PURE DATA EXTERNAL FOR REACTIVE HMM-BASED SPEECH AND SINGING SYNTHESIS

Maria Astrinaki, Alexis Moinet, Nicolas d'Alessandro, Thierry Dutoit

TCTS Lab., Numediart Institute, University of Mons, Belgium

{maria.astrinaki, alexis.moinet, nicolas.dalessandro, thierry.dutoit}@umons.ac.be

ABSTRACT

In this paper, we present the recent progress in the MAGE project. MAGE is a library for reactive HMM-based speech and singing synthesis. Here, it is integrated as a Pure Data external, called **mage~**, which provides reactive voice quality, prosody and identity manipulation combined with contextual control. **mage~** brings together the high-quality, natural and expressive speech of HMM-based speech synthesis with high flexibility and reactive control over the speech production level. Such an object provides a basis for further research in gesturally-controlled speech synthesis. It is an object that can "listen" and reactively adjust itself to its environment. Further in this work, based on **mage~** we create different interfaces and controllers in order to explore the realtime, expressive and interactive nature of speech.

1. INTRODUCTION

The purpose of a Text-To-Speech synthesiser (TTS) is to convert a textual input into an acoustic signal. This acoustic signal should be perceived by any listener as the reading of the inputed text by a given speaker in a intelligible and natural way. Nowadays we even expect such a virtual speaker to be expressive and reactive. Expressivity relates to the ability of the system to "convey emotions" in the production of the text, typically moods. Reactivity supposes that the speech production is not a unilateral process but can rather "react" to the context, such as ambient noise, listener's facial, vocal or postural responses or varying emotional state. In that sense, expressivity and reactivity are intrinsically related. These two features enable the adaptive process of human communication.

In the history of speech synthesis, the target of creating "more human-like" speech sound has mainly been pursued through the improvement of the intelligibility, naturalness and expressivity of the resulting acoustic waveforms, i.e. signal-wise. The system-wise problem of reactivity has always been very marginally considered. Indeed it gathers considerations that stretch outside the speech research itself and are rather encountered in the fields of software design or human-computer interaction.

In formant synthesis [1], the produced waveform is intelligible but it sounds robotic and artificial. However formant synthesizers are small and memory efficient programs without the need of big computational power. They can be easily embedded in systems and are easy to control. In diphone synthesis [2], a small footprint is used and there is space for reactivity and controllability, but the quality of the final output is degraded and the synthetic speech sounds robotic and without expression. The articulatory approach in speech synthesis [3] also results in rather poor speech quality compared to other synthesis methods since the more accurate the model used is, the closer it follows the human physiology but at the same time it becomes more difficult to control, requiring great computational power. Unit selection technique [4] is the dominant approach to speech synthesis, since it provides the greatest intelligibility and naturalness. There is need for large high quality databases, which is difficult and costly. Unit selection is cannot be parametrized with content-oriented speech production.

In statistical parametric speech synthesis [5], there are no real speech samples used at synthesis runtime, the generated speech is synthesized based on vocoding techniques, which results in "buzzy" and "muffled" output. However, this approach allows easy modification of voice characteristics [5] and more importantly it is very intelligible and flexible. It results in a very memory efficient system with a small footprint which can be integrated in devices with small computational power and memory capacity. As we see from this method comparison, the platform that would be more suitable for our research is the statistical parametric speech synthesis. This is because it combines good output quality and intelligibility with well defined statistical models.

We see that, by design, TTS produces static output, whereas, speech is a realtime, dynamic gestural phenomenon that is constantly influenced by the changes of the environment. This "deaf" design is, by its nature, limited, since it is incapable of adapting to the user and to the environment. It has inefficient use and it is lacking the naturalness that comes from the user interaction, without letting any space for user expression and creativity. Current TTS application are strongly bound to its static nature, but the last years there is an emerging interest in applications that need reactivity and expressivity in speech production.

An approach to tackle this issue comes with MAGE, a platform for reactive HMM-based speech and singing synthesis. MAGE allows reactive control of prosody, context and speaking style, although it is still difficult to produce meaningful expressivity without an intermediate mapping layer or an underlying model to enable a high-level control representation. In Section 2, we describe briefly MAGE, the platform used for reactive HMM-based speech and singing synthesis. Then we present the integration of MAGE as an external for Pure Data [6], in Section 3. In Section 4, we discuss the conclusions of this work.

