A multi-layer F0 model for singing voice synthesis using a B-spline representation with intuitive controls

Luc Ardaillon, Gilles Degottex, Axel Roebel

IRCAM - UMR STMS (IRCAM - CNRS - Sorbonne Universités UPMC Paris 06)
Paris, France

luc.ardaillon@ircam.fr, gilles.degottex@ircam.fr, axel.roebel@ircam.fr

Abstract

In singing voice, the fundamental frequency (F0) carries not only melody, but also music style, personal expressivity and other characteristics specific to voice production mechanism. The F0 modelling is therefore critical for a natural-sounding and expressive synthesis. In addition, for artistic purposes, composers also need to have control over expressive parameters of the F0 curve, which is missing in many current approaches. This paper presents a novel parametric F0 model for singing voice synthesis with intuitive control of expressive parameters. The proposed approach considers the various F0 variations of the singing voice as separate layers using B-splines to model the melodic component. This model has been implemented in a concatenative singing voice synthesis system and its perceived naturalness has been evaluated through listening tests. The validity of each layer is first evaluated independently, and the full model is then compared to real F0 curves from professional singers. The results of these tests suggest that the model is suitable to produce natural and expressive F0 contours.

Index Terms: singing voice synthesis, F0 model, concatenative synthesis

1. Introduction

Interest for singing voice synthesis has considerably grown these last few years, popularized by products like Vocaloïd [1]. Althought the aestethical qualities of current synthesis systems make it already suitable for use in music production, some more efforts are required to achieve voice quality comparable to that of a professional singer. Concatenation-based methods are well-known for generating high-quality speech and have for some years been widely used for singing voice synthesis [1, 2, 3]. While HMM-based systems like [4] are more flexible in terms of speaker identity or singing style, using model adaptation [5], their quality is still limited by current vocoding techniques. Conversely, concatenative systems, with a minimum of transformations, lead to high-quality synthesis. Besides the synthesis method itself, a major problem for singing voice is to generate appropriate control parameters. The main question is then : How, with a simple score and a few expressive controls as input, can we generate parameters suitable for synthesizing natural and expressive singing voice? Althougth phoneme lengths, energy, and timbre should be considered, fundamental frequency is especially important as it conveys not only the melody, but also expressivity, style, and some mechanical characteristics. F0 models should thus have the ability to generate natural contours, and also to allow flexible and intuitive control of expressivity to meet a particular style or musical idea of the composer.

Several methods have already been developed for generating F0 curves for singing voice, among which HMM [6], 2nd order linear systems [7], and unit-selection based models [8]. Althought these methods may be appropriate to synthesize natural F0 curves, they don't provide means for the composer to edit the curve locally and easily modify the expressivity. For this purpose, it would be usefull to offer meaningful parameters related to expressive F0 variations like preparation, overshoot, or vibrato [7]. The 2nd order linear system-based method in [7] is parametric. However, eventhough the model parameters are physically meaningful, they are not from a composer point of view. For characterization of observed curves without synthesis purpose, Bezier curves have been used in [9]. Examples like [10], using B-splines to model speech F0 contours, suggest that it may provide an appropriate tool for generating smooth and expressive curves for singing voice synthesis. A particular advantage of B-splines compared to Bezier curves is to offer a better local control of the curve shape. While B-splines have been widely used in computer graphics, very few applications can be found in the field of sound processing. Multi-layer additive approaches to F0 modelling have been used in speech synthesis [11] as a mean to model independently different components of voice prosody. Althought there are strong differences between speech and singing, we can similarly split the F0 variations in singing in separate layers, such as vibrato, jitter, or melodic components.

In this paper, we propose to use B-splines in order to build a melodic component model with meaningful parameters, on top of which add vibrato, jitter and phonetic layers to fully represent all F0 variations of singing voice. The proposed parameters are directly related to expressive fluctuations of the F0 in terms of duration and frequency, and can thus be easily manipulated for reshaping the curve. Therefore, in this study, we only evaluate the naturalness of the model, assuming that its controllability is ensured by the proposed approach.

In a first section, a quick overview of our singing voice synthesis system and of the used databases is presented. Then, our F0 model will be detailed, before evaluating its perceived naturalness by means of a listening test, the results of which will be presented in the fourth section.

