PERFORMANCE-DRIVEN CONTROL FOR SAMPLE-BASED SINGING VOICE SYNTHESIS

Jordi Janer, Jordi Bonada, Merlijn Blaauw

Music Technology Group
Universitat Pompeu Fabra, Barcelona
{jjaner|jbonada|mblaauw}@iua.upf.edu

ABSTRACT

In this paper we address the expressive control of singing voice synthesis. Singing Voice Synthesizers (SVS) traditionally require two types of inputs: a musical score and lyrics. The musical expression is then typically either generated automatically by applying a model of a certain type of expression to a high-level musical score, or achieved by manually editing low-level synthesizer parameters. We propose an alternative method, where the expression control is derived from a singing performance. In a first step, an analysis module extracts expressive information from the input voice signal, which is then adapted and mapped to the internal synthesizer controls. The presented implementation works in an off-line manner processing user input voice signals and lyrics using a phonetic segmentation module. The main contribution of this approach is to offer a direct way of controlling the expression of SVS. The further step is to run the system in real-time. The last section of this paper addresses a possible strategy for real-time operation.

1. INTRODUCTION

One of the challenges of artificial voice synthesizers is to generate a realistic, human-like sound. Speech techniques pursue realistically synthesizing a given emotion (“sadness”, “fear”, “surprise”, etc.). In singing voice synthesis, besides emotion, the goal is to achieve natural sounding musical expression. Several systems have addressed this issue. These usually follow a strategy that consists in first training a model from real performances and then to apply this model in the synthesizer. These models will usually only describe one or a limited set of performance styles. Although good results are achieved, there is still an important issue that remains unaddressed: flexible control.

Control plays an important role in musical virtual instruments. In practical implementations of human voice synthesis systems the available amount of control and the voice quality achieved are bound to each other, leading to a trade-off between *explicit modeling techniques* and *sample-based techniques* [1]. The former allows very flexible control while the voice quality will generally be unconvincing. The latter, in contrast achieves a good voice quality, but with limited control. The challenge in the next years is to combine both techniques, permitting a wider control range while maintaining a good and natural voice quality.

Focusing on sample-based synthesizers, their control has an additional trade-off between database size and transformation post-processing required. We might devise different strategies. One could be building a large database with complete phrases of different expressive performances; another could be using only small database of a single set of phoneme samples with a “flat” expression. Most Singing Voice Synthesizers (SVS) require two types of inputs: a melody, usually a MIDI score, and lyrics. To control the expression, we can either augment this MIDI score with an expressive model or allow the user to manually edit the control parameters using the synthesizers GUI. If we use expressive models, the type of control parameters is often very high-level, such as “happy or sad”, “bluesy”. The difficulty lies in defining a useful, preferably small, set of such parameters that allow meaningful combinations and a wide range of expression. Another big difficulty with this approach is mapping these parameters to low-level synthesizer controls such as the timing of phonetics, pitch, dynamics, etc., which are difficult to control directly by the user.

This work uses the sample-based spectral concatenation SVS introduced by Bonada and Loscos in [2]. This synthesizer picks samples from a database and concatenates one after the other, while applying different kinds of transformations. Good results are achieved by using a corpus of diphonemes instead of single phonemes. Using diphonemes, the samples not only contain the phonemes themselves, but also the transitions or articulations between phonemes. The diphoneme samples are further labeled automatically to indicate sustained parts of the samples and transitional parts of the sample to aid synthesis. The SVS reads a musical score, in the form of a MIDI score, a phonetic transcription of the lyrics and additional expressive parameters such as vibrato, type of attack, etc. The experiments use a singer database in Spanish. Although the described system runs off-line, our goal is to adapt it for real-time operation, as briefly addressed in the last section.

Having briefly introduced the limitations of controlling the expression of Singing Voice Synthesizers, we propose to control the expression of a sample-based Singing Voice Synthesizer’s output voice with another voice. We extract a set of descriptors of the input voice, which are then mapped to the synthesizer’s internal parameters. These parameters are: energy, fundamental frequency and vibrato. The phonetic timing of the synthesis output is also derived from the input voice. In the proposed system, the phonetic timing is generated by *automatic phonetic segmentation* using a similar approach as [3].

