Expression Control of Singing Voice Synthesis: Modeling Pitch and Dynamics with Unit Selection and Statistical Approaches

Martí Umbert Morist

TESI DOCTORAL UPF / 2015

Directors de la tesi:

Dr. Jordi Bonada Sanjaume
Dr. Xavier Serra Casals
Dept. of Information and Communication Technologies
Universitat Pompeu Fabra, Barcelona, Spain

Dissertation submitted to the Deptartment of Information and Communication Technologies of Universitat Pompeu Fabra in partial fulfillment of the requirements for the degree of

DOCTOR PER LA UNIVERSITAT POMPEU FABRA,

Music Technology Group (http://mtg.upf.edu), Dept. of Information and Communication Technologies (http://www.upf.edu/dtic), Universitat Pompeu Fabra (http://www.upf.edu), Barcelona, Spain.

A la meva família,
a la Vicky.

Acknowledgements

I guess there are numerous ways people decide to pursue a PhD. From very vocational researchers to people that have an interest to broaden their knowledge. I guess my place is somewhere in the middle, although I hesitated to pursue the PhD for a long time after finishing my studies. Having lived the academic career initially at the UPC, I finally joined the Sound and Music Computing master at the Music Technology Group (MTG) as a starting point for the PhD. In the following paragraphs I would like to acknowledge those who helped me accomplish such a huge goal. I hope not to leave anyone out!

First and foremost I want to thank my PhD supervisor Jordi Bonada for his guidance and dedication, and to Xavier Serra for giving me the opportunity to join the MTG. This thesis has also been possible thanks to Merlijn Blaauw. Special thanks to Jordi Janer, Oscar Mayor, Ricard Marxer, Graham Coleman, and Sašo Muševič, we shared office and so many coffee breaks.

I would like express appreciation to the Vocaloid team at the YAMAHA Corporate Research and Development Center for giving me the opportunity to do a 5 months research stay and for allowing me to attend the monthly Vocaloid videoconferences during the last years.

Julián Urbano and Perfecto Herrera have been key for a successful design and analysis of the perceptual evaluation. Cárthach Ó Nuanáin has been incredibly helpful to build the evaluation website and so have the participants. I also want to thank the co-authors of my publications Masataka Goto, Tomoyasu Nakano, and Johan Sundberg, as well as the reviewers that provided numerous comments. I would also like to acknowledge Alastair Porter and Sienna Ramos for proofreading parts of this thesis and publications.

I am especially thankful to Emilia Gómez, Mireia Farrús, and Azadeh Faridi, with whom I have shared the teaching side of the academic life. My gratitude extends to Sonia Espí, Cristina Garrido, Alba Rosado, Lydia Garcia, Bea Abad, Vanessa Jiménez, Jana Safrankova, and Joana Clotet for helping me out in so many things. Special thanks go to the university guards, who have seen me around so often that have even learned my name.

José Lastras and Alberto Fernández have been really helpful in some of the publications of this thesis by providing demos and feedback of our proposed systems. I would also like to thank Inma Gomes for providing her voice to record our expression databases.

During my PhD I have shared talks, lunches, advices, and teaching classes with so many colleagues: Giuseppe Bandiera, Dmitry Bogdanov, Juan José Bosch, Mathieu Bosi, Julio Carabias, Rafael Caro, Georgi Dzhambazov, Carles F. Julià, Ángel Faraldo, Andrés Ferraro, Frederic Font, Daniel Gallardo, Sergio

Giraldo, Stanislaw Gorlow, Enric Guaus, Sankalp Gulati, Martín Haro, Sergi Jordà, Gopala Krishna Koduri, Nadine Kroher, Esteban Maestre, Marco Marchini, Agustín Martorell, Sebastián Mealla, Marius Miron, Waldo Nogueira, Sergio Oramas, Panos Papiotis, Alfonso Pérez, Hendrik Purwins, Rafael Ramírez, Gerard Roma, Oriol Romaní, Justin Salamon, Álvaro Sarasúa, Sertan Sentürk, Joan Serrà, Mohamed Sordo, Ajay Srinivasamurthy, Zacharias Vamvakousis, and José R. Zapata.

Last but not least, I would like to thank my friends and family for their continuous support. And most of all, to Vicky for her endless patience and encouragement.

Abstract

Sound synthesis technologies have been applied to speech, instruments, and singing voice. These technologies need to take two aspects into account. On the one hand, the sound representation needs to be as close to the original sound as possible. On the other hand, the control of the sound synthesis should also be able to reproduce the characteristics of the original sound. Thus, we refer about emotional speech synthesis, expressive performances of synthesized instruments, as well as expression in singing voice synthesis. Actually, the singing voice has some commonalities with both speech (the sound source is the same) and instruments (concerning musical aspects like melody and expression resources).

This thesis focuses on the control of a singing voice synthesizer to achieve natural expression similar to a real singer. There are many features that should be controlled to achieve natural expression related to melody, dynamics, rhythm, and timbre. In this thesis we focus on the control of pitch and dynamics with a unit selection-based system and two statistically-based systems. These systems are trained with two possible expression databases that we have designed, recorded, and labeled. We define the basic units from which the databases are built of, which are basically sequences of three notes or rests.

Concerning the unit selection-based system, we define the cost functions for unit selection as well as the unit transformations and concatenation steps. Regarding the statistically-based system, we define the context-dependent information to model both sequences of notes and sequences of note transitions and sustains. The first type of sequences are trained with the absolute pitch values, while the second type of sequences are trained with the pitch fluctuations around a nominal score. A third system is also proposed as a combination of the two previously defined systems.

Modeling singing voice expression is a difficult task, since as humans, we are completely familiarized with the singing voice instrument, and thus we easily detect whether artificially achieved results are similar to a real singer or not. The wide variety of contributing features make achieving natural expression control a complex task. Our perceptual evaluation compares the proposed systems with other systems to see how these relate to each other. The objective evaluation focuses on the algorithms efficiency.

This thesis contributes to the field of expression control of singing voice synthesis: a) it provides a discussion on expression and summarizes some expression definitions, b) it reviews previous works on expression control in singing voice synthesis, c) it provides an online compilation of sound excerpts from different works, d) it proposes a methodology for expression database creation,

e) it implements a unit selection-based system for expression control, f) it proposes a modification on statistical-based systems for expression control, g) it combines the two previous systems on a hybrid system for expression control, h) it compares the proposed systems with other state of the art systems, i) it proposes another use case in which the proposed systems can be applied, j) it provides a set of proposals to improve the evaluation.

Resum

Les tecnologies de síntesi de so s'han aplicat a diversos camps, com a la parla, a instruments musicals, i a la veu cantada. Aquestes tecnologies han de tenir en compte dos aspectes. Per una banda, la representació del so ha de ser el més propera possible a l'original. Per l'altra banda, el control del so sintetitzat ha de poder reproduir les característiques del so original. Així, podem parlar de síntesi expressiva de parla, d'actuacions expressives d'instruments, així com de síntesi expressiva de veu cantada.

Aquesta tesi es centra en el control dels sintetitzadors de veu cantada per aconseguir una expressivitat natural semblant a la d'un cantant real. Hi ha moltes caractarístiques que s'haurien de controlar per aconseguir una expressivitat natural relacionades amb la melodia, la dinàmica, el ritme i el timbre. En aquesta tesi ens centrem en el control de la freqüència fonamental i de la dinàmica amb un sistema basat en selecció d'unitats i dos sistemes estadístics. Aquests sistemes són entrenats amb dues possiblees bases de dades expressives que hem dissenyat, enregistrat i etiquetat. Hem definit les unitats bàsiques a partir de les quals les bases de dades s'han construit i que són bàsicament seqüències de tres notes o silencis.

Pel que fa al sistema de selecció d'unitats, hem definit les funcions de costos per a la selecció d'unitats així com els passos per la transformació i concatenació d'unitats. Respecte als sistemes estadístics, hem definit la informació que depèn dels contextos per modelar tant seqüències de notes com seqüències de transicions i sosteniments. El primer tipus de seqüències són entrenades amb valors absoluts del pitch, mentre que el segon tipus de seqüències són entrenades a partir de les fluctuacions del pitch al voltant de la partitura nominal. Finalment, també presentem un tercer sistema que combina els dos anteriors tipus sistemes.

Modelar l'expressivitat de la veu cantada és una tasca difícil, ja que nosaltres els humans estem totalment familiaritzats amb l'instrument en qüestió, de manera que podem detectar fàcilment si els resultats obtinguts artificialment són similars a un cantant real o no. A més a més, la gran varietat de caractarístiques que hi participen fan del control natural de l'expressivitat una tasca complexa. La nostra avaluació perceptual compara els sistemes proposats amb altres sistemes per tal de veure com els podem relacionar. L'avaluació objectiva es centra en l'eficiència dels sistemes.

Aquesta tesi contribueix en el camp del control de l'expressivitat de la síntesi de veu cantada: a) analitzem la discussió actual sobre l'expressivitat i en resumim algunes de les definicions, b) repassem diversos treballs anteriors en el control de l'expressivitat de la síntesi de la veu cantada, c) presentem un recull

online de sons que mostren els resultats de diversos treballs, d) proposem una metodologia per la creació de bases de dades expressives, e) implementem un sistema basat en selecció d'unitats pel control de l'expressivitat, f) proposem la modificació dels sistemes estadístics pel control de l'expressivitat, g) combinem els dos sistemes anteriors per obtenir un sistema híbrid pel control de l'expressivitat, h) comparem els sistemes proposats amb altres sistemes actuals, i) proposem un altre cas d'ús on aplicar els sistemes proposats, i finalment, j) proporcionem una sèrie de propostes per millorar l'avaluació de sistemes de síntesi de veu cada.

Contents

List of figures

List of tables

CHAPTER 1

Introduction

This chapter aims to provide the context to the research described in the subsequent chapters. First, we explain the reasons that motivate this work on expression control in singing voice synthesis. Based on (Umbert et al., 2015), this context is presented with several systems using singing voice synthesis technologies, showing where these could be applied, and highlighting the importance of expression in such cases. The research carried out at the Music Technology Group is also presented. We explain how the Vocaloid singing voice synthesizer inspired the research that we have carried out. Next, we also provide specific details on the author's own trajectory. Then, expression is defined and put into context in the case of the singing voice. Also, a short excerpt is analyzed in order to illustrate the concept we are studying. Next, expression is related to the singing voice and other fields like speech and music performance. After that, we provide an overview of the proposed systems. Finally, we describe the goals and organization of this dissertation.

1.1 Motivation

1.1.1 Singing voice synthesis systems

Ins recent decades, several applications have shown how singing voice synthesis technologies can be of interest for composers (Cook, 1998; Rodet, 2002). Technologies for the manipulation of voice features (mostly pitch, loudness, and timbre) have been increasingly used to enhance tools for music creation and post-processing, singing live performance, to imitate a singer, and even to generate voices difficult to produce naturally (e.g. castrati). More examples can be found with pedagogical purposes or as tools to check acoustic properties of the voice as a way to identify perceptually relevant voice properties (Sundberg, 2006). These applications of the so-called music information research field may have a great impact on the way we interact with music (Goto, 2012).

Expression control is a particular aspect of such systems that aims to manipulate a set of voice features related to a particular emotion, style, or singer.

Research projects	Website
Cantor	http://www.virsyn.de
Cantor Digitalis	http://www.cantordigitalis.limsi.fr
ChaNTeR	https://chanter.limsi.fr
Flinger	http://www.cslu.ogi.edu/tts/flinger
Lyricos	http://www.cslu.ogi.edu/tts/demos
Orpheus	http://www.orpheus-music.org/v3
Sinsy	http://www.sinsy.jp
Symphonic Choirs	http://www.soundsonline.com/Symphonic-Choirs
VocaListener	https://staff.aist.go.jp/t.nakano/VocaListener
VocaListener2	https://staff.aist.go.jp/t.nakano/VocaListener2
Vocaloid	http://www.vocaloid.com
VocaRefiner	https://staff.aist.go.jp/t.nakano/VocaRefiner
VocaWatcher	https://staff.aist.go.jp/t.nakano/VocaWatcher
Commercial products	**Website**
Melodyne	http://www.celemony.com/
Utau	http://www.utau-synth.com
CeVIO	http://cevio.jp
Sinsy (integrated in Band-in-a-Box)	http://www.pgmusic.com/bbwin.new.htm
VocaListener (product version)	http://www.vocaloid.com/lineup/vocalis
Vocaloid (integrated in Cubase)	http://www.vocaloid.com/lineup/cubase

Table 1.1: Projects using singing voice synthesis technologies.

In the context of singing voice synthesis, these features are generated either automatically or through the user interaction. Also known as performance modeling, expression control has been approached from different perspectives and for different purposes, and different projects have shown a wide extent of applicability.

Examples of research projects and commercial products using singing voice synthesis technologies are listed in Table 1.1. In Figs. 1.1 and 1.2, we show the interfaces of the Sinsy and the Vocaloid[1] synthesizers. In both cases the lyrics of a song are synthesized following the indications of a score which specifies the notes at which each phoneme or syllable has to be reproduced. In the first project, the score is introduced with a MusciXML file [2], and in the second one the user introduces notes and lyrics (either manually via the piano roll or by importing MIDI files). Different technologies are used for voice synthesis and different degrees of interaction with the user may be allowed, from just setting vibrato properties to the possibility of manually tuning a wide set of control parameters in order to generate a voice as natural and expressive as possible.

There are several possible applications one can imagine where the singing voice synthesis technologies could be applied. Concerning music notation software or score writers, like Sibelius[3] or Finale[4] amongst others, they offer the

[1] http://es.vocaloid.wikia.com/wiki/Vocaloid3/
[2] http://www.musicxml.com/
[3] http://www.sibelius.com/
[4] http://www.finalemusic.com/

Figure 1.1: Sinsy interface

Figure 1.2: Vocaloid interface

functionality of reproducing the score that it is being edited, which is usually done with an instrumental sound like a piano. However, singing voice libraries have not been included so far despite the realism that would represent using a voice that sings the edited songs not only in a natural but also in an expressive way. There is one case in this direction, which is the integration of Sinsy into the Band in a Box [5] music arranger software. To our knowledge, the closest attempt in other available software has been to replace the lyrics of the song by a single vowel which is then used to sing the notes of the song. The usage of expressive singing voice synthesis can be applied not only to new created songs, but also to listen to long collections of songs so to have a "previsualization" of how these sound when a recording is not available (for instance, in online score repositories like MuseScore[6]) so that composers can listen to their compositions in a straightforward way.

Beyond singing voice synthesis, expression control can be useful in music production for instance. Imagine that the recorded voice track of a singer could be slightly modified, not only in terms of intonation, but also following a particular singing style, something would require time and a skillfull user if it is done manually. To some extent this is what Melodyne[8] aims to (corrections on intonation, timing, phrasing, and dynamics). However, it could also be envisaged the possibility of improving the singer's expression by providing a modification of the pitch contour for a given note, phrase, or even the whole song by combining the singer specific expression with automatically generated features which improve the original performance. The voice quality would be another relevant aspect to be modified, for instance by changing the voice to sound with a growl effect.

In these applications based on singing voice synthesis technologies we have emphasized the importance of expression, whose control is the topic of this dissertation. Expression appears to highly contribute in the overall quality of

[5] http://www.pgmusic.com/
[6] https://musescore.com or Choral Public Domain Library[7]
[8] http://www.celemony.com/en/melodyne/what-is-melodyne

synthesized sounds (musical, speech, or singing voice) together with the sound quality itself. Expression shapes a musical sound, speech, or singing voice so to convey the message content more effectively. More details on expression are provided in section 1.2. In short, in this dissertation we have envisaged a system that can "mimic" the style of a particular singer so that it can be reproduced when synthesizing any other song by a virtual singer.

1.1.2 Research at the Music Technology Group

This research has been carried out at the Music Technology Group (MTG)[9] of the Universitat Pompeu Fabra (UPF)[10] in Barcelona, founded in 1994 by Dr. Xavier Serra. It is part of the Department of Information and Communication Technologies[11], and focuses its research on sound and music computing. More concretely, around 50 researchers make the MTG a multidisciplinary environment where fields like signal processing, machine learning, semantic technologies, and human computer interaction meet to cover 4 research teams:

- **Music and multimodal interaction lab:** this line of research currently focuses its research on tabletop and tangible interaction. More specifically, it focuses on the study on how these interfaces can favor multi-dimensional and continuous real-time interaction, exploration and multi-user collaboration.

- **Audio signal processing lab:** where their work is focused to develop audio signal processing techniques, and more concretely to model sounds and music by using signal processing methods as well as contextual cultural and social information.

- **Music and machine learning lab:** where their main interest is modeling expression in music performances and the use of emotions in brain-computer (music) interfaces.

- **Music information research lab:** focused in music information retrieval and in voice and audio processing. In the first area of research, the team is involved in the study of descriptors that represent features like rhythm, timbre, tonality, melody, and structure in musical signals. Concerning the second area, the team focuses on the study of singing voice synthesis, voice transformation, audio source separation, music and audio processing, and automatic soundscape generation.

Regarding the voice and audio signal processing team within the Music information research lab, lead by Dr. Jordi Bonada, for more than 15 years

[9] http://www.mtg.upf.edu/
[10] http://www.upf.edu/
[11] http://www.upf.edu/dtic/en/

the MTG has collaborated with Yamaha Corporation[12]. As a result, several projects have been jointly researched and some of them commercialized given the group's focus on technology transfer:

- **Kaleivoicecope[13]:** it is a library with signal processing algorithms that convert and modify the human voice, based in a set of transformations (like vibrato, changes in the fundamental frequency and amplitude, control of the spectral and physical voice characteristics, and timbre modifications) that preserve its natural quality.
- **Elvis[14]:** although it is no longer maintained, this singing voice impersonator project is a voice morphing system is able to transform in real-time (using the Spectral Modeling Synthesis technique) the voice of an amateur singer and make it resemble the voice of a professional singer.
- **Vocaloid:** it is a sample-based singing voice synthesizer (Bonada & Serra, 2007; Kenmochi & Ohshita, 2007), where diphones and triphones are selected from the singer database recordings according to a cost criteria that measures the degree of time and frequency transformations applied to each sample. The selected units are then transformed and concatenated in order to generate the output waveform.

1.1.3 The source of inspiration

The Vocaloid synthesizer has been the main tool used in this dissertation to synthesize singing voice performances. Actually, this tool and its limitations inspired the research carried out in this dissertation. As introduced in section 1.1.1, Vocaloid synthesizes songs according to the lyrics and the notes introduced with the piano roll. In order to achieve a realistic virtual singer performance in terms of naturalness and expressive resources, the user can tune a wide set of control parameters. However, this is a difficult task which requires time and skills to obtain the desired results. Therefore, it becomes desirable a system to automatically tune such control parameters, which can besides represent the style of a particular singer and achieve much better results than done manually. The outcome of such a system can represent a starting configuration which is much richer than the synthesizer's default expression in terms of the expressive resources used by the virtual singer. Therefore, it does not exclude the manual task of fine tuning the control parameters as a last step.

The sample-based system behind the Vocaloid synthesizer inspired our first approach for expression control based on unit selection. In our case, the main difference is that units are not directly voice samples but they correspond to pitch and dynamics contours. The subsequent statistically based approaches

[12] http://www.yamaha.com
[13] http://www.mtg.upf.edu/project/kaleivoicecope
[14] http://mtg.upf.edu/10years-yamaha/demos.htm

are inspired by this first approach, since these keep working with a similar idea of unit.

1.1.4 Personal trajectory

The research presented in this dissertation spans over the last 4 years. Besides the work presented here, I have participated in teaching tasks and supervised several undergraduate and master thesis within the MTG Sound and Music Computing Master (SMC)[15]. In this subsection I provide some details on these tasks and finally comment on my own trajectory prior to the PhD research. Concerning the subjects that I have taught within the UPF Degree in Audiovisual Engineering Systems[16]:

- **Teaching:**
  1. Lab sessions of "Senyals i Sistemes" (Signals and Systems).
  2. Lab sessions of "Processament de la Parla" (Speech Processing).
- **Undergraduate thesis:**
  1. "Síntesis de voz cantada y canto coral: Herramienta de ensayo para integrantes de coros clásicos" (Justel Pizarro, 2014)[17]
  2. "Síntesis de voz cantada y canto coral: criterios musicales y estadísticos" (Iserte Agut, 2014)[18]
  3. "Talking summaries" (L. Díaz, 2015)
- **Master thesis:**
  1. "Expressive speech synthesis for a Radio DJ using Vocaloid and HMM's" (Floría, 2013)[19]
  2. "F0 Modeling For Singing Voice Synthesizers with LSTM Recurrent Neural Networks" (Ozer, S. 2015)

Concerning the personal background, my academic and professional career has been related to speech processing since the my undergraduate thesis in the Technical University of Catalonia (UPC)[20] in 2004 on pitch estimation. Since then, I have been working in speech technologies both at the UPC and in the private sector at Verbio Technologies[21]. In 2010, I obtained the SMC master

[15] http://www.upf.edu/smc/
[16] http://www.upf.edu/esup/en/titulacions/grau-eng_audiovisuals/presentacio/
[17] http://repositori.upf.edu/handle/10230/22897
[18] http://repositori.upf.edu/handle/10230/22885
[19] http://mtg.upf.edu/node/2835
[20] http://telecombcn.upc.edu/en/
[21] http://verbio.com

degree with a thesis entitled "Emotional Speech Synthesis for a Radio DJ: Corpus Design and Expression Modeling" (Umbert et al., 2010). Before starting the PhD research, I made a 5 months research stay with the Vocaloid team at the YAMAHA Corporate Research and Development Center[22] in Hamamatsu, Japan, where I worked on generating the growl effect to the singing voice. During these last years I have also improved my musical skills by learning music theory and by joining a Gospel Choir.

1.2 Expression in music

1.2.1 Definition

Expression is an intuitive aspect of a music performance, but complex to define. In Kirke, Alexis, Miranda (2013), it is viewed as "*the strategies and changes which are not marked in a score but which performers apply to the music*" (p. 2). In Canazza et al. (2004), expression is "*the added value of a performance and is part of the reason that music is interesting to listen to and sounds alive*" (p. 1). A quite complete definition is given in Widmer (2001), relating the liveliness of a score to "*the artist's understanding of the structure and 'meaning' of a piece of music, and his/her (conscious or unconscious) expression of this understanding via expressive performance*" (p. 150).

From a psychological perspective, Juslin defines it as "*a set of perceptual qualities that reflect psychophysical relationships between 'objective' properties of the music, and 'subjective' impressions of the listener*" Juslin (2003) (p. 276). With respect to these objective properties of the music, an extensive summary of acoustic cues for a selection of emotions can be found in Juslin & Laukka (2003). The authors also pose the question of what is the message the performer expresses in a music performance. This is actually analyzed in Gabrielsson & Juslin (1996), where the authors identify the key elements in a performance. These are the composer (with a musical intention containing a certain emotion), the musical score (that encodes that emotion, not present in case of improvisation), one or several performers (who evoke an emotion in a performance that may vary in some aspects compared to the score), the actual sounding music, and the listener (who perceives emotions expressed in the music).

Expression has a key impact on the perceived quality and naturalness. As pointed out by Ternström, "*even a single sine wave can be expressive to some degree if it is expertly controlled in amplitude and frequency*" (Ternström, 2002). Ternström says that musicians care more about instruments being adequately expressive than sounding natural. For instance, in Clara Rockmore's performance of Vocalise by Sergei Vasilyevich Rachmaninoff a skillfully controlled Theremin expresses her intentions to a high degree, despite the limited

[22] http://www.yamaha.com/about_yamaha/research/vocaloid/

degrees of freedom. This audio file and all other sounds mentioned in this thesis have been collected in a single website[23]. The corresponding audio file to the mentioned performance can be found in the *Signal Processing Magazine 2015* section in the website.

1.2.2 Expression control in singing voice

In the case of the singing voice, achieving a realistic sound synthesis implies controlling a wider set of parameters than just amplitude and frequency, as mentioned in section 1.2.1 for the case of a sinusoid. These parameters can be used by a singing voice synthesizer or to transform a recording. From a psychological perspective, pitch contour, vibrato features, intensity contour, tremolo, phonetic timing, and others related to timbre are the main control parameters that are typically used to transmit a message with a certain mood or emotion (Juslin & Laukka, 2003) and shaped by a musical style (Thalén & Sundberg, 2001).

Nominal values for certain parameters can be inferred from the musical score, such as note pitch, dynamics and note duration and its articulation like staccato or legato marks. However, these values are not intrinsically expressive per se. In other words, expression contributes to the differences between these values and a real performance.

It is important to note that there is more than one acceptable expressive performance for a given song (Friberg et al., 2009; Rodet, 2002; Sundberg, 2006). Such variability complicates the evaluation and comparison of different expression control approaches.

In this dissertation, we adopt a signal processing perspective to focus on the acoustic cues that convey a certain emotion or evoke a singing style in singing performances. As mentioned in Juslin & Laukka (2003), "*vocal expression is the model on which musical expression is based*" (p. 799), which highlights the topic relevance for both the speech and the music performance community. Expression has also been studied in speech and instrumental music performance, as presented in the section 1.2.4.

1.2.3 Singing voice performance analysis

The precise elements that contribute to expression in singing voice are studied in detail in section 2.3. The idea of the current section is to provide introductory insights on expression by processing a singing performance to visually present some of these features.

