RAMCESS FRAMEWORK 2.0 REALTIME AND ACCURATE MUSICAL CONTROL OF EXPRESSION IN SINGING SYNTHESIS

Nicolas d'Alessandro [1], *Onur Babacan* [2], *Barış Bozkurt* [2], *Thomas Dubuisson* [1], *Andre Holzapfel* [3], *Loïc Kessous* [4], *Alexis Moinet* [1], *Maxime Vlieghe* [1]

[1] Signal Processing Laboratory, Polytechnic Faculty of Mons (Belgium)
[2] Electrical and Electronics Engineering Dpt, Izmir Institute of Technology (Turkey)
[3] Computer Science Dpt, University of Crete, Heraklion (Greece)
[4] Speech, Language and Hearing, Tel Aviv University (Israel)

ABSTRACT

In this paper we present the investigations realized in the context of the eNTERFACE 3rd summer worshop on multimodal interfaces. It concerns the development of a new release of the RAMCESS framework (2.x), preserving the accurate and expressive control of voice quality dimensions available in the existing system (from 2006), but adding coarticulated speech possibilities. This work will be achieved by the analysis, separation and modeling of the glottal source component from a limited-size and adapted singing voice database.

1. INTRODUCTION

Expressivity is nowadays one of the most challenging topics studied by researchers in speech processing. Indeed, recent synthesizers provide acceptable speech in term of intelligibility and naturalness but the need to improve human/computer interactions has brought researchers to develop systems that present more "human", more expressive skills.

Speech synthesis research seems to converge towards applications where multiple databases are recorded, corresponding to a certain number of labelled expressions (e.g. happy, sad, angry, etc.). At synthesis time, the expression of the virtual speaker is set by choosing the units in the corresponding database, then used in well-known unit selection algorithms.

Recently remarkable achievements have also been reached in singing voice synthesis. We can highlight naturalness and flexibility of Bonada *et al.* [1] algorithms where singing frames are structured at a high performance level. These kinds of technologies seem mature enough to allow for the replacement of human vocals with synthetic, at least for backing vocals.

However existing singing systems suffer from two restrictions: they are aiming at mimicking singers rather than offering real creative voice timbre possibilities, and they are generally limited to note-based MIDI controllers.

In this context, we propose to investigate an original option. We postulate that, even if the use of databases is strategic in order to preserve naturalness, voice modeling has to reach a higher level. These improvements have to meat particular needs, such as more realistic glottal source/vocal tract estimation, manipulation of voice quality features at a perceptual and performance level, and strong real-time abilities. The other issue concerns mapping strategies that have to be implemented in order to optimize the performer/synthesizer relation.

In this paper we present the work that have been achieved during the 3rd eNTERFACE workshop. This work is in the continuity of both 2005 and 2006 work about voice quality manipulation. Thus, after a short introduction of the mixed-phase model of voice production (cf. section 2), we present the existing framework, called RAMCESS 1.x, which was used as a starting point for our investigations (cf. section 3). Then we describe the analysis routines, implemented in order to separate glottal source and vocal tract contributions on a given database (cf. section 4). We also give a comment about the use of SDIF encoding (cf. section 5). Finally we give an overview of the RAMCESS 2.x framework (cf. section 6).

2. MIXED-PHASE MODEL OF VOICE PRODUCTION

The mixed-phase speech model [2] [3] is based on the assumption that speech is obtained by convolving an anti-causal and stable source signal with a causal and stable vocal tract filter. The speech signal is a mixed-phase signal obtained by exciting a minimum-phase system (vocal tract system) by a maximum-phase signal (glottal source signal). (It should be noted that the return phase component of the glottal source signal is included in the vocal tract component since it also has minimum-phase characteristics.) The mixed-phase model assumes that speech signals have two types of resonances; anti-causal resonances of the glottal source signal and causal resonances of the vocal tract filter.

The mixed-phase modeling plays an important role in both analysis and synthesis in our study. In analysis, estimation of glottal flow is achieved by ZZT (Zeros of Z-Transform) decomposition (cf. section 4) which decomposes the signal into anti-causal and causal components. In synthesis, mixed-phase signals are synthesized by exciting minimum phase filters with maximum phase excitation (cf. sections 3 and 6). These result in achieving a natural timber for the output sound which is very important for a digital instrument.

3. RAMCESS 1.X SYNTHESIS ENGINE

In this section we describe the existing framework, developped during preceding eNTERFACE workshops. This project, called RAMCESS (i.e. *Realtime and Accurate Musical Control of Expression in Singing Synthesis*) can be used as an original tool for studies on voice quality, analysis by synthesis or by gesture, and the performance of high quality singing voice. This section exposes new elements of the synthesis software library (which becomes progressively a collaborative project, called VQCLIB [4]), improvements achieved in the modeling of glottal source and vocal tract in order to preserve expressivity and achieve realtime production, and a comment about dimensionnal control of voice quality.

3.1. Voice Quality Control Library

The *Voice Quality Control Library* (or VQCLIB) [4] is a collaborative project that aims at developing a large set of modules for realtime environments (at this time MAX/MSP, and PURE DATA in the pipeline), focusing on the accurate and expressive realtime control of voice timbre features. Indeed voice production patches can be implemented step-by-step, opening large possibilities on glottal signal configuration, vocal tract filtering, or multiple representations of voice signal features. This initiative aims at attracting scientific research or computer music communauties in interactive forums, in order to share knowledges and know-how about realtime voice synthesis. The first version has been released, and a major update is in the pipeline. VQCLIB serves now as the basis of all further investigations on realtime voice synthesis. An example patch is illustrated in the Figure 1.

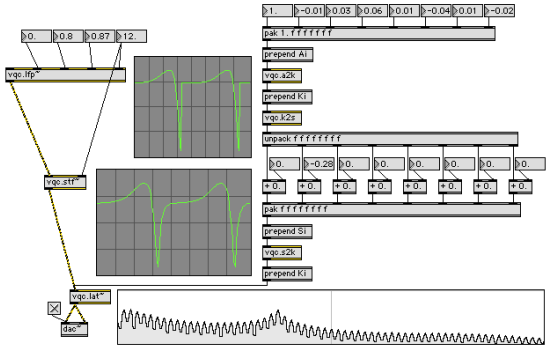

Figure 1: *Example of* MAX/MSP *patch using several objects of* VQCLIB.

