MAXMBROLA : A MAX/MSP MBROLA-BASED TOOL FOR REAL-TIME VOICE SYNTHESIS

Nicolas D'Alessandro, Raphaël Sebbe, Baris Bozkurt, Thierry Dutoit

Circuit Theory and Signal Processing Laboratory (TCTS Lab), Faculté Polytechnique de Mons (FPMs)
Parc Initialis, 1, Copernic Avenue, B-7000 Mons, Belgium, http://tcts.fpms.ac.be

ABSTRACT

In this paper, we present the first step of a project that is able to perform both speech and singing synthesis controlled in real-time. Our aim is to develop a flexible application allowing performers to produce complex and versatile singing as well as speech. Thus, we have adapted an existing speech synthesizer, the MBROLA software, to real-time singing constraints. The work presented in this paper concerns the integration of the MBROLA speech synthesizer into the Max/MSP real-time environment through the development of an external object. We present real-time control functions of the object and discuss its limitations. We further discuss about perspectives in the development of MBROLA-based computer music applications.

1. INTRODUCTION

Speech and singing both result from the same production system : the voice organ. However, the signal processing techniques developed for their synthesis evolved quite differently. One of the main reasons for this deviation is : the aim for producing voice is different for the two cases. The aim of speech production is to exchange messages. For singing, the main aim is to use the voice organ as a musical instrument. Therefore a singing synthesis system needs to include various tools to control (analyze/synthesize or modify) different dynamics of the acoustic sound produced : duration of the phonemes, vibrato, wide range modifications of the voice quality, the pitch and the intensity, etc. some of which are not needed in most of the speech synthesis systems. A pragmatic reason for that separation is that singing voice synthesizers target almost exclusively musical performances. In this case, "playability" (flexibility and real-time abilities) is much more important than intelligibility and naturalness. Discussions about various issues of singing synthesis can be found in [1] [2].

As described in [3], frequency-domain analysis/modifications methods are frequently preferred in singing synthesis research due to the need to modify some spectral characteristics of actual recorded signals. The most popular application of such a technique is the phase vocoder [4], which is a powerful tool used for many years for time compression/expansion, pitch shifting and cross-synthesis.

To increase flexibility, short-time signal frames can be modeled as sums of sinusoids (controlled in frequency, amplitude and phase) plus noise (controlled by the parameters of a filter which is excited by a white noise). HNM (Harmonic plus Noise Model) [5] provides a flexible representation of the signal, which is particularly interesting in the context of unit concatenation. That representation of signals is thus used as a basis in many singing synthesis systems [6] [7] [8] [9].

Another approach is to use the source/filter model. The LF (Liljencrants and Fant) model of the glottal pulse (or derivative glottal pulse) [10] is often used as the source signal for voice production because of its high control abilities of the voice quality. Some differences appear in the method used to compute the vocal tract transfer function. Some systems [11] compute the formants from the magnitude spectrum : a series of resonant filters (controlled by formants frequencies, amplitudes and bandwidths). Some other systems compute an acoustic representation of the vocal tract, as a cascade of acoustic (variant-shape) tubes. For example, the SPASM synthesizer [12] uses digital waveguides [13] to model acoustic features of oral, nasal cavities and throat radiation (driven by a frequency-domain excitation model). The model was extended to variable length conical sections by Välimäki and Karjalainen [14].

There exist also some particular approaches like FOF (*Formes d'Ondes Formantiques*) synthesis [15], used in CHANT [16], which performs synthesis by convolving a pulse train with parallel *formant wave functions* (time-domain functions corresponding to individual formants resonance).

Our aim is to develop a flexible real-time application based on the MBROLA speech synthesizer [17] allowing performers to produce complex and versatile singing – as well as speech – in many languages. Thus, we start from a speech synthesizer and work on the adaptation of that system to real-time singing constraints. We use that particular approach for its high quality synthesis abilities. In this paper, we present the first step of that project which is the integration of the MBROLA software into the Max/MSP (4.5) real-time environment. We first describe MBROLA (section 2) and Max/MSP (section 3) softwares. Then, we explain the development of the MaxMBROLA external object, which we have made publicly available. Finally, we discuss about real-time issues of that object and give some perspectives for further work.

