GENERATING SINGING VOICE EXPRESSION CONTOURS BASED ON UNIT SELECTION

Martí Umbert **Jordi Bonada** **Merlijn Blaauw**
Music Technology Group, Universitat Pompeu Fabra
Barcelona, Spain
{marti.umbert, jordi.bonada, merlijn.blaauw}@upf.edu

ABSTRACT

A common problem of many current singing voice synthesizers is that obtaining a natural-sounding and expressive performance requires a lot of manual user input. This thus becomes a time-consuming and difficult task. In this paper we introduce a unit selection-based approach for the generation of expression parameters that control the synthesizer. Given the notes of a target score, the system is able to automatically generate pitch and dynamics contours. These are derived from a database of singer recordings containing expressive excerpts. In our experiments the database contained a small set of songs belonging to a single singer and style. The basic length of units is set to three consecutive notes or silences, representing a local expression context. To generate the contours, first an optimal sequence of overlapping units is selected according to a minimum cost criteria. Then, these are time scaled and pitch shifted to match the target score. Finally, the overlapping, transformed units are crossfaded to produce the output contours. In the transformation process, special care is taken with respect to the attacks and releases of notes. A parametric model of vibratos is used to allow transformation without affecting vibrato properties such as rate, depth or underlying baseline pitch. The results of a perceptual evaluation show that the proposed approach is comparable to parameters that are manually tuned by expert users and outperforms a baseline system based on heuristic rules.

1. INTRODUCTION

Modeling expressive speech and singing voice has attracted the interest of researchers for many years now. One of the main problems with many of the current singing voice synthesizers that are widely available, such as Vocaloid [1], is that the included models of expression are relatively simple and often are not sufficient for providing a natural-sounding and expressive synthesis "out-of-the-box". To improve results, users have to manually edit control parameters such as pitch bend, dynamics and vibrato, making the process very time-consuming and requiring expert skills.

To alleviate this problem, this paper aims to automatically generate better expression contours from a high-level, symbolic input score. The scope of this article is limited to pitch and dynamics (loudness or singing intensity) evolution over time within a local context. Other aspects of musical expression, such as phrasing, timing deviations, timbrical variations or ornaments, are not considered.

One of the most basic approaches, and the one that is typically used in current singing voice synthesis systems, is based on heuristic rules. For instance, in [2] a simple parametric model is used, based on anchor points, which are manually derived from a small set of arpeggio recordings. The advantage of these approaches is that they are relatively straight-forward and completely deterministic. On the other hand, the main drawback is that either the models are based on very few observations that don't fully represent a given style, or they are more accurate but become unwielding due to the complexity of the rules.

Copy synthesis is another basic type of approach which avoids the need for modeling. In this case, expressive parameters are directly taken from parallel recordings and used to control the synthesis. For instance, [3] applies this concept to singing voice synthesis, where a singing performance directly controls the synthesized expression. The same approach is also applied for generating prosody in speech synthesis [4]. Timbre is set by an emotional database of diphones, and pitch and diphone durations are obtained by copying them from a real utterance. For the Vocaloid synthesis engine, a similar system has been released commercially [5], which has resulted in what is generally considered a very significant increase in synthesis quality. The main disadvantages of copy synthesis approaches are the need for parallel recordings to capture expression controls.

Statistical models, such as HMMs [6], avoid the main drawbacks of the approaches mentioned above. These have been used to model and to produce emotional speech and different speaking styles. Some related applications are style adaptation (from neutral to a target style), interpolation between two or more styles, intensity control and style identification [7]. A well-known project where these techniques have been implemented is HTS [8].

With respect to singing voice synthesis, modeling singing style using HMMs has also been studied. In [9] the singing style is modeled statistically. It focuses on relative pitch, vibrato and dynamics using context dependent HMMs. In this work, notes are considered to contain up to 3 regions (beginning, sustained and end) in order to reflect their expression. HMMs are used to model the singing expression

parameters of these regions or behaviors.