To illustrate the contribution of the acoustic features to expression, we analyze a short excerpt[24] of a real singing performance. The result of the

[23] http://www.mtg.upf.edu/publications/ExpressionControlinSingingVoiceSynthesis

[24] Excerpt from "Unchain my heart" song: http://www.mtg.upf.edu/publications/ExpressionControlinSingingVoiceSynthesis

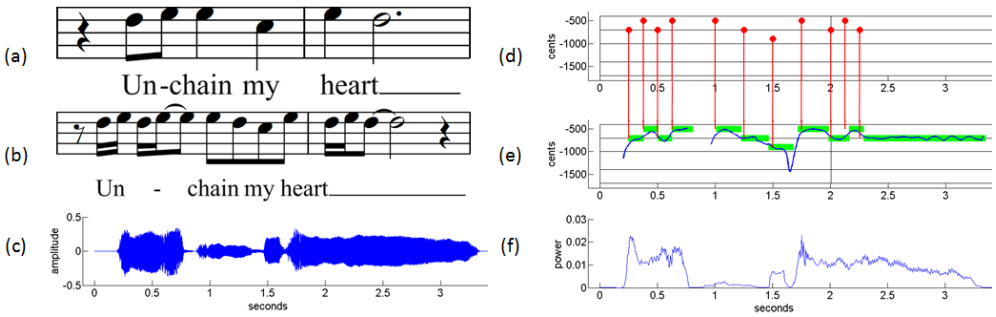


Figure 1.3: Expression analysis of a singing voice sample (a) score, (b) modified score, (c) waveform, (d) note onsets and pitch, (e) pitch and labeled notes, (f) extracted energy.

analysis is shown in Figs. 1.3 and 1.4. The excerpt contains clear expressive features like vibrato in pitch, dynamics, timing deviations in rhythm, and growl in timbre. The original score and lyrics are shown in Fig. 1.3a, where each syllable corresponds to one note except the first and last ones, which correspond to two notes. The singer introduces ornamentation and syncopation changes, shown in Fig. 1.3b. The recorded waveform is shown in Fig. 1.3c.

In Fig. 1.3d the note pitch is specified by the expected frequency in cents and the note onsets are placed at the expected time using the note figures and a 120 bpm tempo. Fig. 1.3e shows the extracted F0 contour in blue and the notes in green. The micro-prosody effects can be observed, for example in a pitch valley during the attack to the 'heart' word (around 1.6 seconds). At the end, vibrato is observed. The pitch stays at the target pitch for a short period of time, especially in the ornamentation notes.

In a real performance, tempo is not generally constant throughout a score interpretation. In general, beats are not equally spaced through time, leading to tempo fluctuation. Consequently, note onsets and rests are not placed where expected with respect to the score. In Fig. 1.3e, time deviations can be observed between the labeled notes and the projection colored in red from the score. Also, note durations differ from the score.

The recording's energy extracted from the waveform, aligned to the estimated F0 contour, is drawn in Fig. 1.3f. The intensity contour increases/decays at the beginning/end of each segment or note sequence. Energy peaks are especially prominent at the beginning of each segment, since a growl voice is used and increased intensity is needed to initiate this effect.

We can take a closer look at the waveform and spectrum of a windowed frame, as in Fig. 1.4. In the former, we can see the pattern of a modulation in amplitude or macro-period which spans over several periods. In the latter we can see that, for the windowed frame, apart from the frequency components related to F0 around 320 Hz, five sub-harmonic components appear between

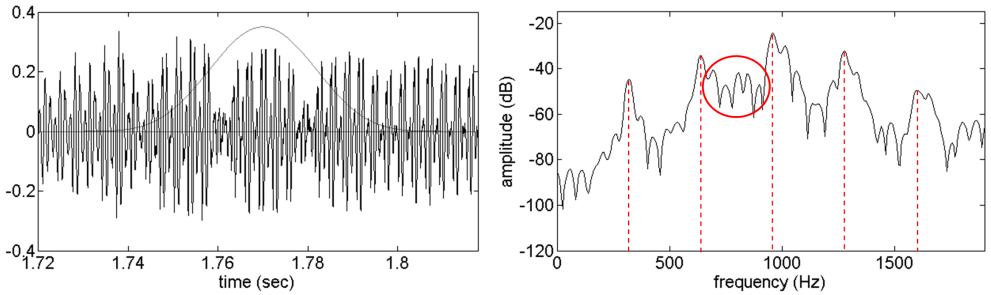


Figure 1.4: Growl analysis of a singing voice sample: waveform and spectrum.

F0 harmonics, which give the 'growl' voice quality. Harmonics are marked with a dashed line and sub-harmonics between the second and the third harmonics with a red circle.

If this set of acoustic features is synthesized appropriately, the same perceptual aspects can be decoded. Several approaches that generate these features are presented in the literature review (Sec. 2.4).

We may think of other elements which may have an influence on them beyond the actual melody of the score. Lyrics, that is to say their meaning, in the context of a given singing style do probably also play an important role. Relevant words in the song lyrics in a given phrase of the melody may be emphasized for example changing the voice quality from modal voice to breathy or growl voice (as is the case in the analyzed excerpt). This is not studied in this dissertation, although it may be worth considering it for future research as written in the conclusions (Sec. 8.4).

1.2.4 Connection to other fields

There are several aspects in performing expressively in singing voice that are common to speech utterances and musical instruments performances. In this section we shortly review how expression has been tackled in these other fields which are close to the topic of this dissertation.

Emotional speech and prosody

In speech, the five acoustic attributes of prosody have been widely studied (Obin, 2011), for instance to convey emotions (Schröder, 2009). The most studied attribute is the fundamental frequency (F0) of the voice source signal. Timing is the acoustic cue of rhythm and it is a rather complex attribute given the number of acoustic features it is related to (Obin, 2011) (p. 43). Other attributes are intensity, voice quality (related to the glottal excitation), and articulation (largely determined by the phonetic context and speech rate). Emotional speech synthesis has been approached via formant synthesis, con-

catenative synthesis, and with statistical methods and prosody has been controlled to convey emotions (Widmer & Goebl, 2004).

An emotional space was set in Russell (1980) and Posner et al. (2005), where emotions are placed in a two-dimensional space (circumplex model of affect) relating them to two main processes: arousal (or activation or alertness) and valence (how positive or negative the emotion is). Also, in Ilie & Thompson (2006) the extended version of the emotion space to three dimensions of affect (energy arousal, tension arousal and valence) was used to compare acoustic parameters in both music and speech. The authors studied the degree of overlap between affective qualities in music and speech by directly comparing intensity, pitch and tempo. They conclude that there is a general mechanism that links acoustic features (like the ones modified in their experiments) to emotions. However, some differences were found in the behaviour of features with respect to dimensions and emotions, which could be taken into account. Nonetheless, these differences show that different strategies may be used for speech and music.

In Schröder (2001) a review of how emotional speech is approached in the different techniques. It is worth mentioning that in unit selection good results are obtained using separated databases for each emotion, and therefore selecting units according to the emotion willing to synthesize. In this case, the voice quality is determined by the database, and the control parameters contours can be directly extracted from a real utterance (copy synthesis). Also explicit prosody models/rules are used to modify pitch, duration and loudness by setting general settings for each emotion (F0 level and range, tempo and loudness level, their relationship to phonemes and syllables).

Within the statistical speech synthesis, in Tachibana et al. (2005) emotions are modelled using a Hidden Markov Model (HMM) framework and at the same time it is possible to interpolate between 2 different styles. In this case, separate models per emotion are created, and control parameters like pitch, timbre and loudness are predicted.

Instrumental musical performance

Expressive music performance with instruments has also been widely studied. In this subsection we mention the basic characteristics of different works. An exhaustive review can be found in Kirke, Alexis, Miranda (2013), where 30 systems are classified into non-learning methods, linear regression, artificial neural networks, rule/case-based learning models among others. Several computational models are reviewed in Widmer & Goebl (2004), like the KTH model, which is based "*on performance rules that predict the timing, dynamics, and articulation from local musical context*" (p. 205). The Todd model links the musical structure to a performance with simple rules like measurements of human performances. The Mazzola model analyzes musical structure features like tempo and melody and iteratively modifies expressive parameters of a synthesized performance. Finally, a machine-learning model discovers pat-

terns within large amounts of data, it focuses for instance on timing, dynamics, and more abstract structures like phrases, and manipulates them via tempo, dynamics, and articulation.

In Mion et al. (2010), different expressive intentions are analyzed and both acoustical and perceptual commonalities of instrumental songs are studied. Machine learning techniques are applied to observe how expressive intentions are organized. PCA technique is applied to get a visual 2D representation.

An approach for modelling and controlling the expressiveness can be found in Canazza et al. (2004). The authors apply morphing techniques to change expressive intentions continuously working both at high (symbolic) and low (features) levels. Still with the idea of synthesis control and how expression is mapped, there is the work of Maestre (2009). In this case, it is worth mentioning that the control parameters are the bowing contours and are used to get a natural violin sound, extracted from an annotated input score. Two sound synthesis approaches (physical modelling synthesis and sample-based synthesis) were taken into consideration.

In Lindemann (2007), reconstructive phrase modelling (RPM), the approach used in the Synful Orchestra, is explained. It combines additive synthesis with concatenative synthesis. The first one is used to represent sounds as combination of "time-varying harmonic plus noise elements", for example, rapidly varying components are separated from slowly varying ones in each harmonic envelope. The second one is used to realistic sound quality of sampling. It differs from the traditional technique in the sense that it captures the transition between notes. In this framework, the fine details from pitch and amplitude are stored in the phrase database. When searching for a matching phrase in real time performance, slow varying features are directly mapped from the MIDI control stream, and rapidly varying details from the database. RPM also uses the relationship of timbre with pitch and loudness in order to predict separately slow varying amplitudes of each harmonic based on neural networks. Rapid variations of the harmonics are stored in the database by subtracting the predicted harmonic contour form the original harmonic.

Finally, case-based reasoning (CBR) has also been used for the generation of expressive performances (Arcos et al., 1998). CBR is an approach to problem solving and learning where new previously solved problems are used to solve new ones. It needs first to retrieve solved problems using some similarity criteria and then it adapts the corresponding solutions to the current problem to solve. In SaxEx, the musical knowledge for the model is provided by musical perception and understanding theories. SaxEx uses Spectral Modelling Synthesis to extract the expressive parameters and to apply transformations to an inexpressive performance. In this framework, predictions of expressive performances are done based on how other similar pieces were played by musicians.

1.3 Proposed Systems

The aim of the section is to provide a general idea of the approaches we have been working on and their building blocks. We give this comprehensive overview in order to make the details of the remaining chapters easier to read and to help see how these are related. We cover from the fundamental concepts of our work to the designed databases and how we model the recorded expression from unit selection and Hidden Markov Model perspectives.

1.3.1 Basic ideas

In this section, we first introduce the basic ideas behind our work, which are the expression contours and what we consider as units. Based on these concepts, we build the expression databases that are used by all our approaches (Sec. 1.3.2). Next, we introduce the main building blocks of the unit selection-based approach (Sec. 1.3.3) and the Hidden Markov-based approach (Sec. 1.3.4). Finally, the hybrid approach which combines elements of the two previous approaches is presented (Sec. 1.3.5).

Expression contours

In section 1.2.3 we have introduced that expression in singing voice performances can be analyzed and partly visualized by plotting the evolution over time of some features. The pitch, dynamics, timing, and the subharmonics in growl voices of a recorded performance are visualized, which represent one of the possible ways a song can be expressively sung.

The aim of any of the proposed systems is to simulate the behavior of such expression features so to control a singing voice synthesis system. We have devoted our efforts into generating pitch and dynamics controls, as an initial step to a more comprehensive approach that controls also timing and timbre aspects of the voice. Therefore, our aim is to model the time evolution of pitch and dynamics at frame level. The singing voice synthesizer will then use the provided values in order to generate an expressive performance.

These contours represent a (virtual) singer rendition of a given target song, which is defined by a sequence of notes and rests, with their durations and pitch values. According to music theory, if we focus on any sequence of three notes, we can distinguish several topologies which are next detailed.

Unit representation: from Narmour to triphones

The basic element in our work are units, which we define as a sequence of three notes or rests. We can think of it in terms of a central note and the surrounding ones which provide contextual information. For instance, the transition or attack to a central note from a silence is generally different than that from a note.

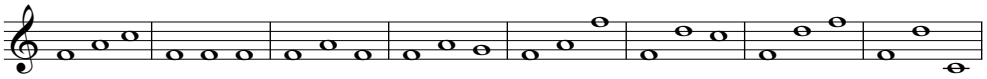


Figure 1.5: Narmour group structures.

We can relate units to music theory aspects, like the basic grouping structures on which the Narmour's Implication-Realization Model (Narmour, 1990, 1992) is founded. This model, as summarized by Mantaras & Arcos (2002), allows to analyze the melody of a piece based on the basic units of the listener's perception and the fulfillment of the expectations. As shown in Fig. 1.5, the patterns described by these structures cover the different trends a sequence of notes may follow and are typically defined by the distances between note pitches. The direction of these intervals may be all ascending, descending, or interleaved directions and the magnitude of such intervals is also used to differentiate them between steps (small intervals) or leaps (large intervals).

Melodies can be segmented into a sequence of Narmour structures or units, as done in Arcos et al. (1998). In the SaxEx project, these structures are identified in the target score, and then used to retrieve similar examples from an expressive database based on the assumption that notes with a similar Narmour structure should be played in a similar way.

If we set to three the number of notes of these structures we can find a similarity between the generic concept of units and the one used in our approaches. In Fig. 1.6 we show a symbolic representation of the unit concept with three labeled notes and the corresponding pitch contour. In the following sections we explain several approaches we have been working on in which these units are being modeled either individually with the unit selection-based approach or either statistically with the Hidden Markov Model-based approach to generate a longer sequence of pitch and dynamics contours.

We can also relate our unit concept to how units are typically defined in speech synthesis. Several unit types are being used by concatenative text-to-speech systems, which may range from simple phonemes, to phoneme transitions or diphones, to three phonemes or triphones, or even to longer units.

1.3.2 Expression contours database creation

Our approaches need to work with an expression database that fulfills very specific requirements, which range from the coverage of different combinations of note durations and pitch intervals, to the lyrics' content. Given these requirements, we have designed, recorded, and labeled two databases ourselves. These steps are detailed in Chapter 3.

In short, the requirements related to coverage that we have adopted imply that we want that our database contains different combinations of note durations, pitch interval and note strength, which a measure of the beat accent of a note taking into account its onset within a measure. For simplicity, we have left

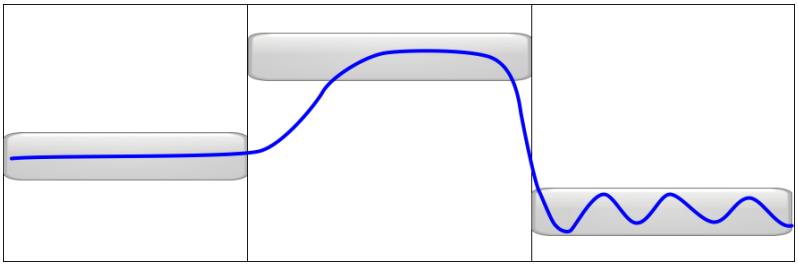


Figure 1.6: Unit: 3 consecutive notes and pitch contour.

out the lyrics and restricted our interest to the pitch and dynamics contours of the singer performance of the recorded scores. However, these parameters are affected by phonetics by what is known as the microprosody effects. For instance, unvoiced consonants produce pitch segments without pitch content, and velar consonants produce pitch valleys which are not related to expression but to phonetics. Therefore in our case it is preferable not to use any lyrics in the recordings, hence we will use simply vowels.

The main idea behind the expression databases is that we want to capture how a singer interprets expressively a set of melodies. As we have explained, we are not recording lyrics but vowels. We have approached the database design from two perspectives. On one hand, we have recorded a database of jazz songs, where we have changed the original lyrics to vowels. On the other hand, we have prepared a set of systematic exercises that cover several combinations of note pitch intervals, durations, and note strengths.

We have designed a methodology for labeling these recordings after pitch and dynamics estimation. Note onsets, its durations, and note transitions and sustains are estimated in a semi-automatic way. That is to say, we have designed an algorithm that manually segments these data, but we manually refine the boundaries. Vibratos first are manually segmented and afterwards rate and depth are automatically estimated by an algorithm that we have designed as well.

The output of the labeling process is on the one side the sequence of notes and rests per song and the sustains and transition segmentation. On the other side we also have the dynamics contour and the pitch information which is split into three contours: the baseline pitch (from which we have extracted the vibrato), the vibrato depth and rate (which are both null when no vibrato is present).

The resulting databases with the labeling information are used by both the unit selection and the statistical approach we have designed. In the first case, the pitch contour is directly generated from the selected units. In the second case, the expressive contours are statistically modeled and then used at synthesis.

1.3.3 A unit selection-based system

The unit selection-based approach aims to generate expressive singing voice contours by capturing the fine details of the recorded databases. This is done following the typical steps in unit selection approaches, but in our case the units are segments of pitch and dynamics contours. Therefore, other voice features like timbre are not represented. First, units are selected based on the cost criteria we have defined. Then, units are transformed to match the target score. Next, the transformed units are concatenated. Finally, the output sound is rendered using the Vocaloid synthesizer. These steps are next summarized.

Unit selection

Given a target score to synthesize, its set of notes and rests can be expressed as a sequence of units. In the unit selection step we want to select a sequence of units from the expression database which is as similar as possible to the target sequence. The similarity measure is provided by the transformation and concatenation costs that we have defined. The transformation costs measure how much the unit notes need to be transformed in time (duration) and frequency (intervals) to match the target score. Note that we are not using the absolute pitch values to measure the amount of transformation in frequency, since we can easily transpose a pitch contour and reuse it with a given offset difference. By contrast, the concatenation costs favor the selection favor the selection of units from close contexts. This is done both by favoring the selection of consecutive units in the database and also by favoring the selection of units of the same phrase.

Unit transformation

Once the sequence of units has been selected, the unit transformation step aims at transforming the units' pitch and dynamics contours to match the corresponding target units. Our representation of the pitch contour allows the system to do a separate transformations in time and frequency. The pitch contour is decomposed into the baseline pitch on one side and the vibrato rate and depth contours on the other side. The baseline pitch is an estimation of the pitch without vibratos. The vibrato features are 0 when there are no vibratos, and their values are estimated for the vibrato segments.

Regarding the note duration, the transformation process is done mainly in the sustain segments and to keep the pitch contour transitions duration as much as possible. Concerning the pitch transformation, the baseline pitch and vibrato depth and rate contours are time-scaled to preserve their shape in the target note durations. The vibrato model we are using allows us to recreate a new vibrato pitch oscillation that preserves the properties of the original vibrato adapted to the new note duration.

Unit concatenation

Once all units have been transformed, we need to concatenate them. This step mainly keeps the shape of the transition and sustain part of the central note of each unit. To do so, the transformed contours are masked before cross-fading. These masks are basically weights equal to 1 during the parts that we want to preserve and 0 otherwise (with a smooth transition between these two areas). The weights are complementary between consecutive masks so that there are no discontinuities when cross-fading.

Contour generation

After unit concatenation, the dynamics contour is already generated by overlapping the transformed unit contours weighted by the corresponding masks. However, the final pitch contour requires one more step. At this point, we have three contours that need to be joined: the baseline pitch, the vibrato depth, and the vibrato rate. First, the baseline pitch can be tuned if necessary in case there is some deviation with respect to the target pitch during the sustain part. Then, the vibrato features are combined to generate the oscillation which is then added to the baseline pitch, resulting in the final pitch contour.

Sound generation

The last step is the generation of the Vocaloid readable files (or VSQX format). These contain all the information of the generated contours for pitch and dynamics. The VSQX files contain the sequence of notes and the lyrics phonetic transcription which is automatically generated. In this thesis we have worked with Spanish and English databases, and therefore the songs we synthesize are in these languages.

1.3.4 A statistical system

The Hidden Markov model-based approach aims to statistically and jointly model the behaviour of the expression contours. In this case we are adapting the HTS framework[25] for speech synthesis to singing voice synthesis. Therefore, the main adaptation steps are to define the contextual factors and also the actual contour data to be modeled in the training step. In the synthesis step, the contextual data for the target song is used by the trained models to generate the output contours. These steps are reviewed here below.

Contextual data

The contextual data used by the HTS framework is an extended version of the unit concept. It uses information related to a central note and the previous

[25] http://hts.sp.nitech.ac.jp

and succeeding ones. It also uses pitch intervals between these three notes as well as their durations. HMM-based systems extend the contextual data by adding more information like the number of notes in the song to extend the information.

Model training

Another difference with respect to the unit selection-based approach is the input training contour data. One possibility would be to use the absolute pitch values. However, this would force us to cover a wide pitch range of several octaves for any possible song we think it might be synthesized. An alternative is to use the pitch difference between the absolute pitch contour and a theoretical pitch contour which is computed as a piecewise cubic interpolation from the sequence of notes and transitions.

The models that we train are different from what is typically done in speech. In our case we do not model phonemes, nor notes, but sequences of note transitions and sustains. Within the transition models, we make differences depending on the pitch interval direction (ascending, descending, or similar).

Contour synthesis

In order to synthesize the target pitch and dynamics contours, the same format of the contextual data is used for the target song. Since we have trained the pitch difference, we can synthesize any sequence of notes even if the absolute pitch was not present in the training data.

The generate data is on the one hand the dynamics, and on the other hand the baseline pitch vibrato depth, and vibrato rate which have to be combined as explained in order to generate the pitch contour.

1.3.5 A hybrid system

The hybrid approach attempts to combine both the unit selection-based and the Hidden Markov model-based approaches into a single one. First, we run the statistical approach. Then, its output is used to enrich the subcost functions of the unit selection step. More concretely, the statistical approach guides the unit-selection approach by providing a baseline of the pitch and dynamics contours. These steps are reviewed here below.

Combination of approaches

We have realized that the unit selection-based approach has a set of subcost functions in order to select the units that will contribute to generate the output contours. However, we can only use the labeling data (note durations and pitch intervals) to measure the cost of unit transforming and concatenating units.

The cost functions could be enhanced if we had a target pitch contour which we want to be similar to. Such an improvement can be done using the HMM-based approach to generate an initial baseline of the dynamics and baseline pitch contours which can be included in the unit selection step.

Extended unit selection

During the computation of the cost functions, the candidate units from the expressive database are compared to the statistically generated baseline pitch. A distance measure can be computed to complement the other subcosts. In our case, we use the dynamic time warping (DTW) cost value between the unit baseline pitch (without vibrato fluctuations) and the proposed baseline pitch from the HMM-based approach as the distance measure.

1.4 Goals and organization of the thesis

As introduced in sections 1.1.1 and 1.1.2, the main objective of this thesis is to develop new systems that reproduce the expressive style of a particular singer when synthesizing a song sung by virtual singers. We focus our research on basic units of 3 notes, where a central note is contextualized by the preceding and succeeding notes. This contextual data surrounds the relevant part of a unit: the transition and sustain of the central note. Our hypothesis is that starting from such working unit, we can use unit selection-based and statistical methods to generate the expression control parameters of any target song. Units are obtained from analyzing singer recordings, and stored in labeled databases, which contain not only the pitch and dynamics from recordings but also information on which are the notes pitches, start and end times, vibrato features and their start and end times, and other score information like note strength.

In all the proposed systems, these contextual data are used either to retrieve, transform, and concatenate units, or to train statistical systems. It is important to remind at this point that the output of the proposed methods are pitch and dynamics contours which are meaningful for the target song, and are used to control the singing voice synthesizer.

The organization of the remainder of this thesis is as follows. We start by providing the literature review on the main scientific background which is relevant for this dissertation in **Chapter 2**. First, we describe how the singing voice is produced, both physically in the human body and artificially in singing voice synthesis systems (Sec. 2.2). Then, we go through the different control parameters that have an effect on expression (Sec. 2.3). Next, we provide an up to date classification, comparison, and description of a selection of approaches to expression control (Sec. 2.4). Finally, we describe and discuss on how these methods are currently evaluated (Sec. 2.5).

The following chapters proceed to detail the different elements introduced in Section 1.3. We provide the block diagram Fig. 1.7 to help understand the flow of the thesis and to visualize how the chapters are interrelated. **Chapter 3** is devoted to the creation of the expression databases. First, we define a set of requirements prior to the design of the recordings (Sec. 3.2). Then, the designs for the timing deviation, the systematic expression, and the song expression databases are detailed (Secs. 3.3 and 3.4, respectively). The common labeling methodology for all these databases is finally described (Sec. 3.5).

The unit selection-based approach is explained in **Chapter 4**. In this case, the first step is to select units according to a set of cost functions (Sec. 4.2). Then, the selected units are transformed preserving note transition shapes and vibrato features (Sec. 4.3.3), and finally concatenated (Sec. 4.4) before synthesizing the sound (Sec. 4.5). A hidden markov model approach is explained in **Chapter 5**. Its main components are the contextual data used to describe the training data (Sec. 5.2), the training process, and the synthesis of the expression contours. These steps are slightly different in the two methods that we describe, a baseline HMM-based system which models note sequences (Sec. 5.3) and our proposal of a modification of the HMM-based method which models transition and sustain sequences (Sec. 5.4). In **Chapter 6** we present how the unit selection-based and the HMM-based approaches can be combined in a hybrid approach. The HMM-based system is used to generate expression contours (Sec. 6.2) which are then used to extend the cost functions in the unit selection-based approach (Sec. 6.3). In **Chapter 7** we evaluate and compare several synthesized performances. Both perceptual (Sec. 7.2) and efficiency (Sec. 7.3) evaluations have been conducted. We also consider some more use case in which the proposed systems could be applied (Sec. 7.4) and discuss on possible aspects that the community should face to improve the evaluation of singing voice synthesis systems (Sec. 7.5).

Finally, in **Chapter 8** we provide the conclusions of this dissertation. First, we summarize the contributions (Sec. 8.2), then we discuss the future perspectives (Sec. 8.3), and finally describe the challenges that we currently foresee (Sec. 8.4).

