MAGE 2.0: New Features and its Application in the Development of a Talking Guitar

Maria Astrinaki
Institute for New Media Art Technology, University of Mons, B-7000, Belgium
maria.astrinaki@umons.ac.be

Nicolas d'Alessandro
Institute for New Media Art Technology, University of Mons, B-7000, Belgium
nda@numediart.org

Loïc Reboursière
Institute for New Media Art Technology, University of Mons, B-7000, Belgium
loic.reboursiere@umons.ac.be

Alexis Moinet
Institute for New Media Art Technology, University of Mons, B-7000, Belgium
alexis.moinet@umons.ac.be

Thierry Dutoit
Institute for New Media Art Technology, University of Mons, B-7000, Belgium
thierry.dutoit@umons.ac.be

ABSTRACT

This paper describes the recent progress in our approach to generate performative and controllable speech. The goal of the performative HMM-based speech and singing synthesis library, called MAGE, is to have the ability to generate natural sounding speech with arbitrary speaker's voice characteristics, speaking styles and expressions and at the same time to have accurate reactive user control over all the available production levels. MAGE allows to arbitrarily change between voices, control speaking style or vocal identity, manipulate voice characteristics or alter the targeted context on-the-fly and also maintain the naturalness and intelligibility of the output. To achieve these controls, it was essential to redesign and improve the initial library. This paper focuses on the improvements of the architectural design, the additional user controls and provides an overview of a prototype, where a guitar is used to reactively control the generation of a synthetic voice in various levels.

Keywords

speech synthesis, augmented guitar, hexaphonic guitar

1. INTRODUCTION

For the last few years, the numediart institute has specialized in creating speech and singing synthesizers that are highly intelligible, natural-sounding and performative. The intelligibility and naturalness of computer-generated voices have always been main concerns in the speech community. Nowadays there is no doubt among experts that non-uniform unit selection [7] is the best category of algorithm that can be used for producing human-like speech. However these systems largely fail at proposing a more interactive experience (conversational or performative) where users or performers can freely interact with the virtual speaker or singer. We see our purpose of building speaking/singing NIMEs [1] as a design-oriented strategy to find new solutions to this 15-year-old issue of interactivity and expressivity in computer-generated voices.

NIME'13, May 27 – 30, 2013, KAIST, Daejeon, Korea.

Our approach could be summarized as that one sentence: "back to modeling". Indeed, the well-known non-uniform unit selection approach concatenates segments of speech waveform to create original content. This technique prevents from subtly changing vocal features and the time scale that leads to good results typically reaches a few seconds, preventing any kind of real-time interaction. An emergent approach called HMM-based statistical parametric modeling [20] tries to model the knowledge contained in the speech data and uses this knowledge to generate original content. For the last few years, our work has been to make this statistically-driven generation process absolutely real-time (i.e. generative statistics happen at runtime) and reactive (i.e. most of the modeling stages can be adjusted through on-the-fly user inputs).

Along the development of our performative HMM-based speech synthesis software library, called MAGE, the main target and the expectations have kept shifting for the better. Indeed, the first iteration aimed at "unlocking" the statistically-driven generation process from its default offline behaviour, as encountered in Text-to-Speech [2]. At that stage, showing some basic user inputs reaching the modeling process with a limited delay was an achievement. With this version 2.0, we reach a zero-phoneme delay architecture, enabling MAGE to be used in real performative cases. In this paper, we first give a bit more context on statistical parametric synthesis in Section 2. Then we describe the MAGE 2.0 architecture in Section 3. Section 4 briefly explains how to integrate MAGE into an application. Finally we describe how we "got a guitar to talk", i.e. we controlled MAGE with a guitar, in Section 5.

2. RELATED WORK

For a long time, the idea of producing artificial speech and singing has been imbricated with creating new instruments for controlling the vocal sounds. One of the most famous examples of artificial speech or singing production is the von Kempelen machine [17], dated in the 18^{th} century. More recent examples are GRASSP [13] and DiVA [5], where the parallel formant synthesis is combined with various neural network designs, proposing an adaptive instrument mapping for speech synthesis, mainly by means of cyberglove gestures. This approach provides continuous and very flexible controls, but its output quality is rather limited and sometimes not intelligible. Another example is RAMCESS [3] and HandSketch [4]. RAMCESS is a hybrid approach where only one part of the speech frame is replaced by a

generative model of the glottal source that can be controlled in realtime. This approach, where singing multi-phones get spectrally interpolated, results in intelligible and good quality output but also in discrete control. Lastly, there is SPASM [12], which is a musical interface to a physical model of the human vocal tract, using a waveguide-based articulatory model. SPASM provides meaningful mappings and controls, but it is not highly intelligible.