3.2. Computation of the Glottal Flow Signal

Glottal pulse signal can be synthesized with many different models. Typical temporal and spectral representations of one period of the derivative of the glottal flow (with usual descriptive parameters: T_0, GCI, O_q, α_m and T_L) are illustrated in Figure 2. In term of flexibility and quality, we can particularly highlight LF [5] and CALM [6] models. However none of them is really suitable for realtime processing. On the one hand, LF parameters are the solution of a system of 2 implicit equations[1] which is known to be unstable. On the other hand, CALM is linear filter processing but one of the filters has to be computed anticausally. This is possible in realtime but with a limited flexibility [7].

The improvement that we propose can be seen as a compromise between both LF and CALM models, or a kind of *spectrally enhanced LF model*. In order to avoid the resolution of implicit equations, only the left part of the LF model is used. It is computed using the left part (cf. equation (1)) of the normalized glottal flow model (GFM) described in [8].

$$n_g(t) = \frac{1 + e^{at}(a\frac{\alpha_m}{\pi} sin(\pi t/\alpha_m) - cos(\pi t/\alpha_m))}{1 + e^{a\alpha_m}} \quad (1)$$

where t evolves between 0 and 1, and is sampled in order to generate the $O_q \times \frac{F_s}{F_0}$ samples of the opened phase (O_q : open quotient, F_0 : fundamental frequency, F_s : sampling rate); α_m is the asymetry coefficient and $a = f(\alpha_m)$ is the pre-processed buffer of solutions of the equation (2).

[1] The LF model gives equations of temporal shapes of both curves on the left and the right of the GCI. The conditions are then 1) the integration of the whole period has to be 0, and 2) left and right curves have to be connected at the position of the GCI.

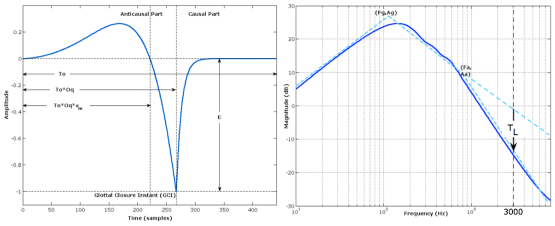

Figure 2: *Temporal (left) and spectral (right) representation of the derivative of one period of glottal flow, with usual descriptive parameters: T_0 (fundamental period), GCI (glottal closure instant), O_q (open quotient), α_m (asymetry coefficient) and T_L (spectral tilt).*

$$1 + e^{a}(a\frac{\alpha_m}{\pi} sin(\frac{\pi}{\alpha_m} - cos(\frac{\pi}{\alpha_m}))) \quad (2)$$

Then the right part (the return phase) is generated in spectral domain, which means that the left LF pulse is filtererd by the spectral tilt low-pass first order filter presented in [6]. This option is also prefered because a long filter-generated return phase smoothly overlaps with following pulses, thus avoiding discontinuities. The complete process, also integrating the derivation of the pulse and the normalization (in order to control separately spectral tilt and energy of the pulse), is illustrated in Figure 3.

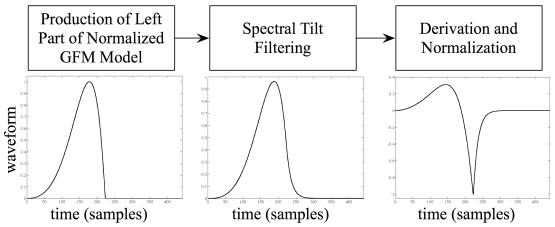

Figure 3: *Synthesis of the glottal pulse by combination of LF left part time-domain generation, spectral tilt filtering, derivation and normalization.*

3.3. Computation of the Vocal Tract Filter

The vocal tract is computed with a simple tube model. LPC coefficents a_i are converted into reflection coefficients k_i, and then into area (or section) coefficients S_i, defining geometrical properties of vocal tract. A complete coefficient conversion framework have been developed in order to jointly manipulate multiple representations (spectral and physical) of the vocal tract. This approach is powerful in order to create typical voice quality effects: vowel interpolation, obstructions, singer formant, etc [7]. A representation of the vocal tract by its sections (S_i) is illustrated in Figure 4.

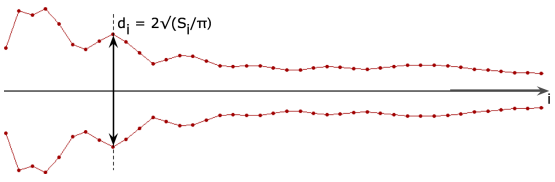

Figure 4: *Geometrical representation of the vocal tract, thanks to its S_i.*

3.4. Dimensional Study of Voice Quality

On the top of typical synthesis parameters that VQCLib can manipulate (F_0, O_q, α_m, T_l, spectral and geometrical features of vocal tract), it is interesting to build a layer which is able to provide more perception-based control to the performer. This dimensional study of voice quality have been achieved, resulting in a set of dimensions and their corresponding mapping equations with synthesis parameters [7]. As an example, we describe here the way we define the behavior of the open quotient (O_q) as a function of *tenseness* (T), *vocal effort* (V) and *registers* (M_i) in equations (3), (4) and (5).

$$O_q = O_{q_0} + \Delta O_q \tag{3}$$

$$O_{q_0} = \begin{cases} 0,8 - 0,4\,V & \text{if } M_i = M_1 \text{ (modal)} \\ 1 - 0,5\,V & \text{if } M_i = M_2 \text{ (chest)} \end{cases} \tag{4}$$

$$\Delta O_q = \begin{cases} (1 - 2\,T)\,O_{q_0} + 0,8\,T - 0,4 & \text{if } T \leq 0,5 \\ (2\,T - 1)\,O_{q_0} + 2\,T + 1 & \text{if } T > 0,5 \end{cases} \tag{5}$$

4. EXPRESSIVE VOICE ANALYSIS

This section describes investigations that have been realized in order to work on database contents instead of pure rule-based modeling. Different steps are described. First we discuss choices that have been made in the design and the segmentation of the first database. Then we propose the use of a sequence of different algorithms (ZZT, LF fitting, ARX-LF and spectral compensation) in order to achieve a high-quality decomposition of speech signals in its glottal source and vocal tract contributions, and the reliable modeling of these contributions. Finally we propose a suitable format, called SDIF for the encoding of the parameters of these models.