2. MAGE : REACTIVE HMM-BASED SPEECH AND SINGING SYNTHESIS

MAGE [7] is a platform for reactive HMM-based speech and singing synthesis. It is based on HTS [5] while providing the required framework for reactivity and interaction. In traditional HTS, the output generation takes into account all the available contextual information at a time, which results in an actual delay of one utterance. In contrast, MAGE uses only one phonetic label at a time as contextual information, resulting in a much smaller time window, allowing user interaction while preserving the output quality.

2.1. Reactive speech parameter generation

MAGE has a training and a synthesis part; the training is common with HTS but the synthesis is realized by using reduced contextual input. In details, for each phonetic label only the corresponding context-dependent HMMs of that label are used in order to generate the speech parameters; sequences of spectral and excitation parameters. For the generation of the speech parameter trajectories we use the maximization in Equation 2, as presented in [5]. Consequently, the resulting parameter trajectories are not generated based on the overall maximum probability of the final utterance, but only based on the locally-maximized speech parameters at the phonetic label level.

$$q^\star = \underset{q}{\operatorname{argmax}}\, P(q \mid \boldsymbol{\lambda}^\star, \hat{\boldsymbol{T}}) \tag{1}$$

$$\hat{\boldsymbol{O}} = \underset{O}{\operatorname{argmax}}\, P(O \mid q^\star, \boldsymbol{\lambda}^\star, \hat{\boldsymbol{T}}) \tag{2}$$

where O and q are respectively the sequence of speech parameters and the sequence of states, $q^\star$ and $\lambda^\star$ respectively the estimated sequence of states and the concatenated left-to-right HMMs corresponding to the current label, $\hat{O}$ the sequence of locally-maximized generated speech parameters, and $\hat{T}$ is the time frame corresponding to the label on which $\hat{O}$ is computed.

Indeed, by using only the current phonetic label to generate the speech parameter trajectories, MAGE reduces the accessible time scale from the sentence level to just the label level. As illustrated in Figure 1, for every single phonetic label available to the system the speech parameter trajectories are generated from the HMMs for that single label, by simultaneously taking into account the user control. Then the corresponding speech samples are generated and stored independently. Note here that for the sample generation we have zero label delay, taking into account past dependencies if existing but no future label dependencies, and still preserve an output quality comparable with the original system.

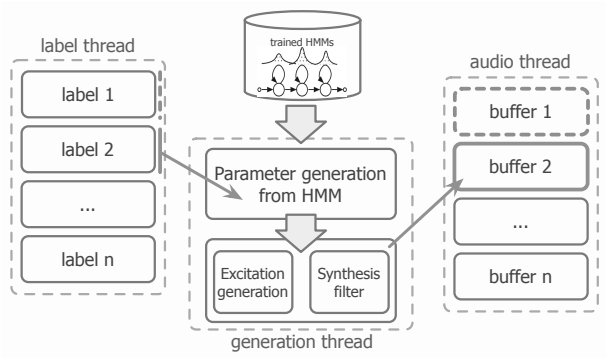

Figure 1: MAGE synthesis, using a single phonetic label to generate the speech parameter trajectories and audio buffers.

2.2. Realtime architecture and multithreaded control

In order to achieve user reactivity and provide realtime controls the above described reactive parameter generation is essential but not enough; a realtime architecture that supports multiple threads is required. MAGE itself is able to provide these important features so that it can be embedded into reactive and realtime frameworks, such as Pure Data. The multithreaded architecture allows user control over several speech and singing production levels always preserving a maximum delay of a single label.

As presented in Figure 2, MAGE consists of three main threads: the *label* thread, the *parameter generation* thread and the *audio generation* thread. The *label* thread allows the user to control the context of the output. By controlling the input sequence of the phonetic labels it is possible to achieve realtime contextual control. The provided user control is on the phonetic level, and therefore the delay introduced is the current phonetic label.