In the speech synthesis domain, the statistical parametric speech synthesis approach has gained in popularity in the past decade. Hidden Markov Models (HMMs) have been widely used in this speech synthesis approach. The HTS system [20], is a system for statistical parametric speech synthesis using HMMs as the underlying statistical model. Note here that in HMM-based speech synthesis no real speech samples are used at synthesis runtime. Instead, it uses the knowledge contained in pre-recorded speech databases. The database is analyzed and various production features are extracted (i.e. the magnitude spectrum, the fundamental frequency and the duration of the phonemes). Then the extracted features are used to train a statistical model, and the targeted speech waveforms are generated from the models themselves based on the maximum likelihood criterion. This results in a system with small synthesizer footprint. It also gives more flexible control of synthesis methods, combined with good speech quality and possibilities for portable applications. On the other hand, the fact that there are no real speech samples to be used during the synthesis time introduces small degradation in the final speech quality compared to the unit selection approach [7]. Since such a system is statistically well defined, it is possible to successfully apply either speaker adaptation [11] [16], speaker interpolation [19], or eigenvoice technique [15] in order to modify various voice characteristics. Moreover, speaking styles and emotional voices can be constructed by re-estimating existing average voice models with only a few utterances using adaptation techniques [18]. Another controllable feature brought in HTS is the articulatory modeling and control [9] as well as the vowel creation [10].

However, such a system is not designed to be reactive or have any user control during the synthesis, since the full linguistic specifications of a whole utterance is required in order to generate the speech parameters from the context-dependent phonemes. Therefore, the time scale on which the user can influence the sound can not be smaller than a whole sentence. At this point it is essential to use a system that will convert the conventional parameter generation framework to a more flexible one that allows reactivity. Such a system is MAGE [1], a recently proposed platform suitable for reactive HMM-based speech and singing synthesis.

3. MAGE 2.0: NEW FEATURES

MAGE 1.00, as described in [1], is the proof of concept that the original system, HTS [20] can be thread safe, engine independent and it can allow users to have control over several speech and singing production levels. It provided the required framework first to study the contextual tailoring of the HMMs that are taken into account for the production features and second to enable the reactive manipulation of the waveform output. MAGE 2.0[1] is rather a system-wise reorganisation of how the computation of the production features is achieved and how the user controls are applied. All the synthesis part is redesigned as a multithread software that addresses each problem within its own processing

[1]tutorial video : https://vimeo.com/57359293

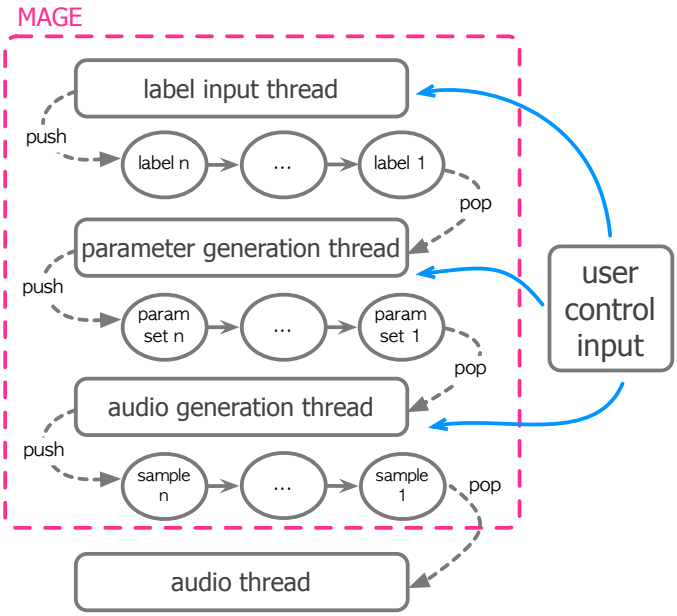

Figure 1: Multithread architecture of MAGE: speech synthesis thread makes the connection between the user control and the audio thread.

queue, making the overall execution of the existing statistical parametric algorithm as reactive as it can be[2].