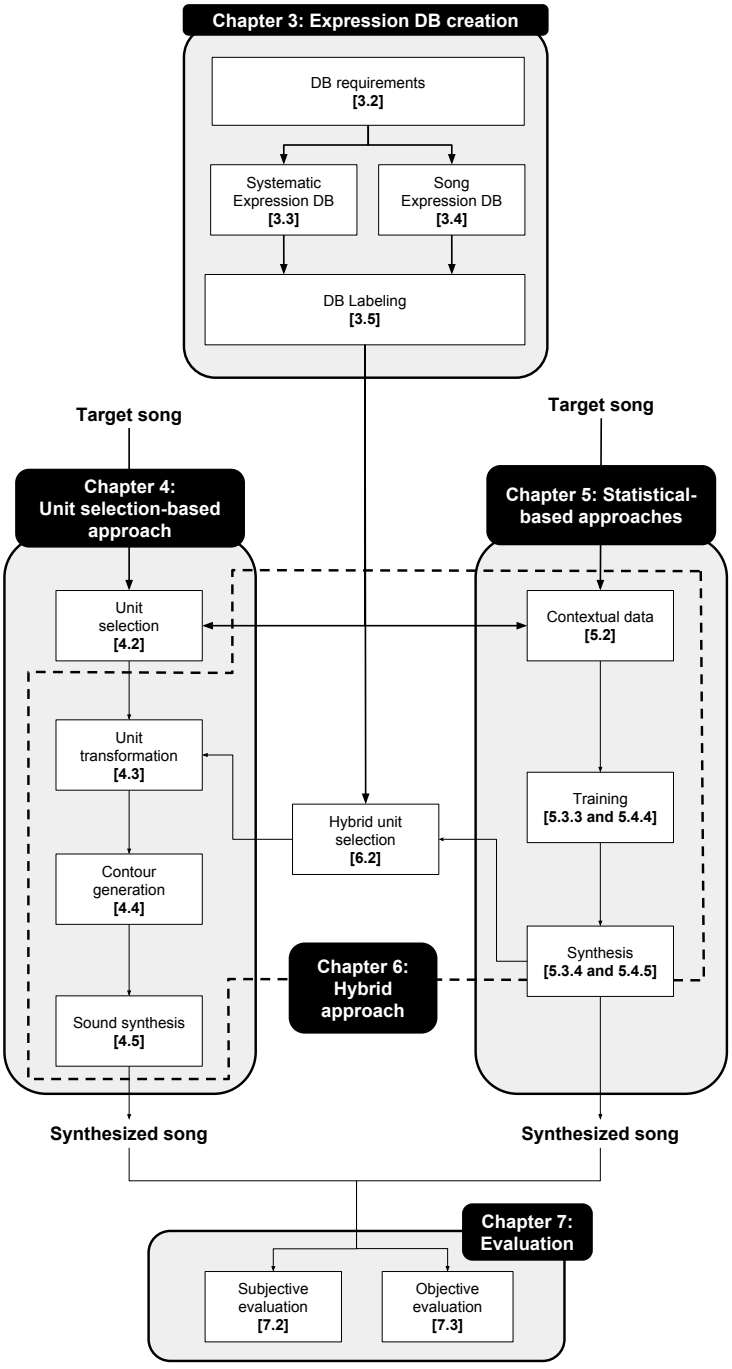


Figure 1.7: Thesis layout (numbers represent chapters and sections).

CHAPTER 2

Literature review

In this second chapter we provide the state of the art as the required scientific background for the next chapters. It is mainly divided into four parts: the explanation of the production mechanism of the singing voice, the main features that control its performance, a categorization of the approaches that have been typically used to control the singing voice expression, and finally the evaluation strategies.

The succeeding chapters of this dissertation present several approaches that aim to broaden the amount of categories within the topic of expression control in singing voice synthesis.

2.1 Introduction

This literature review, mainly based on Umbert et al. (2015), starts by introducing the mechanism of singing voice production (Sec. 2.2), both from a physical and a synthesis perspectives. Next, we present the commonly studied set of voice parameters that, from a perception perspective, have an effect on expression (Sec. 2.3). Then, we provide an up to date classification, comparison, and description of a selection of approaches to expression control (Sec. 2.4). Next, we describe and discuss how these methods are currently evaluated (Sec. 2.5). Finally, we conclude the main ideas presented in this chapter (Sec. 2.6).

2.2 The singing voice

In order to better understand the signal we are dealing with and how it has been modeled, in this section we describe the generation of the singing voice. First, we explain the physical mechanism of the air coming from the lungs until the voice sound is generated (Section 2.2.1). Then, we overview the main blocks of the singing voice synthesis systems (Section 2.2.2).

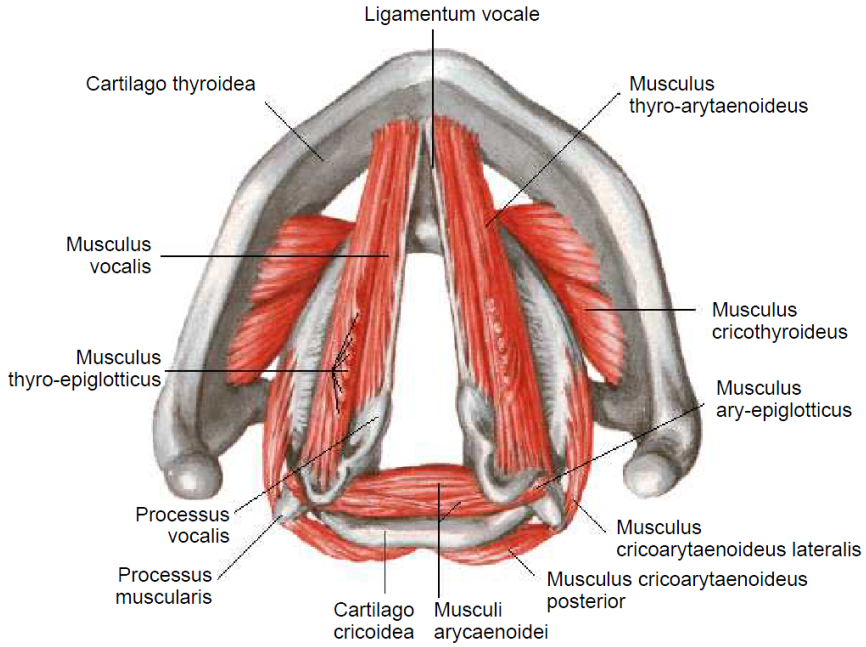


Figure 2.1: Vocal folds representation from Kob (2002)

2.2.1 How is the singing voice produced?

The voice organ anatomy

The principal systems of the voice organ are the breathing apparatus, the vocal folds (see Fig. 2.1), and the vocal tract. Here we provide a short overview, the reader is referred to (Sundberg, 1987) for a comprehensive description of all the elements in the voice organ.

The first system (breathing apparatus) is formed by the lungs, which are connected to the vocal folds through the trachea. The sound production starts by a compression of the lungs that send air to the vocal folds and vocal tract. The vocal folds (or vocal cords) are a set of muscles protected by a membrane. The length of the vocal folds is related the pitch (the longer the vocal folds the lower the pitch range) and it is correlated to the perimeter of the neck. The glottis is the opening between the vocal folds. These may be brought together by the so called adduction movement (the vocal folds vibrate), or separated by the abduction movement. Depending on the balance between these two movements the output sounds may be a combination of voiced and unvoiced phonemes (e.g. flow/breahy phonation).

The vocal folds (from the glottis) are joined to the vocal tract through the larynx and pharynx tubes. The vocal tract starts at the pharynx and continues with the mouth and nasal cavities. When we produce sounds, the air may pass

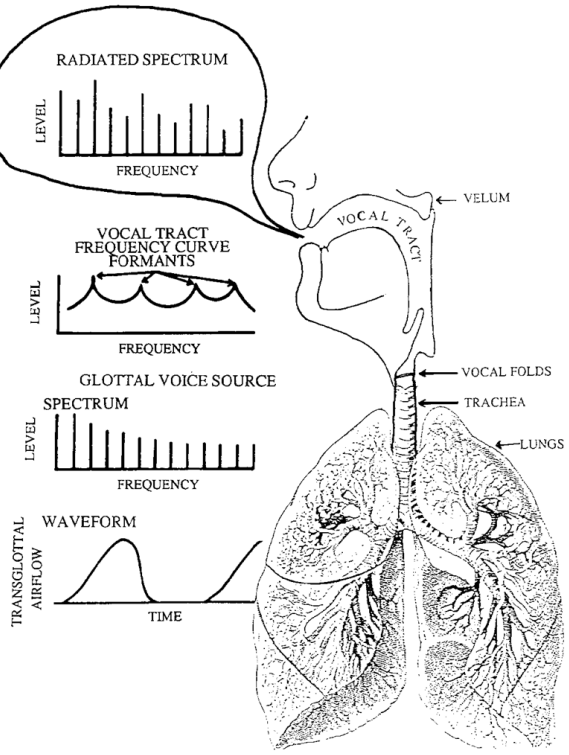


Figure 2.2: Vocal folds[1]

through the nasal cavity producing nasal sounds.

Speech and singing voice production mechanism

The singing voice is produced in the voice organ, which also produces speech. Therefore, both speech and singing voice are quite similar. The singing voice is a broader phenomena that includes speech and modifications of speech sounds (notes), but both are generated by the same mechanism (Sundberg, 1987).

Simply put, in voiced phonemes, the air coming from the lungs triggers the vocal folds vibration. The vibration of the vocal folds is periodic and results in what we call the fundamental frequency or pitch. Pitch refers to a perceptual characteristic, but it is broadly used as equivalent to fundamental frequency. The temporal evolution of the pitch, is related to prosody in speech and to melody in singing voice. This voice source signal is shaped by the larynx constriction together with the filtering applied in the vocal tract generating a signal with time-varying properties. The variation of vocal trach filter depends broadly on the jaw opening, the tongue position, or whether the air pass through the nose. The vocal tract filter can be described by a set of

[1]Figure from Sundberg (1987) reproduced with the author's permission.

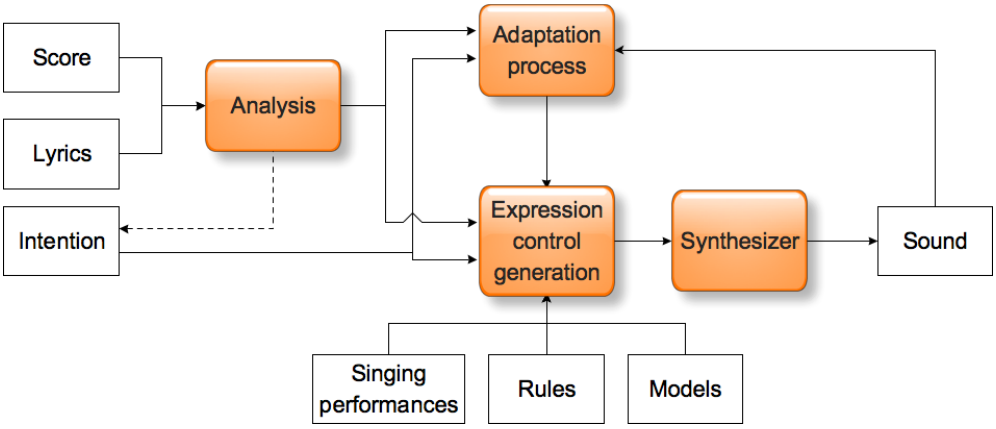


Figure 2.3: Generic framework blocks for expression control.

emphasized frequencies which are named formants. More details on the formant frequencies are given in section 2.3.5. In unvoiced phonemes, the filtered source signal is approximated by noise. These features have a great impact on the way singing performance expression is perceived.

We can also have a look at the different signals that intervene in this mechanism and specially to how their spectrum is being shaped at each step. These signals are shown in Fig. 2.2. First, the vibration of the vocal folds generates a set of pulses. This signal is shown at the bottom part of the figure, showing how the pressure at the vocal folds increases and decreases periodically, at the same rhythm these open and close. This signal spectrum has a fundamental frequency and the corresponding multiple frequencies or harmonics. Their amplitude decreases as the frequency increases. This signal is filtered by the vocal tract filter. In the radiated signal the amplitude of the harmonic frequencies depend therefore on both the vocal tract shape and on the voice source characteristics.

2.2.2 How is the singing voice synthesized?

Synthesis systems' building blocks

The generic framework of the singing voice synthesis systems is represented in Fig. 2.3, based on Kirke, Alexis, Miranda (2013). The input may consist of the score (e.g. note sequence, contextual marks related to loudness, or note transitions), lyrics, and the intention (e.g. the style or emotion). Intention may be derived from the lyrics and score content (dashed line).

The input may be analyzed to get the phonetic transcription, the alignment with a reference performance, or contextual data. The expression control generation block represents the implicit or explicit knowledge of the system as either a set of reference singing performances, a set of rules, or statistical mod-

Signal models		Physical models
Time domain	Frequency domain	
PSOLA, MBROLA	formant synthesis, FM, FOF, LPC, HMMs, spectral modeling synthesis (SMS), sinusoidal plus residual, (phase) vocoder	acoustic tube models, mass model, wave propagation models, finite differential equations

Table 2.1: Voice model classification.

els. Its output is used by the synthesizer to generate the sound, which may be used iteratively to improve the expression controls.

Synthesis systems' voice model

A key element of such technologies is the singer voice model. Although it is out of the scope of this dissertation to describe it in depth, Table 2.1 shows the groups in which these are typically classified (Bonada & Serra, 2007; Cook, 1998; Rodet, 2002; Schwarz, 2007) and the corresponding synthesizer control parameters. These are organized in waveform synthesizers (distinguishing between perceptual or production mechanisms), and concatenative synthesizers.

The main difference between perceptual perspective (signal models) and production perspective (physical models) is found in the type of controls. In the former, controls are related to perceptual aspects such as pitch and dynamics, while in the latter ones controls are related to physical aspects of the voice organ. In concatenative synthesis, samples (called units) retrieved from a corpus are transformed and then concatenated to generate the output utterance according to some concatenation-cost criteria. Units may cover a fixed length (e.g. diphones cover the transition between two phonemes), or a more flexible and wider scope. Inspired by the speech synthesis community, a wide variety of techniques can be found in the literature, from acoustic tubes, (phase) vocoder, linear prediction coding (LPC), frequency modulation (FM), spectral modeling synthesis (SMS), formant wave functions (FOF), and formant synthesis to combinations such as sinusoidal modeling with PSOLA (SM-PSOLA) or sinusoidal modeling with glottal excitation and resonances in the frequency domain. Finally, statistical methods have also been used to train Hidden Markov Models (HMMs) and to generate a singing voice signal. In Table 2.1 we classify these voice models, and relate them to what is being modeled (signal vs. physical mechanism) and the type of representation (time vs. frequency domain).

	Model-based synthesis		Concatenative synthesis	
	Signal models	Physical models	Fixed length	Non uniform
Parameters	F0, resonances (centre frequency and bandwidth), sinusoid frequency, phase, and amplitude, glottal pulse spectral shape, phonetic timing	Vocal apparatus related parameters (tongue, jaw, vocal tract length, and tension, subglottal air pressure, phonetic timing)	F0, amplitude, timbre, phonetic timing	

Table 2.2: Singing voice synthesis systems and control parameters.

Synthesis systems' control parameters

For the purpose of this dissertation, it is more interesting to classify singing synthesis systems with respect to the control parameters. As shown in Table 2.2, those systems are classified into model-based and concatenative synthesizers. While in signal models the control parameters are mostly related to a perception perspective, in physical models these are related to physical aspects of the vocal organs. In concatenative synthesis, a cost criterion is used to retrieve sound segments (called units) from a corpus which are then transformed and concatenated to generate the output utterance. Units may cover a fixed number of linguistic units, e.g. diphones that cover the transition between two phonemes, or a more flexible and wider scope. In this case, control parameters are also related to perceptual aspects.

Within the scope of this dissertation, we focus on the perceptual aspects of the control parameters which are used to synthesize expressive performances by taking a musical score, lyrics or an optional human performance as the input. This work therefore, does not discuss voice conversion and morphing in which input voice recordings are analyzed and transformed (Doi et al., 2012; Kawahara et al., 2009). In these cases, a real voice recording, playing the role of the voice model, is analyzed and transformed (e.g. timbre and prosodic features). This transformation in some cases is done via statistical methods such as Gaussian Mixture Models (GMMs) or Hidden Markov Models (HMMs).

2.3 Singing voice performance features

In Section 1.2.2 we introduced a wide set of low-level parameters for singing voice expression. In this section we relate them to other musical elements. Then, the control parameters are described, and finally we illustrate them by analyzing a singing voice excerpt.

2.3.1 Feature classification

As in speech prosody introduced in Section 1.2.4, music can also be decomposed into various musical elements. The main musical elements such as melody, dynamics, rhythm, and timbre are built upon low-level acoustic features. The

<table>
<tr><th>Melody</th><th>Dynamics</th><th>Rhythm</th><th>Timbre</th></tr>
<tr><td colspan="2">Vibrato and tremolo (depth and rate)</td><td>Pauses</td><td>Voice source</td></tr>
<tr><td colspan="2">Attack and release</td><td>Phoneme time-lag</td><td>Singer's formant</td></tr>
<tr><td>Articulation</td><td colspan="2">Phrasing</td><td>Sub-harmonics</td></tr>
<tr><td>F0 contour</td><td>Intensity contour</td><td>Note/phone onset/dur</td><td>Formant tuning</td></tr>
<tr><td>F0 frame value</td><td rowspan="2">Intensity frame value</td><td>Timing deviation</td><td rowspan="2">Aperiodicity spectrum</td></tr>
<tr><td>Detuning</td><td>Tempo</td></tr>
</table>

Table 2.3: Singing voice expression features' classification.

relationships between these elements and the acoustic features can be represented in several ways (Lesaffre, 2006) (p. 44). Based on this, Table 2.3 relates the commonly modeled acoustic features of singing voice to the elements to which they belong. Some acoustic features spread transversally over several elements. Some features are instantaneous such as F0 and intensity frame values, some span over a local time window like articulation and attack, and others have a more global temporal scope like F0 and intensity contours, or vibrato and tremolo features. Next, for each of these four musical elements, we provide introductory definitions to their acoustic features.

2.3.2 Melody related features

The F0 contour, or the singer's rendition of the melody (note sequence in a score), is the sequence of F0 frame-based values (Salamon et al., 2014). F0 represents the "*rate at which the vocal folds open and close across the glottis*", and acoustically it is defined as "*the lowest periodic cycle component of the acoustic waveform*" (Juslin & Laukka, 2003) (p. 790). Perceptually it relates to pitch, defined as "*the aspect of auditory sensation whose variation is associated with musical melodies*" (Plack & Oxenham, 2005) (p. 2). In the literature, however, pitch and F0 terms are often used indistinctly to refer to F0.

The F0 contour is affected by micro-prosody (Saino et al., 2010), that is to say, fluctuations in pitch and dynamics due to phonetics (not attributable to expression). While certain phonemes like vowels may have stable contours, other phonemes such as velar consonants may fluctuate due to articulatory effects.

A skilled singer can show the expressive ability through the melody rendition and modify it more expressively than unskilled singers. Pitch deviations from the theoretical note can be intentional as an expressive resource (Sundberg, 2006). Moreover, different articulations, that is to say the F0 contour in a transition between consecutive notes, can be used expressively. For example, in 'staccato' short pauses are introduced between notes. In Section 2.3.6 the use of vibratos is detailed.

2.3.3 Dynamics related features

As summarized in Juslin & Laukka (2003), intensity (related to the perceived loudness of the voice) is a "*measure of energy in the acoustic signal*" usually from the waveform amplitude (p. 790). It "*reflects the effort required to produce the speech*" or singing voice, and is measured by energy at a frame level. A sequence of intensity values provides the intensity contour, correlated to the waveform envelope and the F0 since energy increases with the F0 so to produce a similar auditory loudness (Sundberg, 1987). Acoustically, vocal effort is primarily related to the spectrum slope of the glottal sound source rather than to the overall sound level. Tremolo may also be used, as detailed in Section 2.3.6.

Micro-prosody has also an influence on intensity. The phonetic content of speech may produce intensity increases as in plosives or reductions like some unvoiced sounds.

2.3.4 Rhythm related features

Perception of rhythm involves cognitive processes such as "*movement, regularity, grouping, and yet accentuation and differentiation*" (Scheirer, 1998) (p. 588), where it is defined as "*the grouping and strong/weak relationships*" amongst the beats, or "*the sequence of equally spaced phenomenal impulses which define a tempo for the music*". Tempo corresponds to the number of beats per minute. In real life performances, there are timing deviations from the nominal score (Juslin & Laukka, 2003).

Similarly to the role of speech rate in prosody, phoneme onsets are also affected by singing voice rhythm. Notes and lyrics are aligned so that the first vowel onset in a syllable is synchronized with the note onset and any preceding phoneme in the syllable is advanced (Saino et al., 2006; Sundberg, 2006).

2.3.5 Timbre related features

Timbre depends mainly on the vocal tract dimensions and on the mechanical characteristics of the vocal folds which affect the voice source signal (Sundberg, 1987). Timbre is typically characterized by an amplitude spectrum representation, and often decomposed into source and vocal tract components.

The voice source can be described in terms of its F0, amplitude, and spectrum (vocal loudness and mode of phonation). In the frequency domain, the spectrum of the voice source is generally approximated by an average slope of -12 dB/octave, but typically varies with vocal loudness (Sundberg, 1987). Voice source is relevant for expression and used differently among singing styles (Thalén & Sundberg, 2001).

The vocal tract filters the voice source emphasizing certain frequency regions or formants. Although formants are affected by all vocal tract elements,

some have a higher effect on certain formants. For instance, the first two formants are related to the produced vowel, with the first formant being primarily related to the jaw opening and the second formant to the tongue body shape. The next three formants are rather related to timbre and voice identity, with the third formant being particularly influenced by the region under the tip of the tongue and the fourth to the vocal tract length and dimensions of the larynx (Sundberg, 1987). In western male operatic voices the 3rd, 4th, and 5th typically cluster, producing a marked spectrum envelope peak around 3 kHz, the so-called singer's formant cluster (Sundberg, 1987). This makes it easier to hear the singing voice over a loud orchestra. The affected harmonic frequencies (multiples of F0) are radiated most efficiently towards the direction where the singer is facing, normally the audience.

Changing modal voice into other voice qualities can be used expressively (Loscos & Bonada, 2004). Rough voice results from a random modulation of the F0 of the source signal (jitter) or of its amplitude (shimmer). In growl voice sub-harmonics emerge due to half periodic vibrations of the vocal folds and in breathy voices the glottis does not completely close, increasing the presence of aperiodic energy.

2.3.6 Transverse features

Several features from Table 2.3 can be considered transversal given that they spread over several elements. In this section we highlight the most relevant ones.

Vibrato is defined (Sundberg, 1987) as a nearly sinusoidal fluctuation of F0. In operatic singing, it is characterized by a rate that tends to range from 5.5 to 7.5 Hz and a depth around 0.5 or 1 semitones. Tremolo (Sundberg, 1987) is the vibrato counterpart observed in intensity. It is caused by the vibrato oscillation when the harmonic with the greatest amplitude moves in frequency, increasing and decreasing the distance to a formant, thus making the signal amplitude vary. Vibrato may be used for two reasons (Sundberg, 1987) (p. 172). Acoustically, it prevents harmonics from different voices from falling into close regions and producing beatings. Also, vibratos are difficult to produce under phonatory difficulties like pressed phonation. Aesthetically, vibrato shows that the singer is not running into such problems when performing a difficult note or phrase like high pitched notes.

Attack is the musical term to describe the pitch and intensity contour shapes and duration at the beginning of a musical note or phrase. Release is the counterpart of attack, referring to the pitch and intensity contour shapes at the end of a note or phrase.

As summarized in (Mantaras & Arcos, 2002), grouping is one of the mental structures that are built while listening to a piece that describes the hierarchical relationships between different units. Notes, the lowest-level unit, are grouped into motifs, motifs into phrases, and phrases into sections. The piece is the

highest-level unit. Phrasing is a transversal aspect that can be represented as an "*arch-like shape*" applied to both tempo and intensity during a phrase (Friberg et al., 2009) (p. 149). For example, a singer may increase tempo at the beginning of a phrase or decrease it at the end for classical music.

2.4 Expression control approaches

In Section 2.3, we defined the voice acoustic features and related them to aspects of music perception. In this section we focus on how different approaches generate expression controls. First, we propose a classification of the reviewed approaches and next we compare and describe them. As it will be seen, acoustic features generally map one-to-one to expressive controls at the different temporal scopes, and the synthesizer is finally controlled by the lowest-level acoustic features (F0, intensity, and spectral envelope representation).

2.4.1 Classification of approaches

In order to see the big picture of the reviewed works on expression control, we propose a classification in Fig. 2.4. Performance-driven approaches use real performances as the control for a synthesizer, taking advantage of the implicit rules that the singer has applied to interpret a score. Expression controls are estimated and applied directly to the synthesizer. Rule-based methods derive a set of rules that reflect the singers' cognitive process. In analysis-by-synthesis, rules are evaluated by synthesizing singing voice performances. Corpus-derived rule-based approaches generate expression controls from the observation of singing voice contours and imitating their behavior. Statistical approaches generate singing voice expression features using techniques such as Hidden Markov Models (HMMs). Finally, unit selection-based approaches select, transform, and concatenate expression contours from excerpts of a singing voice database. Approaches using a training database of expressive singing have been labeled as corpus-based methods.

The difficulties of the topic studied in this dissertation center on how to generate control parameters which are perceived as natural. The success of conveying natural expression depends on a comprehensive control of the acoustic features introduced in Section 2.3. Currently, statistical approaches are the only type of system that jointly model all the expression features.

2.4.2 Comparison of approaches

In this section we review a set of works which model the features that control singing voice synthesis expression. Physical modeling perspective approaches can be found for instance in Kob (2003).

Within each type of approach in Fig 2.4, there are one or more methods for expression control. In Table we provide a set of items we think can be

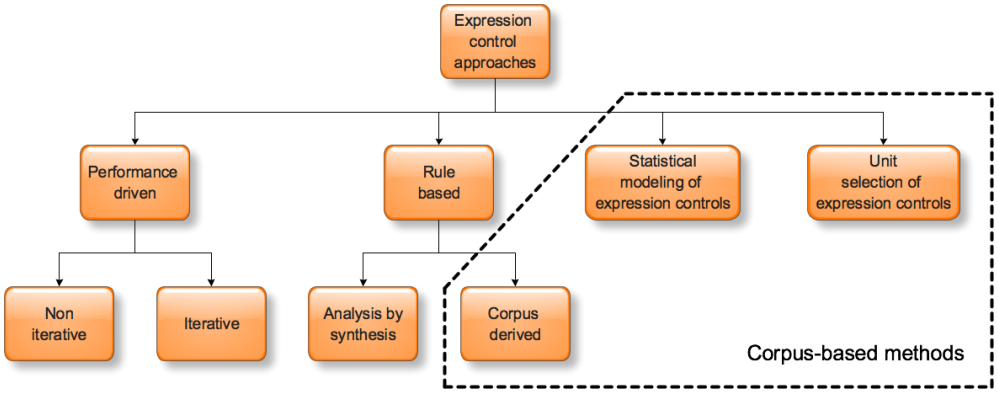


Figure 2.4: Classification of Expression Control Methods in Singing Voice Synthesis.

useful for comparison. From left to right, Type refers to the type of expression control from Fig. 2.4 to which the Reference belongs. In Control features we list the set of features that the approach deals with. Next, we provide the type of Synthesizer used to generate the singing voice, followed by the emotion, style or sound to which the expression is targeted. Also, we detail the Input to the system (score, lyrics, tempo, audio recording, etc). The last column lists the language dependency of each method, if any.