2. System overview

For the purpose of our research, a concatenative singing voice synthesis system has been built, which uses SuperVP [12, 13] with the SHIP algorithm [14] as a processing engine for high-quality voice transformations. We first present here the singing databases we use for the synthesis, and then address some issues due do the concatenation process to avoid phase and spectral envelope discontinuities.

2.1. The singing database[1]

Compared to speech, the possible variations in singing, in terms of pitch, loudness, and timbre, cover a much wider range of possibilities. It is thus impossible to capture all this variability with recordings, and we have to rely on transformation techniques in order to cover a range as wide as possible with a limited database. In order to synthesize any possible lyrics, the minimum requirements for our system's database is to cover all the diphones of the French language (about 1200). A set of 900 real words has been chosen for ensuring this coverage. Those words are sung on a single pitch with constant intensity.

In order to define the units to be concatenated from the raw recordings, a segmentation step is necessary. Two kinds of segmentations are used in our system :

- A phonemes segmentation.
- A "stable parts" segmentation, which defines the parts of the sound where its spectral caracteristics are almost constant.

This segmentation allows the system to adjust the length of the selected units according to the target notes lengths, while preserving the co-articulation parts [15]. These segmentations are done automatically, with manual post-correction. From these two segmentations, we can then select in the database the samples to be concatenated and define the time stretching curve for generating the desired phoneme durations. Only diphones are used for now, but this segmentation strategy also allows the use of longer units that may be considered for later improvements.

2.2. Phase interpolation at junctions

During the transformation step, the use of the SHIP algorithm [14] ensures high-quality voice transformations by minimizing phasiness effects. This algorithm is well adapted for continuous sounds, but in the case of units concatenation, vertical phase alignment discontinuities may occur at junctions, which can lead to audible distortions. The solution we adopted for solving this problem is to continuously spread these phase discontinuities along an undetermined number of frames for each harmonic, as summerized by the following equations :

$$\Delta\varphi = \varphi_R^i - \varphi_L^i \tag{1}$$

$$while \mid \Delta\varphi \mid > \Delta\varphi_{max} \Rightarrow \varphi_R^i = \varphi_L^i + \frac{\Delta\varphi}{\mid \Delta\varphi \mid} \cdot \Delta\varphi_{max} \tag{2}$$

where φ_L^i and φ_R^i denote the phase of the i^{th} harmonic at the point where the phase of the fundamental is 0, for the left (L) and right (R) units respectively; and $\Delta\varphi_{max}$ is a maximum phase gap chosen empirically such that, below this threshold, the discontinuity is not audible.

Unlike the solution presented in [2], this phase correction is independant from the pitch-shifting factor applied, and is only applied if the discontinuity is big enough ($> \Delta\varphi_{max}$).

2.3. Envelope interpolation at junctions

At junctions, spectral envelopes are never equal, eventhough the concatenated phonemes are the same. In order to avoid these discontinuities, we simply linearly interpolate spectral envelopes on the stable parts around the junction points, with a minimal interpolation duration of 0.06s.

[1]The creation of our databases has been established in collaboration with our project partner Acapela Group

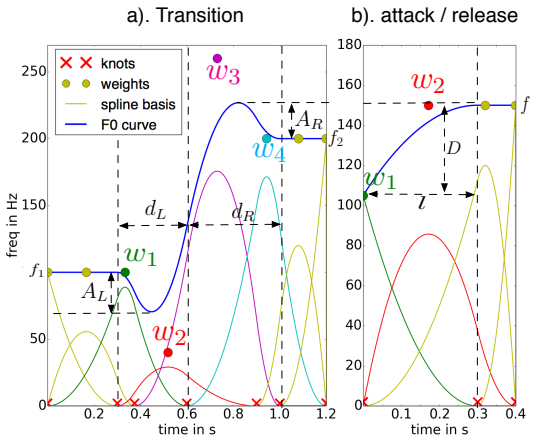

Figure 1: *transition and attack/release models*

3. F0 Modeling

The system presented hereabove is already capable of synthesizing high-quality singing voice, which gives us a good basis to work on more specific problems. This section presents the F0 model we developed for driving the synthesis.

3.1. Model overview

In this paper, we consider the following F0 components :

- a melodic component, comprising the sustained notes, attack and release parts, and transitions between notes
- the vibrato, that occurs during sutained notes
- a phonetic component, representing the F0 inflexions induced by the prononciation of voiced consonants [16]
- the jitter, that corresponds to random uncontrolled variations of the F0 such as described in [17]

the first two being related to expressivity and style, and the former two being due to uncontrolled behaviors. We thus model the F0 as an additive curve resulting from these 4 components.