The multithreaded architecture of MAGE 2.0 is illustrated in Figure 1. As shown, MAGE consists of three main threads: the *label* thread, the *parameter generation* thread and finally the *audio generation* thread. First, the *label* thread is responsible for the contextual control of the system. Indeed, the context of the targeted output can be easily manipulated in realtime by simply controlling the order in which the phonemes available for processing will be inputed into the system.

Then, the *parameter generation* thread generates the corresponding spectral and excitation parameters for a single phoneme at a time. This is achieved by locally maximising the probability of the inputed speech parameter sequence. However, compared to the first implementation of MAGE where the use of a sliding window of minimum two phonemes was essential for the correct estimation of the required parameters, here every phoneme is processed individually. There is zero phoneme delay, there are no future dependences and therefore the system becomes even more reactive.

Finally the *audio generation* thread will generate speech samples corresponding to the inputed phoneme. It is important to note here that in contrast to MAGE 1.00, where the access and the controls were applied over buffers of samples, now the user can access and manipulate every speech sample separately. Consequently, the user control is much more reactive and accurate.

Due to the multithread architecture of MAGE 2.0 it is possible to allow user interaction on every production level. Indeed, it is possible to have contextual control by controlling the phonetic input, model control by controlling the parameter generation, for example reactive control of the articulation degree [2] and finally vocoder parameter control at the sample generation level.

[2]The new MAGE baseline is "Change speech one sample at a time..."

4. INTEGRATION: C++, OF, MAX, PD

MAGE is a C++ library and can be integrated within an application in various ways detailed in the following Sections.

4.1 C++ API

MAGE comes with a comprehensive API declared in the file `mage.h`. An external C++ application can access it directly by including this file and calling its functions appropriately. Note that every function and class of MAGE belongs to the namespace `MAGE`.

```
#include "mage.h"
using namespace MAGE;
...
Mage *mage = new Mage();
mage->addEngine( "slt", "slt.conf" );
...
mage->setAlpha(0.42);
...
float * buffer = new float[bsize];
for( int k = 0; k < bsize; k++ )
  buffer[k] = mage->popSamples();
...
delete[] buffer;
delete mage;
```

The openFrameworks application, described in Section 4.2, and the PureData and Max/MSP externals, described in Section 4.3, are actually examples of C++ programs embedding and using MAGE directly through its API.

4.2 openFrameworks

A simpler approach to MAGE integration within a program is to create an openFrameworks application based on one of the several examples distributed with the source code. These small control examples illustrate the various ways through which MAGE parameters can be modified on-the-fly while the speech is synthesized. Besides a more complete example is available as `ofOscControl` which, when started, launches the MAGE threads and permits to control each parameter of the speech synthesis through OSC messages sent from the Pure Data or the Max/MSP patch examples.

4.3 Pure Data and Max/MSP

A third possibility is to use the `mage~` object within a Pure Data or Max/MSP patch. `mage~` starts the MAGE threads at initialization time and can then be controlled through different messages such as:

```
alpha 0.42
pitchscale 2
speedoverwrite 0.75
```

which respectively mean: "change α value to 0.42", "multiply pitch by two" and "reduce speed to 75%". Some other messages make it possible to manipulate the sequence of labels in real-time:

```
label alice.lab
labelnext
labelinsert 0
labelreplace 42
labelswitch 17
```

where the first message loads a list of label from file alice.lab and `labelnext` sends a label from that list to MAGE and goes to the next label of the list. `labelinsert N` sends the N^{th} label of the list to MAGE, `labelreplace N` sends the N^{th} label of the list to MAGE and makes `labelnext` jump to its next label. `labelswitch N` sends the N^{th} label of the list and makes `labelnext` point to label $N + 1$. If a command reaches the end of the list, it loops back to the start.