2. THE MBROLA SYNTHESIZER

MBROLA (Multi-Band Re-synthesis OverLap Add) [17] is a widely used synthesizer for diphone-based speech synthesis. A diphone is a speech segment, which starts in the middle of the stable part (if any) of a phoneme, and ends in the middle of the stable part of the next phoneme. When the diphone is used as the basic unit, the concatenation points are at stable parts of the phonemes. This facilitates some smoothing operation to be performed at synthesis time, which reduces possible discontinuities at concatenation points.

To create such a synthesizer, first the diphones of a given language are recorded in carrier words and segmented. The MBROLA system performs an off-line pre-processing of the

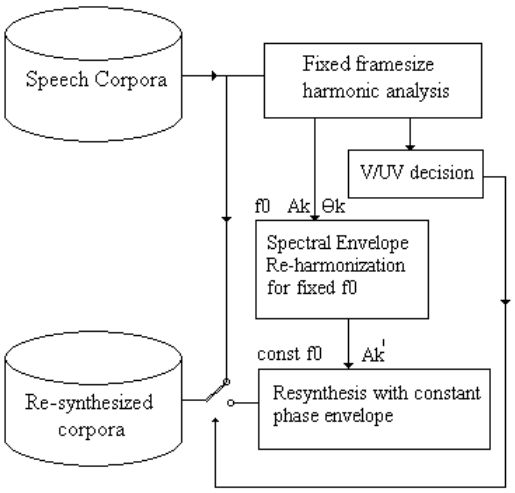

FIG. 1 – MBROLA database processing

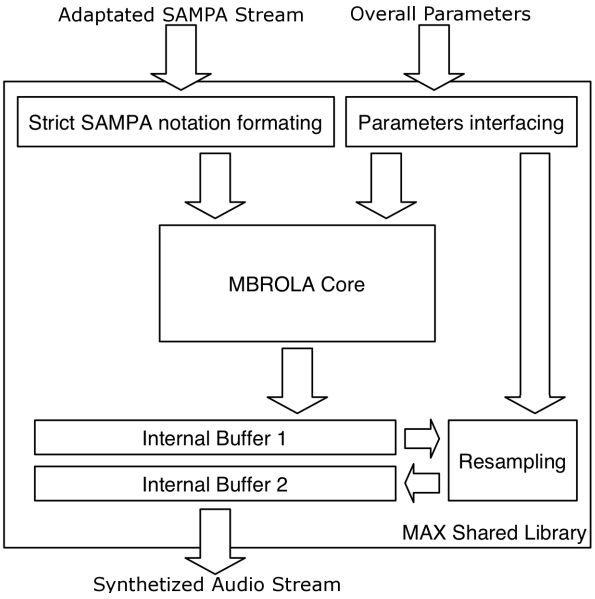

FIG. 2 – Description of the encapsulation of the MBROLA core in the real-time external object : floating point values and symbols formating for inputs, samples routing and resampling for output.

diphone database to combine the computational efficiency of time-domain synthesis with the flexibility of a harmonic model (see Fig. 1) : speech units are first submitted to harmonic analysis with constant frame length and frame shift. Then voiced frames are re-synthesized with constant pitch (at the average pitch of the database) and constant phase envelope (for the low frequency part of the speech spectrum) with a harmonic synthesizer. This operation includes computation of new harmonic amplitudes by resampling the spectral envelope obtained from the actual harmonic amplitudes and resetting low frequency harmonic phases to constant phase values copied from a steady-state speech frame. The re-synthesis procedure is applied to voiced frames only and unvoiced frames are directly copied. This operation facilitates frame synchronization in the overlap add (OLA) part of the algorithm and reduces pitch mismatches. Another important advantage of the phase reset is that the spectral envelope interpolation can be performed by direct temporal interpolation instead of performing interpolation in frequency domain which is an important aspect regarding real-time synthesis. During synthesis, the smoothing operation is applied to stationary voiced frames by distributing the difference of boundary frames linearly to the neighbor stationary voiced frames on the left and right units.

At run time, once the synthesizer receives some phonetic input (phonemes, phonemes durations, intonation curves) from the natural language processing (NLP) module, it sets up a list of required diphones, together with their required duration and fundamental frequency contour. Speech is synthesized from the re-synthesized speech segments using the OLA operation to impose target prosody.