In addition to this "vertical" decomposition in multiple layers, we also defined an "horizontal" decomposition to model the evolution of the F0 across time according to the input score. From this temporal point of view, we model the F0 curve as a succession of 5 basic types of segments : attacks, sustains, transitions, releases, and silences, in a similar way to [18]. This temporal segmentation applies to the melodic component which models each of those segments in a parametric way using B-splines, such as described in the following section.

3.2. Melodic component model

Some preceding experiments [7] have pointed out the importance of F0 variations such as preparation and overshoot in singing. Thus, we built a transition model that allows such fluctuations, as described in Figure 1 a). Transitions are split in 2 parts around the center, which can be delayed, with a parameter Δt, relatively to the note onset. The length of the left (L) and right (R) parts are determined by the parameters d_L and d_R respectively. The amplitudes of the preparation and overshoot are deduced from the parameters A_L and A_R.

From these parameters, B-splines are used in order to generate the transition curve. For this purpose, 5 knots are positionned as described in figure 1 a). The 1^{st}, 3^{rd} and 5^{th} knots

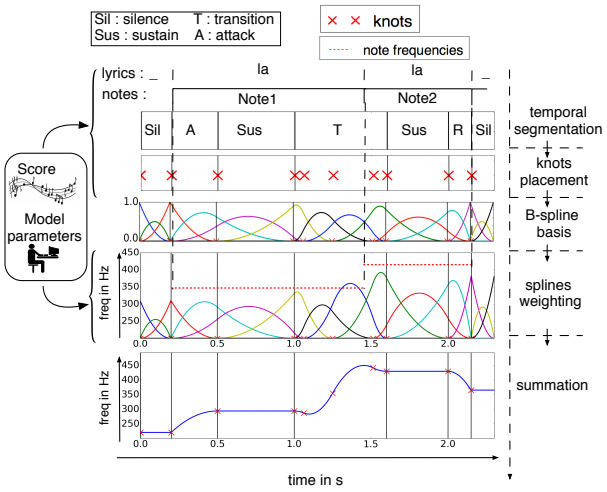

Figure 2: *melodic component model*

are placed at the start, middle and end times of the transition. The 2^{nd} and 4^{th} knots are placed at $0.75 \cdot d_L$ and $0.75 \cdot d_R$ from the middle knot. From these knots positions, we generate a set of 3^{rd} order B-spline basis vectors, which are then weighted in order to shape the transition. The B-splines and their associated weight values are illustrated in figure 1 a). The weights values are determined as follows, from the model parameters :

$$w_1 = f1 \quad ; \quad w_2 = f1 - \triangle f \cdot A_L$$

$$w_4 = f2 \quad ; \quad w_3 = f2 + \triangle f \cdot A_R$$

where $f1$ is the target frequency of the left-side note, $f2$ is the target frequency of the right-side note, and $\triangle f = f2 - f1$. The curve is finally generated by summing all the weighted spline vectors along the time axis.

With this method, one could then easily imagine a simple user interface where the transition could be shaped by moving only 3 control points in a 2D time-frequency space : one to control the transition center (Δt), one 2D handle for (d_L,A_L), and another 2D handle for (d_R, A_R).

Attacks and releases are defined in a similar way. Like in [19], they are characterized only by their length l and depth D, as shown in figure 1 b). Knots are placed at the start and end times of attacks, and the weights of the spline vectors are :

$$w_1 = f - D \cdot f \quad ; \quad w_2 = f$$

where f is the target frequency of the attacked note, and D is defined as a percentage. Releases are modelled symmetrically to attacks. The user interface would require only one 2D handle to control (l,D).

Figure 2 shows a detailed exemple of generating the melodic component from a given score. First, a temporal segmentation is built from the note durations and the temporal parameters (d_L, d_R, Δt , and l). An attack is set after a silence, a release before a silence, and a transition between each pair of notes. From this temporal description, we can then place the knots (as explained hereabove), generate the splines, and weight them according to note frequencies and the amplitude parameters (A_L, A_R and D). The use of 3^{rd} order B-splines automatically ensures a C^1 continuity of the generated curve.