5. USE CASE: TALKING GUITAR

In [14], we developed algorithms for most of the guitar playing techniques, i.e, hammer-on, pull-off, slide, palm muting, bend, natural harmonic notes as well as detection of the plucking point. Pitch of the played note is also detected as well as its start and end. Different types of musical gesture could have been used to make the guitar talk, e.g [8] and [6], but for this article, we focused on the effective gesture listed above in order to use an expert practice of the guitar to the control of speech. The detected gesture we used can be divided in two parts:

- continuous gesture: bend, slide and plucking point.
- triggering gesture: pitch, note on, note off, hammer-on, pull-off, palm muting, natural harmonic

The mappings detailed above present a first attempt to make an electric guitar talk, i.e to control a MAGE synthesized text by a guitar used as a controller as well as an instrument. The example listed below have been used in performative context .The text used is a synthesized version of the poem *The Fish*, by Elisabeth Bishop.

5.1 Mapping examples

5.1.1 Playing generates labels

The first mapping idea that appeared was to generate labels as the guitar was played in order to keep a basic direct correlation between guitar sound and produced sound. We simply mapped one-to-one both amplitudes and both pitches and any new attack was used to trigger a new label. Quickly, a first improvement was to generate a series of labels rather than one as the latter is a too small linguistic and sonic unit to be understandable. A empirically-defined series of three labels gave the guitarist a reactivity that matched closely its playing.

The second improvement to this basic correlation was to use an envelope instead of a one-to-one mapping between both amplitudes. Indeed, the reactive mapping forces the melody and the voice to be two tied discourses dependent on the guitarist's will. Using an envelope gave the guitarist a wider range in the guitar/voice relationship. The reactive setting could still be used with a short envelope and a less correlated setting could be defined with a longer envelope. Longer envelopes have been used to enable the guitarist to play something else while the text was being pronounced, a notes on a defined-string being used as a labels trigger. It has to be noticed that the envelope was simply modeled with two ramps (upward and downward) and that the three-labels series was, it that case, replaced by a envelope-long series of labels.

This second improvement expands the possibility of this basic mapping by giving the instrumentalist the possibility to go from a correlated relationship between the voice and the guitar to an uncorrelated, less direct relationship.

5.1.2 Navigating inside the text

Apart from the controls, affecting the way the text is pronounced, being able to change the order and the connection the words and sentences had in the first place seemed to be a very motivating research axis, artistically speaking.

MAGE implementation, as previously mentioned in 4.3, enables to manipulate the sequence of labels with messages e.g, labelinsert, labelreplace, labelswitch.

As a first attempt to modify poem's structure, attack of note was mapped to a random poem's line read or not in loop (the in and out labels of each poem's lines had been defined by hand and poem's lines are accessed through label-switch message). Despite the fact that a specific sequence can't be redone with this system, randomness forces the guitarist to listen to which poem's line is triggered in order to react to it (i.e, let the loop play or go to a new line). Indeed, due to loop and randomness functionality, the guitarist is not anymore only the sound generator but also part of a question / answer game with the MAGE system. Both loop possibility and random triggered text adds notions of rhythmic and responsive game between the guitar and the voice.

5.1.3 Continuous controls

In this third mapping proposition, our attention was focused on the way the text is pronounced by using vocal tract length and speed parameters as well as continuous data extracted from guitar detection (contrary to the trigger data used in the previous mapping examples).

Here, bends (after fretting a note with the left-hand finger, the same finger pushes or pulls the string without lifting) have been mapped to vocal tract length and speed. In the first case, bending the string enabled the guitarist to define the character of the voice (e.g, going from a normal voice to a tiny voice). On the second case, bend was used to slow down the voice in order to reach its micro scale. In both cases, a certain correlation between the physical act of the guitarist and the alterations applied on the voice is present.

It has to be noticed that other parameters such as duration of labels and scale or shift of the pitch have been used by hand during the performance, but no mapping with the guitar have been tried yet.

6. CONCLUSIONS

We have presented a very promising approach to realize reactive control in HMM-based speech synthesis which can result in speech expressivity and speaking style variability based on user interaction. First, we described the modifications and improvements that were applied to the initial system in order to create a framework that can successfully support reactive HMM-based speech and singing synthesis. Secondly, we presented the various integrations of MAGE into several reactive environments, such as openFrameworks, PureData and Max/MSP. Thirdly, we presented a prototype for reactively generating speech using a guitar as a controller to manipulate several aspects of speech, such as context, speed, pitch. We used the challenge of creating a talking guitar to develop our second major version of MAGE and bring more voice-related materials in the hands of guitarists. Our future work will focus on investigation using more complex controls combined with meaningful user controls as well as conducting user studies for its evaluation.