We have collected samples from most of the approaches in order to help to easily listen to the results of the reviewed expression control approaches. The reader will observe several differences among them. First, some samples consist of a cappella singing voice, and others are presented with background music which may mask the synthesized voice and complicate the perception of the generated expression. Second, samples correspond to different songs, which makes it difficult to compare approaches. Concerning the lyrics, though in most cases these belong to a particular language, in some the lyrics are made by repeating the same syllable, such as /la/. We believe that the evaluation of a synthesized song can be performed more effectively in a language spoken by the listener. Finally, the quality of the synthetic voice is also affected by the type of synthesizer used in each sample. The difficulties in comparing them and the subsequent criticism are discussed in section 2.5 as well as in Chapter 7.

2.4.3 Performance driven approaches

These approaches use a real performance to control the synthesizer. The knowledge applied by the singer, implicit in the extracted data, can be used in two ways. In the first one, control parameters like F0, intensity, timing, etc from the reference recording are mapped to the input controls of the synthesizer

Type	Reference	Control features	Synthesizer	Style or emotion	Input	Language
Performance-driven	Meron (1999)	Timing, F0, intensity, singer's formant cluster	Unit-selection	Opera	Score, singing voice	German
	Janer et al. (2006)	Timing, F0, intensity, vibrato	Sample-based	Generic	Lyrics, singing voice, MIDI notes	Spanish
	Nakano & Goto (2009)	Timing, F0, intensity	Sample-based	Popular Music (RWC database)	Lyrics, singing voice	Japanese
	Nakano & Goto (2011)	Timing, F0, intensity, timbre	Sample-based	Music Genre (RWC database)	Lyrics, singing voice	Japanese
	Saitou et al. (2007)	Timing, F0, timbre	Resynthesis of speech	Children's songs	Score, tempo, speech	Japanese
Rule-based	Sundberg (2006)	Consonant duration, vowel onset, timing, timbre changes, formant tuning, overtone singing, articulation silence to note	Formant synthesis	Opera	Score, MIDI, or keyboard	Any
	Alonso (2004)	Note timing, micro-pauses, tempo and phrasing, intensity, pitch, vibrato and tremolo, timbre quality	Sample-based	Angry, sad, happy	Score, lyrics, tempo, expressive intentions	Swedish, English
	Bonada (2008)	Timbre (manual), phonetics, timing, intensity, musical articulation, sustains, vibrato and tremolo (rate and depth)	Sample-based	Generic	Score, lyrics, tempo	Any
Statistical modeling	Saino et al. (2006)	Timbre, pitch, timing (time-lag)	HMM-based	Children's songs	Score and lyrics	Japanese
	Oura & Mase (2010)	Pitch, vibrato and tremolo, timbre, source, timing	HMM-based	Children's songs	MusicXML score	Japanese, English
	Saino et al. (2010)	Baseline pitch (relative to note), vibrato rate and depth (not tremolo), intensity	Sample-based	Children's songs	Score (no lyrics to create models)	Japanese

Table 2.4: Comparison of approaches for Expression control in Singing Voice Synthesis.

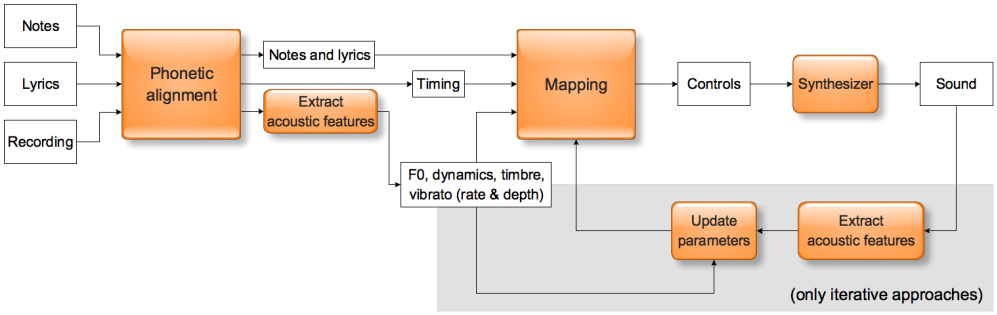


Figure 2.5: General framework for performance-driven approaches.

so that the rendered performance follows the input signal expression. Alternatively, speech audio containing the target lyrics is transformed in order to match pitch and timing of the input score. Fig. 2.5 summarizes the commonalities of these approaches on the inputs (reference audio, lyrics, and possibly the note sequence) and intermediate steps (phonetic alignment, acoustic feature extraction, and mapping) that generate internal data like timing information, acoustic features, and synthesizer controls used by the synthesizer.

In Table 2.5 we summarize the correspondence between the extracted acoustic features and the synthesis parameters for each of these works. The extracted F0 can be mapped directly into the F0 control parameter, processed into a smoothed and continuous version, or split into the MIDI note, pitch bend, and its sensitivity parameters. Vibrato can be implicitly modeled in the pitch contour, extracted from the input, or selected from a database. Energy is generally mapped directly into dynamics. From the phonetic alignment, note onsets and durations are derived, mapped directly to phoneme timing, or mapped either to onsets of vowels or voiced phonemes. Concerning timbre, some approaches focus on the singer's formant cluster and in a more complex case the output timbre comes from a mixture of different voice quality databases.

Approaches using estimated controls achieve different levels of robustness depending on the singing voice synthesizers and voice databases. In the system presented in Meron (1999), a unit selection framework is used to create a singing voice synthesizer from a particular singer's recording in a nearly automatic procedure. In comparison to sample-based system, where the design criterion is to minimize the size of the voice database with only one possible unit sample (e.g. diphones), the criterion in unit selection is related to redundancy in order to allow the selection of consecutive units in the database, at the expense of having a larger database. The system automatically segments the recorded voice into phonemes by aligning it to the score and feeding the derived segmentation constraints to an HMM recognition system. Units are selected to minimize a cost function that scores the amount of time, fre-

Acoustic features	Meron (1999)	Janer et al. (2006)	Nakano & Goto (2009)	Nakano & Goto (2011)	Saitou et al. (2007)
F0	F0	Smoothed and continuous pitch	MIDI note number, pitch bend and sensitivity	MIDI note number, pitch bend and sensitivity	F0
Vibrato	Included in F0 implicitly	Vibratos from input or from DB singer	Included in F0 implicitly	Included in F0 implicitly	Included in F0 implicitly
Energy	Dynamics	Dynamics	Dynamics	Dynamics	Dynamics
Phonetic alignment	Phoneme timing	Onsets of vowels or voiced phonemes	Note onset and duration	Note onset and duration	Phoneme timing
Timbre	Singer's formant cluster amplitude	Not used	Not used	Mixing different voice quality DBs	Singer's formant cluster amplitude and AM of the synthesized signal

Table 2.5: Mapping from acoustic features to synthesizer controls.

quency, and timbre transformations. Finally, units are concatenated. In this approach, the main effort is put on the synthesis engine. Although it uses a unit selection-based synthesizer, the expression controls for pitch, timing, dynamics, and timbre like the singer's formant are extracted from a reference singing performance of the target score. These parameters are directly used by the synthesizer to modify the selected units with a combination of sinusoidal modeling with PSOLA called SM-PSOLA. Editing is allowed by letting the user participate in the unit selection process, change some decisions, and modify the unit boundaries. Unfortunately, this approach only manipulates the singer's formant feature of timbre so that other significant timbre related features in opera singing style are not handled.

In Janer et al. (2006), the followed steps are: extraction of acoustic features like energy, F0, and automatic detection of vibrato sections, mapping into synthesis parameters, and phonetic alignment. The mapped controls and the input score are used to build an internal score that matches the target timing, pitch, and dynamics, and minimizes the transformation cost of samples from a database. However, this approach is limited since timbre is not handled and also because the expression features of the synthesized performance are not compared to the input values. Since this approach lacks a direct mapping of acoustic features to control parameters, these differences are likely to happen. On the other hand, the possibility of using a singer DB to produce vibratos other than the extracted ones from the reference recording provides a new degree of freedom to the user.

Toward a more robust methodology to estimate the parameters, in Nakano & Goto (2009) the authors study an iterative approach that takes the target singing performance and lyrics as. The musical score or note sequence is automatically generated from the input. The first iteration provides an initialization of the system similar to the previous approach (Janer et al., 2006). At this point these controls can be manually edited by applying pitch transposition, correction, vibrato modifications, and pitch and intensity smoothing. The

iterative process continues by analyzing the synthesized waveform and adjusting the control parameters so that in the next iteration the results are closer to the expected performance. In Nakano & Goto (2011), the authors extend this approach by including timbre. Using different voice quality databases from the same singer, the corresponding versions of the target song are synthesized as in the previous approach. The system extracts the spectral envelopes of each one to build a 3-dimensional voice timbre space. Next, a temporal trajectory in this space is estimated from the reference target performance in order to represent its spectral timbre changes. Finally, singing voice synthesis output is generated using the estimated trajectory to imitate the target timbre change. Although expression control is more robust than the previous approach thanks to iteratively updating the parameters and by allowing a certain degree of timbre control, these approaches also have some limitations. First, it cannot be assured that the iterative process will converge to the optimal set of parameter values. Secondly, the timbre control is limited to the variability within the set of available voice quality databases.

In Saitou et al. (2007), naturally-spoken readings of the target lyrics are transformed into singing voice by matching the target song properties described in the musical score. Other input data are the phonetic segmentation and the synchronization of phonemes and notes. The approach first extracts acoustic features like F0, spectral envelope, and the aperiodicity index from the input speech. Then, a continuous F0 contour is generated from discrete notes, phoneme durations are lengthened, and the singer's formant cluster is generated. The fundamental frequency contour takes into account four types of fluctuations, namely, overshoot (F0 exceeds the target note after a note change), vibrato, preparation (similar to overshoot before the note change), and fine fluctuations. The first three types of F0 fluctuations are modeled by a single second-order transfer function that depends mainly on a damping coefficient, a gain factor and a natural frequency. A rule-based approach is followed for controlling phoneme durations by splitting consonant-to-vowel transitions into three parts. First, the transition duration is not modified for singing. Then, the consonant part is transformed based on a comparative study of speech and singing voices. Finally, the vowel section is modified so that the duration of the three parts matches the note duration. Finally, with respect to timbre, the singer's formant cluster is handled by an emphasis function in the spectral domain centered at 3 kHz. Amplitude modulation is also applied to the synthesized singing voice according to the generated vibratos parameters. Although we have classified this approach into the performance-driven section since the core data is found in the input speech recording, some aspects are modeled like the transfer function for F0, rules for phonetic duration, and a filter for the singer's formant cluster. Similarly to Meron (1999), in this approach timbre control is limited to the singer formant, so that the system cannot change other timbre features. However, if the reference speech recording contains voice quality variations that fit the target song, this can add some naturalness to

Acoustic features	Dependencies
Consonant duration	Previous vowel length
Vowel onset	Synchronized with timing
Formant frequencies	Voice classification
Formant frequencies	Pitch, if otherwise F0 would exceed the first formant
Spectrum slope	Decrease with increasing intensity
Vibrato	Increase depth with increasing intensity
Pitch in coloratura passages	Each note represented as a vibrato cycle
Pitch phrase attack (and release)	At pitch start (end) from (at) 11 semitones below target F0

Table 2.6: Singing voice related KTH rules' dependencies.

the synthesized singing performance.

Performance-driven approaches achieve a highly expressive control since performances implicitly contain knowledge naturally applied by the singer. These approaches become especially convenient for creating parallel database recordings which are used in voice conversion approaches (Doi et al., 2012). On the other hand, the phonetic segmentation may cause timing errors if not manually corrected. The non-iterative approach lacks robustness because the differences between input controls and the extracted ones from the synthesized sound are not corrected. In Nakano & Goto (2011) timbre control is limited by the number of available voice qualities. We note that a human voice input for natural singing control is required for these approaches, which can be considered as a limitation since it may not be available in most cases. When such a reference is not given, other approaches are necessary to derive singing control parameters from the input musical score.

2.4.4 Rule-based approaches

Rules can be derived from work with synthesizing and analyzing sung performances. Applying an analysis-by-synthesis method an ambitious rule-based system for Western music was developed at KTH in the 1970s and improved over the last three decades (Sundberg, 2006). By synthesizing sung performances, this method aims at identifying acoustic features that are perceptually important either individually or jointly (Friberg et al., 2009). The process of formulating a rule is iterative. First a tentative rule is formulated and implemented and the resulting synthesis is assessed. If its effect on the performance needs to be changed or improved, the rule is modified and the effect of the resulting performance is again assessed. On the basis of parameters such as phrasing, timing, metrics, note articulation, and intonation, the rules modify pitch, dynamics, and timing. Rules can be combined to model emotional expressions as well as different musical styles. Table 2.6 lists some of the acoustic features and their dependencies.

The rules reflect both physical and musical phenomena. Some rules are compulsory and others optional. The Consonant duration rule, which length-

ens consonants following short vowels, applies also to speech in some languages. The Vowel onset rule corresponds to the general principle that the vowel onset is synchronized with the onset of the accompaniment, even though lag and lead of onset are often used for expressive purposes (Sundberg & Bauer-Huppmann, 2007). The Spectrum slope rule is compulsory, as it reflects the fact that vocal loudness is controlled by subglottal pressure and an increase of this pressure leads to a less steeply sloping spectrum envelope. The rule Pitch in coloratura passages implies that the fundamental frequency makes a rising-falling gesture around the target frequency in legato sequences of short notes (Sundberg, 1981). The Pitch phrase attack, in the lab jargon referred as the "Bull's roaring onset", is an ornament used in excited moods, and would be completely out of place in a tender context. Interestingly, results close to the KTH rules have been confirmed by machine learning approaches (Marinescu & Ramirez, 2011).

A selection of the KTH rules (Friberg et al., 2009) has been applied to the Vocaloid synthesizer (Alonso, 2004). Features are considered at note level (start and end times), intra and inter note (within and between note changes) and to timbre variations (not related to KTH rules). The system implementation is detailed in Bresin & Friberg (2000), along with the acoustic cues which are relevant for conveying basic emotions such as anger, fear, happiness, sadness, and love-tenderness (Juslin & Laukka, 2003). The rules are combined in expressive palettes indicating to what degree rules need to be applied to convey a target emotion. The relationship between application level, rules, and acoustic features is shown in Table 2.7. As an example of the complexity of the rules, the punctuation rule at note level inserts a 20 milliseconds micropause if a note is three tones lower than the next one and its duration is 20% larger. Given that this work uses a sample-based synthesizer, voice quality modifications are applied to the retrieved samples. In this case, the timbre variations are limited to rules affecting brightness, roughness, and breathiness, and therefore do not cover the expressive possibilities of a real singer.

Apart from the KTH rules, in corpus-derived rule-based systems heuristic rules are obtained to control singing expression by observing recorded performances. In Bonada & Serra (2007), expression controls are generated from high-level performance scores where the user specifies note articulation, pitch, intensity, and vibrato data which is used to retrieve templates from recorded samples. This work, used in the Vocaloid synthesizer (Kenmochi & Ohshita, 2007), models the singer's performance with heuristic rules (Bonada, 2008). The parametric model is based on anchor points for pitch and intensity, which are manually derived from the observation of a small set of recordings. At synthesis, the control contours are obtained by interpolating the anchor points generated by the model. The number of points used for each note depends on its absolute duration. The phonetics relationship with timing is handled by synchronizing the vowel onset with the note onset. Moreover, manual editing is permitted for the degree of articulation application as well as its duration, pitch and dynamics contours, phonetic transcription, timing, vibrato and

Level	Rules	Affected acoustic features
Note	Duration contrast	Decrease duration and intensity of short notes placed next to long notes
	Punctuation	Insert micro-pauses in certain pitch interval and durations combinations
	Tempo	Constant value for the note sequence (measured in bpm)
	Intensity	Smooth/strong energy levels, high pitch notes intensity increases 3 dB/octave
	Transitions	Legato, staccato (pause is set to more than 30% of inter-onset interval)
	Phrasing arch	Increase/decrease tempo at phrase beginning/end, same for energy
	Final ritardando	Decrease tempo at the end of a piece
Inter note	Attack	Pitch shape from starting pitch until target note, energy increases smoothly
	Note articulation	Pitch shape from the starting to the ending note, smooth energy
	Release	Energy decreases smoothly to 0, duration is manually edited
	Vibrato and tremolo	Manual control of position, depth, and rate (cosine function, random fluctuations)
Timbre	Brightness	Increase high frequencies depending on energy
	Roughness	Spectral irregularities
	Breathiness	Manual control of noise level (not included in emotion palettes)

Table 2.7: Selection of rules for singing voice: level of application and affected acoustic features.

tremolo depth and rate, and timbre characteristics.

The advantage of these approaches is that they are relatively straightforward and completely deterministic. Random variations can be easily introduced so that the generated contours are different for each new synthesis of the same score, resulting in distinct interpretations. The main drawbacks are that either the models are based on few observations that do not fully represent a given style, or they are more elaborate but become unwieldy due to the complexity of the rules.

2.4.5 Statistical modeling approaches

Several approaches have been used to statistically model and characterize expression control parameters using Hidden Markov Models (HMMs). They have a common precedent in speech synthesis (Yoshimura et al., 1999), where the parameters like spectrum, F0 and state duration are jointly modeled. Compared to unit selection, HMM-based approaches tend to produce lower speech quality, but they need a smaller dataset to train the system without needing to cover all combinations of contextual factors. Modeling singing voice

HMM-based approaches	Levels	Contextual factors
Saino et al. (2006)	Phoneme	P/C/N phonemes
	Note	P/C/N note pitches, durations, and positions within the measure
Oura & Mase (2010)	Phoneme	Five phonemes (central and two preceding and succeeding)
	Mora	Number of phonemes in the P/C/N mora
		Position of the P/C/N mora in the note
	Note	Musical tone, key, tempo, length, and dynamics of the P/C/N note
		Position of the current note in the current measure and phrase
		Ties and slurred articulation flag
		Distance between current note and next/previous accent and staccato
		Position of the current note in the current crescendo or decrescendo
	Phrase	Number of phonemes and moras in the P/C/N phrase
	Song	Number of phonemes, moras, and phrases in the song
Saino et al. (2010)	Note region	Manually segmented behaviour types (beginning, sustained, ending)
	Note	MIDI note number and duration (in 50 ms units)
		Detuning: model pitch by the relative difference to the nominal note

Table 2.8: Contextual factors HMM-based systems (P/C/N stands for: Previous, Current, and Next).

with HMMs amounts to using similar contextual data as for speech synthesis, adapted to singing voice specificities. Moreover, new voice characteristics can be easily generated by changing the HMM parameters.

These systems operate in two phases: training and synthesis. In the training part, acoustic features are first extracted from the training recordings like F0, intensity, vibrato parameters, and mel-cepstrum coefficients. Contextual labels, that is to say, the relationships of each note, phoneme, phrase with the preceding and succeeding values, are derived from the corresponding score and lyrics. Contextual labels vary in their scope at different levels, such as phoneme, note, or phrase, according to the approach, as summarized in Table 2.8. This contextual data is used to build the HMMs that relate how these acoustic features behave according to the clustered contexts. The phoneme timing is also modeled in some approaches. These generic steps for the training part in HMM-based synthesis are summarized in Fig. 2.6. The figure shows several blocks found in the literature, which might not be present simultaneously in each approach. We refer to Yoshimura et al. (1999) for the detailed computations that HMM training involves.

In the synthesis part, given a target score, contextual labels are derived as in the training part from the note sequence and lyrics. Models can be used in two ways. All necessary parameters for singing voice synthesis can be generated from them, therefore state durations, F0, vibrato and mel-cepstrum

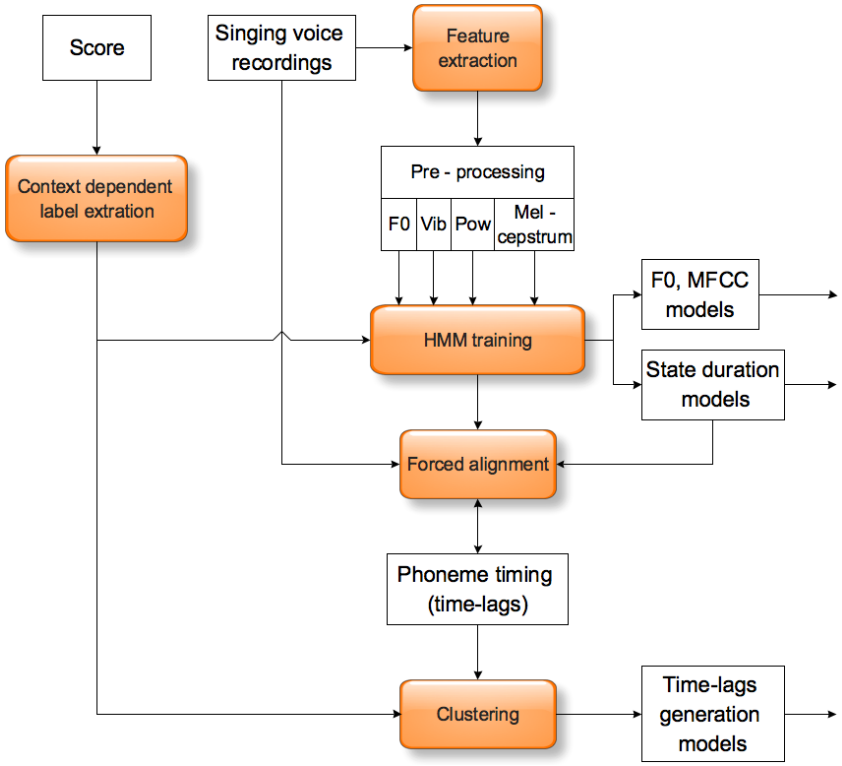


Figure 2.6: Generic blocks for the training part of HMM-based approaches.

observations are generated to synthesize the singing voice. On the other hand, if another synthesizer is used, only control parameters such as F0, vibrato depth and rate, and dynamics need to be generated which are then used as input of the synthesizer.

As introduced in Section 2.4.1, statistical methods jointly model the largest set of expression features among the reviewed approaches. This gives them a better generalization ability. As long as singing recordings for training involve different voice qualities, singing styles or emotions, and the target language phonemes, these will be reproducible at synthesis given the appropriate context labeling. Model interpolation allows new models to be created as a combination of existing ones. New voice qualities can be created by modifying the timbre parameters. However, this flexibility is possible at the expense of having enough training recordings to cover the combinations of the target singing styles and voice qualities. In the simplest case, a training database of a set of songs representing a single singer and style in a particular language would be enough to synthesize it. As a drawback, training HMMs with large databases tends to produce smoother time series than the original training data, which may be perceived as non-natural.

In Saino et al. (2006), a corpus-based singing voice synthesis system based

Approach	DB style	Length	Feature extraction
Saino et al. (2006)	60 Japanese children's songs (male singer)	72 min	sampling at 16 KHz, 25ms Blackman window, 5ms shift 0-24 MFCCs, log F0 spectral and log F0 feature vectors also include delta and delta-deltas
Oura & Mase (2010)	70 Japanese children's songs (female singer)	70 min	sampling at 48 KHz, windowed, 5ms shift 0-48 STRAIGHT MFCCs, log F0 (Âś a halftone pitch shifts), vibrato depth (cents) and rate (Hz) spectral, log F0 and vibrato feature vectors also include delta and delta-deltas
Saino et al. (2010)	5 Japanese childen's Songs (deep bendy)	5 min	25ms window, 5ms shift melody, vibrato shape and rate, dynamics feature vectors include delta and delta-deltas

Table 2.9: Training DBs and extracted features in HMM-based systems.

on HMMs is presented. The contexts are related to phonemes, note F0 values, and note durations and positions, as we show in Table 2.8 (dynamics are not included). Also, synchronization between notes and phonemes needs to be handled adequately, mainly because phoneme timing does not strictly follow the score timing; and phonemes might be advanced with respect to the nominal note onsets (negative time-lag).

In this approach, the training part generates three models. One for the spectrum where MFCCs are estimated with STRAIGHT and excitation (F0) parts, extracted from the training database, another for the duration of context-dependent states, and a third one to model the time-lag. The latter ones model note timing and phoneme durations of real performances, which are different to what can be inferred from the musical score and its tempo. Time-lags are obtained by forced alignment of the training data with context-dependent HMMs. Then, the computed time-lags are related to their contextual factors and clustered by a decision-tree. Feature extraction and training configuration details are shown in Table 2.9.

The singing voice is synthesized in five steps. First, the input score (note sequence and lyrics) is analyzed to determine note duration and contextual factors. Then, a context-dependent label sequence of contextual factors as shown in Table 2.8 is generated. Then, the song HMM is generated and its state durations are jointly determined with the note time-lags. Next, spectral and F0 parameters are generated, which are used to synthesize the singing voice. The authors claim that the synthesis performance achieves a natural singing voice which simulates expression elements of the target singer such as voice quality and the singing style (F0 and time-lag).

In this work, the training database consists of 72 minutes of a male voice singing 60 Japanese children's songs in a single voice quality. These are the characteristics that the system can reproduce in a target song. The main limitation of this approach is that contextual factors scope is designed only to cover phoneme and note descriptors. Longer scopes than just the previous and next note are necessary to model higher level expressive features such

as phrasing. Although we could not get samples from this work, an evolved system is presented next.

The system presented in Saino et al. (2006) has been improved, and is publicly available as Sinsy, an online singing voice synthesizer (Oura & Mase, 2010). The new characteristics of the system include reading input files in MusicXML format[2] with F0, lyrics, tempo, key, beat, and dynamics, also extended contextual factors used in the training part, vibrato rate and depth modeling, and a reduction of the computational cost. Vibrato is jointly modeled with the spectrum and F0 by including depth and rate in the observation vector in the training step.

The new set of contexts, automatically extracted from the musical score and lyrics, used by the Sinsy approach are also shown in Table 2.8. These factors describe the context such as previous, current, and next data at different hierarchical levels, namely, phoneme, mora (the sound unit containing one or two phonemes in Japanese), note, phrase, and the entire song. Some of them are strictly related to musical expression aspects, such as musical tone, key, tempo, length and dynamics of notes, articulation flags, or distance to accents and staccatos.