4.1. Database Design

In order to achieve coarticulated speech synthesis, we need to record a database containing at least one instance of each diphone (i.e. a two-phoneme combination) of a language, whatever it is. However, as we are investigating first steps in this research, our target is to be able to synthesize a few sentences with high quality and realtime control over pitch and voice quality parameters (as a proof of concept). Therefore we think that we do not need to record now a whole database but only sentences we have decided to produce. Consequently database design is made as follows.

- Each sentence is recorded with constant pitch and with voice quality as constant as possible, so as to constrain the range of subsequent analysis steps.
- Each sentence is recorded two times. In the first recording, the speech is slightly slower than in natural speech while the second recording is made at a normal rate. The first recording is made to ensure that each vowel has a non-coarticulated central region which will allow us to easily connect diphones together during synthetic voice production without loosing or mismatching the natural coarticulation.
- Each sentence is a succession of voiced and unvoiced phonemes and, due to their complexity, we don't use voiced consonants (e.g. /v/, /z/, /b/, /g/, ...) nor liquids (e.g. /R/, /l/, ...).
- We choose two phonemic sequences, one with different plosive and fricative consonants and one with the same consonant interleaved with various vowels. Their SAMPA transcriptions are respectively:

$$/t\,a\,k\,a\,p\,a\,f\,a\,S\,e/$$
$$/t\,a\,t\,2\,t\,i\,t\,o\,t\,u/$$

The recordings are made on one channel at 44100 Hz and each sample is stored in 16 bits PCM format. A MAX/MSP application which is illustrated in the Fig 5 helps the speaker maintaining pitch, voice quality and syllabic rate as constant as expected.

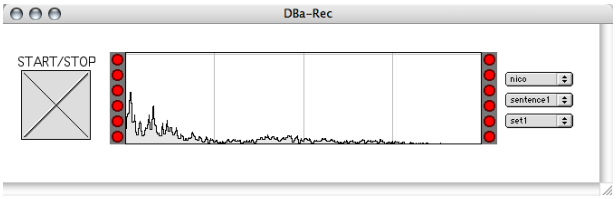

Figure 5: *Graphical User Interface for the recording of the database, with an auditory feedback helping the speaker maintaining the pitch, voice quality and syllabic rate as constant as possible.*

4.2. Database Segmentation

The four phonemic sequences are manually segmented into two sets: the vowels and the consonants. Each vowel is in turn divided in three parts. The first and third parts are the regions of the vowel that are coarticulated respectively with the preceding and following consonant (or silence). These are the left and right transient parts of the vowel. These coarticulated parts of speech usually contain a few (less than 10) periods of the vowel. The central part, also called the steady part is the region of speech that we can consider as coarticulation-free and thus actually quasi-stationnary (cf. Figure 6).

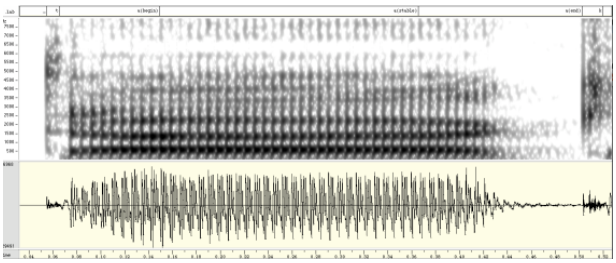

Figure 6: *Typical example of segmentation on a syllable of the database.*

Next, for each voiced region, the first GCI (GCI_1 : corresponding to the GCI of the first glottal pulse of a voiced island, meaning not overlapped with preceding vocal tract response, and thus clearly observable) is automatically approximated as the position of the minimum of the speech waveform in a region of $1.2 \times T_0$ after the unvoiced-to-voiced (e.g. /t/ to /a/) segmentation point. Finally positions of these GCIs are checked and corrected if necessary.

4.3. ZZT Representation

In order to find precise position of following GCIs and a first estimation of the glottal flow parameters, the ZZT representation

[3] of speech is used. For a series of N samples $(x(0), x(1), \cdots, x(N-1))$ taken from a discrete signal $x(n)$, this representation is defined as the set of roots (zeros of the polynomial) $(Z_1, Z_2, ..., Z_m)$ of its corresponding Z-Transform $X(z)$ (cf. equation (6)).

$$X(z) = \sum_{n=1}^{N-1} x(n) z^{-n} = x(0) z^{-N+1} \prod_{m=1}^{N-1} (z - Z_m) \quad (6)$$

This representation implies to compute roots of polynomials [9] of which the degree increases with the sampling frequency, introducing errors on the estimation of zeros in high frequencies. That is why we perform the analysis at 16000 Hz, thus first downsampling the waveforms of the database.

According to the mixed-phase model of speech, the ZZT representation of a speech frame contains zeros due to the anticausal component (mainly dominated by the glottal source) and to the causal component (mainly dominated by the vocal tract response) [10]. Consequently zeros due to the anticausal component lie outside the unit circle, and zeros due to the causal component inside the unit circle. Under some conditions about the location, the size and the shape of the window analysis, zeros corresponding to both anticausal and causal contributions can be properly separated by sorting them according to their radius in the z-plane. The waveform and the spectrum of these contributions are then computed by the Discrete Fourier Transform (equation (7)).

$$X(e^{j\phi}) = G e^{(j\varphi)(-N+1)} \prod_{m=1}^{N-1} (e^{(j\varphi)} - Z_m) \quad (7)$$

A typical windowed speech frame is displayed in Figure 7, and results of decomposition based on zeros separation are displayed in Figure 8, illustrating relevant shapes of derivative of glottal flow and vocal tract impulse response.

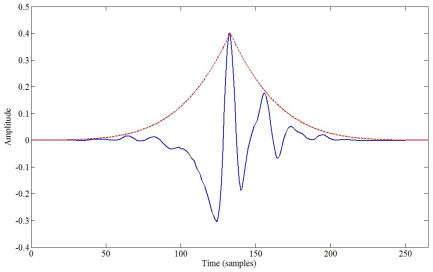

Figure 7: *Example of a typical speech frame (blue), weighted by a Hanning-Poisson window (red). With that shape, the signal is prepared to be processed by the ZZT-based decomposition algorithm.*

4.4. GCI Adjustment

The location, the size and the shape of the window chosen for the analysis have a huge effect on the ZZT representation, thus on the estimation of anticausal and causal components. A review of these different conditions and their effects can be found in [3, 11]. We can highlight that the analysis window has to be centered on a GCI in a really precise way, as the ZZT-based decomposition is really sensitive to wrong positioning.