The main drawback of HMM-based synthesis is the over-smoothing of parameters, since statistical averaging affects quality and the perceived emotion [6]. On the other hand, it has proven to be flexible to change voice characteristics, speaking styles and emotions (e.g. interpolation, extrapolation). Another advantage of statistical models is that they are like "black boxes", simply trained with data, without requiring much prior knowledge about the underlying expressive mechanisms.

Finally, an approach inspired by copy synthesis, but avoiding its main disadvantages, is unit selection. It has been used for synthesizing expressive speech utterances. For instance in [10] unit selection is applied to concatenate variable-size units taken from a large database of emotional speech of a single speaker. In this case the units will contain the audio segments used in synthesis, including inherent prosody. In [11], unit selection is used for the transformation of prosody in intra-speaker emotional voice conversion. First, a mapping is made between parallel neutral and emotional pitch contours, segmented into units. In this case the units used are "accent groups"; an accented syllable and its surrounding syllables. Then, for a given neutral input unit, the closest unit in the map is selected and the mapped emotional prosody is applied in its place.

The main drawback of concatenative-based synthesis is the lack of flexibility with respect to statistical approaches (e.g. interpolation). On the other hand, due to the lack of statistical modeling in unit selection, its best-case examples are generally considered to outperform the HMM-based ones [6].

This paper applies the unit selection approach to expression contour generation for singing voice synthesis. The presented approach is concatenative, taking units from a database of a capella singing voice recordings. Thus, the idea is to capture the original expression from short excerpts of this set of recorded songs by keeping the fine details of the control parameters.

The remainder of this paper is organized as follows. Section 2 explains in detail the proposed system, describing the expression database and the process for the generation of the expression contours. In section 3, the evaluation setup is explained and the results are discussed. In section 4 the conclusions are presented.

2. PROPOSED SYSTEM

2.1 Overview

The system is related to the user's input and the synthesis engine as shown in figure 1. Given a target score as input, represented in terms of the notes and timing provided by the user, the system (within the gray area) generates expression contours to control the synthesizer. The expression database stores a set of processed and labeled singing voice recordings. The local contexts of this database are the basic elements or units of the presented approach.

Units are melodic contexts of three notes or silences, with their associated dynamics and pitch contours. Choosing units of such length allows to capture a note attack and release contours with the previous and following notes contexts. Depending on the target song, a different sequence of units may be retrieved to be transformed and overlapped to generate the expression contours. At synthesis, the engine uses these expression contours to control the samples from the voice database.

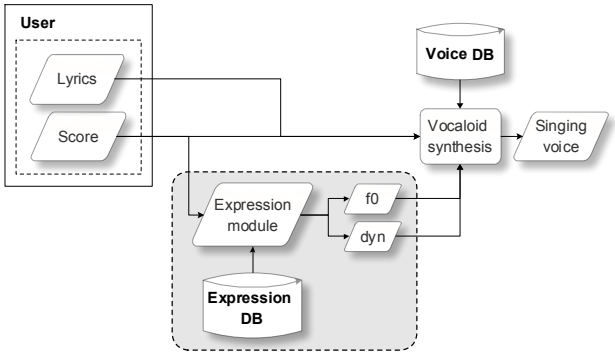

Figure 1. System interaction with the synthesis engine.

The following subsections explain more in detail how the database is created, the optimal sequence of units is selected, the units are transformed (to match target pitch and duration) and concatenated.

2.2 Expression database creation

2.2.1 Recordings

The ideal expression database would need to cover the complete sonic space in terms of possible note durations and pitch intervals. If this was the case, units would require no (or little) transformation to match the target score. Thus, in the proposed approach, just a few songs were recorded and labeled to get an initial idea of the system's performance. Details on the recorded database are given in section 3.1.

The generated expression contours must control the synthesizer for any target lyrics. Therefore, it is important that the expression contours in the database do not contain unvoiced sections or micro-prosody due to phonetics. That is to say, in order to avoid fluctuations in the recorded pitch and dynamics due to phonetics (not attributable to expression), the lyrics of the recorded songs were modified to only vowels. Thus, any sequence of notes was sung as /ua-i-a-i/, where /a/ and /i/ vowels are alternated at every note change and /ua/ diphthong is used to attack a note from a silence.