The *parameter generation* thread that generates the corresponding speech parameters of the current label has a similar delay. The *parameter generation* thread allows also reactive user control over the available models. Indeed, it is possible with a delay over the currently synthesized phoneme to reactively interpolate between models of different speakers controlling the speaker identity or models of different degrees of vocal effort or articulation [8].

Lastly, the *audio generation* thread will generate the actual speech samples. However, here the delay over the user control drops to the sample level, since the vocoder allows certain parameter modifications for every single sample. For example controls over the pitch trajectory, over the speed or the vocal tract length parameter are applied to every sample separately. Such a feature allows instant, very reactive and accurate control over the synthesized output.

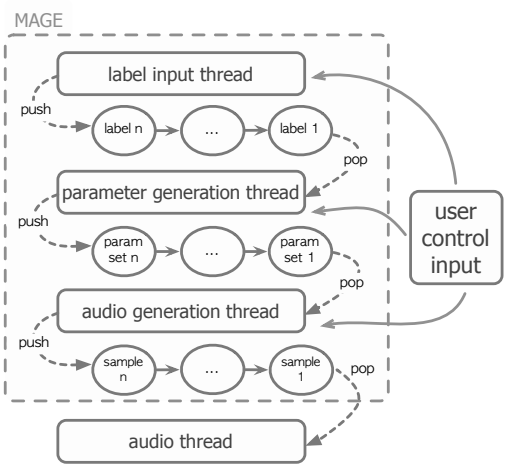

Figure 2: Multithread architecture of MAGE: the *label thread*, the *parameter generation* thread and the *audio generation* thread connected to the user control providing speech samples to the system's audio thread.

3. INTEGRATION OF MAGE IN PURE DATA

MAGE is a C++ library and can be integrated within several reactive and realtime frameworks. In this Section we present the integration of MAGE into Pure Data as an external called **mage~**. Such an integration combines reactive speech synthesis with open sound control (OSC) [9] and a first simple user interface.

3.1. Implementation and controls of the mage~ object

The **mage~** external object is implemented with one input and one output. The input is used to receive control messages from the user

whereas the output gives raw speech samples. It starts and initializes the three MAGE threads, as described in the previous section. It also creates and initializes all the functions in order to receive and apply the user inputs. Basically, there are two kind of controls in **mage~** that can be reactively applied to affect the synthesized output. First, the reactive manipulation of the context, that can be applied on the phoneme level. Second, the reactive control over the voice quality, prosody and speaker identity, that can be applied either on the phoneme level or on the sample level, depending on the nature of the control and the thread that is applied.

In details, the available controls in the **mage~** object for context manipulation are:

```
label alice.lab
labelfill
labelnext
labelinsert 10
labelreplace 54
labelswitch 7
```

where the message `label alice.lab` loads a list of phonetic labels from the label file alice.lab and `labelfill` send all the loaded labels into the **mage~** object so that the processing will start, using one label at a time allowing the user to focus on other controls and not only on the contextual control of the voice. Similarly, `labelnext` sends a label from that list to MAGE and goes to the next label of the list. `labelinsert N` sends the Nth label of the list to MAGE, `labelreplace N` sends the Nth label of the list to MAGE and makes `labelnext` jump to its next label. `labelswitch N` sends the Nth 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. The usage of the messages is presented in Figure 3.

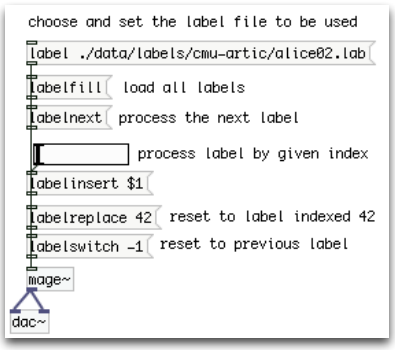

Figure 3: Messages for context manipulation in **mage~**.