Similarly to the previous work, in this case the training database consists of 70 minutes of a female voice singing 70 Japanese children's songs in a single voice quality. However, it is able to reproduce more realistic expression control since vibrato parameters are also extracted and modeled. Notes are described with a much richer set of factors than the previous work. Another major improvement is the scope of the contextual factors shown in Table 2.8, which spans from the phoneme level up to the whole song and therefore being able to model phrasing.

In Saino et al. (2010), a statistical method is able to model singing styles. This approach focuses on baseline F0, vibrato features like its extent, rate, and evolution over time, not tremolo, and dynamics. These parameters control the Vocaloid synthesizer, and so timbre is not controlled by the singing style modeling system, but is dependent on the database.

A preprocessing step is introduced after extracting the acoustic features like F0 and dynamics in order to get rid of the micro-prosody effects on such parameters, by interpolating F0 in unvoiced sections and flattening F0 valleys of certain consonants. The main assumption here is that expression is not affected by phonetics, which is reflected in erasing such dependencies in the initial preprocessing step, and also in training note HMMs instead of phoneme HMMs. Also, manual checking is done to avoid errors in F0 estimation and MIDI events like note on and note off estimated from the phonetic segmentation alignment. A novel approach estimates vibrato shape and rate, which at synthesis is added to the generated baseline melody parameter. The shape is represented with the low frequency bins of the Fourier Transform of single

[2] http://www.musicxml.com/

vibrato cycles. In this approach, context-dependent HMMs model the expression parameters which are summarized in Table 2.8. Feature vectors contain melody, vibrato shape and rate, and dynamics components.

This last HMM-based work focuses on several control features except timbre, which is handled by the Vocaloid synthesizer. This makes the training database much smaller in size. It consists of 5 minutes of 5 Japanese children's songs, since there is no need to cover a set of phonemes. Contextual factors are rich at a note level, since the notes are divided into 3 parts (begin, sustain, and end), and the detuning is also modeled relatively to the nominal note. On the other hand, this system lacks of the modeling of wider temporal aspects such as phrasing.

2.4.6 When to use each approach?

The answer to this question has several considerations: from the limitations of each approach, to whether singing voice recordings are available or not since these are needed in model training or unit selection, the reason for synthesizing a song which could be for database creation or rule testing, or flexibility requirements like model interpolation. In this section we provide a brief guideline on the suitability of each type of approach.

Performance-driven approaches are suitable to be applied, by definition, when the target performance is available, since the expression of the singer is implicit in the reference audio and it can be used to control the synthesizer. Another example of applicability is the creation of parallel databases for different purposes like voice conversion (Doi et al., 2012). An application example for the case of speech to singing synthesis is the generation of singing performances for untrained singers, whose timbre is taken from the speech recording and the expression for pitch and dynamics can be obtained from a professional singer.

Rule-based approaches are suitable to be applied to verify the defined rules and also to see how these are combined, for example to convey a certain emotion. If no recordings are available, rules can still be defined with the help of an expert, so that these approaches are not fully dependent on singing voice databases.

Statistical modeling approaches are also flexible, given that it is possible to interpolate models and to create new voice characteristics. They have the advantage that in some cases these are part of complete singing voice synthesis systems, that is to say, the ones that have the score as input and that generate both the expression parameters and output voice.

Similarly to rule-based and statistical modeling approaches, unit selection approaches do not need the target performance, although they can benefit from it. On the other hand, unit selection approaches share a common characteristic with performance-driven approaches. The implicit knowledge of the singer is contained in the recordings, although in unit selection it is extracted from

shorter audio segments. Unlike statistical models, no training step is needed, so that the expression databases can be improved just by adding new labeled singing voice recordings.

In the following section we review the evaluation strategies of the expression control approaches, identify some deficiencies, and finally propose a possible solution.

2.5 Evaluation

2.5.1 Current strategies

In Section 1.2.2, we introduced that a score can be interpreted in several acceptable ways, which makes expression a subjective aspect to rate. However, "*procedures for systematic and rigorous evaluation do not seem to exist today*" (Rodet, 2002) (p. 105), especially if there is no ground-truth to compare with. In this section, we first summarize typical evaluation strategies.

Expression control can be evaluated from subjective or objective perspectives. The former typically consists of listening tests where participants perceptually evaluate some psychoacoustic characteristic like voice quality, vibrato, and overall expressiveness of the generated audio files. A common scale is the mean opinion score (MOS), with a range from 1 (bad) to 5 (good). In pairwise comparisons, using two audio files obtained with different system configurations, preference tests rate which option achieves a better performance. Objective evaluations help to compare how well the generated expression controls match a reference real performance by computing an error. Within the reviewed works, subjective tests outnumber the objective evaluations. In Table 2.10 the evaluations are summarized. For each approach, several details are provided like a description of the evaluation (style, voice quality, naturalness, expression, and singer skills), the different rated tests, and information on the subjects if available. Objective tests are done only for performance-driven approaches, that is to say, when a ground-truth is available. In the other approaches, no reference is directly used for comparison, so that only subjective tests are carried out. However, in the absence of a reference of the same target song, the generated performances could be compared to the recording of another song, as is done in the case of speech synthesis.

2.5.2 Discussion

In our opinion, the described evaluation strategies are devised for evaluating a specific system, and therefore focus on a concrete set of characteristics particularly relevant for that system. For instance, the evaluations summarized in Table 2.10 do not include comparisons to other approaches. This is due to the substantial differences between systems, which make the evaluation and comparison between them a complex task. These differences can be noted in

		Tests		
Type	**Approach**	**Evaluation**	**Description**	**Subjects**
Performance-driven	Meron (1999)	Subjective	Rate voice quality with pitch modification of 10 pairs of sentences (SM-PSOLA vs TD-PSOLA)	10 subjects
	Janer et al. (2006)	Subjective	Informal listening test	Not specified
	Nakano & Goto (2009)	Objective	Two tests: lyrics alignment and mean error value of each iteration for F0 and intensity compared to target	No subjects
	Nakano & Goto (2011)	Objective	Two tests: 3D voice timbre representation and Euclidean distance between real and measured timbre	No subjects
	Saitou et al. (2007)	Subjective	Paired comparisons of different configurations to rate naturalness of synthesis in a 7 step scale (-3 to 3)	10 students with normal hearing ability
Rule-based	Sundberg (2006)	Subjective	Listening tests of particular acoustic features	15 singers or singing teachers
	Alonso (2004)	None	None	None
	Bonada (2008)	Subjective	Listening tests ratings (1-5)	50 subjects with different levels of musical training
Statistical modelling	Saino et al. (2006)	Subjective	Listening test (1-5 ratings) of 15 musical phrases. Two tests: with and without time-lag model	14 subjects
	Oura & Mase (2010)	Subjective	Not detailed (based on Saino et al. (2006))	Not specified
	Saino et al. (2010)	Subjective	Rate style and naturalness listening tests ratings (1-5) of 10 random phrases per subject	17 subjects with different levels of musical training

Table 2.10: Conducted subjective and objective evaluations per approach.

the audio excerpts of the accompanying website to this dissertation, which have been introduced in Section 2.4.2. At this stage, it is difficult to decide which method more efficiently evokes a certain emotion or style, performs better vibratos, changes the voice quality in a better way, or has a better timing control. There are limitations in achieving such a comprehensive evaluation and comparing the synthesized material.

2.6 Conclusion

This chapter was devoted to review the scientific background on singing voice synthesis and expression control. We have explained the singing voice production mechanism both from a physical perspective and an artificial point of view. We have also detailed the main expression features related to melody, dynamics, rhythm, and timbre. The approaches for expression control have been described, compared, and classified into performance-driven, rule-based, and statistical models. Finally, the evaluation strategies have been reviewed and discussed.

Throughout this chapter, we also addressed the advantages and disadvantages of the selected approaches. The drawbacks of the reviewed approaches show the requirements for any new proposed system. It should not suffer from requiring the target song to control the synthesizer, it should not be too complex, and it should avoid the smoothing issues from the statistical methods. In the next chapter we proposed the unit-selection based method as a possible solution.

CHAPTER 3

Expression database creation

In the previous chapters we have provided an overview of the systems we propose in this thesis as well as the state of the art context on expression control for singing voice synthesis. As we have shown in Fig. 1.7, several expression databases (DB) are shared between the building blocks. In this chapter we describe how each database is designed, recorded, and labeled.

3.1 Introduction

In the proposed systems, several expression databases are used to control pitch and dynamics. To do so, we have considered handling pitch and dynamics jointly taking the corresponding contours from the same recordings.

The aim of this chapter is to explain the design of the recording scripts that the singer sang in the studio and from which pitch and dynamicsneed to be estimated to model the singer's particular style. In this thesis we have selected jazz as the target style. Such scripts need to fulfill several requirements as explained in section 3.2. One of them is related to how well the sequences of notes in the script represent the target style that are designed to cover. Another requirement refers to the lyrics content, which should help to have continuous pitch and dynamics contours which are not affected by microposody.

We have devised two strategies to build the expression database for pitch and dynamics. The note sequences can be generated automatically from the study of several scores belonging to the same style (Sec. 3.3) deriving into a set of melodic exercises that contain the most common note combinations in terms of its duration, pitch, and position in the measure. On the contrary, real songs from the same style can be directly taken as melodic exercises (Sec. 3.4). These two databases, namely the Systematic and the Song expression databases respectively, are summarized in Table 3.1.

In section 3.5 we detail the methodology we have followed to label each database, which basically aims to obtain a set of note characteristics (note pitch values, start and end times, note transition times, and the note strength)

DB name	# files	Duration (mm:ss)
Systematic DB	70	11:59
Song DB	17	18:29

Table 3.1: Summarized data of the Systematic and the Song expression databases.

and also vibrato related characteristics (start and end times, and depth and rate evolution over time). Finally, in section 3.6 we explain the advantages and disadvantages of the 2 proposed expression databases.

3.2 Database design requirements

There are several conditions that we impose as requirements to be met by the expression databases, either related to the content or the way of recording. We have identified 3 requirements, namely, the notes coverage (Sec. 3.2.1), the need to avoid microprosody effects on the extracted features (Sec. 3.2.2), and the usage of a musical background to convey the appropriate style (Sec. 3.2.3). These requirements are explained in this section.

3.2.1 Coverage

Our interest is that the expression databases contain as many elements (and their combinations) that can appear in a target score as possible. By elements we especially mean several properties of notes, like pitch, duration, and strength (related to the note position within the measure).

In the unit selection proposed system, covering a high amount of combinations of note properties implies that the selected notes would, in theory, suffer from less amount of transformation, since the selected units will be closer to the target score. In the statistical based systems, a high coverage implies that the system can be provided with enough observations for the training step.

Such a high coverage can be achieved in several ways. The simplest way is by recording lots of data, although this is a time consuming task both in recording time and especially to label all the data. However, regarding pitch coverage, we can generate new contexts by pitch shifting the pitch contours. Although the note intervals and durations do not change with such transformation, adding the shifted contours provide new contours as if these had been recorded at a higher or lower pitch. The second option is the one we have chosen to increase the coverage.

3.2.2 Lyrics and microprosody

Our aim is to estimate pitch and dynamics contours from the recordings which are continuous and with the least amount of fluctuations not attributable to

Figure 3.1: Recording room

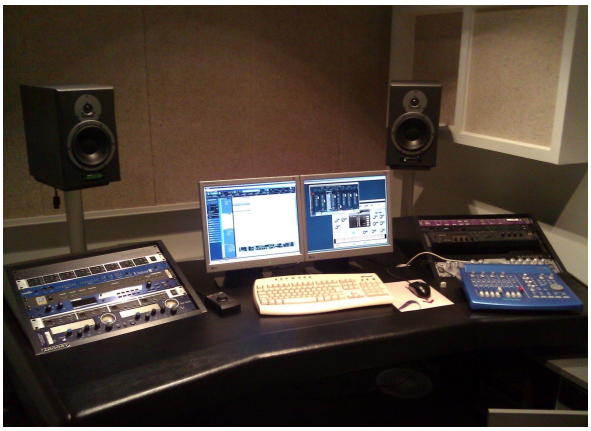


Figure 3.2: Sound studio

expression. In other words, we want to avoid microprosody effects on the extracted features, that is fluctuations originated by the sequence of phonemes sung.

Lyrics, and more concretely their corresponding phonetics, have an effect on the estimated pitch and dynamics contours from the recordings. For instance, in unvoiced phonemes (like /s/) the pitch contour cannot be extracted, and in velar sounds (like /g/) pitch valleys are produced.

Taking these into account, we have decided to record our scripts without normal lyrics. Instead, we have used vowels, which are interleaved at every note change. These timbre changes are used to semi-automatically segment the pitch into notes. This is explained in section 3.5. For instance, the vowels we would use for a sequence of 4 notes would be /ua-i-a-i/, where the first note starting from silence would be the diphtong /ua/. We did some experiments with the syllable /na/ per note, which might useful for note onset detection but we discarded them because the consonant /n/ also introduces microprosodic effects as well.

3.2.3 Recordings

As we have introduced, the scripts recorded in the studio represent a particular style, which can either be songs from that style or melodic exercises generated automatically. However, acapella singing of these scripts with no external help may produce out of tune singing, variable tempo throughout a piece, and it may become difficult to evoke the singing style the scripts represent.

In order to try to avoid these problems and help the singer, we use background music during the recording session which is listened by the singer through the headphones. The melodic exercises share the same type of background music, since the sequence of notes are taken from the same scale and

Figure 3.3: Singer at the studio.

the same chord sequence is used. In the songs, although it may be easier to convey the style than in melodic exercises since they are known by the singer, it is still useful to use the background music to keep the tempo and sing in tune. To generate the accompaniments we have used the Sibelius score editor. One of its functionalities is to create the harmony for a sequence of notes.

The studio recordings have been held at the UPF facilities of La Nau building[1] at the Communication Campus in Poblenou neighbourhood, with a recording[2] and control rooms[3]. In Fig. 3.1, Fig. 3.2 we show images of the control and recording rooms. In Fig. 3.3 we show how the singer was placed with respect to the microphone.

3.3 Systematic expression database

In this section we explain the generation of melodic exercises not from real repertoires but automatically by looking at which note properties should be covered. Songs from real repertoires typically have the disadvantage of being redundant, so only a portion of an entire song introduces new note sequences. Also, in order to select which parts of a song to include as an exercise, it should be carefully studied.

[1] http://www.upf.edu/campus/en/comunicacio/nau.html
[2] http://www.upf.edu/bibtic/serveis/audiovisuals/edlanau/lanau01.html
[3] http://www.upf.edu/bibtic/serveis/audiovisuals/edlanau/lanau02.html

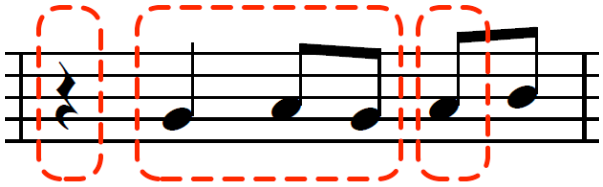


Figure 3.4: Unit of three notes with preceding silence and following note.

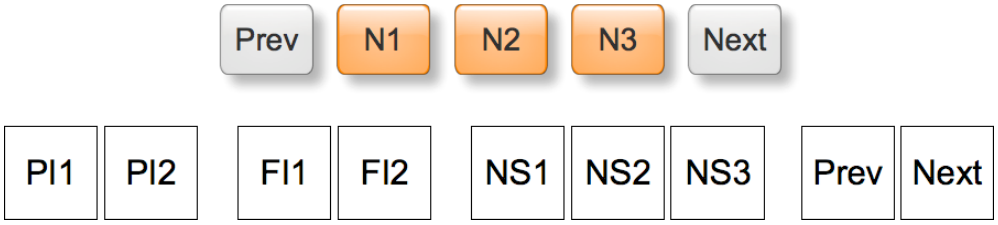


Figure 3.5: Unit and context features.

In the process of creating the expression database described in this section, the aim is to obtain melodic exercises by concatenating short melodic units generated in a systematic way, also including musical knowledge. First, a set of scores is statistically analyzed in order to know which feature values (note strengths and figures and pitch intervals in semitones) should be covered, their relevance and how these are connected. Then, dynamic programming is applied in order to generate melodic exercises as sequences of concatenated units. This section is based on Umbert et al. (2013b).

3.3.1 Units versus contexts

The basic elements of our systematic process of melodic exercises creation are units made up as sequences from one to three notes surrounded by a previous and following note or silence. An example is shown in Fig. 3.4. In this dissertation a note is defined mainly by the following properties: note strength, note duration (seconds), and the figure and pitch interval with the next one. Note strength (NS) is a measure for the accentuation of a note beat within a bar. Figure interval (FI) refers to the relationship between two consecutive note durations and the same applies to pitch interval (PI) with respect to the note frequencies. This data is shown in Fig. 3.5. We can see that for a sequence of 3 notes, there are 2 pitch intervals, 2 figure intervals, 3 note strength, and a previous and succeeding note or silence.

For each note property there are many possible combinations, which imply a great amount of units, especially in the case of sequences of three notes. This relates to the goal of the systematic database, which is to cover a high amount

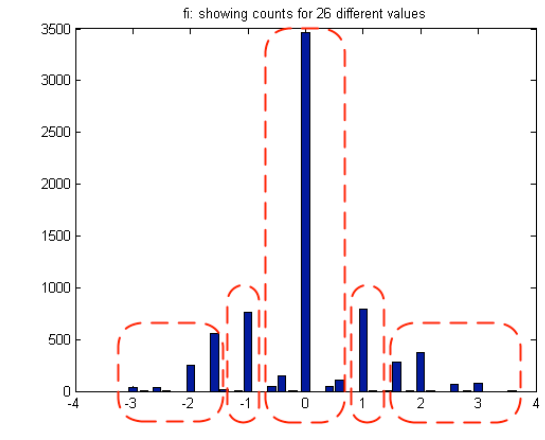


Figure 3.6: Figure interval distribution (in octaves) and clusters.

Cluster	Range of FI values
1	[-3, -1.585]
2	[-1.41, -1]
3	[-0.585, 0.585]
4	[1, 1.415]
5	[1.585, 3.585]

Figure 3.7: Figure interval cluster values.

of relevant note combinations. Therefore, the coverage criteria is not defined with respect to the units but related to a higher abstract unit or context. Each context comprises several possible units.

Thus, the relationship between units and contexts has to be defined by grouping the set of values of each note property into clusters. Once the clusters are set, it is possible to statistically analyze the transition probabilities between contexts according to the analyzed database. These probabilities are used to generate the systematic melodic exercises. Next, we explain both steps.

3.3.2 Statistical analysis and clustering

In order to study the values of the note properties that need to be covered, a set of songs belonging to the same style have been processed using Music21 (Cuthbert & Ariza, 2010), a Python toolkit to process music in symbolic form.

Since most of the processed units are three notes long, and each note is defined in terms of its strength, duration, and figure and pitch intervals, the possible number of units is enormous. As previously explained, in order to reduce the amount of units to cover, these are clustered into similar contexts.

In general, clusters have been organized so that close values are represented by the same cluster. In the case of pitch interval clusters, it has also been taken into account that within the same cluster all pitch intervals correspond to only ascending or descending intervals since we do not want to transform an ascending pitch contour to synthesize a descending one (and vice versa). Therefore, an interval of zero semitones (same consecutive notes) is grouped in a separate cluster. In the case of the figure interval, clusters do not need to follow the same constraint concerning the direction of the interval (ascending

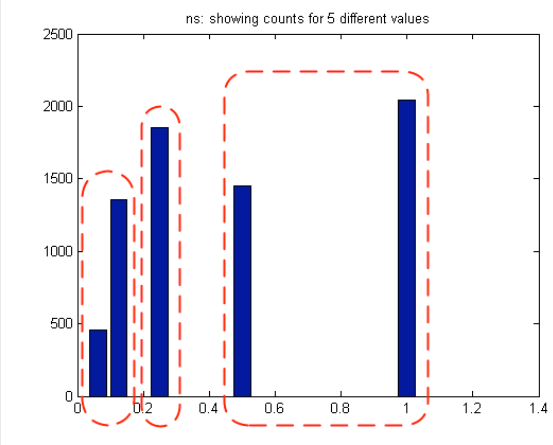


Figure 3.8: Note strength distribution and clusters.

Cluster	Range of NS values
1	[0.5, 1]
2	[0.25]
3	[0.125, 0.625]

Figure 3.9: Note strength cluster values.

or descending). Note strength clusters have been grouped according to the note accentuation within a measure.

In Fig. 3.6 and Table 3.7, the values distribution for the figure interval and their clustering is shown. Similarly, the note strength data is presented in Fig. 3.8 and Table 3.9, and the pitch interval data in Fig. 3.10 and Table 3.11.

Using this cluster representation, the context frequencies have been counted and the 90% most common ones have been selected to be covered, generating a list of 993 contexts of three notes. Also, the amount of connections between these selected contexts (by overlapping two or one notes or just concatenating them) has been computed to measure the transition probabilities among contexts. These contexts are a higher level representation of 1480 units.

3.3.3 Melodic exercises generation

Next, we proceed to explain the process followed to generate the melodic exercises as sequence of three note long units by dynamic programming (Viterbi algorithm). In a similar way exercises of two and one notes were generated. In these cases, the previous and following notes are considered to be silences, so the Viterbi algorithm was no longer necessary since unit overlapping does not apply. These exercises were generated in a more straightforward manner by taking one value per cluster to generate the contexts to cover.

Note strength grid

The Viterbi algorithm has been used in order to generate the sequence of melodic exercises of the systematic database. The temporal resolution, or tick, of each melodic exercise is defined by the minimum note length. In our case

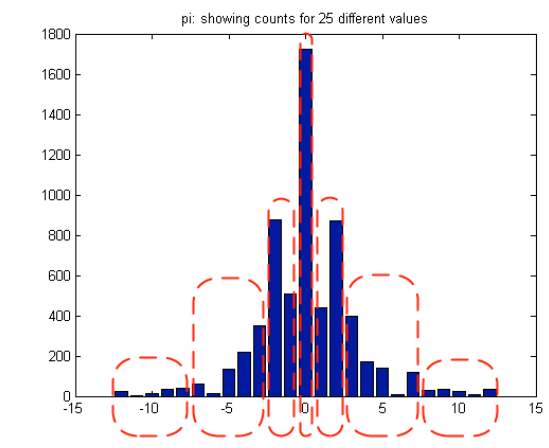


Figure 3.10: Pitch interval distribution (in semitones) and clusters.

Cluster	Range of PI values
1	[-12, -8]
2	[-7, -3]
3	[-2, -1]
4	[0]
5	[1, 2]
6	[3, 7]
7	[8, 12]

Figure 3.11: Pitch interval cluster values.

we have used a tick of an eighth note. The sequence of ticks defines a note strength grid which is used in order to know which units fit at each position in time.

Given the minimum note length that will be used in the systematic score, a grid can be generated which sets where notes can be placed and which their note strengths are at those positions. The length of this grid is related to the amount of measures per exercise.

For a minimum note length of an eighth note, the note strength grid for a single measure (4 beats, 8 ticks) is musically defined as shown in the following vector:

$$[1, 0.125, 0.25, 0.125, 0.5, 0.125, 0.25, 0.125] \tag{3.1}$$

Cost measures

At each (forward) step of the Viterbi algorithm, the cumulated cost of inserting a given database unit at a certain tick is computed using a set of cost functions. These cost functions handle the transitions between units according to the statistical information at context level computed (section 3.3.2). The cost functions also measure whether an instance fits in the grid and reusing a context is penalized. Harmony is managed by the preset accompaniment chords of the melodic exercises and how these and the unit notes match. Inserting silences in the middle of the exercise is also favored considering readability, in order to help the singer to breath in the middle of the performance. Also, the generated note pitches are constrained to the singers tessitura in order to facilitate singing the exercises. The cumulated cost for an evaluated node of the Viterbi matrix is obtained by adding these cost measures and are next detailed.

Table 3.2: Harmony costs.

Bar	Chord	C	D	E	F	G	A	B
1	C7	0	1	0	2	0	1	0
2	Am7	0	1	0	2	0	0	1
3	Dm	1	0	1	0	2	0	2
3	G7	2	0	2	0	0	1	0
4	C7	0	1	0	2	0	1	0.5

- Note strength cost: The first computed cost checks whether the note strengths features of the unit match the note strengths related to the tick position where is is intended to be inserted. If the unit does not fit, then it is not necessary to check all the other costs, and the total cost is set to infinity. For units that do fit, the cost is set to zero.

- Unit transition cost: The second computed cost relates to the transition between units. The result of the statistical analysis (the transition probability cost) provides this cost for an overlapping of two, one or zero notes (concatenation). These transition is computed for the current selected unit with respect to all possible previous units.

- Context repetition: Since the aim is to have the highest coverage possible with the minimum amount of melodic exercises, context repetition is taken account for penalization. Therefore, a history of all previously selected contexts is kept, so that if in the currently evaluated node path there is a context repetition, a cost proportional to the amount of repetitions is added. Although some context repetitions may appear in the final score, this cost favors the selection of different contexts. We handle this cost with an array that counts the number of times each context appears.

- Harmony cost: The harmony cost takes into account the chords for the melodic exercises. The same sequence of chords has been predefined for all exercises in order to make it easy for the singer: C7 (1st bar), Am7 (2nd bar), Dm (3rd bar 1st half), G7 (3rd bar 2nd half), C7 (4th bar). Those notes with cost zero are the ones belonging to the chord. Otherwise, it is more costly to add notes which do not match with the chord note information. In Table 3.2 the harmony costs are shown relating which notes are favored (zero cost) per chord and which ones are more penalized (non-zero cost).