3.3. Vibrato

Vibrato in singing voice has been widely studied [20, 21, 22], mostly from an analysis point of view. Some phenomenons such as an increase of the vibrato rate on the last cycles [20] have been reported. While some authors have seeked to precisely characterize the vibrato shape, rate and amplitude [16, 19], the necessity of such a precise description of the vibrato for synthesis purpose, from a perceptual point of view, has not been attested. [21] suggests that a good vibrato is nearly sinusoïdal and that changes in its shape along time is not perveived by listeners. These assumptions thus encouraged us to use in our system a very simple vibrato model, consisting in a sinusoid with fixed extent (amplitude) and rate (frequency), the amplitude being scaled by an ASR (Attack-Sustain-Release) curve, similar to the one presented in [23] for violin vibrato. An offset parameter can also be used to shift the starting time of the vibrato segment respectively to the start of the sustain part.

3.4. Phonetic component

The pronunciation of voiced consonants induces some inflexions in the observed F0 contours [16]. As these inflexions are inherent to the pronunciation of those phonemes, they are not controlled by the singer. Thus, we decided to treat this component using an F0 profile template for each voiced consonant. We analyzed the F0 profiles of all occurences of each voiced consonant in our singer database, and computed median templates. As the inflexions are not always strictly constrained inside the limits of the consonant, the limits of the profiles are considered from half the consonant's length before its begining to half its length after the end of the consonant. All extracted profiles are normalized in time and frequency before computing the median template. The template is then scaled during synthesis to the target length of the consonant and the target frequency given by the melodic component.

3.5. Jitter

Similar to the phonetic component, we assume jitter to be related to the naturalness of a singing voice and not controlled by the singer. We thus do not need to parametrize it and we also use a template-based approach to model jitter. For this purpose, we analyzed the F0 curves of multiple sustained notes without vibrato contained in our singer database. We normalized them in frequency according to the median frequency of the segment, and stored them as jitter templates (without context), in a similar way to [15]. For the synthesis, we then concatenate randomly chosen templates (with 200ms cross-fades at junction) for the whole length of the synthesized extract and scale the resulting curve according to the frequency given by the melodic component.

4. Results and evaluation

In order to evaluate the naturalness of the curves generated by the suggested model, we conducted a 3 parts listening test. First, we evaluate the relevance of the different layers, and we then confront the complete model to real F0 curves. The test was conducted on 46 participants listening with headphones or earphones, through a web interface, using a CMOS preference test to compare pairs of synthesized sound files, as described in [24]. For all parts of the test, in order to avoid any bias due to other parameters in the evaluation, only the F0 curve was different between the 2 synthesized files of a pair, all other sound characteristics (durations, spectral envelopes, ...) remaining the same. For each test, similar examples were synthesized using both a

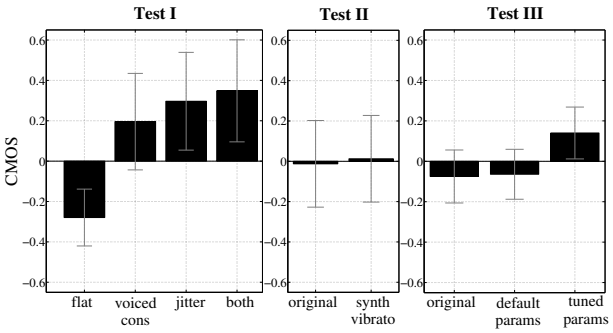

Figure 3: *results of listening test*

man and a woman voice, using the only 2 databases available at the time of this study. Nevertheless, the lack of voice variability does not seem critical here, since we do not aim at evaluating the overall quality of the synthesis, or timbre characteristics, but only the F0 model. The details for each part of the test are given below, and the results are shown in figure 3, with the confidence interval of 95%. All sounds used in the test can be found in the web page [25].

4.1. Test I : jitter and phonetic component

The first test was aiming at evaluating the usefulness of modelling both jitter and phonetic components to improve the naturalness of the synthesized voice. For this purpose, 2 very simple examples with long sustained notes were generated for each voice. One of the examples consisted of a non-sense sentence comprising only vowels and voiced consonants sung on a single note; the 2nd example was an actual french sentence sung on a very simple melody with no expression [25]. The pitch was adapted to each database's mean frequency.

For each example, a flat version (i.e. with fixed F0) was compaired to the same example with only jitter, only phonetic component, or both layers.

From the results showed in figure 3, we can conclude that adding the jitter and phonetic components to the curve improves the perceived naturalness of the voice, at least in the case of sustained notes without vibrato.