Additionally, some example of the available controls in the **mage~** object for voice quality, prosody and model manipulation are:

```
reset
alpha 0.42
speed 2 scale
pitchshift 128
interpolate awb 0.75
interpolate slt 0.25
```

where the message `reset` when received reset the **mage~** object to its default values. `alpha` sets the value of the vocal tract length parameter to 0.42. `speed 2 scale` multiplies the current speed of the speech output by two. `pitchshift 128` shifts / adds 128 Hz to the currently produced pitch trajectory. `interpolate awb 0.75` sets the interpolation weight of the speaker "awb" to 75% and respectively `interpolate slt 0.25` sets the interpolation weight of the speaker "slt" to 25%. Here we see that, we can reactively change the speaker identity by switching gradually and mixing between two voices (e.g. from 100% awb to 100% slt). Such and action has a delay of only one phonetic label. The usage of the messages is presented in Figure 4.

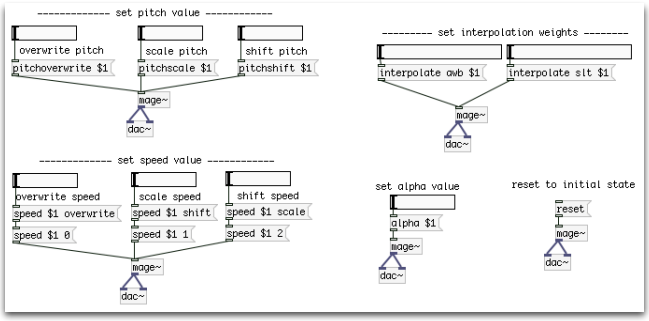

Figure 4: Messages for voice quality, prosody and model manipulation in **mage~**.

3.2. Interfaces combined with the mage~ object

An interesting feature of MAGE and consequently of **mage~** is that it supports OSC messages. Such a functions allows easy and fast prototyping with various environments, controllers and interfaces. Here we will present briefly different controllers that we have successfully combined with **mage~** in order to gesturally control speech production.

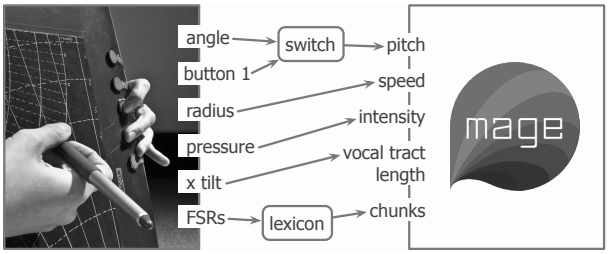

Figure 5: Mapping of the HandSketch control space to the parameters of the **mage~** object.

One of our first attempts to gesturally control speech was to combine **mage~** with HANDSKETCH [10], a new musical instrument prototype [7]. In Figure 5 we illustrate the mapping between **mage~** and the available pen-based gesture controls. The polar coordinates on the fan (angle, radius), control the pitch and speed respectively. By using the front button of the pen and moving along the angle on the fan, the user can select to overwrite the generated pitch trajectory or to deviate it by a given ratio. The pen pressure controls the speech intensity and the pen x tilt modifies the vocal

tract length. The Force Sensing Resistors (FSRs) on the side are used to trigger different controls over the context of the generated voice.

Another application built combines **mage~** with FaceOSC [11]. Here, based on realtime face tracking we control reactively the speech production. The head tilt controls the speech speed and pitch shift of the output whereas the opening and closing of the mouth controls the context. When tilting the head right or left the speed of the generated speech is accelerated or decelerated, respectively. When tilting the head up or down the generated pitch trajectory is shifted up or down. When the opening of the mouth is detected then a random phoneme is triggered and processed. Figure 6 illustrates the facial controls over the generated output.

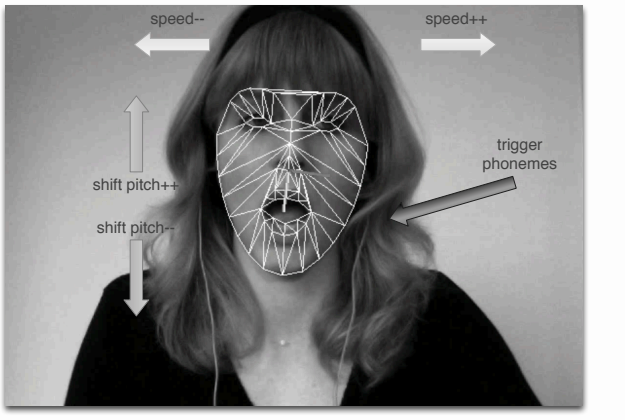

Figure 6: Mapping of three parameter of FaceOSC to control pitch, speed and context of the **mage~** object.