2.2.2 Labeling

In the current approach, the recorded songs were labeled in a semiautomatic procedure. The information needed to represent units are the song pitch and dynamics contours, note values and timing as well as vibrato parameters.

Pitch is estimated based on the spectral amplitude correlation (SAC) algorithm described in [12]. In terms of dynamics, the extracted energy sample values are normalized and smoothed using a sliding window of 0.5 seconds. This is to keep the tendency of dynamics instead of the energy at frame level.

The segmentation of the recordings provided the note pitch and timing information. Since recordings were done with the modified lyrics, this task is easier than by score following or detecting pitch changes. Given that notes and vowel changes are strictly related, note segmentation is equivalent to vowel change detection. GMM models were trained for clustering and regression. The data used for training were MFCCs extracted from sustained vowel recordings. The outcome of the segmentation was manually checked.

Note to note transition times are needed to preserve note transition shape during transformation. These are estimated as the time instants when pitch deviates a threshold from the labeled note pitch. The threshold is set to 10% of interval (with a minimum of a quarter semitone).

The vibrato parameters allow resynthesis keeping the shape of the original vibrato at any note pitch and duration. The extracted parameters are depth, rate, baseline pitch and reconstruction error. The estimation of these parameters is semiautomatic, where the first step is to manually indicate the first and last peak or valley for each vibrato. Although the way these parameters are estimated is out of the scope of this paper, their relationship to the reconstructed pitch contour with vibrato $\tilde{F0}(n)$ is:

$$\tilde{F0}(n) = \bar{F0}(n) + d(n)sin(\varphi(n) + \varphi_{sign}) \quad (1)$$

$$\varphi(n) = \sum_{k=0}^{n-1} 2\pi r(k)\Delta_t + \varphi_{correc}(n) \quad (2)$$

where, in equation 1, $\bar{F0}(n)$ is the estimated baseline pitch (no vibrato) at frame n, φ_{sign} is a constant value that indicates whether the sinusoid's initial phase is 0 or π, $d(n)$ is the pitch deviation (depth) with respect to the baseline, and $\varphi(n)$ is the sinusoid phase. In equation 2, $r(k)$ is the vibrato rate at frame k, Δ_t is the frame shift time and $\varphi_{correc}(n)$ is the reconstruction error.

In figure 2, we show an example of vibrato parameters extraction and resynthesis. The top most subfigure represents the original pitch, its resynthesis and the baseline estimated parameters are plot. In the other three subfigures, depth, rate and reconstruction error are shown respectively.

2.3 Unit selection

Unit selection aims to retrieve short melodic contexts from the expression database that, ideally, match the target contexts or units. Since perfect matches are unlikely, this step retrieves the optimal sequence of units according to a cost function.

The Viterbi algorithm is used to select the set of units that minimize the cost value. This value is the combination of different cost measures. The aim of these costs is that the sequence of units is the least transformed as possible, with units easy to overlap. These costs also consider the introduction of score variations as well as the selection of consecutive units from the database.

2.3.1 Transformation cost

The transformation cost measures how much a source unit u_i has to be modified to match a target unit t_i. It can be

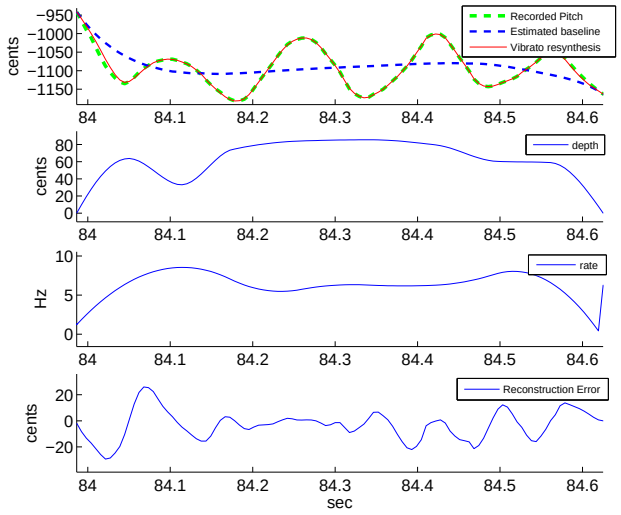

Figure 2. Vibrato resynthesis and parameters: depth, rate, reconstruction error and baseline pitch.