A potential application of this system is voice impersonation. Methods such as timbre mapping and transposition achieve good results for changing the character of the voice, e.g. gender change. However, with state-of-the-art voice processing techniques, it is difficult to impersonate a particular target singer by audio transformation solely. Although the timbre of the target can be reached in stationary sounds, most of the singer characteristics lie in the phonetic articulations, which are hard to achieve by means of transformation. Another use of this system can be for singing in a foreign language not mastered by the singer, where a bad pronunciation might induce a negative effect on the singing voice quality. Using sample-based synthesizers we can improve the pronunciation with ease.

On the other hand, this work can be considered as the natural succession of the singing voice morphing system presented by Cano et al. [4]. In this real-time karaoke-oriented system, the user controlled the timing of a given song, while the synthesized voice was a morph between the user and the target professional singer. However, for each song to be performed it was required to record the professional performance for that specific song. Using a SVS based on samples of the target singer and the presented control, we would overcome this issue.

2. CONTROLLING VOICE SYNTHESIS EXPRESSION

Expressive sound synthesis control is mainly tackled from two different viewpoints. The first adds expression to a neutral musical score by means of expression models. The second looks at the interfaces that a performer uses to control Digital Musical Instruments (DMI). In our approach, we use an alternative method for controlling sound synthesis.

2.1. Control with expression models

Part of the research on musical expression deals with characterizing and modeling expressive performances in order to apply a given expression (e.g. “sad”, “happy”, “dark”, etc.) to neutral performance, or even to reproduce the playing style of a given performer [5]. First approaches worked solely at an abstract level, such as MIDI scores. Using machine learning techniques the system learns different types of manually annotated performances for a given score. Then it modifies the neutral performance to another synthesized one that conveys a particular expression. Recent studies tackle the modeling and control of expressive performances at a signal level. In these systems, given a neutral performance the user navigates in a two-dimensional expression space, controlling the mood of the transformed sound [6].

This type of control is very high-level. User actions are therefore limited to select among predefined expressive performances or at most to control the amount of morphing between two of them.

2.2. Control with input interfaces

Another way of applying expression is by means of a musical interface. Musical interfaces, also referred as *Musical Controllers* are mainly employed in performance situations and work in real time. The control of Digital Musical Instruments has been traditionally bound to the limitations of the MIDI protocol. Interfaces capture *musical gestures*; which gestures and how they are parameterized will determine the synthesized sound. A wide variety of musical controllers are available nowadays, ranging from traditional keyboards to sophisticated artificial vision systems.

In commercial singing voice synthesizers such as Yamaha’s Vocaloid system[1], we can use a MIDI interface for recording a MIDI score. In addition, the control parameters of the synthesizer can be manually edited using a GUI. The user can draw pitch and loudness contours, or add high-level expressive gestures such as vibrato. Also, the type of articulation (e.g. “legato”, “strong accent”) can be specified. A problem of this type of control is that there is no *Musical Controller* we know appropriate for singing voice synthesis, as there are for other types of synthesis such as MIDI wind controllers for wind instrument synthesizers. This fact restricts the generation of expressive MIDI score suited to the voice. On the other hand, manually parameter editing does not solve the problem completely since it is a slow and tedious process, although very flexible.

2.3. Performance-driven control

An alternative way of applying expressive control to a synthesizer is to use an audio-stream as control. This concept is often referred in the literature as “Indirect Acquisition”. Several systems addresses this issue from different perspectives, either using a general audio-signal as control [7, 8], or specifically using the singing voice as control for other sounds [9]. In [3], Meron uses singing performances as input for an automatically trained system that produces high-quality singing voice synthesis.

In our approach the performance descriptors have a direct control of the synthesizer parameters. This type of control avoids the intermediate level of the MIDI score, which reduces the resolution of the musical gestures. With respect to the mapping, it can be kept simple since voice is controlling voice.