In the framework of this project, we first use this sensitivity in order to get precise location of GCIs as instants where the ZZT-based decomposition is well achieved. As illustrated in Figure 9, a first approximation of GCIs can be extrapolated from

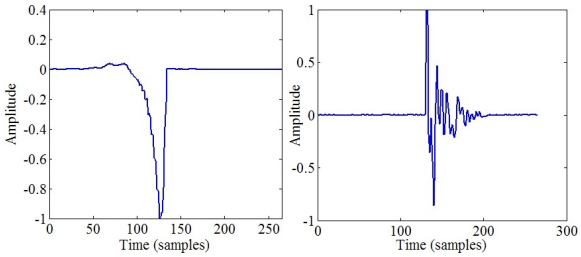

Figure 8: *Results of ZZT-based decomposition for a speech frame: the anticausal component, close to the derivative of glottal flow (left) and the causal component, close to the vocal tract impulse response (right).*

the first GCI of a voiced island marked in the segmentation task (GCI_1) and by marking next ones thanks to an estimation of the pitch contour (computed e.g. by autocorrelation).

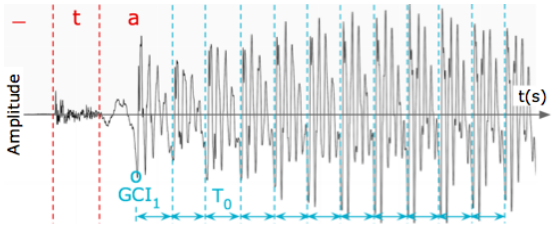

Figure 9: *First approximation of GCI positions taken from the first onset of the voice island (with a appearing GCI: GCI_1) and extrapolation thanks to a pitch contour estimation (e.g. by autocorrelation).*

Recent experiments have shown that it is possible to obtain reliable glottal source and vocal tract estimates by shifting few samples around each estimated GCI [12]. If the maximum shift is set e.g. to 4 samples, it gives us, for each estimated GCI, 9 candidates for the glottal source. The challenge is to find which shift gives the best decomposition, in terms of proximity to the LF model. By comparing a correct glottal pulse and a wrong one in the spectral domain (cf. Figure 10), we can observe that their behaviour is quite the same below 2 kHz and significantly different in higher frequencies.

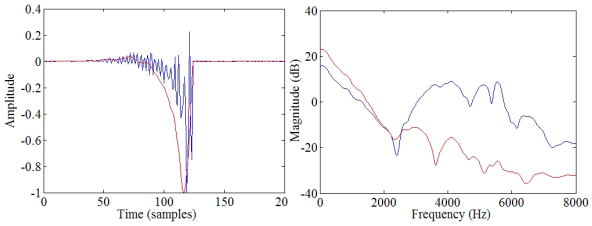

Figure 10: *Good glottal pulse (red) vs wrong glottal pulse (blue): illustration of the spectral criterion, which is based on the clear increasing of high frequencies when decomposition fails.*

In order to choose the best one among all candidates, we define a spectral criterion as the ratio between the energy in 0-2 kHz frequency band and the energy in the whole spectrum (0-F_s/2).

$$Criterion = \frac{Energy_{[0-2000Hz]}}{Energy_{[0-8000Hz]}} \quad (8)$$

Thus for each GCI, the best glottal pulse is chosen as the one which maximises this criterion among all the possible candidates. The location of every GCI can thus be refined (and the pitch estimation as well). Figure 11 shows the result of decomposition after shifting and taking the best candidate.

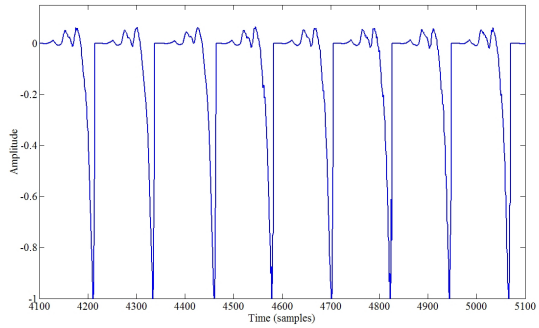

Figure 11: *Result of decomposition (anticausal contribution) after the procedure of GCI shifting (4), the computation of the spectral criterion, and the choice of the best candidate.*

4.5. Estimation of O_q and α_m

Once the best glottal sources are obtained for all the GCIs, the open quotient (O_q), the asymmetry coefficient (α_m) and the onset time of the glottal source can be estimated by the anticausal component to the LF model. For each GCI, this fitting is performed as:

- Computation of all the LF waveforms with the same pitch period and energy as the one obtained from the analysed speech frame at the considered GCI, and a defined range for O_q (0.3-0.9) and α_m (0.6-0.9), thus defining a codebook of LF waveforms.
- Windowing of the LF codebook in order to take into account the effect of the window analysis on the estimated glottal source;
- Synchronization between the estimated glottal sourcet obtained by the ZZT-based decomposition and the LF codebook, performed by an alignement between the GCI of the glottal source and the GCI of the LF waveforms;
- Computation of the square error between each windowed LF waveform of the codebook and the estimated glottal source;
- O_q and α_m are defined as the couple of parameters corresponding to the LF waveform closest to the glottal source;
- Once O_q is computed, the onset time (defined as the beginning of the opening phase of the vocal folds) can be estimated.

The result of the fitting and the estimation of the onset time are displayed in Figure 12 for the glottal sources shown in Figure 11. We can observe that estimated LF waveforms are quite close to the estimated glottal sources. Then Figure 13 shows the evolution of O_q and α_m for the whole part */ta/* of the database. These evolutions also confirm that the constant voice quality we had targeted at recording time has been somehow achieved by our speaker.