- Silence insertion: Finally, since melodic exercises are four measures long (plus one as a break between exercises), and in order to make them less exhausting to sing, a silence has been included in the middle, at the end

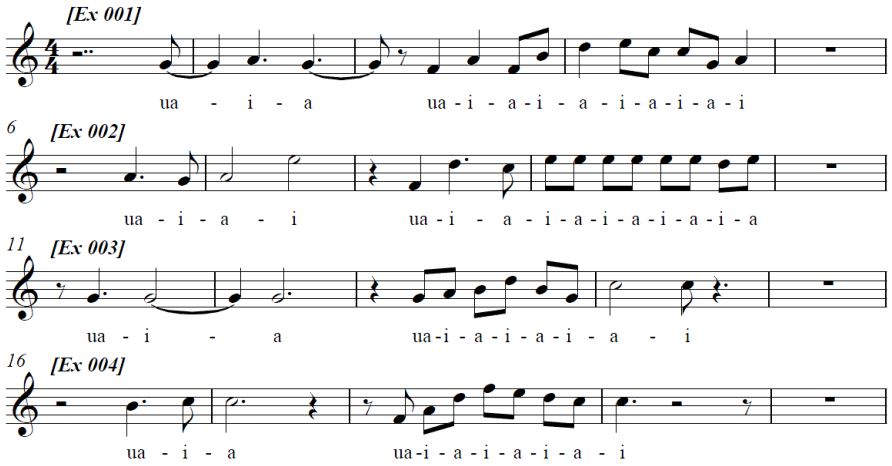


Figure 3.12: First systematic exercises.

of the second measure and at the beginning of the third one. Several tick candidates for inserting the pause are considered in the Viterbi paths and the least costly one is chosen.

Stop criteria

The algorithm stops generating melodic exercises depending on two conditions. The first one is related to the coverage. If all 993 contexts have been selected (one unit per context is enough) after the generation of a melodic exercise, the generation of exercises is stopped. This is controlled by the history of selected contexts as explained in the previous section.

The second stop criteria is related to the available recording session duration and the tempo of the generated score. If the accumulated duration of all exercises reaches the recording time, given the amount of measures per exercise and the bpm, then no more melodic exercises are generated.

Results

The systematic script has been generated by taking 57 jazz standard songs, setting the tessitura to one octave, a tempo of 71 bpm and a limit for the recording time of one hour. These constraints generate a recording script of 236 exercises and a coverage of 82% of contexts.

The generated melodic exercises as concatenation of three note long units can be downloaded in pdf and audio files are online [4] for Umbert et al. (2013b). The first 4 exercises of the systematic database are shown in Fig. 3.12. The

[4] http://mtg.upf.edu/publications/ExpressionControlinSingingVoiceSynthesis/

Index	Song name	Index	Song name
1	A foggy day	10	Polka dots and moonbeams
2	Alone together	11	Skylark
3	Angel eyes	12	Summertime
4	But not for me	13	Stella by starlight
5	Body and soul	14	The days of wine and roses
6	Everything happens to me	15	The nearness of you
7	Like Someone In Love	16	Time after time
8	Misty	17	When I fall in love
9	My funny valentine		

Table 3.3: List of songs in the Song expression database.

database that had time to record in the studio corresponds to the first 70 melodic exercises and the voice material lasts 11 minutes and 59 seconds, as shown in Table 3.1.

3.4 Song expression database

The song expression database is the second type of database introduced in section 3.1. In this case, the recording script has not been systematically created with melodic exercises that cover as much contexts as possible. Instead, a group of jazz standard songs has been selection without analyzing the coverage of its notes pitches, figures, and strength.

The list of songs from this database is shown in Table 3.3. As a whole, these 17 songs last 18 minutes and 29 seconds as shown in Table 3.1. The songs in this expression database where selected by the singer from a much longer list. The only criteria was to record the songs that she already knew by heart, in order to make it easy for her to sing them in jazz style.

The songs score were available in Musical XML format. Therefore, the musical accompaniment was generated with the Sibleius software as explained in section 3.2.3.

3.5 Labeling

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, note strength as well as vibrato parameters. The following subsections describe how these data are extrected. This section is based on Umbert et al. (2013a).

3.5.1 Feature extraction

Pitch is estimated based on the spectral amplitude correlation (SAC) algorithm described in Gómez & Bonada (2013). 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.

3.5.2 Note segmentation

The segmentation of the recordings provides the note pitch and timing information. Since recordings were done with the modified lyrics (only vowels), 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.

In order to detect the vowel changes, GMM models were trained for clustering and regression (in our case we used 3 different GMM components given that we want to segment the /a/, /i/, and /ua/ vowels). The data used for training were 13 MFCCs extracted from sustained vowel recordings that were done at the beginning of the recording session. We asked the singer to sing sustained vowels (/a/, /i/) covering all her vocal range. Since the automatic segmentation is not completely correct, its outcome was manually checked and corrected. The code we used for the GMM clustering and regression

3.5.3 Transitions segmentation

Note to note transition times are needed to preserve note transition shape during transformation in the unit selection-based systems. Also, note transition times are important for the HMM-based approach since we model sustain and transition sequences.

Transitions 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). We have also manually refined the automatically detected transitions boundaries. An example of the note transition segmentation is shown in Fig. 3.13, where the vertical lines show the pitch transition boundaries.

3.5.4 Note strength estimation

Similarly to the note durations, although the note strength values can be extracted from the score, if we compare the finally recorded melodies with the score there usually is some delay due to the performance itself. Thus, the note strength values can be estimated from the note onset position of the labeled notes.

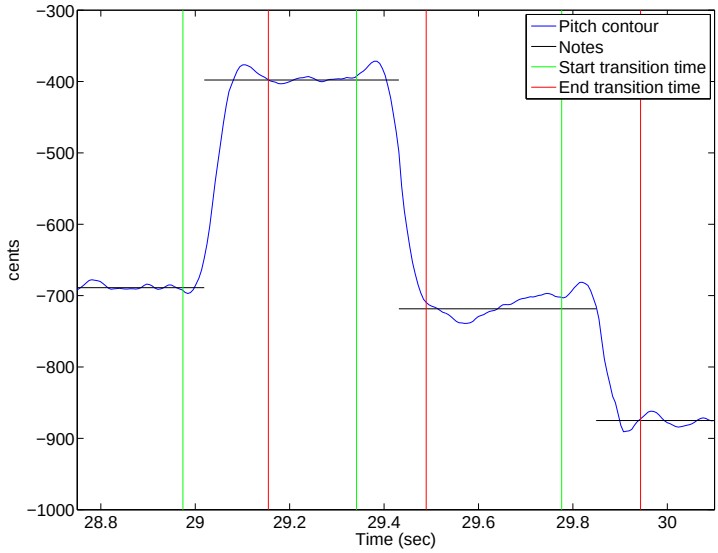


Figure 3.13: Transition segmentation.

To this purpose, for each measure we generate the note strength curve in Fig. 3.14 (in this case for a $\binom{4}{4}$ time signature). First, the anchor points for note strength are defined as in eq. 3.1 from the note strength grid. Note that the first frame has the highest note strength (1), and in the middle point note strength is 0.5, and at a fourth part of the measure the note strength is 0.25. Then, the note strength curve is generated by interpolating these points for each time frame. Finally, the note strength are sampled from this curve at the note onset times. This process is done both for the expression database songs and also for the target songs to synthesize.

3.5.5 Vibrato modeling and baseline pitch estimation

In this section we explain the methodology we follow to separate the vibrato features (depth and rate) and the baseline pitch. The baseline pitch corresponds to the pitch without the modeled fluctuations of the vibrato regions.

Basic idea

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. The relationship of these parameters to the reconstructed pitch contour with vibrato $\tilde{F0}(n)$ is:

$$\tilde{F0}(n) = \bar{F0}(n) + d(n)sin(\varphi(n) + \varphi_{sign}) \tag{3.2}$$

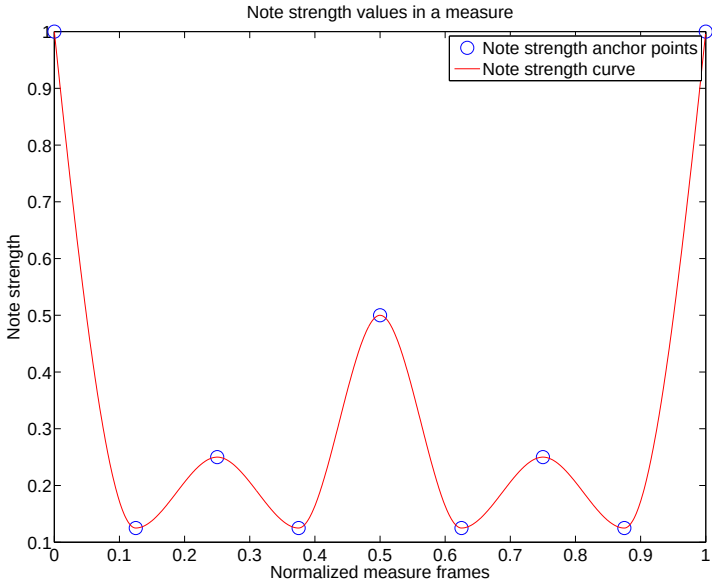


Figure 3.14: Note strength curve for a single measure.

$$\varphi(n) = \sum_{k=0}^{n-1} 2\pi r(k)\Delta_t + \varphi_{correc}(n) \tag{3.3}$$

where, in equation 3.2, $\bar{F0}(n)$ is the estimated baseline pitch (without 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 3.3, $r(k)$ is the vibrato rate at frame k, Δ_t is the frame shift time and $\varphi_{correc}(n)$ is the reconstruction error.

In Fig. 3.15, 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.

In the following subsection we detail how vibrato features are estimated. Initially, the first and last peaks or valleys are manually indicated, and a set of constraints are imposed. Then, vibrato rate and the baseline pitch are iteratively estimated to refine the results. Finally, vibrato depth and the phase are estimated.

Initialization

Before starting the iterative process that estimates and refines depth and rate, we need to detect where vibratos are present. We do so by manually indicating the first and last time instants where there is a peak or a valley. We impose as

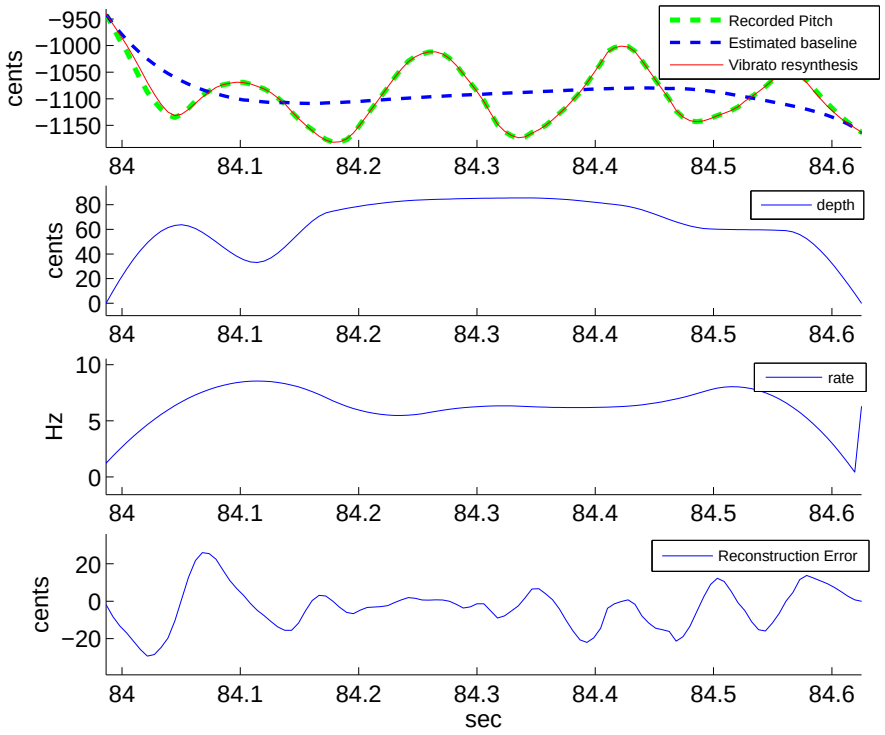


Figure 3.15: Vibrato resynthesis and parameters: depth, rate, reconstruction error and baseline pitch.

constraints for vibratos to have at least one cycle and a depth above a certain threshold (just a few cents would not be considered a vibrato). It is also worth mentioning that rate is initialized to a constant value (13 Hz in our case).

Iterative feature estimation

The baseline pitch and vibrato features estimation involves iterating over three steps, namely, 1) the detection of peaks and valleys within the vibrato segment, 2) the rate estimation from the peaks and valleys time instants, and 3) the baseline pitch estimation as the pitch curve placed between peaks and valleys.

Regarding peaks and valleys, their computation is illustrated in Fig. 3.16. Their time position (or anchor times) is set as the derivative zero-crossings. The derivative is computed by convolving the pitch with a sinusoidal kernel in order to avoid false detections due to pitch irregularities or estimation errors. The kernel is composed of a half negative cycle followed by a half positive cycle. Its length corresponds to one cycle of the estimated rate at each frame, so it can be different for consecutive frames. Next, we compute each peak or valley pitch values using a polynomial regression over a third part of a period.

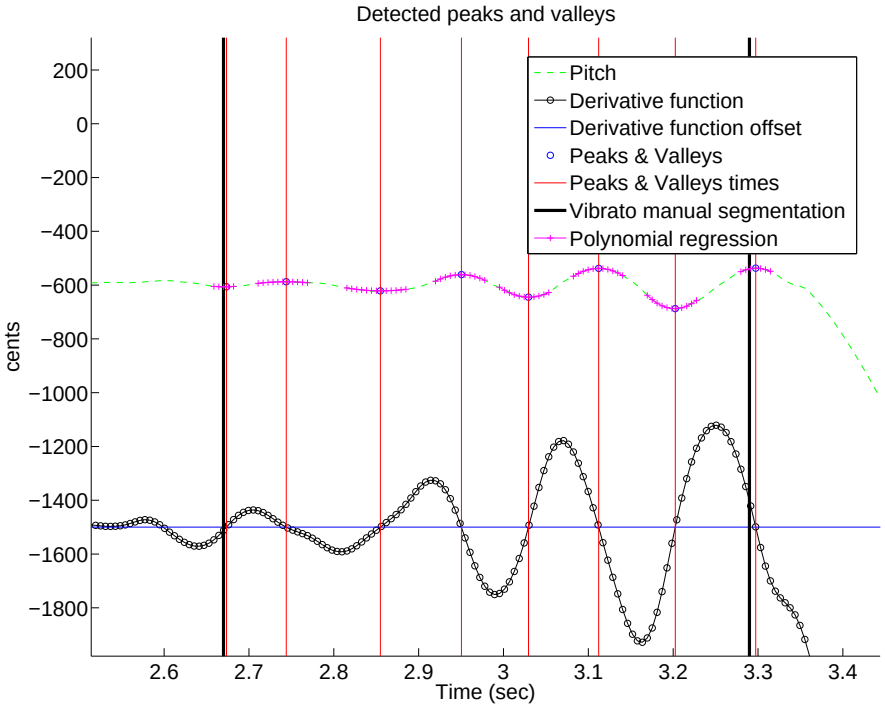


Figure 3.16: Vibrato model: peaks and valleys computation. We have added an offset of -1500 cents to the pitch derivative for visualization purposes.

The rate contour is first estimated as half the inverse of the time between consecutive anchor times, and afterward smoothed by convolving it with a Gaussian window of 61 frames length.

The baseline pitch in a vibrato segment is obtained by smoothing the pitch with a Gaussian window that spans over 2.5 rate cycles. In Fig. 3.17 we show the original pitch contour, the estimated baseline pitch, and the estimated vibrato rate.

Next, we refine the estimated features iterating again over the previous three steps. After this iteration, the final baseline and depth estimations are computed as follows. Since a vibrato does not start or end at peaks or valleys, we extend the manual segmentation of the vibrato segment by a quarter of a period according to the rate values at boundaries. Next, as illustrated in Fig. 3.18, we compute a set of anchor points as the mean time and pitch values of consecutive peaks and valleys pairs. An intermediate baseline pitch (dashed black line) is obtained by a spline regression over these anchor points and the pitch frames outside the vibrato segment. Note that the first and last anchor points are left out. Next, a smooth baseline pitch (cyan dashed line) is computed by convolving the intermediate baseline pitch with a gaussian window. The final baseline estimation (red line) is obtained interpolating the two pre-

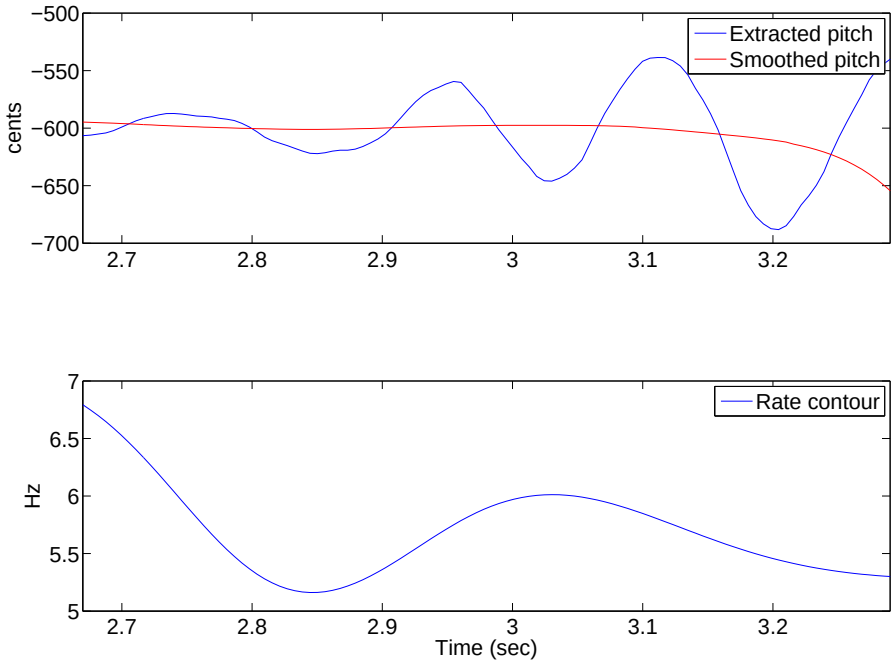


Figure 3.17: Vibrato model: baseline pitch computation.

vious baseline contours so that the central part of the vibrato corresponds to the smooth baseline, and special care is taken to ensure smooth transitions in the first and last vibrato cycles.

The depth contour is computed interpolating the absolute differences between the original pitch and the final baseline at peaks and valleys, as shown in Fig. 3.19.

Finally, we apply one more step to refine the results. First, in the case of the vibrato example we are showing in the figures, the initial phase φ_{sign} (eq. 3.2) is set to π since the first peak/valley has a lower amplitude than the estimated baseline pitch. Then we check that the phase at the peaks and valleys is the expected value. That is to say, a peak or a valley in a sinusoid should have a phase value equal to $\varphi_{sign} + k \times \frac{\pi}{2}$, with $k = 0, 1, 2...$ For each peak and valley, we compute the difference between the cumulated phase and the expected one. In Fig. 3.20 we show the computed phase error from the original pitch and detected peaks and valleys. With the phase error at the peaks and valleys we can generate the phase error contour (middle subplot). The phase error difference between consecutive frames is the phase correction we add to the previously computed rate contour as a way to compensate the phase error within the rate contour.

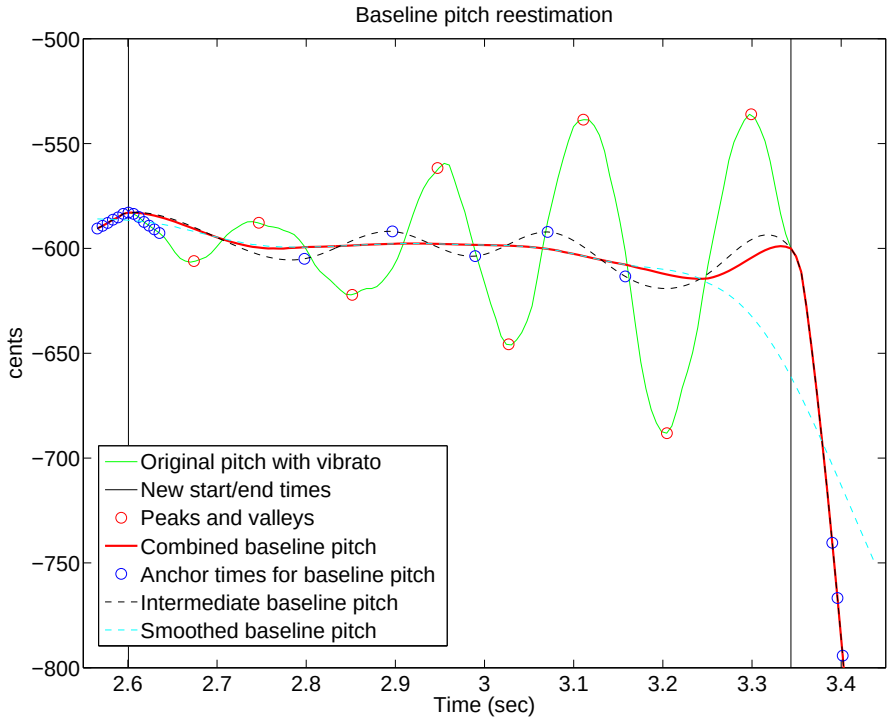


Figure 3.18: Vibrato model: baseline pitch reestimation.

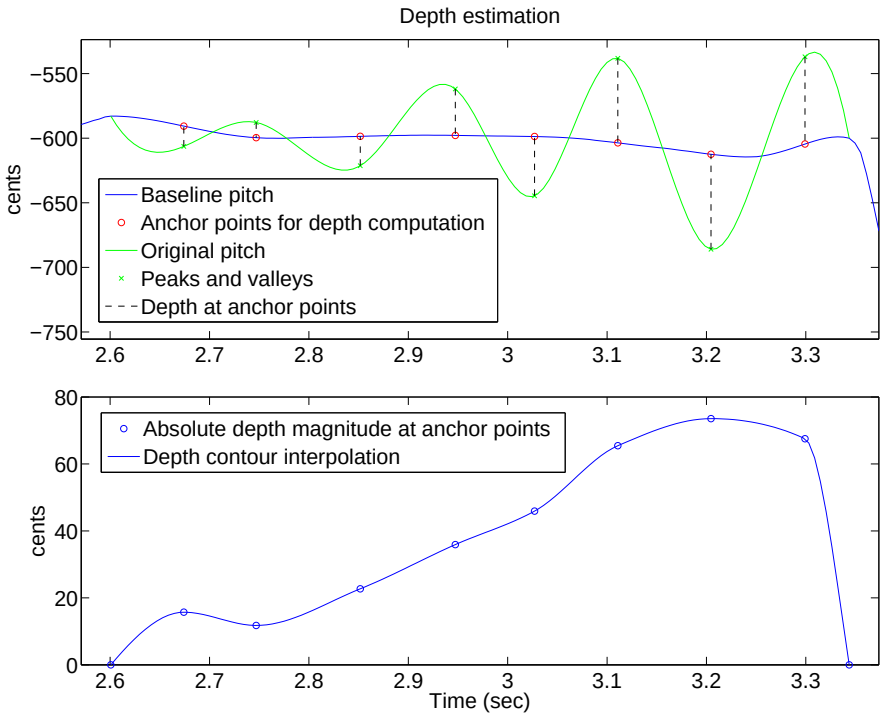


Figure 3.19: Vibrato model: depth estimation.

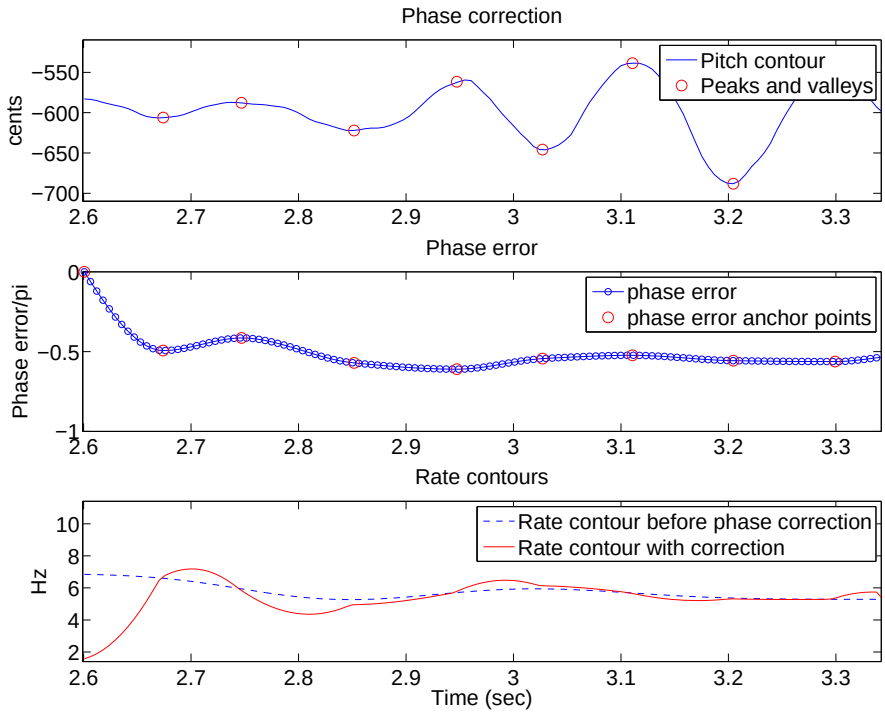


Figure 3.20: Vibrato model: phase correction.

3.6 Conclusion

In this section we have explained our method to design, record, and label the expression databases used by our methods. Both the unit selection-based and the HMM-based methods need a set of features (dynamics, pitch, and vibrato depth and rate) and metadata which is automatically estimated (and then manually refined). These metadata are the note characteristics (onset, duration, pitch, and note strength), as well as the note transitions start and end times.

Each strategy for the creation of the expression databases has its own advantages and disadvantages. The Systematic database aims to cover a set of note features combinations, so that any target song can be represented by units or contextual data that its not very different. A high coverage means that units are not transformed too much in the unit selection-based approaches, or that any target song can be statistically well represented, in the case of the HMM-based methods. Besides the difficulty of building such type of database, another disadvantage is that systematic databases are difficult to record, not only because of the limited time that may shorten the amount of melodic exercises finally recorded, but also because the songs are not known by the singer and cannot be learned by heart either. There are no lyrics and the melodies are short, random, the same chord progression is followed by all of

them, and there are too many exercises, so that these are difficult to remember.