3. SYSTEM IMPLEMENTATION

Our implementation consists of three independent modules: *Performance Analysis*, *Phonetic Alignment* and the actual *Singing Voice Synthesizer*. The inputs are a sound file and a phonetic transcription of the lyrics in text format. In this section we introduce the different modules that allow us to control the synthesis with a performance. First, we describe the performance analysis and the phonetic alignment modules. Second, we look at the control layer of the synthesizer, observing how it affects the generated singing.

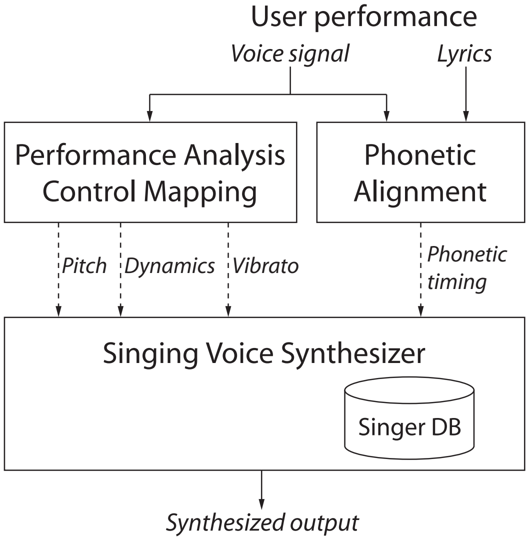

Figure 1: *Block diagram of the complete system: Performance Analysis, Phonetic Alignment, and Singing Voice Synthesizer.*

[1] http://www.vocaloid.com/

3.1. Performance analysis and mapping

We analyze the voice signal and extract a set of descriptors that are mapped to the synthesizer's control. Expression in singing voice could be described in a low level by energy, fundamental frequency and timbre, which all vary dynamically along the time axis. Along with the mentioned descriptors, a higher-level analysis allows us to extract the phonetics and musical gestures such as vibrato or type of articulation.

From the set of low-level descriptors, we take energy and fundamental frequency. From the high-level analysis, we use the vibrato. Energy is computed directly from the windowed input voice signal. Several techniques and methods for fundamental frequency estimation have been extensively studied in the past decades. In our implementation we use a frequency-domain method based on work by Cano et al. [10]. Detecting the presence of vibrato in the signal is done by post-processing the fundamental frequency curve. The method implemented gives a measure of the probability of vibrato, and its depth and rate values. In brief, the method searches typical vibrato patterns (i.e. a quasi-sinusoidal component with a frequency in a range from 3 Hz to 8 Hz) after filtering out slow fundamental frequency variations due to intonation. Extracted descriptors are vibrato rate and vibrato depth. This method is still under development and needs to be studied further.

How input parameters are assigned to the synthesis parameters is known as mapping. In our case, the mapping is one-to-one, and the mapping layer consists basically in adapting the energy and fundamental frequency signals to the requirements of the synthesizer control layer. The energy curve is smoothed in order to fit the *dynamics* control of the synthesizer. The fundamental frequency curve also has to be adapted. The first requirement for the pitch control parameter is to be continuous. Thus, in transition and unvoiced phonemes when fundamental frequencies are missing, the curve is filled by smoothly interpolating the boundary values. Also, the rapid and small fluctuations of fundamental frequency not resulting from musical articulations, should ideally be removed. The reason is that these fluctuations are caused by the phonetics and how they are pronounced. Thus, they vary from one singer to another, and will be already present in the database samples. As a result, the computed pitch control is actually a smoothed and continuous version of the fundamental frequency analysis where musical gestures such as articulation and vibrato are kept. In addition, we have two options for controlling the vibrato: either we use the *pitch curve* extracted from the user performance, or a *pitch curve* of an actual vibrato by the recorded singer. The choice will depend on the quality of the user performance, since untrained singers will hardly produce a good vibrato. Finally, the parameters that are sent to the synthesizer are shown in the table 1.

3.2. Phonetic alignment

Up to this point, we have addressed the control of the musical expression described by pitch and dynamics contours. However, an important part of the singing voice expression belongs to phonetic information. In western classical singing, scores assign a phoneme to each note. For long notes, a vowel sound is sustained, while consonant sounds take place in the transitions. This concept is maintained in the used SVS [2]. In the phonetic timing of the internal score (see section 3.3), stationary voiced phoneme samples are looped while the duration of most remaining phoneme samples is kept unaltered.