The MBROLA synthetic speech quality is often graded as highly intelligible but to some level computer-like since a diphone database hardly represents all the variability of natural human speech. Still, the naturalness of the synthesized speech is higher than that of the rule-based synthesizers. Thanks to ten years of collaboration with various research laboratories, the MBROLA system is now capable of producing speech in 34 languages (and 67 voices).

3. THE MAX/MSP ENVIRONMENT

The Max graphical development environment [18] and its MSP audio processing library [19] are well known by the computer music community. This software is a powerful tool in many fields of music like real-time sound processing, control mapping, composition and performance abilities enhancement, etc. It is a rare example of an intuitive interface (design of personalized modules by the building of graphs of simple functions, called *objects*) and a high level of flexibility (functions accepting and modifying numbers, symbols, audio and video stream, etc) at the same time. The skills of that software increase every day with the help of an active developers community providing new *external* objects (or *externals*).

4. THE MAXMBROLA EXTERNAL OBJECT

The first step in the design of our real-time voice production tool is to bring the MBROLA technology into the workspace of Max. This task is accomplished by the development of an external object (called MaxMBROLA~) encapsulating overall features of the speech synthesis system (Figure 2).

Max objects work as small servers. They are initialized when they are imported inside the workspace. They contain a set of dedicated functions (methods) which are activated when the object receive particular messages. These messages can be simple numbers, symbols or complex messages with a header and arguments. Considering that real-time request-based protocol of communication between objects, the second thing we have to do is to define a particular set of messages (header and arguments). As shown in figure 2, we can separate the possible requests in two main channels. On one side, there is parameter modification, which influence the internal state of the synthesizer. On the other side, there is the phonetic/prosodic stream, which generate speech instan-

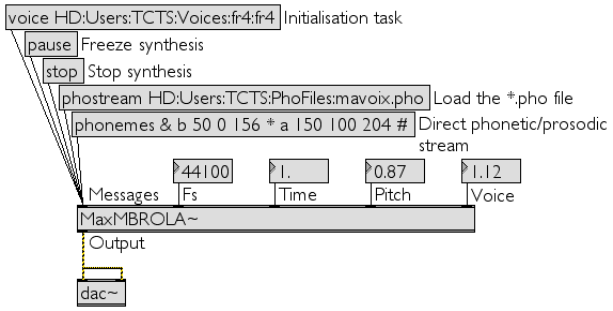

FIG. 3 – Conventions used in the naming of inlets and supported messages : *Messages, Fs, Time, Pitch, Voice*.

taneously.

4.1 Internal state modifications

Some particular modifications of the internal state of the MBROLA synthesizer can be applied with Max/MSP requests. Here is a description of the supported actions. The labels used to name inlets (from left to right : *Messages*, *Fs*, *Time*, *Pitch* and *Voice*) and examples of the supported messages are illustrated on Figure 3.

- The use of the synthesizer always starts with the initialization task (*Messages* inlet). That function starts the MBROLA engine, loads the requested database of diphones and set all the internal parameters to their default values. All the existing MBROLA databases are compatible with the external.
- The stream provided by the external can be frozen (*Messages* inlet). It means that the phonetic/prosodic content stays in memory but the MBROLA engine stops the synthesis task.
- The MBROLA engine can be stopped (*Messages* inlet). That function flushes the phonetic/prosodic content, stops the synthesis process and sets all the internal parameters to their default values. The database of diphones stays loaded.
- *Fs* inlet receives a floating point number. It controls the output sampling rate. Indeed, the original sampling rate depends on the database (16000Hz or 22050Hz). Linear interpolation is performed allowing the use of that external object with all possible sampling rates.
- The inlets *Time*, *Pitch* and *Voice* each receive a floating point number. These values are respectively the time ratio (deviation of the reference speed of speech), the pitch ratio (deviation of the reference fundamental frequency of speech) and voice ratio (compression/dilation ratio of the spectrum width).

4.2 Phonetic/prosodic stream processing

Here are the requests that generate speech in the Max environment. All following messages are sent into the *Messages* inlet.

A loading request allows to use a standard `*.pho` file (which include the list of phonemes to be produced and the target prosody) to perform synthesis. Examples are available together with MBROLA voices and complete explanations about standard SAMPA[1] notation is given in [20].