Finally, we used an electric guitar as a controller for MAGE speech synthesis, as described in [12]. We used algorithms to detect several different guitar playing techniques [13] and, combined with appropriate mapping, we managed to control the **mage~** object. Here, instead of a one-to-one mapping between the sound of the guitar and the synthesized speech we used an envelope between both amplitudes, which gives a wider range in the guitar / voice relationship. Considering now the contextual control, every attack of note was mapped to a random set of phonetic labels, over which the guitarist would apply the desired prosody controls.

4. CONCLUSIONS

In this work we present a pure data object for reactive HMM-based speech and singing synthesis, called **mage~**. Some first prototypes using the **mage~** object are also demonstrated. Each prototype tries to explore a different aspect in the framework of reactive speech and singing synthesis and at the same time to prove that reactive control over different speech production levels is possible.

Although it is rather easy and straight forward to combine different controllers and **mage~**, issues of formal evaluation rise. In order to properly evaluate such prototypes it is essential to conduct user studies. However, we realize that even if the different mapping and controls are meaningful for the user, still the expected naturalness and expressivity is not delivered or experienced. One of the reasons could be that controlling simultaneously both context and prosody or voice quality is overwhelming for the user. Maybe focusing on one aspect of the speech at a time could enable the user to generate the speech output as intended. Also, manipulated the contextual information at the "phonetic label" level on the one had delivers the needed reactivity but on the other hand it is too short as a unit for a human to manipulate within the required timeframes. 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.

5. ACKNOWLEDGMENTS

M. Astrinaki's work is supported by a PhD grant funded by UMONS and ACAPELA-GROUP SA.

6. REFERENCES

[1] R. Carlson and B. Granstrom, "A text-to-speech system based entirely on rules", in *Proc. of ICASSP*, 1976.

[2] F. Charpentier and M. Stella, "Diphone synthesis using an overlap-add technique for speech waveforms concatenation", in *Proc. of ICASSP*, 1986, vol. 11, pp. 2015–2018.

[3] B. Brent and W. J. Strong, "Windbag – a vocal-tract analog speech synthesizer", *Acoustical Society of America*, vol. 45, 309(A), 1969.

[4] A. Hunt and A. Black, "Unit selection in a concatenative speech synthesis system using a large speech database", in *Proceedings of IEEE International Conference of Acoustics, Speech, and Signal Processing*, 1996, pp. 373–376.

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

[6] Miller Puckette, "Pure data", September 2009.

[7] M. Astrinaki, N. d´Alessandro, and T. Dutoit, "MAGE - a platform for tangible speech synthesis", in *Proceedings of the 12th Conference on New Interfaces for Musical Expression (NIME´12)*, 2012, pp. 353–356.

[8] M. Astrinaki, N. d´Alessandro, B. Picart, T. Drugman, and T. Dutoit, "Reactive and continuous control of HMM-based speech synthesis", in *Proceedings of the IEEE Workshop on Spoken Language Technology (SLT 2012)*, 2012, pp. 353–356.

[9] A. Schmeder, A. Freed, and D. Wessel, "opensoundcontrol.org", September 2009.

[10] N. D´Alessandro and T. Dutoit, "Handsketch bi-manual controller: Investigation on expressive control issues of an augmented tablet", in *Proceedings of the 7th International Conference on New Instruments for Musical Expression (NIME'07)*, 2007, pp. 78–81.

[11] K. McDonald, "Faceosc", September 2012.

[12] M. Astrinaki, N. d´Alessandro, L. Reboursière, A. Moinet, and T. Dutoit, "Mage 2.0: New features and its application in the development of a talking guitar", in *Proceedings of the 13th Conference on New Interfaces for Musical Expression (NIME´13)*, 2013.

[13] L. Reboursière, O. Lähdeoja, T. Drugman, S. Dupont, C. Picard, and N. Riche, "Left and right-hand guitar playing techniques detection", in *Proceedings of the 12th Conference on New Interfaces for Musical Expression (NIME´12)*, 2012.