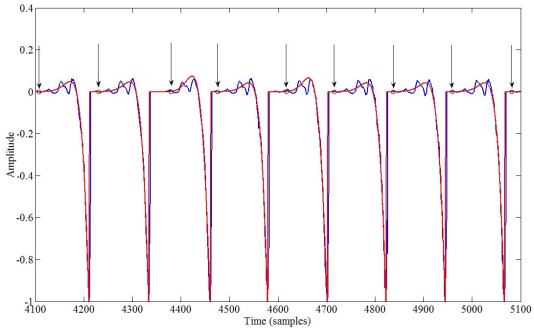

Figure 12: *Fitting in time domain between the anticausal component coming from ZZT-based decomposition (blue) and the LF model of glottal pulse (red). Black circles and arrows show onset instants.*

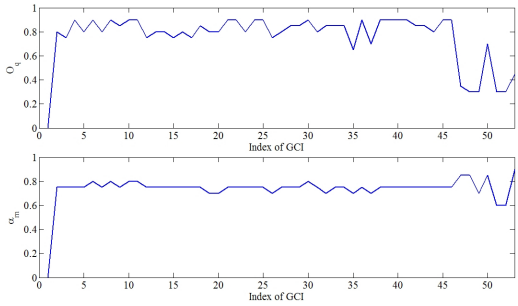

Figure 13: *Evolution of estimated open quotient (O_q) (upper panel) and asymetry coefficient (α_m) (lower panel), coming from the time-domain fitting, for the /ta/ sound in the database.*

4.6. ARX-LF Optimization on Sub-Codebook

In the source/filter model [13], a sample $y(n)$ of speech is modeled by the AR equation :

$$y(n) = -\sum_{i=1}^{p} (a_n(i)y(n-i)) + b_n x(n) + e(n), \qquad (9)$$

where $x(n)$ is the source, $e(n)$ is the residual and $a_n(i)$ and b_n are the AR filter coefficients (these coefficients represent the vocal tract which is varying over time instants n and thus their values evolve accordingly).

In the usual LP modelization, the source $x(n)$ is supposed to be impulse train or white noise and, given this assumption, some mathematical developments lead to the Yule-Walker equations which can be solved to obtain the AR filter coefficient. This is the classical LP analysis.

In [14] and [15] Vincent *et al.* developped a more complex model which assumes that the source $x(n)$ is a glottal flow in the voiced segments of speech. Therefore the hypothesis on the nature of the source has been changed and Yule-Walker equations cannot be used anymore. Instead, one can write equation (9) for successive values of n and solve the resulting set of linear equations the obtain a set of prediction coefficients minimizing the residual energy.

As explained in subsection 4.4 the results of analysis on voiced speech are strongly dependent upon the choice made to build the analysis window. Accordingly we work with particular position and length configurations for that window. It is GCI-centered and has a length equal to two periods ($2\,T_0$). Moreover, the analysis window is Hanning-weighted.

Consequently, for the k^{th} GCI, we can write equation (9) for 2 T_0 values of n ($[GCI_k - T_0 + 1 \dots GCI_k + T_0]$) and we solve the system of equations in $p+1$ unknowns : $a_k(i)$ and b_k, considered as constant around GCI_k, instead of varying from sample to sample:

$$Y = MA + E, \tag{10}$$

where Y is a vector of $y(n)$, M is the concatenation of a matrix of $-y(n-i)$ values and a vector of $x(n)$, all of them for $n = [GCI_k - T_0 + 1 \dots GCI_k + T_0]$. A is the vector of unknown values $a_k(i)$ and b_k. E is a vector of residuals $e(n)$, that is to say a vector of modelization errors that we want to minimize when computing A.

We then solve these equations for different glottal flows x_w (particular values of x, called the *codebook* hereafter). The glottal flow minimizing the modelization error (see below) is considered as the most correct estimate of the actual glottal flow produced by the speaker. In the next paragraphs we will explain how we create the codebook and compute the corresponding errors.

The glottal flows are built using a spectrally enhanced LF model [16] and are driven by three parameters: O_q (open quotient), α_m (asymetry coefficient) and T_l (spectral tilt). Nevertheless a codebook of glottal flows based on the possible variations of these three parameters would be rather bulky and solving (10) for all the waveforms stored in that codebook would be CPU expensive.

Fortunately O_q and α_m estimates are already known (thanks to ZZT analysis and LF fitting techniques) which allows us to select a part of the codebook that we call a sub-codebook. T_l is the only varying parameter of that sub-space of LF pulses. Moreover, although we are confident in the estimate of O_q and α_m, we can refine these results by selecting a somehow larger sub-codebook, allowing some slight variations of O_q and α_m around the initial estimates.

Let us say there are W glottal flows x_w in the sub-codebook. As said above, for each one of them we can compute the $a_k(i)$ and b_k coefficients and therefore re-synthesize an approximation y_w of the speech frame y by applying equation (9).

The error for word x_w is then measured as the Euclidean distance between the re-synthesized speech frame y_w and the original analysis window y. Note that both y_w and y are Hanning-windowed.

$$E_w = \sqrt{\sum_{n=1}^{2T_0} (y(n) - y_w(n))^2} \tag{11}$$

However before actually computing errors, two important points remain: the GCI position and the filter stabilization.

Indeed, the estimate of each GCI position is provided by the ZZT analysis. Although that position fits very well for ZZT decomposition, it's not necessarily the best one for ARX decomposition. For that reason one more step is added to the algorithm explained above: we do not consider only the analysis window y centered on the GCI approximation coming from ZZT but also windows centered a few points on the left and on the right of that location.

In our implementation we look three points before and after the position of the current GCI. Henceforth we will have $7W$ error measurements and not only the minimum error will give us the best guess for the glottal flow parameters but also for the GCI optimal position.

Finally, although the Levinson-Durbin method that solves the Yule-Walker equations guarantees that the AR filter has all of its poles inside the unit circle and therefore is stable, this is no longer the case when solving equation (10). Consequently, the last step before synthesizing any of the y_w is to reflect the outside poles inside the unit circle and adapting the value of parameter b accordingly [13].

All these steps are performed at a sample rate of 8kHz which allows us to get reliable estimates of T_l and the positions of GCIs (as well as an estimate of the filter parameters for that rate). However high quality singing synthesis is produced at higher sampling rate such as 44.1 or 48 kHz.

The glottal flow parameters O_q, α_m and T_l are independent of the sampling frequency and therefore they can be used as is. On the contrary the filter coefficients rely upon the sampling rate and need to be recomputed. The task is fast and easy since we just have to solve equation (10) once with a different sampling frequency for $y(n)$ (for which we have the original recording at 44.1kHz) and $x(n)$ (which is the LF/Klatt model and thus can be produced at any given rate for the fixed O_q, α_m and T_l).