Voice Descriptor	*Synthesizer Parameter*
Fundamental Frequency	Pitch
Energy	Dynamics
Vibrato Depth	Vibrato Depth
Vibrato Rate	Vibrato Rate

Table 1: *Performance descriptors extracted from the voice signal and corresponding synthesis control parameter.*

In our approach, the phonetic timing is also driven by an input voice. Therefore, the duration of the phonemes are derived from those in the input performance. To extract the phonetic timing from the voice signal we use the segmentation module of the *Julius* Automatic Speech Recognition system [11]. The segmentation process is based on Hidden Markov Models (HMM), and uses acoustic models for Japanese in HTK format. Although the performances are in Spanish, the automatic segmentation works good enough since the phonetics of both languages are to some extent similar. However, we need to convert appropriately the Spanish phonetics symbols to the Japanese ones.

At first, we aligned all the phonemes in the lyrics to the corresponding phonemes from the input performance. We observed that with the used SVS, while time-scaling voiced phonemes does not significantly affect the synthesis quality, the time-scaling process of unvoiced sound produced unrealistic and artificial sound. In order to improve the sound quality, we introduced two alternatives modes of generating the phonetic timing. In the first, "voiced onset" mode, only voiced phoneme onsets are aligned, while in the second, "vowel onset" mode, only vowel onsets are aligned. Both options ensure that unvoiced sounds have the same duration as in the database, i.e. they are not time-scaled. In the figure 2, we observe the waveforms of the input performance and the two proposed alternative phonetic alignments. A first listening evaluation showed that the most natural sounding results are achieved with "voiced onset" alignment.

3.3. Synthesizer control layer

Internally the SVS used in this work has a small number of low-level controls that define the synthesis target. The *phonetic timing* defines which phoneme sounds at which time and its duration. *Pitch* and *dynamics* define target pitch and target singing intensity respectively and can be controlled directly or be derived from higher-level controls (MIDI notes and velocities) using models of pitch and dynamics curves. A number of other expressive controls control other sample transformations such as "roughness" or "breathiness" quality of the voice. Finally, vibrato controls, depth and rate, can be used to easily control the vibrato of the synthesized voice using a signal analysis of actual vibratos uttered by the recorded singer.

From these low-level controls an internal score is built which describes a sequence of samples and their transformation. The optimal sequence of samples is chosen so the overall transformation required is minimized. The cost function used is derived from the compression or expansion of samples needed to fit the given phonetic timing, the pitch and dynamics transformations needed to

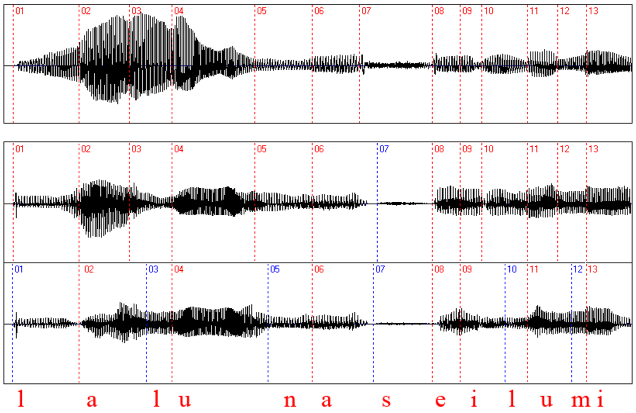

Figure 2: *Two different modes of phonetic alignment: The second plot shows the synthesized output with all voiced phonemes aligned to the input segmentation (top plot). The bottom plot shows the synthesized output with only vowels aligned to the input segmentation (top plot).*

reach the target pitch and dynamics curves and the continuity of the sample sequence.

4. A STRATEGY FOR REAL-TIME OPERATION

As we have mentioned, the current implementation runs off-line. However, our final aim is to have a system running in real-time. In this section we introduce some of the issues involved in a real-time operation.

The goal is to perform phoneme aloignment to a text, which generates the phonetic timing information sent to the synthesizer. Our strategy will be based on the approach presented by Loscos et al. in [12]. This system uses HMM for the phoneme recognition, a Finite State Network (FSN) and a Viterbi algorithm to perform the text alignment.

