giving the acoustic vector. Let P be the articulatory vector (represented by the seven Maeda's artic-

ulatory model parameters) that we are searching for. We use the same principle as that used for the acoustic interpolation. We make the hypothesis that the solution is near the vertex P_0 which is determined as the vertex which leads to formants the closest to those of the point to be inverted. To perform inversion, we solve the following equation which is deduced from (Eq. 1)):

$$F - F_0 = \nabla F.(P - P_0) \tag{2}$$

or:

$$\begin{bmatrix} F^1 - F^0 \\ F^2 - F^0 \\ F^3 - F^0 \end{bmatrix} = \begin{bmatrix} \frac{\partial F^1}{\partial \alpha_1} \frac{\partial F^1}{\partial \alpha_2} \cdots \frac{\partial F^1}{\partial \alpha_7} \\ \frac{\partial F^2}{\partial \alpha_1} \frac{\partial F^2}{\partial \alpha_2} \cdots \frac{\partial F^2}{\partial \alpha_7} \\ \frac{\partial F^3}{\partial \alpha_1} \frac{\partial F^3}{\partial \alpha_2} \cdots \frac{\partial F^3}{\partial \alpha_7} \end{bmatrix} \begin{bmatrix} P^1 - P_0^1 \\ P^2 - P_0^2 \\ \vdots \\ P^7 - P_0^7 \end{bmatrix} \tag{3}$$

Where F^i, F_0^i are components of F and F_0 and P^i, P_0^i those of P and P_0.

As the number of equations is fewer than the number of unknowns, we use an algorithm based on the SVD (*singular value decomposition*) method as described in [3]. We obtain all the solutions of the system. As we look for a solution in the vicinity of the vertex P_0 to respect the hypothesis used to calculate the Jacobian at P_0, we choose the point in the solution space which is the nearest to P_0, in other words, the point of the hypercube where the distance $(P - P_0)$ is minimal. SVD gives the solution in the form of *this* point plus the basis vectors of the null space. Thus SVD gives the inverse point we are looking for. Now, we must verify the hypothesis that we made about the proximity of P regarding the vertex P_0. We carry out the test of the underlying hypothesis for all the vertexes ($2^7 = 128$ vertexes) and the solution is not accepted unless the hypothesis is satisfied.

We apply this procedure for all the hypercubes containing F. As we explain above, three constraints must therefore be verified to accept a point as a solution:

- F_0, the acoustic image of the vertex P_0, is the nearest to F (the acoustic vector to be inverted);
- P_0 must be the nearest vertex to P, the articulatory vector result of the inversion process calculated by interpolation;
- the articulatory vector P result of the inversion must be in the hypercube that we use to perform the inversion.

First and second conditions are concurrent which ensure the proximity of the nearest vertex in both spaces (acoustic and articulatory space), and improves the linearity of the mapping in the hypercube. The third condition eliminates cases where the result is out of the current hypercube. If one of these conditions is not verified, the solution is rejected.

4.2. Experiments

We generated a complete hypercube codebook for the seven parameters of Maeda's articulatory model. The margin errors $\Delta\epsilon$ for the linearity test is 50 Hz for the first formant, 75 Hz for the second formant and 100 Hz for the third. The number of the hypercubes is 412875. Of course, this still very large codebook, but it has the advantage that it represents the whole articulatory space. Thus this hypercube codebook gives all the possible solutions. We synthesized acoustic trajectories using articulatory synthesizer by varying one or more articulatory parameters, the others being kept at the neutral position. We use this synthetic trajectories to compare the result of the inversion to known articulatory trajectories in order to check the accuracy of the inversion method. For that purpose formant parameters of the synthesized acoustic signal are fed in the inversion procedure to obtain all the solutions of the inversion.