4.2. Test II : vibrato model

In this second test, we aimed at evaluating the relevance of our simple vibrato model, using a fixed rate and an ASR extent curve. We thus analyzed F0 contours of 4 extracts with various singing styles (Bizet, Yves Montand, Céline Dion, and Zaz) sung by the 2 same singers who recorded the databases. We then marked the limits of each vibrato parts and flattened manually the F0 curve through the vibrato cycles using Audiosculpt [26, 27]. Then, we generated new vibratos for each of the flattened notes adjusting manually the model parameters (rate, extent, attack and release durations, and initial phase when required) trying to match the original vibrato with the model, and we added these new vibratos on the flattened curve. We then resynthesized each of the analyzed examples with both original curve and modified curve with synthetic vibrato. The subjects were asked to focus on the vibrato sections for this test. The vibrato segments in those examples were from 0.14 (1 cycle) to 2.17 (12 cycles) seconds long. Based on figure 3, the listeners showed no preference between the original vibrato and our model (highly overlapping confidence intervals). Thus, we plan to keep this simple model for our forthcoming works.

4.3. Test III : complete f0 model

In the last section of our listening test, we evaluated the potential of the complete model (all layers) for natural singing voice synthesis, by confronting generated curves to real ones. For this purpose, we analyzed and carefully corrected the F0 curves of 5 extracts of various singing styles sung by both our singers (among which the 4 extracts of test II). In order to apply those F0 curves coherently with the aligned lyrics, we also extracted the phonemes lengths from the original recordings and applied them to the synthesis. For each extract, a score was created specifying the midi notes to be sung. We then generated 2 different F0 curves from the score with our model : the first one using manually chosen default parameters which were the same for all transitions, attacks, releases and vibratos; the second one refining manually the tuning of the parameters, for each transition, attack, release, and vibrato in order to better match the original curve locally. For each example, 3 versions were then synthesized using the 3 curves ("original", "default params", and "tuned params") and each pair of the 3 versions were compaired. In order to keep the test short enough and allow listeners to easily compare the sounds, only short extracts of 4 to 8 seconds were used, and only 2 randomly chosen extracts were selected for each voice and presented to the user.

In figure 3, the results of the test show that the subjects were not able to make a difference between the original curve and both the generated curves (overlapping confidence intervals). Thus, the main conclusion is that the used model seems appropriate to generate natural F0 contours for singing voice synthesis, for various singing styles, assuming that the tuning of the parameters is appropriate. The positive tendency for the "tuned params" version may be explained by the fact that the generated curve, driven by the midi notes in the score, may correct eventual mistuned notes in the original version. The fact that no difference is made between the "original" and the "default params" versions is quite encouraging for the modelling of singing styles with few parameters. However, we may also expect that a difference would be made for longer examples with more variations, as the default parameters wouldn't be able to reproduce the variety of expressions.

5. Conclusion

In this paper, we presented a novel F0 model for singing voice providing a simple parametrization for the control of expressions such as preparation, overshoot and vibrato. This model has been implemented in a concatenative synthesizer. The results of a subjective listening test first showed that adding jitter and voiced-consonants F0 profiles helped improving the perceived naturalness of the synthesized voice. Then it appeared that listeners did not show any preference between natural vibrato and our simple model consisting of a perfect sinusoid scaled by an ASR envelope. Finally, the full model has been confronted to real F0 contours and seem appropriate to generate natural-sounding F0 curves across a variety of musical styles. A next step in our work will be to learn model parameters from recordings, in order to automatically model a specific singing style and apply it to a given score, thus requiring minimal tuning from the user.

6. Acknowledgements

This work is part of the project ChaNTeR, supported by the ANR, and in collaboration with Acapela Group, the Limsi, and Dualo.

7. References

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

[2] J. Bonada, A. Loscos, and H. Kenmochi, "Sample-based singing voice synthesizer by spectral concatenation," in *Proceedings of Stockholm Music Acoustics Conference*. Citeseer, 2003, pp. 1–4.

[3] M. Macon, L. Jensen-Link, E. B. George, J. Oliverio, and M. Clements, "Concatenation-based midi-to-singing voice synthesis," in *Audio Engineering Society Convention 103*. Audio Engineering Society, 1997.

[4] K. Nakamura, K. Oura, Y. Nankaku, and K. Tokuda, "Hmm-based singing voice synthesis and its application to japanese and english," in *Acoustics, Speech and Signal Processing (ICASSP), 2014 IEEE International Conference on*. IEEE, 2014, pp. 265–269.