The main limitation for implementing the approach presented in this paper in real-time is that the SVS uses diphoneme samples. Therefore, the synthesizer has to wait that the rightmost phoneme is detected in the phonetic segmentation, and then play the complete diphoneme sample. Obviously, this will introduce latency in our system. A way to minimize this effect would be to time-compress the left-hand side phoneme of the diphoneme sample, or to start playing the diphoneme sample with an appropriate offset.

5. CONCLUSIONS

We have introduced an approach that uses a singing performance to control the expression of a Singing Voice Synthesizer. It aims at improving the process of generating expressive synthetic singing, for which most state-of-the-art systems rely on modeling techniques or direct manipulation of low-level parameters. We have introduced the performance analysis module of our proposed system that extracts *pitch*, *dynamics* and *vibrato* information from an input voice signal. Given lyrics and a voice signal, the system can automatically perform phonetic segmentation from which the phonetic timing of the synthesis can be derived. Due to sound quality limitations of the SVS used, we suggested two alternatives for the phonetic alignment besides a one-to-one alignment with the input voice. First experiments showed that best sounding quality and natural sounding singing is achieved aligning voiced phonemes only. Although our final aim is to implement a real-time control of SVS, the current off-line implementation already offers a direct and easy way of controlling synthetic singing. Some audio examples are also available online [2].

6. REFERENCES

[1] X. Rodet, "Synthesis and processing of the singing voice," in *1st IEEE Benelux Workshop on Model based Processing and Coding of Audio (MPCA 2002),* Leuven, Belgium, 2002.

[2] J. Bonada and A. Loscos, "Sample-based singing voice synthesizer by spectral concatenation," in *Proc. Stockholm Music Acoustics Conf. (SMAC'03)* Stockholm, Sweden, 2003, pp. 439–442.

[3] Y. Meron, "High quality singing synthesis using the selection-based synthesis scheme," Ph.D. dissertation, University of Tokyo, 1999.

[4] P. Cano, A. Loscos, J. Bonada, M. de Boer, and X. Serra, "Voice morphing system for impersonating in karaoke applications," in *Proc. Int. Comp. Music Conf. (ICMC'00),* Berlin, Germany, 2000, pp. 109–12.

[5] G. Widmer and W. Goebl, "Computational models of expressive music performance: The state of the art," *J. New Music Research*, vol. 33, no. 3, pp. 203–216, 2004.

[6] S. Canazza, G. De Poli, C. Drioli, A. Roda, and A. Vidolin, "Modeling and control of expressiveness in music performance," *Proc. IEEE*, vol. 92, no. 4, pp. 286–701, 2004.

[7] E. Métois, "Musical sound information: Musical gesture and embedding synthesis," Ph.D. dissertation, Massachusetts Institute of Technology, 1996.

[8] T. Jehan and B. Schoner, "An audio-driven, spectral analysis-based, perceptually meaningful timbre synthesizer," in *110th Conv. Audio Eng. Soc.,* Amsterdam, the Netherlands, Amsterdam, Netherland, 2001. [Online]. Available: citeseer.nj.nec.com/448390.html

[9] J. Janer, "Voice-controlled plucked bass guitar through two synthesis techniques," in *Int. Conf. New Interf. for Musical Expr. (NIME'05),* Vancouver, Canada, 2005, pp. 132–135.

[10] P. Cano, "Fundamental frequency estimation in the SMS analysis," in *Proc. COST-G6 Workshop on Digital Audio Effects (DAFx-98),* Barcelona, Spain, 1998, pp. 99–102.

[11] A. Lee, T. Kawahara, and K. Shikano, "Julius — an open source real-time large vocabulary recognition engine," in *Proc. European Conf. on Speech Communication and Technology*, 2001, pp. 1691–1694.

[12] A. Loscos, P. Cano, and J. Bonada, "Low-delay singing voice alignment to text," in *Proc. Int. Comp. Music Conf. (ICMC'99),* Beijing, China, 1999, pp. 437–440.

[2] http://www.iua.upf.edu/~jjaner/dafx06