7. REFERENCES

[1] M. Astrinaki, N. d´Alessandro, and T. Dutoit. MAGE - a platform for tangible speech synthesis. In *Proc. of NIME´12*, pages 353–356, 2012.

[2] M. Astrinaki, N. d´Alessandro, B. Picart, T. Drugman, and T. Dutoit. Reactive and continuous control of HMM-based speech synthesis. In *Proc. of the SLT12*, pages 353–356, 2012.

[3] N. d´Alessandro, O. Babacan, B. Bozkurt, T. Dubuisson, A. Holzapfel, L. Kessous, A. Moinet, and M. Vlieghe. Ramcess 2.x framework - expressive voice analysis for real- time and accurate synthesis of singing. *Journal of Multi-modal User Interfaces*, 2:133–144, 2008.

[4] N. d´Alessandro and T. Dutoit. Handsketch bi-manual controller: Investigation on expressive control issues of an augmented tablet. In *Proc. of NIME*, pages 78–81, 2007.

[5] S. Fels, R. Pritchard, and A. Lenters. Fortouch: a wearable digital ventriloquized actor. In *Proc. of NIME*, pages 274–275, 2009.

[6] R. Graham. A live performance system in pure data: Pitch contour as figurative gesture. In *Proc.of Pure Data Convention*, Bauhaus-Universität, Weimar, Germany, August 2011.

[7] A. Hunt and A. Black. Unit selection in a concatenative speech synthesis system using a large speech database. In *Proc. of ICASSP*, pages 373–376, 1996.

[8] O. Lähdeoja. An approach to instrument augmentation : the electric guitar. In *Proc. of NIME*, 2008.

[9] Z. Ling, K. Richmond, and J. Yamagishi. Articulatory control of HMM-based parametric speech synthesis using feature-space-switched multiple regression. *Proc. of IEEE Transactions on Audio, Speech, and Language Processing*, 21:207–219, 2013.

[10] Z.-H. Ling, K. Richmond, and J. Yamagishi. Vowel creation by articulatory control in HMM-based parametric speech synthesis. In *Proc. of Interspeech*, 2012.

[11] T. Masuko, K. Tokuda, T. Kobayashi, and S. Imai. Voice characteristics conversion for HMM-based speech synthesis system. In *Proc. of ICASSP*, pages 1611–1614, 1997.

[12] P.Cook. Spasm: A real-time vocal tract physical model editor/controller and singer. *Computer Music Journal*, 17:30–44, 1992.

[13] B. Pritchard and S. Fels. GRASSP: Gesturally-realized audio, speech and song performance. In *Proc. of NIME*, pages 272–271, 2006.

[14] L. Reboursière, O. Lähdeoja, T. Drugman, S. Dupont, C. Picard, and N. Riche. Left and right-hand guitar playing techniques detection. In *Proc. of NIME*, 2012.

[15] K. Shichiri, A. Sawabe, K. Tokuda, T. Masuko, T. Kobayashi, and T. Kitamura. Eigenvoices for HMM-based speech synthesis. In *Proc. of IC- SLP*, pages 1269–1272, 2002.

[16] M. Tamura, T. Masuko, K. Tokuda, and T. Kobayashi. Adaptation of pitch and spectrum for HMM-based speech synthesis using mllr. In *Proc. of ICASSP*, pages 805–808, 2001.

[17] W. von Kempelen. Mechanismus der menschlichen sprache nebst beschreibung einer sprechenden maschine. 1791.

[18] J. Yamagishi, K. Onishi, T. Masuko, and T. Kobayashi. Modeling of various speaking styles and emotions for HMM-based speech synthesis. In *Proc. of Interspeech*, pages 2461–2464, 2003.

[19] T. Yoshimura, K. Tokuda, T. Masuko, T. Kobayashi, and T. Kitamura. Speaker interpolation in HMM-based speech synthesis system. In *Proc. of Eurospeech*, pages 2523–2526, 1997.

[20] H. Zen, K. Tokuda, and A. W. Black. Statistical parametric speech synthesis. *Speech Communication*, 51:1039–1064, 2009.