expressed in terms of the mean of two sub-cost functions (amount of pitch shift ps and time stretch ts) as in equation 3:

$$C^t(t_i, u_i) = \frac{1}{2}\left(C^t_{ts}(t_i, u_i) + C^t_{ps}(t_i, u_i)\right) \quad (3)$$

Both subcosts functions are a weighted sum of note durations *dur* ratios (or unit interval pitch values *int*) between source and target units. The C^t_{ts} cost computation is shown in equation 4:

$$C^t_{ts}(t_i, u_i) = \sum_{n=1}^{3} \omega_{ts}(n) log2\left(\frac{dur(u_i(n))}{dur(t_i(n))}\right) \quad (4)$$

where n is the note index within the unit, and time stretch weights ω_{ts} give more relevance to the central unit note transformation:

$$\omega_{ts} = [0.75,\ 1.5,\ 0.75] \quad (5)$$

The C^t_{ps} cost computation is shown in equation 6:

$$C^t_{ps}(t_i, u_i) = \sum_{n=1}^{2} \omega_{ps}(n) log2\left(\frac{int(u_i(n))}{int(t_i(n))}\right) \quad (6)$$

where n points to the two pitch intervals, and pitch shift weights ω_{ps} give the same importance to both intervals,

$$\omega_{ps} = [1,\ 1] \quad (7)$$

Besides, an extra rule is applied to avoid selecting some source units. We have assumed that an ascending interval should not be used to generate a descending interval (and vice-versa). Also, silences must be present in the same note in the source and target units, otherwise that unit should not be selected. If this requirements are not met, the transformation cost is set to infinity.

2.3.2 *Concatenation cost*

The concatenation cost measures how appropriate two units are for overlapping. Consecutive units in the selected sequence share two notes, and crossfading has to be applied to obtain smooth transitions. If the source units are also consecutive in the expression db, the cost is zero. Otherwise, the cost is measured based on the transition to the central unit notes. For this computation, transitions start and end times are used.

2.3.3 *Alternative score cost*

It measures the possibility to introduce some variations to the original score by erasing or adding silences. Using these variations may offer the possibility to select a sequence of units at a lower cost.

In case of erasing a silence, its length determines the cost. On the other hand, the added silences between notes are very short and not actually synthesized due to synthesizer constraints. Therefore, this variation is not penalized.

2.3.4 *Continuity cost*

With the three costs used up to this point, it is likely that units are selected from very different songs and contexts. However, the more different the contexts are, the higher impact it has on the resulting contour. At a very local context, this is managed by the concatenation cost, although it only takes into account whether two candidate units are consecutive in the database or not. A higher scope of concatenation is managed by the continuity cost, towards the musical concept of phrasing.

Continuity cost is included to favor the selection of a certain amount L of consecutive source units. Thus, more similar contexts and easy to concatenate (already done by the original singer) can be selected. The starting point is set to a silence or from a point where two units are not consecutive in the database. While L consecutive units are not chosen, selecting non-consecutive units is penalized. When L is reached, a new starting point is set.

2.4 Unit transformation

This step deals with the transformation of the selected sequence of units. Source notes have to match target notes in pitch and duration. Source unit dynamics contour is also stretched according to the target unit duration.

Figure 3 shows the basic idea for the expression contours generation. A target sequence of four notes (bottom image), can be generated by overlapping a couple of source units (A and B) which share two notes. The target pitch contour (pink dashed line) is generated by transforming them in time (according to the target note durations) and frequency (target note pitches). After transformation, crossfading is applied between the pitch contours. Vibratos appearing in the source units are also rendered, preserving the original depth and rate and spanning over the target note duration.