We developed a function that directly accepts SAMPA streams inside Max messages to provide user control to interactive speech production. The standard SAMPA notation has been modified to fit to the Max message structure. For example, the following stream :

```
phonemes & b 50 0 156 * a 150 100 204 #
```

begins by initializing the synthesizer, then produces a syllable /*ba*/ of 200 (50 + 150) milliseconds with a fundamental frequency increasing from 156Hz to 204Hz (two pitch points). Finally, it flushes the phoneme buffer. More details about the syntax can be found in the MaxMBROLA tutorial [22].

5. PROBLEMS AND PERSPECTIVES FOR REAL-TIME SINGING SYNTHESIS

Indeed the MaxMBROLA tool answers a need in real-time speech generation by gathering a synthesizer with complex articulation abilities and a versatile interface for live performances (see [21] for an example). As shown above (section 1), singing production involves much more constraints in the synthesis process than in speech synthesis. In this part, we present some limitations of MaxMBROLA in singing synthesis, which define our further challenges in the development of this application.

- Infinite phoneme duration :
  As we explained in section 4, duration of phoneme has to be specified in the request, which will be sent to the object. Thus, MaxMBROLA can't hold long vowels, as we can expect from a singing instrument.
- Unavaibility of "sub-phonemic" pitch control :
  For the same reason as above, it is impossible to modify *a posteriori* the pitch curves of phonemes sent to the object. Thus, phoneme duration is the smallest subdivision of time for pitch control.
- Context-dependency of speech synthesis :
  As a system initially created for speech synthesis – and not singing – MBROLA needs a phonetic context (2 phonemes around the target phoneme) to correctly compute the pitch curve and concatenate units. Thus, the synthesized signal will always be delayed by two phonemes compared with its corresponding request.

Thus, the development of an MBROLA-based singing synthesis system will be driven by the correct balance between two main issues :

- Generation of versatile short-term requests to drive MaxMBROLA with sufficient real-time properties ;
- Creation of post-processing algorithms with limited signal deterioration.

We also developed a musical application based on the MaxMBROLA external object, called *MIDI-MBROLA*. That tool has a full MIDI compatible interface. MIDI *control changes* are used to modify the internal parameters of the MBROLA synthesizer. *Events* from a MIDI keyboard are used to compute the prosody, which is mixed to the phonetic content at the time of performance. As a standard module of the Max/MSP environment, the MIDI-MBROLA digital instrument automatically allows polyphony. Indeed, many voices can readily be synthesized simultaneously because the

[1] SAMPA : Speech Assessment Methods Phonetic Alphabet. It is the machine-readable phonetic alphabet used in many speech synthesizers.

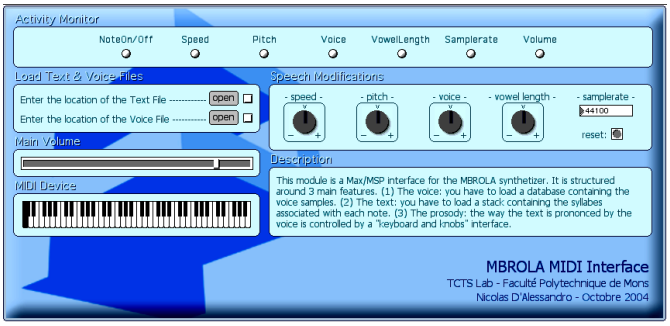

FIG. 4 – Front-end of the MIDI-MBROLA digital instrument allowing to play voice with a MIDI keyboard.

MBROLA synthesis doesn't utilize many CPU resources. It can also be compiled as a standalone application or a VST ("Virtual Studio Technology", digital effect standard developed by Steinberg) instrument. That tool is publicly available [22]. The MIDI-MBROLA front-end is illustrated on Figure 4.

6. CONCLUSIONS

In this paper, we described the integration of the MBROLA text-to-speech synthesizer in a real-time processing platform and discussed further works about its adaptation to the particular constraints of real-time singing synthesis. The first results of the external object are really encouraging. It is being used in live performances and it can synthesize all MBROLA voices almost as well as other standalone applications. Finally, that development step provided a strong tool that will certainly be used as a basis in many further projects that need high articulation abilities in real-time.