Regarding the Song expression database, it is easy to design, it can even be recorded without the score, and the fact that songs are known by the singer favors to sing them in the required style. On the other hand, the units coverage may not be ensured if the songs are not previously analyzed and selected according to the coverage criteria. In our case, the only criteria were that the songs should belong to the same singing style and also known by the singer. This could have been done differently if we had a pool of several hundreds of songs, then selected the ones that the singer knows, and from this subset select the scores that cover a wider variety of note features combinations.

All the labeling information described in this chapter is needed for all methods, either to define the units or the contextual information in the HMM-based methods. Our labeling consists on extracting pitch and dynamics, semiautomatic note segmentation, semiautomatic note transition times annotations, note strength estimation, and extraction of the baseline pitch and the vibrato features.

CHAPTER 4

A unit selection-based system for expression control

This chapter details the approach for expression control of pitch and dynamics based on unit selection. Inspired by unit selection methodologies applied to speech, a unit selection approach typically consists of the selection, transformation, and concatenation of a set of units that match the target utterance. In this chapter, we propose to adapt these building blocks to generate the expression contours.

4.1 Introduction

We have introduced the unit selection approach for expressive contour generation on section 1.3.3. In this chapter we explain the different blocks in which this approach is built upon.

First, units are selected according to a set of cost functions (Sec. 4.2). Then, the selected units are transformed and concatenated (Sec. 4.3). The transformation is done in time and frequency in order to match the target sequence of notes and rests. The final pitch contour is eventually obtained by generating the vibrato shape which is added to the baseline pitch (Sec.4.4). Finally, the voice is synthesised with the Vocaloid synthesizer (Sec. 4.5). For each section, we present some figures that illustrate the described concepts. This chapter is partly based on Umbert et al. (2013a) and Umbert et al. (2015).

4.2 Unit selection

4.2.1 Description

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.

Cost	Description	Computation
Time-scaling	Favour similar source and target unit durations	Octave ratio (source/target unit notes)
Pitch shift	Favour similar source and target unit intervals	Octave ratio (source/target unit intervals)
Note strength	Favour similar source and target unit note strength	Octave ratio (source/target note strength)
Concatenation	Favor compatible units from the DB	Zero if consecutive units, or depends on transition times
Phrasing	Favor selection of groups of consecutive units	Penalize selection of nonconsecutive units

Table 4.1: Unit selection: sub-cost functions.

The cost criterion consists of the combination of several subcost functions, as summarized in Table 4.1. In this case, there are four functions and unit selection is implemented with the Viterbi algorithm. This algorithm is useful to select from the huge amount of units that may be theoretically possible to transform to match the target unit.

The overall cost function considers the amount of transformation in terms of note durations (time-scaling cost) and pitch interval (pitch interval cost) to preserve as much as possible the contours as originally recorded. Note that while the note duration cost is defined in terms of the absolute note durations (in seconds), the pitch interval cost is defined by the pitch difference (in semitones) of consecutive notes and this value is compared in the source and target unit. The absolute pitch difference between the candidate source unit and the target unit is not used because we have considered that a pitch contour can be pitch shifted and reused some semitones higher or lower.

The overall cost function also measures how appropriate it is to concatenate two units (concatenation cost) as a way of penalizing the concatenation of units from different contexts. Finally, the overall cost function also favors the selection of long sequences of consecutive notes (continuity cost), although the final number of consecutive selected units depends on the resulting cost value. This last characteristic is relevant to be able to reflect, to some extent, the recorded phrasing at synthesis. A third subcost function, the note strength cost, computes how well the source unit fits at the measure position of the target unit.

We can easily imagine the Viterbi trellis as a matrix in which each node is placed at a given column and row. Each column represents a position in time, in our case the target units, and the elements in that column are all the possible units from the expression database (described in chapter 3). The unit

selection process links elements from one column to the next one depending on the least cumulated cost up to that point taking into account on the cost functions described in the following section.

4.2.2 Cost functions

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 expressed in terms of the mean of two subcost functions (amount of pitch shift ps and time-scaling ts) as in equation 4.1:

$$C^t(t_i, u_i) = \frac{1}{3}\left(C^t_{ts}(t_i, u_i) + C^t_{ns}(t_i, u_i) + C^t_{ps}(t_i, u_i)\right) \tag{4.1}$$

These subcosts functions are a weighted sum of note durations (in seconds) dur ratios (in the case of the time-scaling cost), note strength values ns ratios (in the case of the note strength cost) between source and target units, and similarly, unit interval pitch values (in semitones) int ratios (in the case of the pitch shift cost) between source and target units. The C^t_{ts} cost computation is shown in equations 4.2 and 4.3:

$$C^t_{ts}(t_i, u_i) = \sum_{n=1}^{3} \|\omega_{ts}(n)\| \min\left(50, x + (x-1)^3\right) \tag{4.2}$$

$$x = \left[log2\left(\frac{dur(u_i(n))}{dur(t_i(n))}\right)\right]^2 \tag{4.3}$$

where x is the actual computation of the octave-based cost, and we have fine tuned it with the third degree function, and set a threshold of 50 in order to avoid to high values in the computation. The note index within the unit is represented by n, and the normalized time-scaling weights $\|\omega_{ts}(n)\|$ are computed by dividing the ω_{ts} weights by their sum. These weights give more relevance to the central unit note transformation:

$$\omega_{ts} = [0.75,\ 1.5,\ 0.75] \tag{4.4}$$

Similarly, C^t_{ns} is computed with the note strength ratios. In this case, this cost is computed using 4.5 and 4.6 as the x and weigths ω_{ns}, respectively:

$$x = \left[log2\left(\frac{ns(u_i(n))}{ns(t_i(n))}\right)\right]^2 \tag{4.5}$$

$$\omega_{ns} = [1,\ 1,\ 1] \tag{4.6}$$

Note that, in the C_{ns}^t computation, we also have 3 note strength values per unit, and that in this case we have considered that each note strength should be equally weighted. There is no specific reason why the weights have change from function to function, a more in depth study could have been done in this respect, probably.

Differently from the 2 first subcost functions, the C_{ps}^t cost involves 2 values (in a unit of 3 notes there are 2 pitch intervals). The x computation is shown in equation 4.7:

$$x = \left[log2 \left(\frac{int(u_i(n))}{int(t_i(n))} \right) \right]^2 \tag{4.7}$$

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

$$\omega_{ps} = [1,\ 1] \tag{4.8}$$

Note that the C_{ts}^t, C_{ns}^t, C_{ps}^t subcost functions are defined in terms of the $log2$ computation. Based on the octave concept, we have used it to define these costs. Therefore, doubling a note duration is equivalent to an octave, or having to change a note interval from 1 semitone to 2 semitones is also related to the octave idea. In the case of the pitch interval, the octave would not refer to the absolute pitch values, but to the ratio between the pitch intervals.

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.

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. The crossfading step (or concatenation in section 4.3.3) is done with a mask that specifies which frames of a given unit contribute to the output pitch contour. This mask generally focuses on the transition to a unit central note, the central note, and the transition to the next note. This cost handles any possible mismatch between the shapes of the crossfading masks of consecutive units.

For example, if the source units in consideration are consecutive in the expression db, this cost is zero, because the notes that are crossfaded share the same transitions. Otherwise, the transition start and end times of the two source units to concatenate are used to penalized a couple of situations.

The ideal situation when source units are not consecutive would be to ensure that the sustain of the central note of the first unit lasts until the end of the transition to the central note of the second source unit. Thus, we want to avoid the situations in which very distant transition times between the first unit and the second unit may derive into unstable crossfading results.

The first situation we want to penalize is when in the first unit the transition to the third note (end of the sustain of the central note) has already started but the transition to the central note of the second unit has not finished yet. The second situation we want to penalize is when the start of the transition to the central note of the second unit starts before the end of the transition to the third note of the first note. What we actually penalize is the time distance between the times values we are comparing, and the cost is directly this time distance having expressed the time instances relatively (as a percentage) within the unit duration.

For efficiency, and given that the computation of this cost does not depend on the target score (transition times are expressed relative to the source unit duration), this concatenation cost is processed and stored before computing all the other costs that depend on the target score (transformation and continuity). Once the expression database is labeled, this cost can be processed and stored in a sqared matrix with the cost values computed for any pair of source units, so that for any target score this subcost can be retrieved.

Phrasing cost

The ideal situation in a unit selection-based system would be to have the target song in the expression database, as in a performance driven approach. Being this unlikely to happen, with the costs used up to this point, the most probable situation is 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 this cost, towards the musical concept of phrasing.

The phrasing 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 in the path of selected units where two units are not consecutive in the database. While L consecutive units are not chosen, selecting non-consecutive units is penalized (the penalization cost is set to 2, following the criteria of octave-based costs as in the transformation cost). When L is reached, a new starting point is set in order not to force very restrictive constraints to the Viterbi costs.

In our case, we have set $L = 3$ in order to favour the selection of sequences of 3 consecutive source units (or 5 notes). Of course, including this cost it

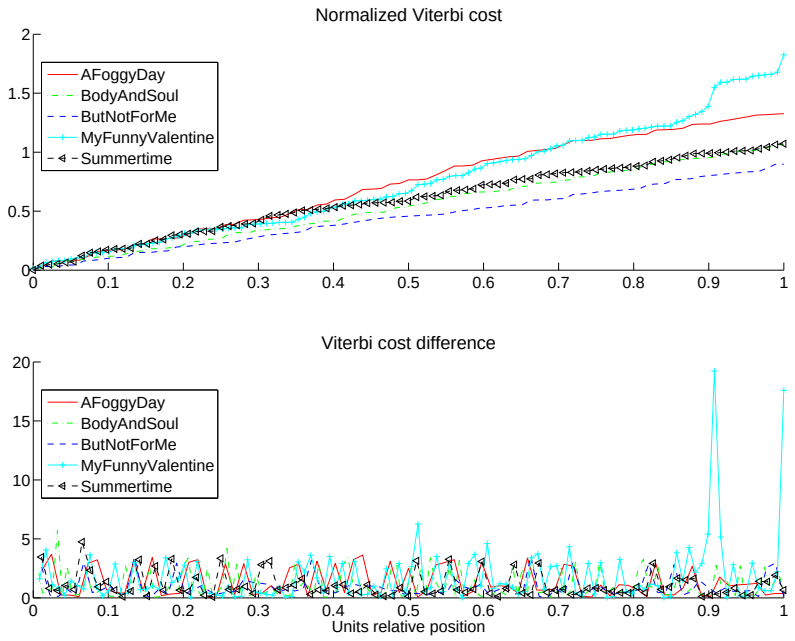


Figure 4.1: Cumulated Viterbi cost.

does not ensure that sequences of such length are present in the selected units, since the this might be to costly compared to the other costs depending on the Viterbi path.

4.2.3 Results

In this section we present some figures on the described costs. First, we present the time evolution of the overall Viterbi cost (the cumulated cost in each node of the Trellis). We have computed it for the 5 songs that we have evaluated in Chapter 7. More details on these songs can be found in this chapter. Besides the cumulated costs, we also present each subcost separately.

In Fig. 4.1 we present the time evolution of the cumulated Viterbi cost for the 5 songs. We have normalized the cost by the total amount of units in each song in order to be able to compare them. Otherwise, longer songs tend to have higher cumulated costs simply because these songs have more notes. The time axis is referred to the units indexes, but these are also normalized to the length of each song, so that all of them are placed between 0 and 1. On the bottom figure we show the cost increment among consecutive nodes, which we can see that have a range of values below 5 in general.

In Figs. 4.2, 4.3, and 4.4 we show the histograms of all values of the 3 unit transformation subcosts related to note duration, note strength, and pitch interval, respectively. The computation has been done for the same 5 songs.

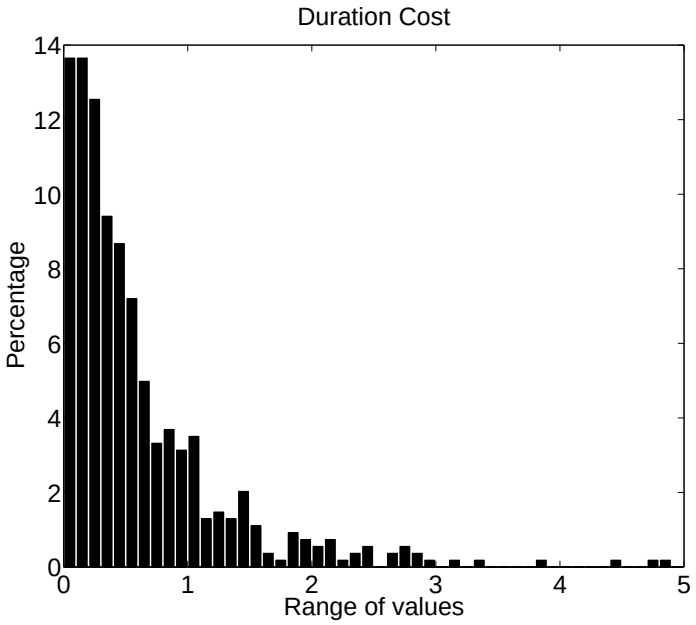


Figure 4.2: Duration cost histogram.

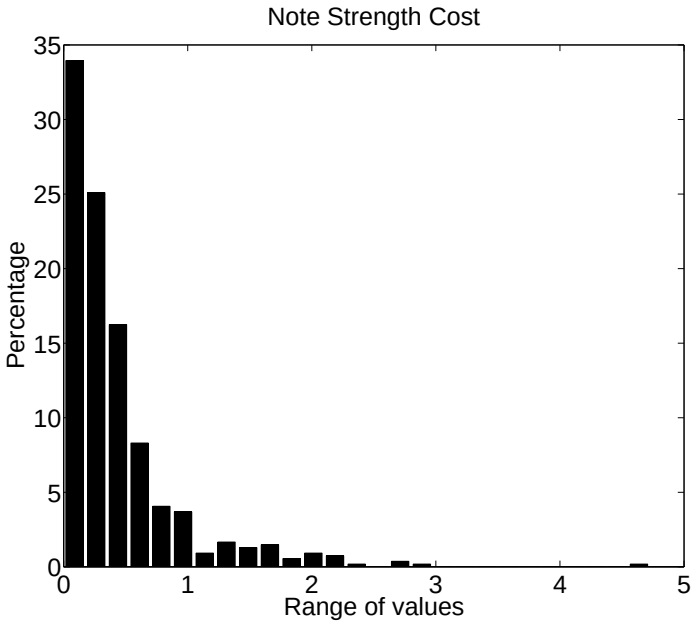


Figure 4.3: Note strength cost histogram.

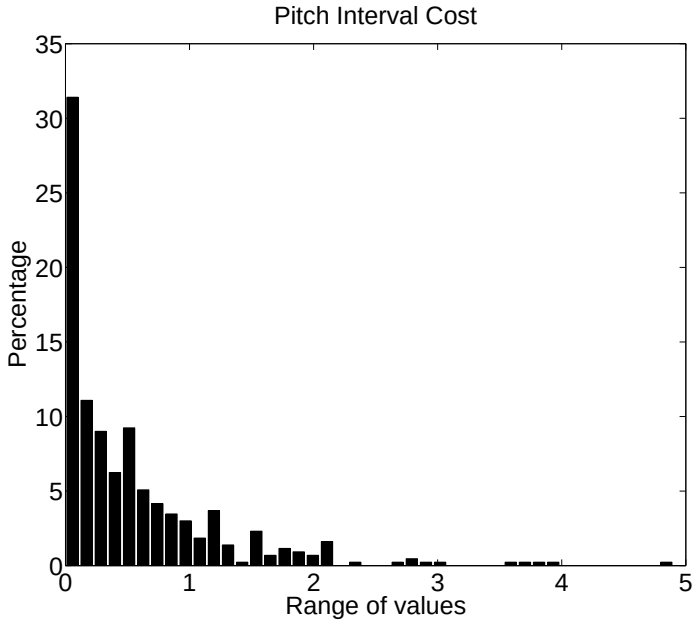


Figure 4.4: Pitch interval cost histogram.

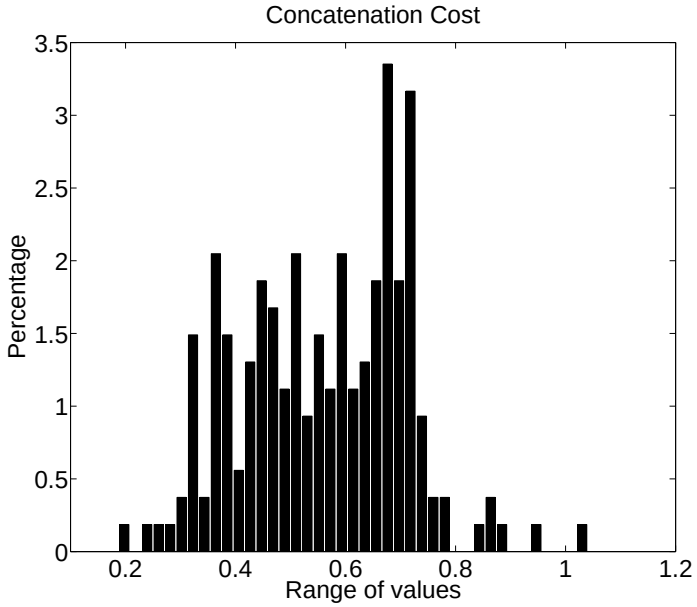


Figure 4.5: Concatenation cost histogram.

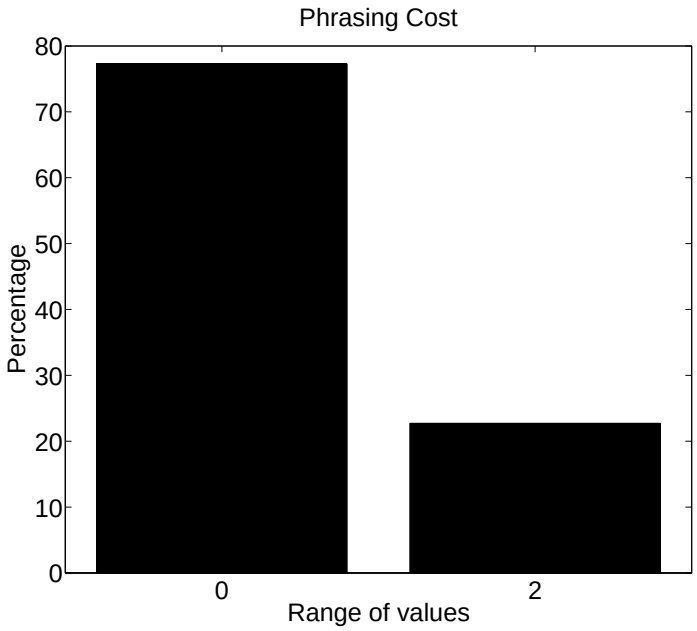


Figure 4.6: Phrasing cost histogram.

In all cases we have positive cost values lower than 5, and the histogram shape tends to decrease with the cost value, showing that most units are transformed at a low cost.

In Fig. 4.5 we show the histogram for the concatenation cost, which has values lower than 1. Finally, in Fig. 4.6 we show the histogram for the "phrasing" cost. In this case, this cost only takes 2 values: 2 is used to penalize taking a unit which is not part from a consecutive phrase in the source unit, and 0 otherwise. In the processed songs, around 20% of the units were selected although these were penalized.

We have also analyzed the effect of the concatenation and phrasing costs from another perspective. If we take the sequences of units which are consecutive in the expression database, we may find sequences of length 1 when a unit is surrounded by units which are from other contexts in the database, but we can also find longer sequences which are consecutive in the database. In Figs. 4.8 and 4.7 we show the length of these sequences. We have used both the Song and the Systematic database to synthesize the same 5 songs.

In the case of the Song database, we have around a 18% of units which taken from a different context than the surrounding ones, and there is around 22% of units which are grouped in pairs (length = 2). The remaining 60% of units have a length of 3 or more. In the case of the Systematic database, single units are a 20%, and paired units a 32%. The remaining 48% are sequences of at least 3 consecutive units in the database.

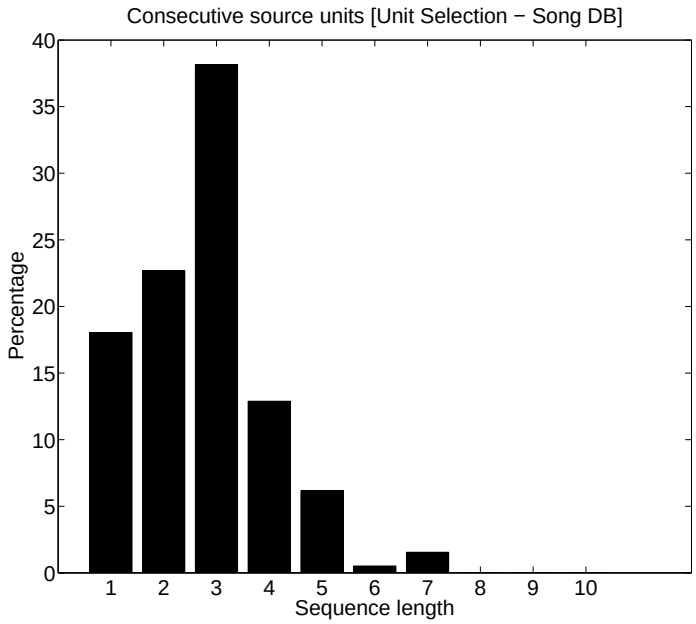


Figure 4.7: Sequences of consecutive units (Song DB).

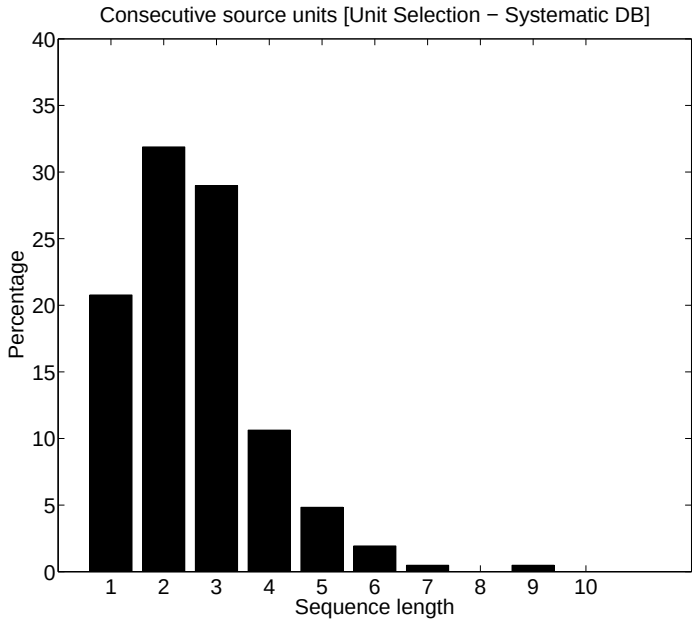


Figure 4.8: Sequences of consecutive units (Systematic DB).

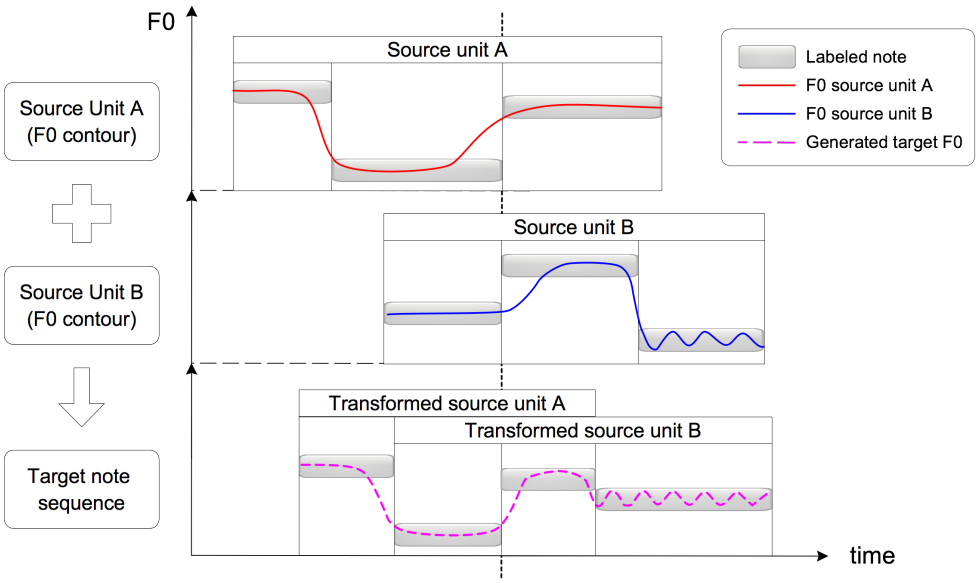


Figure 4.9: The performance feature (F0) generated by unit selection.

From this perspective, we can say that the unit selection-based system is capable of retrieving longer sequences from the Song database than the Systematic database (60% vs 48%, respectively).

4.3 Unit transformation and concatenation

4.3.1 Description

This step deals with the transformation of the selected sequence of units. Source notes have to match target notes in pitch and duration. Therefore, once a sequence is retrieved, each unit is time scaled and pitch shifted. The time scaling is not linear; instead, most of the transformation is applied in the sustain part and keeping the transition (attacks and releases) durations as close to the original as possible. Vibrato is handled with a parametric model, which allows the original rate and depth contour shapes to be kept. Source unit dynamics contour is also scaled according to the target unit duration.

In Fig. 4.9 we show 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). Vibratos appearing in the source units are also rendered, preserving the original depth and rate and spanning over the target note duration. In parallel

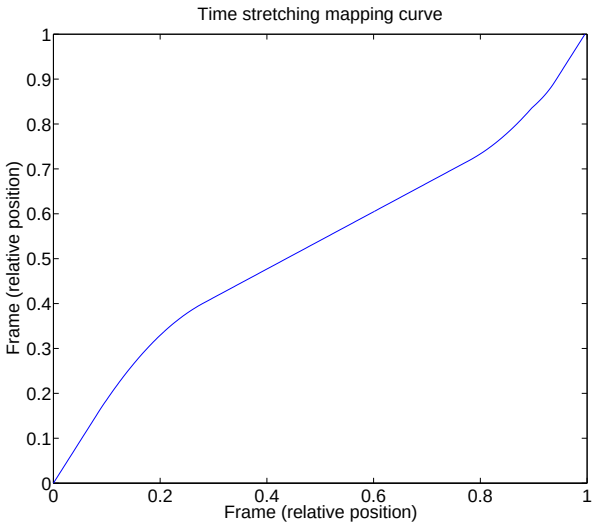


Figure 4.10: Example of unit time-scaling mapping curve.

to unit transformation, crossfading is applied between the transformed units pitch contours in order to generate expression contours.