Unit pitch contour is transformed by adding an offset value per note. This offset is the difference between target and source notes. Offset values during note transitions are interpolated linearly in order to have smooth changes.

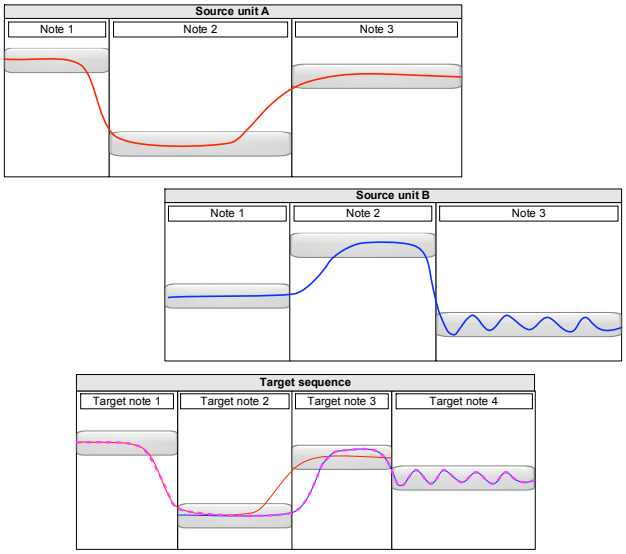

Figure 3. Two overlapping source units (top plots), transformed contours (bottom, solid lines) and concatenated contours (bottom, dashed line)

Unit contours (pitch, dynamics and vibrato parameters) are time stretched in a non-linear way. Most of the transformation is applied during the note sustain in order to preserve note transition shapes. The ratio between source and target note durations determines the amount of transformation.

2.5 Unit concatenation

The expression contours were finally rendered in three steps. First, the basic contours are generated, then the baseline pitch is tuned and finally vibratos are rendered.

2.5.1 *Basic contours*

The overlapping step of the transformed pitch, dynamics and vibrato parameter contours was handled with a crossfading mask. This mask was computed per unit in order to determine the samples that contributed to the output contour. More relevance is given to the attack to the central unit note and its sustain, until next unit central note attack time based on note transition start and end times.

2.5.2 *Baseline pitch tuning*

In order to ensure that sustains were at the right target pitch, the baseline pitch was tuned. A similar process to auto-tuning techniques was followed before rendering the final pitch contour.

This step consists on adding a correction offset to each pitch frame value. First, a sliding window is used to compute local pitch average values through each note duration. The deviation of each frame average value with respect to the target note pitch is weighted in order to get the correction offset. Given the shape of the applied weights (tukey window), boundary note frames are less modified than middle note frames.

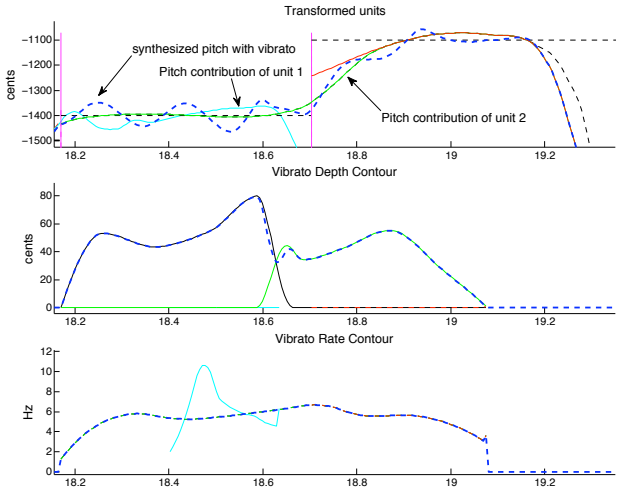

Figure 4. Transformed unit pitches and vibrato control contours concatenation.

2.5.3 Vibrato synthesis

Vibratos were synthesized using the depth, rate and reconstruction error contours generated for the target song. Those frames with depth equal to zero contained no vibrato. Otherwise, the procedure introduced in 2.2.2 was followed for synthesis.