To make things short, equation (10) is first solved at 8kHz for 24 $a(i)$ parameters ($p = 24$) and considering a sub-codebook with O_q and α_m constant and T_l varying between 3dB and 20dB (with a 1dB step). Then it is solved at 44.1kHz for $p = 46$ and O_q, α_m and T_l constant. Results are illustrated in the Figure 14.

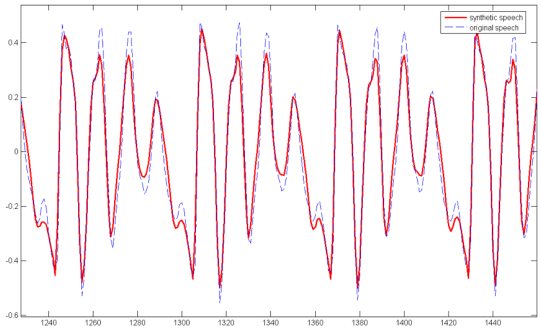

Figure 14: *Superposition of original (blue) and resynthesized (red) signals, after the computation of ARX-LF on a sub-codebook defined by ZZT-based parameters.*

4.7. Vocal Tract Response Compensation

It's observed that synthesis obtained by exciting the ARX filter with the glottal flow results in a certain loss of high frequency components. To compensate for this effect, we devised a simple compensation method via AR filtering. For this, the AR compensation filter is obtained by linear predictive analysis of an impulse response obtained in the following way. The frequency response of the original signal is divided by the frequency response of the synthetic signal, and the inverse Fourier transform of the result is taken. A sample result of the compensation is presented in Figure 15. The obtained AR compensation filter is combined (by cascading) with the ARX filter to obtain a single filter that will perform the synthesis in one stage.

5. SOUND DESCRIPTION INTERCHANGE FORMAT

The analysis tool being MATLAB and the realtime synthesis tool being MAX/MSP, a compatible format must be used. It was found that SDIF tools exist for both of these softwares. SDIF means *Sound Description Interchange Format*. This kind of file does not contain the sound itself, but sets of descriptive parameters e.g. coefficients of an estimated AR filter for speech or values of harmonics for additive synthesis.

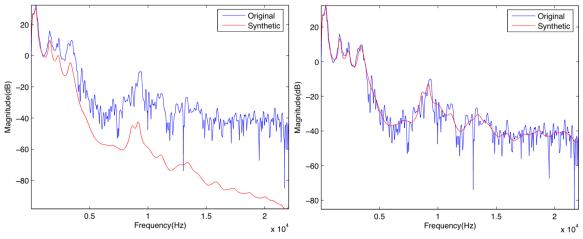

Figure 15: *Spectral envelope of original (blue) and synthetic (red) signals before (left) and after (right) the HF compensation.*

A SDIF file is divided in many simple parts. The biggest entity in such a file is called a *stream*. Streams are series of time-tagged frames and are identified by a stream ID (integer). Frames are identified inside a stream by their corresponding instants on the timeline. They have to be stored in increasing time sequence. Two frames of the same stream may not have the same time tag. In each frame, it is possible to store multiple matrices which differ from each other by their type.

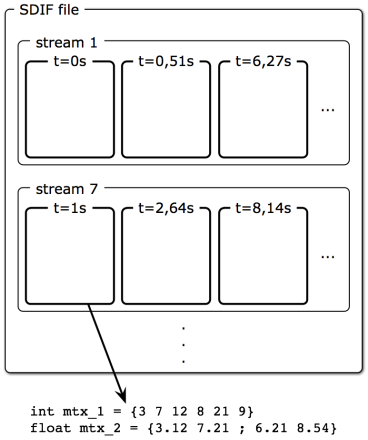

Figure 16: *Structure of a* SDIF *file: streams, time-tagged frames and matrix.*

5.1. Storage of MATLAB Analysis Results

Different functions are available in order to deal with SDIF files in MATLAB [17]. There are two possible ways to read and verify SDIF contents: *loadsdif* and *loadsdiffile*. However the most interesting function in this project is the *sdifwrite*. It allows to encode easily SDIF files frame by frame. Parameters to be specified in each frame are the stream ID, the frame type, the frame time, and type and content for each of the matrices. MATLAB SDIF encoder presented some unexpected results as there was always an unintended stream with a negative stream ID in addition to encoded streams. Knowing this problem and adapting reading/writing algorithms to it, encoded streams work properly.

In this project, we decide to create, for each GCI-centered analysis window, a SDIF frame containing a single vector with parameters of source and tract structured as follow (for $t = t_k$): reflection coefficients (K_i), fundamental frequency (F_0), open quotient (O_q), asymetry coefficient (α_m), spectral tilt (T_L) and and arbitrary index informing about the tag that has been used in the segmentaion file (S_k).

$$\{K_1 ... K_n \; F_0 \; O_q \; \alpha_m \; T_L \; S_k\}|_{t=t_k}$$

Regions of the database corresponding to unvoiced speech are represented in the SDIF file as a island of ten frames, with time tag distributed on the whole unvoiced period, and identified by their S_k values.

5.2. SDIF Interface in MAX/MSP

The voice synthesis is performed on MAX/MSP. Here we present a list of the existing MAX/MSP objects existing to deal with SDIF files [18].

The first SDIF object to use is the *sdif-buffer* object. It allows storing temporarily the content of a SDIF file. There are three main messages understood by *sdif-buffer*. `[streamlist myfile.sdif]` lists the streams existing in the file. `[frame list myfile.sdif]` lists and prints all the existing frames in myfile.sdif. Finally `[read-stream-number myfile.sdif StreamID]` allows to load into the buffer a whole stream, including all the frames and all the matrixes it contains. Once a stream has been read, it is possible to know his properties through the message `[print]`.

Other objects allow manipulating the stream contained in the *sdif-buffer* object. First *sdif-tuples* is able to output the data content of a specified frame. The message which allows outputing such data is `[tuple]`. Arguments that may be specified are the frame time, the matrix type, indexes of columns of the matrix to output, and the format of the output. Possible formats are data outputed by row or concatenated data. The *sdif-range* external is able to scan the stream contained in a buffer to know the maximum number of columns in the matrices of a specified type or the maxima and minima of each column of a specified type of matrix. The *sdif-menu* external is a graphical interface to manage different streams and obtain information about each of them. Finally the *sdif-listpoke* allows the user to write his own matrices in frames and put these together to form a stream in a buffer. At this time it is not possible by a simple way to write data into a sdif file but it should soon be possible through the FTM library [19].