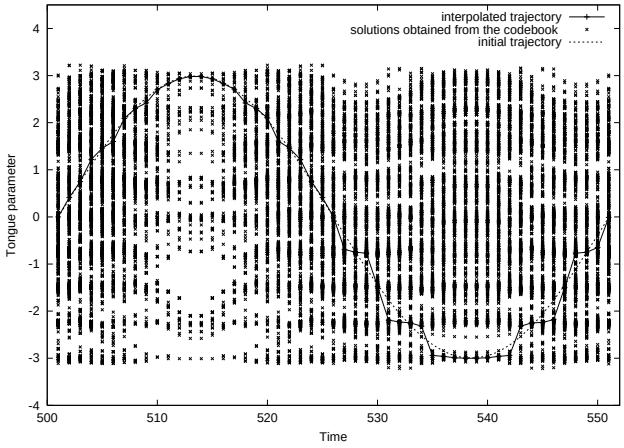

Figure 1: The inversion solutions of a simulated articulatory trajectory. For sake of clarity of the figure, not all the solutions are presented here.The dashed curve is the original trajectory and the continuous curve is the solution trajectory of the inversion.

In Fig. 1 and Fig 2, we present the inversion solutions for a signal which corresponds to synthetic articulatory trajectory in which only the tongue parameter varies. As we can see, for each acoustic entry we obtain a vast number of solutions. For this example, we obtain an average of 550 solutions for each acoustic entry. As we did already explain, our hypercube codebook is a complete representation of the articulatory space, we did not eliminate any hypercube because that would constrain the inversion process. The only eliminated hypercubes are those which are situated in forbidden regions [1] because they do not correspond to any valid vocal tract shape (as explained in [6]). The set of all the possible articulatory trajectories which may give rise to a given acoustic signal can be exploited to study the variability of the production of vowels and the compensatory effects used by a real speaker.

As our first aim is to obtain an articulatory trajectory which is smooth and varies slowly in time, we used a non-linear

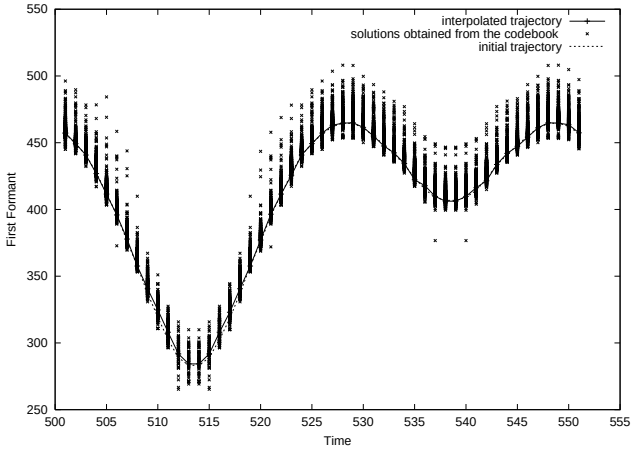

Figure 2: The inversion solutions of a simulated articulatory trajectory in the acoustic space (first formant). All the solutions are in the vicinity of the original acoustic signal(dashed curve).

smoothing method [4] which is based on dynamic programming. This method provides good results. In fact, as we can see in Fig.1, among all the solutions obtained our algorithm succeeds in recovering an articulatory trajectory close to the initial by incorporating a bonus which expresses the fitting in amplitude between synthetic and original formants. In Fig.2, we present the result in the acoustic space for the first formant, to verify whether the acoustic proximity is respected or not. It is clear that all the solutions are in the vicinity of the initial signal and the trajectory solution chosen is very close to the initial one.

5. CONCLUSION

Our hypercube codebook turns out to be a very good representation of the articulatory-to-acoustic mapping which respects its non-linearity. We used an improved interpolation method to extract solutions from the codebook and our non linear smoothing algorithm allows good results to be obtained. Our current work is to reduce the inversion time by parallelizing the inversion procedure. We will exploit this approach of the acoustic-to-articulatory inversion to study the particularities of the articulatory space and more precisely how the vocal tract capacity for compensation is used for coarticulation.

6. 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. 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.

3. G.H. Golub and C.F. Van Loan. In *Matrix Computations*, chapter 12. Johns Hopkins University Press, Baltimore, 1989.

4. Y. Laprie and B. Mathieu. A variational approach for estimating vocal tract shapes from the speech signal. In *Proceedings of the International Conference on Acoustics, Speech and Signal Processing*, volume 2, pages 929–932, Seattle, USA, May 1998.

5. 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, Mai 1979.

6. S. Ouni and Y Laprie. Design of hypercube codebooks for the acoustic-to-articulatory inversion respecting the non-linearities of the articulatory-to-acoustic mapping. In *Proceedings of the 6th European Conference on Speech Communication and Technology*, pages 141–144, Budapest, September, 1999.