An example of the result is shown in figure 4 (dashed line), with most frames belonging to a vibrato segment. The contributing units contours are represented in continuous lines. The top-most subfigure shows the pitch values of the transformed source units and the resulting pitch with vibrato. This vibrato was synthesized with the depth shown in the second sub-figure, where the two contributing units can also be observed. The vibrato rate is shown in the bottom subfigure.

3. EVALUATION

3.1 Experimental setup

We evaluated the achieved expression by conducting a MOS type test with 16 participants. The subjects rated the synthesized performances from 1-5 in terms of naturalness (rather synthetic or human), expressiveness (very inexpressive to very expressive) and singer skills (very bad or good timing, tuning, overall perception).

Three excerpts of 30 seconds were synthesized. For each of these excerpt, three versions were synthesized using different methods of generating expression contours. These were the baseline method based on heuristic rules, manual tuning of dynamics, pitch bend and vibratos, and finally the synthesis using the proposed system. All versions had background music.

The heuristic rules or default configuration was obtained following the algorithm described in [2]. Pitch and dynamics curves are obtained from the interpolation of a set of points generated by normal distributions (derived from real arpeggio performances). The amount of points used in the interpolation depends on the absolute note duration.

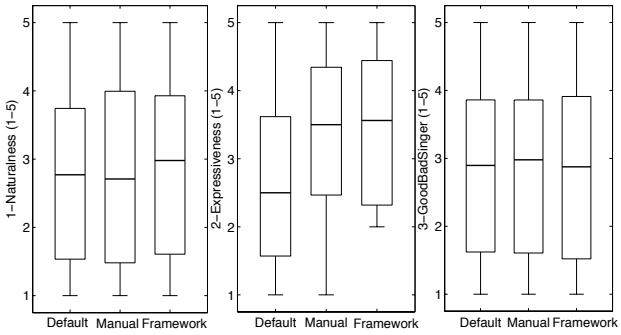

Figure 5. Results of listening tests.

With respect to the manual tuning, it was done by skilled experts who are used to generate singing performances with Vocaloid.

The expression database built for this evaluation contained melodic sections from four recorded songs in soul/pop style. In total, six minutes of a cappella singing voice were recorded by a female trained singer. The target songs were not present in this database.

The subjects first listened the three versions of the song being rated to get an overview of the variability within examples and then listened to them again in order to rate them individually. This was done separately for each song. The order in which songs were listened to was not always the same and versions were presented in a random order. These songs were synthesized using a Spanish voice bank. The rating task took around 15 minutes.

3.2 Results and discussion

In order to evaluate how the three different versions compare to each other, the results are grouped in terms of the control parameter configurations within each rated question. These are shown in figure 5, where the boxplots refer to naturalness, expressiveness and singer skills respectively. The statistics show the mean opinion scores, standard deviations (above and under mean) and minimums and maximums. Paired-samples t-tests were conducted to determine the statistical differences between the evaluated synthesis configurations with respect to a p-value threshold of 0.05.

Concerning naturalness, the three versions have been rated quite similarly. Although the proposed system has a slightly higher mean value, this difference is not statistically significant with respect to the baseline method and the manual tuning.

In terms of expressiveness, it can be observed that the baseline method has the lowest mean rating, followed by the manually tuned version which is slightly improved by our method. In this case, the differences between both the proposed system and the manual configuration with respect to the baseline method are statistically significant ($p=2.64\times10^{-6}$ and $p=3.23\times10^{-6}$, respectively). On the other hand, no statistically significant difference is observed between the proposed system and the manual configuration ($p=0.76$). Therefore, expression was improved using the proposed system and achieved a similar level to the man-

ual configuration.

Finally, with respect to whether the singer is good or bad, the three versions have a similar mean value. The differences between both the proposed system and the manual configuration with respect to the baseline method are not statistically significant.

The sound files used in the listening tests are online at: `http://www.dtic.upf.edu/~mumbert/smac2013/`.