The only objects needed to use a SDIF file inside MAX/MSP are *sdif-buffer* and *sdif-tuples* objects. The first step is to load a stream into a buffer. Then, at each new frame instant, all the parameters are transmitted in a list by the *sdif-tuples* object. Some objects then slice this list in a list of reflection coefficients redirected to the lattice filter (*vqc.lat~*), anticausal glottal parameters redirected to the *vqc.lfp~* object and the spectral tilt redirected to the *vqc.stf~* object. Finally the equalization gain is isolated and used as a multiplier after the lattice filter.

6. OVERVIEW OF RAMCESS 2.X FRAMEWORK

Finally these investigations in expressive voice analysis allow us to draw the first contours of the next step in the development of the RAMCESS framework (version 2). This structure combines the realtime synthesis layer of RAMCESS 1.x system with now the encapulsation of lots of analysis and decomposition processes in a SDIF database, imported in the performance application, with modules communicating by OSC [20].

A gestural sollicitation is captured by sensors, driving the behavior of a typical digital instrument. In our experimentations, the HandSketch [21] has been used. Typical mappings related to phonetic contents and dimensions of voice quality are applied. At this level, information extracted from the database is used in order to generate more relevant syntheis parameters, especially in the realtime production of the coarticulation. Finally sound is produced by the VQCLIB layer. This process is summarized in Figure 17.

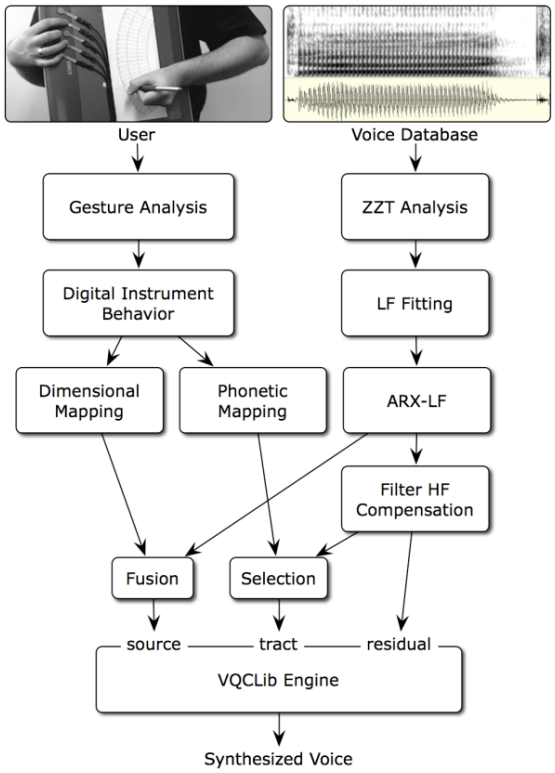

Figure 17: *An overview of RAMCESS 2.x framework.*

7. CONCLUSION

In this paper we described main improvements that have been achieved in order to transform a vowel-only system into a framework which is able to produce coarticulated speech expressively. This work illustrated that the deep analysis of glottal source features and the separation with vocal tract components were achievable on a prepared database. We were also able to validate the use of VQCLIB, and more generally RAMCESS elements in a complex voice production context.

On the top of this current work, two significant axis will be developed further: the achievement of a first (limited but functional) diphone synthesizer, and the extension of these concepts to musical signal synthesis, as recent work have shown that it was clearly possible [22].

8. ACKNOWLEDGMENTS

Authors would first like to thank the organization committee of eNTERFACE'07 summer workshop in Istanbul (Boğaziçi University) for their great involving and support. We also would like to highlight a role of the European Union (through FP6 and SIMILAR) in the achievement of all these tools. Finally we would like to thank our respective laboratories and supervisors for their advices, trust and support.

9. REFERENCES

[1] J. Bonada and X. Serra, "Synthesis of the Singing Voice by Performance Sampling and Spectral Models", *IEEE Signal Processing*, vol. 24, no. 2, pp. 67–79, 2007. 129

[2] B. Bozkurt and T. Dutoit, "Mixed-Phase Speech Modeling and Formant Estimation, using Differential Phase Spectrums", in *Proc. of ISCA ITRW VOQUAL*, pp. 21–24, 2003. 129

[3] B. Bozkurt, *New Spectral Methods for the Analysis of Source/Filter Characteristics of Speech Signals*. PhD thesis, Faculté Polytechnique de Mons, 2005. 129, 132

[4] "VQClib". http://vqclib.blogsite.org. 129, 130

[5] G. Fant, J. Liljencrants, and Q. Lin, "A Four-Parameter Model of Glottal Flow", *STL-QPSR*, vol. 4, pp. 1–13, 1985. 130

[6] B. Doval and C. d'Alessandro, "The Voice Source as a Causal/Anticausal Linear Filter", in *Proc. of Voqual'03, Voice Quality: Functions, Analysis and Synthesis, ISCA Workshop*, 2003. 130

[7] N. d'Alessandro, B. Doval, S. L. Beux, P. Woodruff, Y. Fabre, C. d'Alessandro, and T. Dutoit, "Realtime and Accurate Musical Control of Expression in Singing Synthesis", *Journal on Multimodal User Interfaces*, vol. 1, no. 1, pp. 31–39, 2007. 130, 131

[8] B. Doval, C. d'Alessandro, and N. Henrich, "The Spectrum of Glottal Flow Models", *Acta Acustica*, vol. 92, pp. 1026–1046, 2006. 130

[9] A. Edelman and H. Murakami, "Polynomial Roots from Companion Matrix Eigenvalues", *Mathematics of Computation*, vol. 64, no. 210, pp. 763–776, 1995. 132

[10] B. Bozkurt, B. Doval, C. d'Alessandro, and T. Dutoit, "Zeros of the Z-Transform Representation with Application to Source-Filter Separation in Speech", *IEEE Signal Processing Letters*, vol. 12, no. 4, pp. 344–347, 2005. 132

[11] B. Bozkurt, L. Couvreur, and T. Dutoit, "Chirp Group Delay Analysis of Speech Signals", *Speech Communication*, vol. 49, no. 3, pp. 159–176, 2007. 132

[12] T. Dubuisson and T. Dutoit, "Improvement of Source-Tract Decomposition of Speech using Analogy with LF Model for Glottal Source and Tube Model for Vocal Tract", in *(to appear in) Proc. of Models and Analysis of Vocal Emissions for Biomedical Application Workshop*, 2007. 132