[5] K. Shirota, K. Nakamura, K. Hashimoto, K. Oura, Y. Nankaku, and K. Tokuda, "Integration of speaker and pitch adaptive training for hmm-based singing voice synthesis," in *Acoustics, Speech and Signal Processing (ICASSP), 2014 IEEE International Conference on*. IEEE, 2014, pp. 2559–2563.

[6] S. W. Lee, S. T. Ang, M. Dong, and H. Li, "Generalized f0 modelling with absolute and relative pitch features for singing voice synthesis," in *Acoustics, Speech and Signal Processing (ICASSP), 2012 IEEE International Conference on*. IEEE, 2012, pp. 429–432.

[7] T. Saitou, M. Unoki, and M. Akagi, "Development of an f0 control model based on f0 dynamic characteristics for singing-voice synthesis," *Speech communication*, vol. 46, no. 3, pp. 405–417, 2005.

[8] M. Umbert, J. Bonada, and M. Blaauw, "Generating singing voice expression contours based on unit selection," in *Proc. SMAC*, 2013.

[9] E. Maestre, J. Bonada, and O. Mayor, "Modeling musical articulation gestures in singing voice performances," in *Proceedings of the AES 121st Convention*, 2006.

[10] D. Lolive, N. Barbot, and O. Boeffard, "Melodic contour estimation with b-spline models using a mdl criterion," in *Proceedings of the 11th International Conference on Speech and Computer (SPECOM)*, 2006, pp. 333–338.

[11] S. Sakai, "Additive modeling of english f0 contour for speech synthesis." in *ICASSP (1)*, 2005, pp. 277–280.

[12] M. Liuni and A. Roebel, "Phase vocoder and beyond," *Musica/Tecnologia*, vol. 7, no. 73–89, 2013, http://www.fupress.net/index.php/mt/article/view/13209.

[13] A. Roebel, "Supervp software," http://anasynth.ircam.fr/home/english/software/supervp, 2015.

[14] ——, "A shape-invariant phase vocoder for speech transformation," in *Proc. Digital Audio Effects (DAFx)*, 2010.

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

[16] K. Saino, M. Tachibana, and H. Kenmochi, "A singing style modeling system for singing voice synthesizers," in *Eleventh Annual Conference of the International Speech Communication Association*, 2010.

[17] R. Stables, J. Bullock, and C. Athwal, *Towards a model for the humanisation of pitch drift in singing voice synthesis*. Ann Arbor, MI: MPublishing, University of Michigan Library, 2011.

[18] O. Mayor, J. Bonada, and A. Loscos, "The singing tutor: Expression categorization and segmentation of the singing voice," in *Proceedings of the AES 121st Convention*. Citeseer, 2006.

[19] Y. Ikemiya, K. Itoyama, and H. G. Okuno, "Transcribing vocal expression from polyphonic music," in *Acoustics, Speech and Signal Processing (ICASSP), 2014 IEEE International Conference on*. IEEE, 2014, pp. 3127–3131.

[20] E. Prame, "Measurements of the vibrato rate of ten singers," *The journal of the Acoustical Society of America*, vol. 96, no. 4, pp. 1979–1984, 1994.

[21] R. Maher and J. Beauchamp, "An investigation of vocal vibrato for synthesis," *Applied Acoustics*, vol. 30, no. 2, pp. 219–245, 1990.

[22] J. Bretos and J. Sundberg, "Measurements of vibrato parameters in long sustained crescendo notes as sung by ten sopranos," *Journal of Voice*, vol. 17, no. 3, pp. 343–352, 2003.

[23] E. Schoonderwaldt and A. Friberg, "Towards a rule-based model for violin vibrato," in *Workshop on Current Research Directions in Computer Music*, 2001, pp. 61–64.

[24] I. Recommendation, "1284-1: General methods for the subjective assessment of sound quality," *International Telecommunications Union, Geneva*, 2003.

[25] A. R. L. Ardaillon, G. Degottex, "demo page : http://recherche.ircam.fr/anasyn/ardaillon/ardaillon2015f0model/."

[26] N. Bogaards, A. Röbel, and X. Rodet, "Sound analysis and processing with audiosculpt 2," in *Proc. Int. Computer Music Conference (ICMC)*, 2004.

[27] C. Picasso, "Audiosculpt software," http://forumnet.ircam.fr/product/audiosculpt/, 2015.