REFERENCES

[1] X. Rodet and G. Bennet, "Synthesis of the Singing Voice," *Current Directories in Computer Music Research*, ed. M. V. Mathews and J. R. Pierce, MIT Press, 1989.

[2] X. Rodet, "Synthesis and Processing of the Singing Voice," *Proceeding of the First IEEE Benelux Workshop on Model-Based Processing and Coding of Audio (MPCA-2002)*, Leuven, Belgium, 2002.

[3] P. R. Cook, *Identification of Control Parameters in an Articulatory Vocal Tract Model, with Applications to the Synthesis of Singing*. PhD Thesis, Standford University, 1990.

[4] J. A. Moorer, "The Use of the Phase Vocoder in Computer Music Application," *Journal of the Audio Engineering Society*, vol. 26, no. 1/2, pp. 42-45, 1978.

[5] J. Laroche, Y. Stylianou and E. Moulines, "HNS : Speech Modifications Based on a Harmonic plus Noise Model," *Proceedings of the International Conference on Acoustics, Speech and Signal Processing*, pp. II-550-553, 1993.

[6] M. Macon, L. Jensen-Link, J. Oliviero, M. A. Clements and E. B. George, "A Singing Voice Synthesis System Based on Sinusoidal Modeling," *Proceedings of the International Conference on Acoustics, Speech and Signal Processing*, vol. 1, pp. 435-438, 1997.

[7] K. Lomax, *The Analysis and the Synthesis of the Singing Voice*. PhD Thesis, Oxford University, 1997.

[8] Y. Meron, *High Quality Singing Synthesis Using the Selection-Based Synthesis Scheme*. PhD Thesis, University of Michigan, 2001.

[9] P. Cano, A. Loscos, J. Bonada, M. de Boer and X. Serra, "Voice Morphing System for Impersonating in Karaoke Applications," *Proceedings of the International Computer Msuic Conference*, 2000.

[10] G. Fant, J. Liljencrants and Q. Lin, "A Four-Parameter Model of Glottal Flow," *Speech Transmission Laboratory Quarterly Progress and Statut Report (STL-QPSR)*, 4 :1-13, 1985.

[11] B. Larson, "Music and Singing Synthesis Equipment (MUSSE)," *Speech Transmission Laboratory Quarterly Progress and Statut Report (STL-QPSR)*, (1/1977) :38-40, 1977.

[12] P. R. Cook, "SPASM : a Real-Time Vocal Tract Physical Model Editor/Controller and Singer : the Companion Software System," *Colloque sur les Modèles Physiques dans l'Analyse, la Production et la Création Sonore*, ACROE, Grenoble, 1990.

[13] J. O. Smith, "Waveguide Filter Tutorial," *Proceedings of the International Computer Music Conference*, San Fransisco, USA, pp. 9-16, 1987.

[14] V. Välimäki and M. Karjalainen, "Improving the Kelly-Lochbaum Vocal Tract Model Using Conical Tubes Sections and Fractionnal Delay Filtering Techniques," *Proceedings of the International Conference on Spoken Language Processing*, Yokohama, Japan, 1994.

[15] X. Rodet, "Time-Domain Formant Wave Function Synthesis," *Computer Music Journal*, vol. 8, no. 3, pp. 9-14, 1984.

[16] X. Rodet and J. B. Barriere, "The CHANT Project : From the Synthesis of the Singing Voice to Synthesis in General," *Computer Music Journal*, vol. 8, no. 3, pp. 15-31, 1984.

[17] T. Dutoit and H. Leich, "MBR-PSOLA : Text-to-Speech Synthesis Based on an MBE Resynthesis of the Segments Database," *Speech Communication*, no 13, pp. 435-440, 1993.

[18] D. Zicarelli, G. Taylor, J. K. Clayton, jhno, and R. Dudas, *Max 4.3 Reference Manual*. Cycling'74/Ircam, 1990-2004.

[19] D. Zicarelli, G. Taylor, J. K. Clayton, jhno, and R. Dudas, *MSP 4.3 Reference Manual*. Cycling'74/Ircam, 1997-2004.

[20] `http ://www.phon.ucl.ac.uk/home/sampa/home.htm`

[21] `http ://www.artzoyd.com/fr/rub_cs/cs.htm`

[22] `http ://tcts.fpms.ac.be/synthesis/maxmbrola/`