[13] G. Fant, *Acoustic Theory of Speech Production*. Mouton and Co. Netherlands, 1960. 133, 134

[14] D. Vincent, O. Rosec, and T. Chonavel, "Estimation of LF Glottal Source Parameters Based on ARX Model", in *Proc. of Interspeech*, (Lisbonne), pp. 333–336, 2005. 133

[15] D. Vincent, O. Rosec, and T. Chonavel, "A New Method for Speech Synthesis and Transformation Based on an ARX-LF Source-Filter Decomposition and HNM Modeling", in *Proc. of ICASSP*, (Honolulu), pp. 525–528, 2007. 133

[16] N. d'Alessandro and T. Dutoit, "RAMCESS/HandSketch: A Multi-Representation Framework for Realtime and Expressive Singing Synthesis", in *(to appear in) Proc. of Interspeech*, (Anvers), 2007. 134

[17] D. Schwarz and M. Wright, "Extensions and Applications of the SDIF Sound Description Interchange Format", in *Intl. Computer Music Conf.*, 2000. 135

[18] "SDIF for Max/MSP". http://www.cnmat.berkeley.edu/MAX/downloads/. 135

[19] N. Schnell, R. Borghesi, D. Schwarz, F. Bevilacqua, and R. Muller, "FTM - Complex Data Structures for Max", in *Proc. of Intl. Computer Music Conf.*, 2007. 135

[20] "OpenSoundControl". http://opensoundcontrol.org. 135

[21] N. d'Alessandro and T. Dutoit, "HandSketch Bi-Manual Controller", in *Proc. of NIME*, pp. 78–81, 2007. 135

[22] N. d'Alessandro, T. Dubuisson, A. Moinet, and T. Dutoit, "Causal/Anticausal Decomposition for Mixed-Phase Description of Brass and Bowed String Sounds", in *(to appear in) Proc. of Intl. Computer Music Conf.*, 2007. 136

10. BIOGRAPHIES

Nicolas d'Alessandro holds an Electrical Engineering degree from the Faculté Polytechnique de Mons (FPMs) since 2004. He did the master's thesis in the Faculty of Music of the University de Montréal (UdeM) (supervisor: Prof. Caroline Traube). That work gathered the development of applications based on perceptual analogies between guitar sounds and voice sounds, and a study of mapping strategies between (hand-based) gestures and speech production models (source/filters and concatenative approaches). He started a PhD thesis in September 2004 in the Signal Processing Lab of the Faculté Polytechnique de Mons (supervisor: Prof. Thierry Dutoit) related to the real-time control of unit-based synthesizers.
Email: nicolas.dalessandro@fpms.ac.be

Onur Babacan was born in Izmir, Turkey in 1986. He is currently a senior undergraduate student at Izmir Institute of Technology (IYTE, Izmir, Turkey), studying Electronics and Telecommunications Engineering.
Email: onurbabacan@gmail.com

Barış Bozkurt is currently employed as an Assistant Professor in Izmir Institute of Technology (IYTE, Izmir, Turkey) where he teaches electronics and digital audio/speech/signal processing and continues his research in the same fields. He obtained his Electrical Engineering degree in 1997 and Master of Science degree in Biomedical Engineering in 2000 both from Boğaziçi University, Istanbul, Turkey. After obtaining his PhD degree (Electrical Engineering (speech processing)) in 2005 from Faculté Polytechnique De Mons, Mons, Belgium, he worked as a research engineer in Svox AG, Zurich. He is an associate editor of Journal of Interdisciplinary Music Studies.
Email: barisbozkurt@iyte.edu.tr

Andre Holzapfel received the graduate engineer degree in media technology from the University of Applied Sciences in Duesseldorf, Germany, and the M.Sc. degree in computer science from University of Crete, Greece, where is currently pursuing the Ph.D. degree. His research interests are in the field of speech processing, music information retrieval and ethnomusicology.
Email: hannover@csd.uoc.gr

Thomas Dubuisson was born in Tournai, Belgium, in 1983. He received the Electrical Engineering Degree from the Faculty of Engineering, Mons (FPMs) in 2006. He is currently PhD Student in the Circuit Theory and Signal Processing (TCTS) Lab, FPMs, where he is working on assessment of speech pathologies and source-tract separation of speech, in the framework of the WALEO II Research Project ECLIPSE (Walloon Region, Belgium).
Email: thomas.dubuisson@fpms.ac.be

Loïc Kessous was born in Marseille, France. He received the Master of Science degree in Physics from the Aix-Marseille University and obtained is PhD in 2004 from Paris IX University, and was affiliated at this time to the Mechanics and Acoustics Laboratory at CNRS in Marseille (LMA-CNRS). He realized during his PhD several digital instrument including an gesture controlled singing voice synthesizer. During the same period he was also a research and development associate at GMEM, (National Center for Music Creation) in Marseille. From 2004 to 2007 he was working in the framework of the european HUMAINE (Human-Machine Interaction Network on Emotion) project at Tel Aviv University, Israel. He is currently working at The Computer Sciences Laboratory for Mechanics and Engineering Sciences (LIMSI), CNRS laboratory, in Paris, France.
Email: loic.kessous@limsi.fr

Alexis Moinet holds an Electrical Engineering degree from the FPMs (2005). He did his master thesis at the T.J. Watson research Center of IBM (2005). He is currently working on the IRMA project in the Signal Processing Lab of the FPMs. He is particularly interested in source/filter decomposition of speech and music, phase vocoder, voice activity detection, voice conversion and HMM synthesis.
Email: alexis.moinet@fpms.ac.be

Maxime Vlieghe was born in 1984 in Belgium. When he was 7, he learned academic music for 5 years and played piano for 3 years at the "conservatoire Royal de Tournai". After finishing his college classes in sciences, he wanted to go on in this direction and decided to start studies at the "Faculté Polytechnique de Mons". He studied Electric Engineering for 5 years. In his last year, joining the fields of music and sciences, he wrote his master thesis at the Université de Montréal with Pr. Caroline Traube about the synthesis of singing voice and particularly rules synthesis of plosive consonants. As he was writing this work, he was invited by Pr. Thierry Dutoit to be member of a team of eNTERFACE'07 on a singing voice synthesis project.
Email: maxime.vlieghe@gmail.com