4. CONCLUSIONS

In this paper we have introduced a new method for generating expression contours for singing voice synthesis based on unit selection. It is worth mentioning that our system does not rely on statistical models and therefore it is capable of preserving the fine details of the recorded expression. First, the steps for the expression database recordings and labeling have been detailed. With respect unit selection process, the four costs that are taken into account have been explained. These costs involve unit transformation and concatenation, alternative score generation and continuity cost. Unit transformation in time and frequency, unit concatenation with the crossfading masks, and contours rendering have been described.

From the listening tests, our system is capable to automatically generate a performance which is as expressive and natural sounding as can be achieved by manual tuning of parameters. Also, its naturalness and perceived singer skills are not worse than the baseline rule-based system.

Automatic generation of expression controls for a given target style has several advantages. It contributes to reducing the time a user spends in providing expression to singing performance. Another advantage is that it provides a richer starting point than the default configuration for manual expression tuning. More importantly, the proposed system paves the way towards modeling all of the aspects of expression for a singer in a particular style.

Concerning the future work, the expression database can be improved by realizing a comprehensive study of the note durations and intervals to cover in a given style. The cost functions can be adapted to new labeled data (type of note figures and strength, timing deviations, lyrics). The tremolo effect could be considered by modeling dynamics in a similar way as vibrato instead of the current smoothing step. We also plan to designing objective evaluation tests for unit selection, transformation and concatenation.

5. REFERENCES

[1] H. Kenmochi and H. Ohshita, "Vocaloid - commercial singing synthesizer based on sample concatenation." in *INTERSPEECH*, 2007, pp. 4009–4010.

[2] J. Bonada, "Voice processing and synthesis by performance sampling and spectral models," Ph.D. dissertation, Universitat Pompeu Fabra, Barcelona, 2008.

[3] J. Janer, J. Bonada, and M. Blaauw, "Performance-driven control for sample-based singing voice synthesis," in *Proc. of the Int. Conf. on Digital Audio Effects (DAFx-06)*, Sept. 18–20, 2006, pp. 41–44.

[4] M. Schröder, "Can emotions be synthesized without controlling voice quality," *Phonus 4, Forschungsbericht Institut f ür Phonetik, Universität des Saarlandes*, 1999.

[5] T. Nakano and M. Goto, "Vocalistener: A singing-to-singing synthesis system based on iterative parameter estimation," in *Proceedings of the 6th Sound and Music Computing Conference. Porto*, 2009.

[6] H. Zen, K. Tokuda, and A. W. Black, "Statistical parametric speech synthesis," *Speech Communication*, vol. 51, no. 11, pp. 1039–1064, 2009.

[7] T. Nose and T. Kobayashi, "Recent development of HMM-based expressive speech synthesis and its applications," in *Proc. 2011 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC 2011)*, 2011.

[8] H. Zen, T. Nose, J. Yamagishi, S. Sako, T. Masuko, A. W. Black, and K. Tokuda, "The HMM-based speech synthesis system (HTS) version 2.0," in *Proc. of Sixth ISCA Workshop on Speech Synthesis*, 2007, pp. 294–299.

[9] K. Saino, M. Tachibana, and H. Kenmochi, "A singing style modeling system for singing voice synthesizers." in *INTERSPEECH*. ISCA, 2010, pp. 2894–2897.

[10] A. Iida, N. Campbell, S. Iga, F. Higuchi, and M. Yasumura, "A speech synthesis system with emotion for assisting communication," in *ISCA Tutorial and Research Workshop (ITRW) on Speech and Emotion*, 2000.

[11] D. Erro, E. Navas, I. Hernáez, and I. Saratxaga, "Emotion conversion based on prosodic unit selection," *Audio, Speech, and Language Processing, IEEE Transactions on*, vol. 18, no. 5, pp. 974–983, 2010.

[12] E. Gómez and J. Bonada, "Towards computer-assisted flamenco transcription: An experimental comparison of automatic transcription algorithms as applied to a cappella singing," *Computer Music Journal*, In Press.