4.3.2 Unit transformation

Time scaling: articulations vs. sustains

Time scaling aims to transform the selected notes so to match the duration of the target notes. One consideration is that besides notes, the pitch contour consists of a sequence of sustains and transitions and those can be treated differently. While sustain durations are typically correlated with note durations (so they can have any duration within a wide range), transition durations are less dependent on the note duration and therefore their durations are less variable. Hence, naturalness would be theoretically better preserved if most of the time-scaling transformation is applied to the sustains. With this aim, we apply a non-linear time-scaling transformation through a mapping function between target and source notes. This is illustrated in Fig. 4.10, where we can clearly see different time-scaling factors applied to transition and sustain segments.

Pitch shifting

The main idea behind the pitch shifting step is that a pitch contour can be transposed as if that note sequence had been sung at a higher or lower frequencies. Besides transposing pitch contours, we also might need to change the note intervals in order to match the target note sequence.

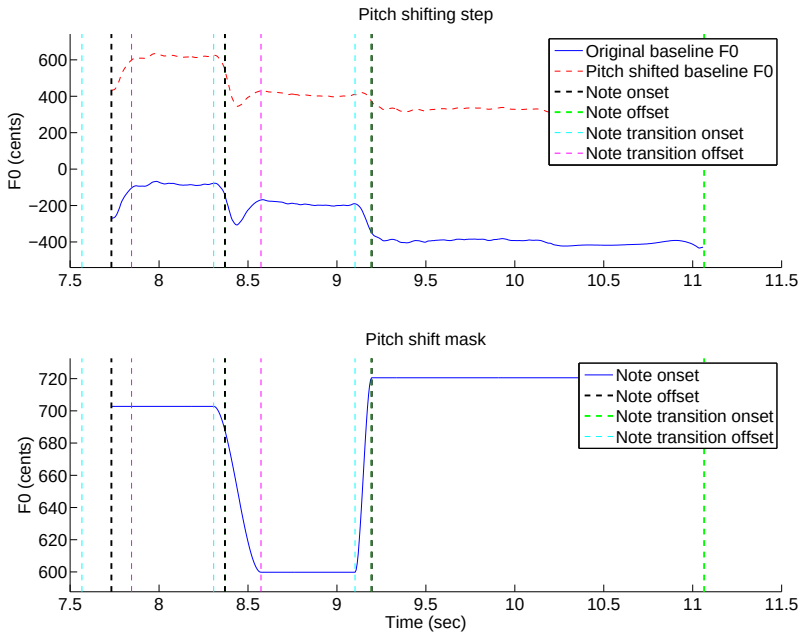


Figure 4.11: Example of unit pitch shifting.

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. In Fig. 4.11 we show an example of the original and transformed baseline pitches (top figure) and the pitch shift mask used for this case. During note sustains the mask has a constant offset. In note transitions, the mask is obtained by interpolating with the cubic method from one note to the next one.

Dynamics offset level

The main transformation applied to dynamics is time-scaling as explained before. We also add an offset level to the source unit dynamics contour. The dynamics contour is placed around 0.6 offset value since the Vocaloid synthesizer treats this value as an average level. By doing this step we ensure that all phonemes will be assigned high enough dynamics to be heard.

4.3.3 Unit concatenation

The transformed units are concatenated in order to generate the expression contours for each expression feature. This process is basically an overlap and add iterative process applied to every consecutive unit.

The overlapping step of the transformed pitch, dynamics and vibrato parameter contours is handled with a crossfading mask. This mask is computed

per unit in order to determine the samples that contribute to the output contour. More relevance is given to the attack to the central unit note and its sustain, until next the start time of the attack to the third unit note.

In Figs. 4.12 and 4.13 we show how the crossfading masks are generated from the transformed unit (after the time-scaling and pitch shifting step explained in the previous section) for the baseline pitch and dynamics expression contours, respectively. In both figures, the top subplot shows the baseline pitch (or dynamics) with the note onsets and transitions start and end times marked with vertical dashed lines. In the middle subplot there is the transformed unit in time and frequency (note that the time axis do not match with the previous one because the target unit is placed at another time instant in the target song). The bottom subplot represents the crossfading mask that is used in the concatenation step. The mask's shape gives more importance to the attack (transition) to the central unit note and the corresponding note sustain. From another perspective, the masks controls the frames' contribution to the final expression contour. Right before the attack the mask reaches 1, and right before the transition to the next note it reaches 0 again. Similarly, we do the same steps for dynamics.

4.3.4 Results

In this section we present some further graphical results for the unit transformation step. In the previous subsection we have already introduced some partial results, like the time-scaling mapping curve, the pitch shifting mask, the transformed baseline pitch, and the transformed dynamics.

We have collected the values of the time-scaling factors (ratio of note duration between the central note of the source and target units) which have been applied to the Song database and to the Systematic database. This information is shown in the histograms of Figs. 4.14 and 4.15. The experiment has been done for the same 5 target songs as in section 4.2.3. In both databases, source units have been time scaled with a factor between 0 and 2. That is to say, in a few cases notes are shortened, in other cases notes' durations are doubled. The average time-scaling factor is 1.16 and 1.18, and the histogram peaks are placed at 0.77 and 0.71 for the Song and Systematic databases, respectively.

Similarly to the time-scaling factors, interval transformations have been applied to the selected units. The semitone difference between the first interval (attack to the central unit note) of selected units and the target units is represented in Figs. 4.16 and 4.17. The average pitch interval is -0.27 and -0.08, and the histogram peaks are placed at 0.12 and -0.71 for the Song and Systematic databases, respectively. In both cases, most of the semitones difference between source and target units is less than 2.5 semitones.

Next, in Fig. 4.18 we present an example of the unit concatenation step with 5 units. On the top figure we show the contours of the 5 transformed units with the effect of each crossfading mask. When the masks reach 1, the

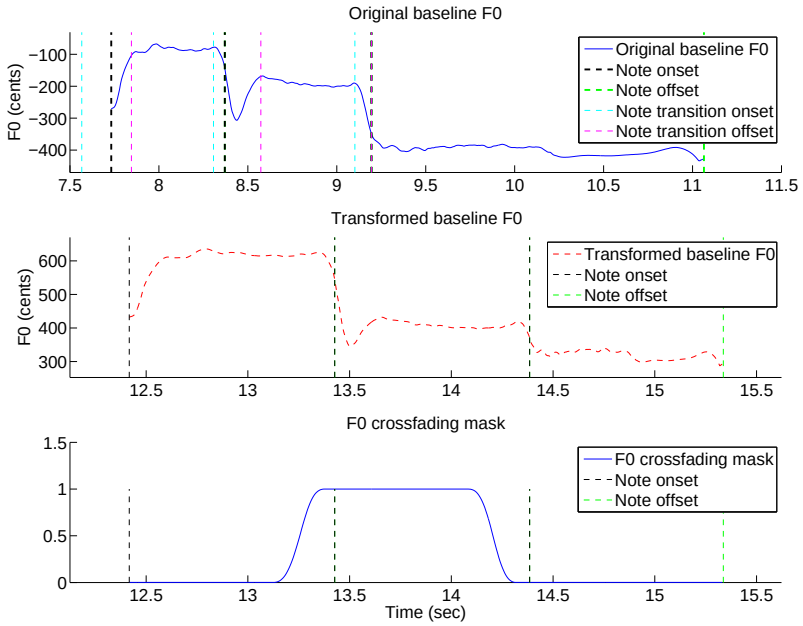


Figure 4.12: Transformed baseline pitch and crossfading mask.

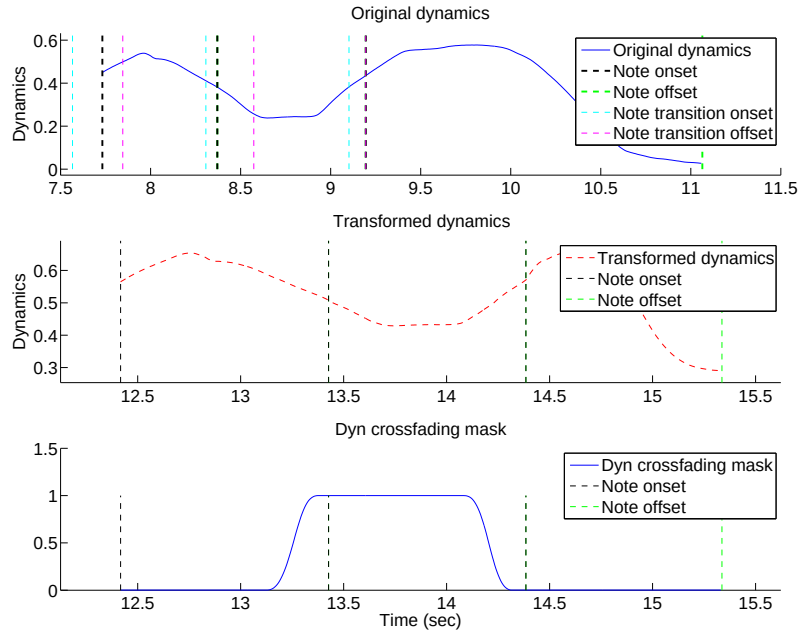


Figure 4.13: Transformed dynamics and crossfading mask.

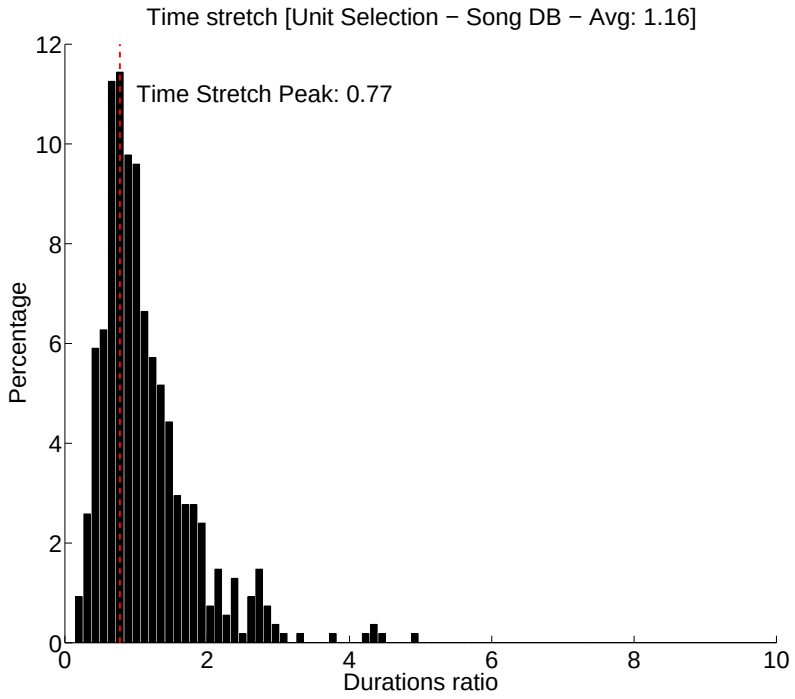


Figure 4.14: Time-scaling factors (Song DB).

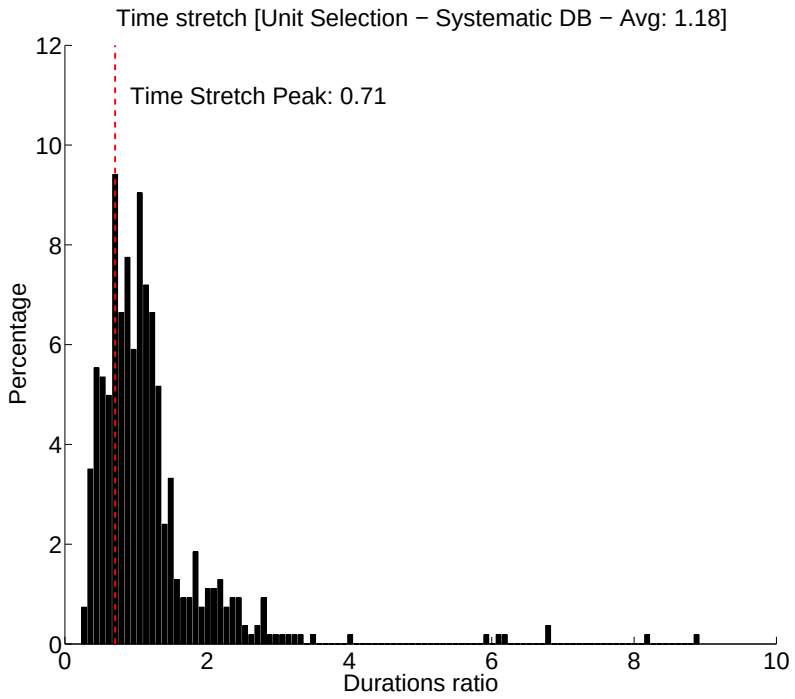


Figure 4.15: Time-scaling factors (Systematic DB).

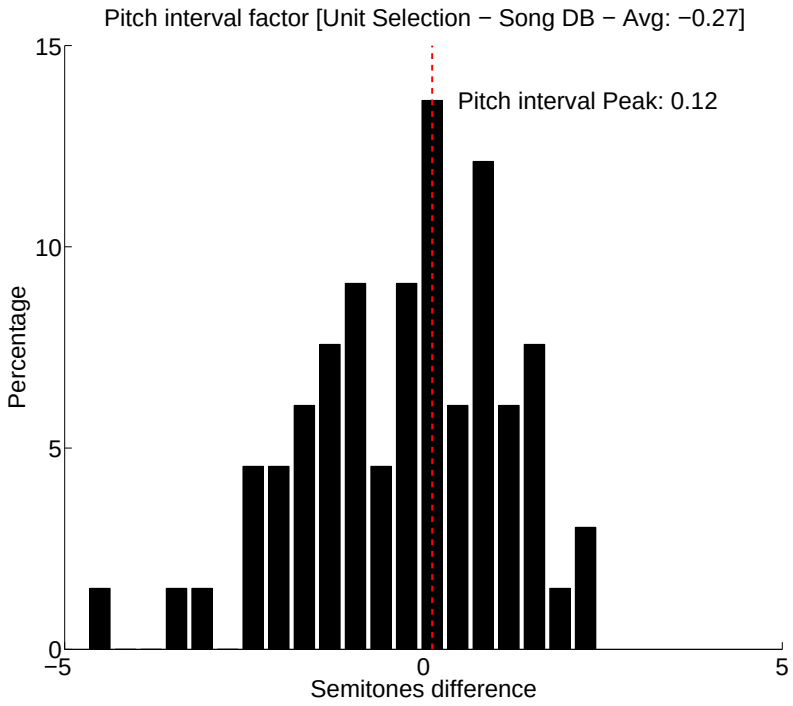


Figure 4.16: Pitch interval difference (Song DB).

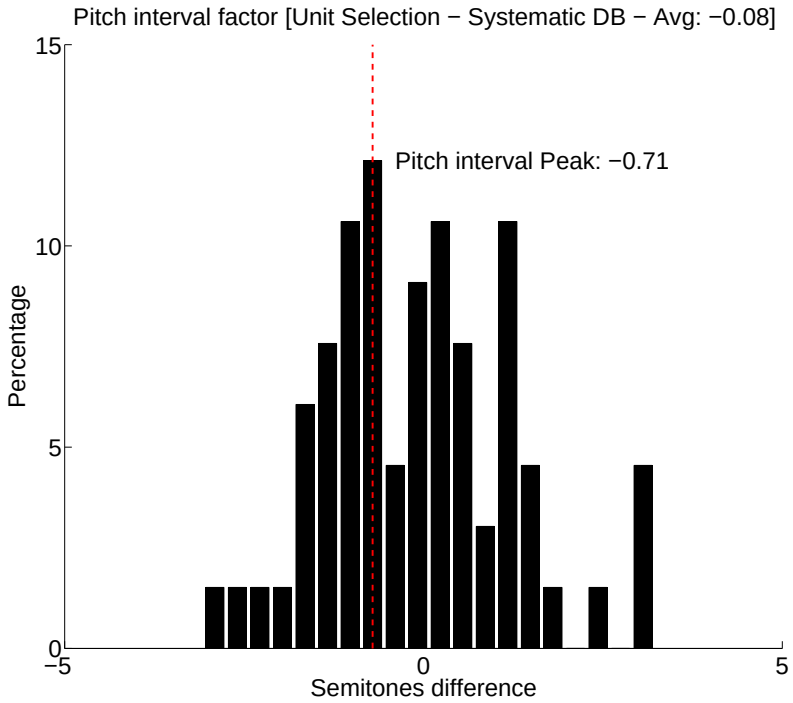


Figure 4.17: Pitch interval difference (Systematic DB).

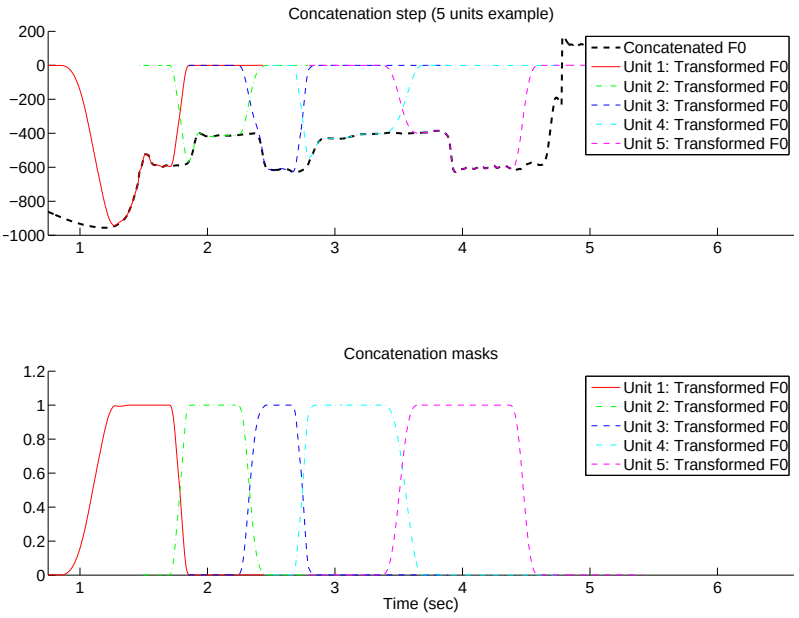


Figure 4.18: Example of cross-fading masks.

transformed unit shapes are preserved (attack and sustain of the unit central note). In the preceding and succeeding frames, the mask tends to 0 and so does the contour. Since consecutive overlapping masks do the transition from 0 to 1 (or viceversa) in same previous frames to the note attack (or release), the crossfading handles the weight or contribution of overlapping frames. The thickest dashed line (in black) in this figure is the result of the concatenated baseline F0.

An example showing the concatenation of the vibrato features are shown in the following section 4.4 on the generation of the expression contours.

4.4 Contour generation

4.4.1 Description

After concatenating the transformed units, we obtain different pitch expression contours that need to be joined. First, the baseline pitch is tuned in the note sustains to correct any possible mistake in the labeling process and to ensure the singer is in tune. Then, vibratos are rendered and added to the baseline pitch. Dynamics are no longer processed since these are obtained in the previous step of unit concatenation.

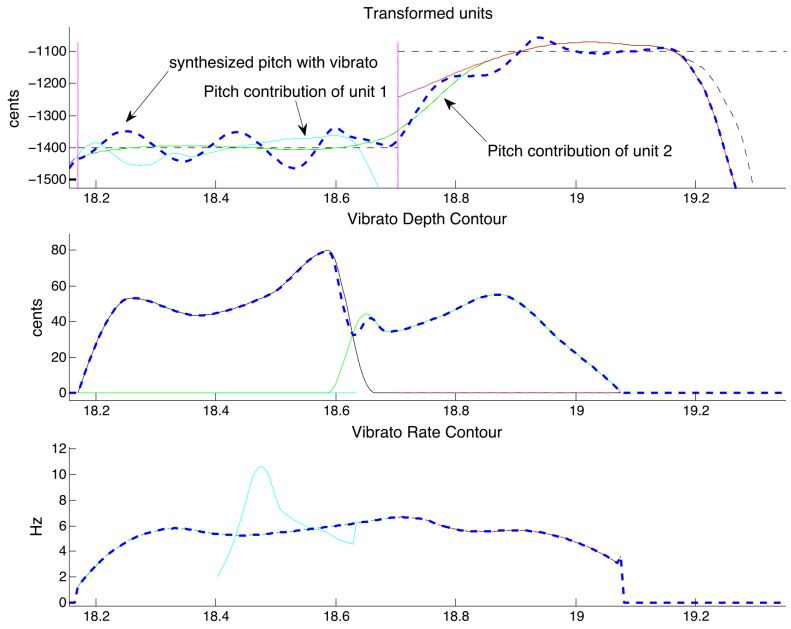


Figure 4.19: Transformed unit pitches and vibrato control contours concatenation.

4.4.2 Baseline pitch tuning

In order to ensure that sustains are at the right target pitch, the baseline pitch is 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.

4.4.3 Vibrato generation

Vibratos are synthesized using the depth and rate generated for the target song. Those frames with depth equal to zero contained no vibrato. Otherwise, the procedure introduced in section 3.5.5 is followed for synthesis.

An example of the result is shown in Fig. 4.19 (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 has been synthesized with the depth shown in the second subfigure,

```
<mCtrl>
  <posTick>32036</posTick>
  <attr id="DYN">55</attr>
</mCtrl>
<mCtrl>
  <posTick>32036</posTick>
  <attr id="BRI">55</attr>
</mCtrl>
<mCtrl>
  <posTick>32036</posTick>
  <attr id="PIT">1210</attr>
</mCtrl>
```

Listing 4.1: Code example: VSQX format for dynamics, brightness, and pitch bend

where the two contributing units can also be observed. The vibrato rate is shown in the bottom subfigure.

4.5 Sound synthesis

4.5.1 Description

In this section we explain the last step for the sound synthesis generation with the Vocaloid singing voice synthesizer, and then we evaluate the generated audio files. In short, in section 4.5.2 we explain the basic file format in which lyrics, notes, and frame pitch bend and dynamics values are specified, and in section 4.5.3 the synthesized files are compared to the expression achieved by the synthesizer default configuration and we also compare it to the expression achieved by manually tuning the expression parameters.

4.5.2 File formatting

In Vocaloid files (*.vsqx) the song score and expressions controls are represented in XML (eXtensible Markup Language) format. Besides a header containing information on the file version and enconding, the most important tags with the score information are:

- *<VoiceTable>*: it contains the configuration of the voice bank.
- *<mixer>*: it specifies the mixer configuration, for instance on the compression or reverberation.
- *<masterTrack>*: it specifies score information like the time signature or the tempo.
- *<vsTrack>*: it specifies values at frame and note level.

```
<note>
  <posTick>4581</posTick>
  <durTick>211</durTick>
  <noteNum>55</noteNum>
  <velocity>64</velocity>
  <lyric><![CDATA[For]]></lyric>
  <phnms><![CDATA[f @` r]]></phnms>
  <noteStyle>
    <attr id="accent">50</attr>
    <attr id="bendDep">8</attr>
    <attr id="bendLen">0</attr>
    <attr id="decay">50</attr>
    <attr id="fallPort">0</attr>
    <attr id="opening">127</attr>
    <attr id="risePort">0</attr>
    <attr id="vibLen">0</attr>
    <attr id="vibType">0</attr>
  </noteStyle>
</note>
```

Listing 4.2: Code example: VSQX format for notes

The part that contains the relevant data is the *<vsTrack>* tag which contains 2 typs of data values. The first set of values are the expression control feature values at frame level, which may be the pitch bend (pitch deviation between the pitch value and the note pitch), dynamics, or brightness (see code example in listing 4.1 with 3 feature values for the same frame). Next, the note information is specified with the note onset, duration, MIDI note number, the lyrics orthographic and phonetic transcription (see code example in listing 4.2 for the word *For*). In this format, frames are indicated by the *posTick* integer which is internally mapped to a time position.

It is important to highlight how the expression control parameters are mapped in the *VSQX* files. Regarding the pitch contour, the $F0$ frame information is provided through the note pitch and the deviation from the note to the frame value (or pitch bend). Concerning dynamics, we are not only mapping dynamics directly to the dynamics feature, but also to the brightness feature since it we can obtain more realistic results by slightly changing timbre as well. Both contours are almost the same. While timbre brightness (BRI) is entirely controlled by the dynamics value, so that higher dynamics values imply more timbre brightness, the synthesizer dynamics (DYN) is handled by the lower values of the generated *dynamics* expression contour according to expression in 4.9:

$$DYN = min(0.5, dynamics) \tag{4.9}$$

The synthesizer interface has an export functionality which allows to generate the audio file from the specified XML format.

4.5.3 Evaluation and results

Aim of the study

The evaluation explained in this section is based on Umbert et al. (2013a). The aim of this perceptual evaluation is to compare the perceived naturalness, expressiveness, and the singer skills of three different methods of controlling the singing voice expression. We have to clarify that by the time of this publication note strength feature was not included, and therefore the corresponding subcost feature is not used.

Before starting the perceptual test, the 3 parameters to rate were explained to them. The naturalness was explained to the participants based on whether the singing voice was perceived rather synthetic or human. Expressiveness could range from very inexpressive to very expressive, and we refer to singer skills as an overall perception also related to elements like a very bad or good timing and tuning.

The three methods we have compared in this evaluation are the baseline method based on heuristic rules, manual tuning of dynamics, pitch bend and vibratos, and finally the synthesis using the proposed unit selection-based system.

Experimental setup

We evaluated the achieved expression by conducting a Mean Opinion Score (MOS) test with 16 participants. The subjects rated the synthesized performances from 1-5 in terms of naturalness, expressiveness, and the singer skills.

Three excerpts of 30 seconds were synthesized. For each of these excerpt, three versions were synthesized using the three different methods of generating expression contours. All versions had background music.

The heuristic rules or default configuration was obtained following the algorithm described in Bonada (2008) and also introduced in section 2.4.4. The manually tunned files have been generated 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. Although the database used in this experiment is neither the song or systematic databased described in chapter 3, it was built following the same principles. It was initially built to test the unit selection-based framework.

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

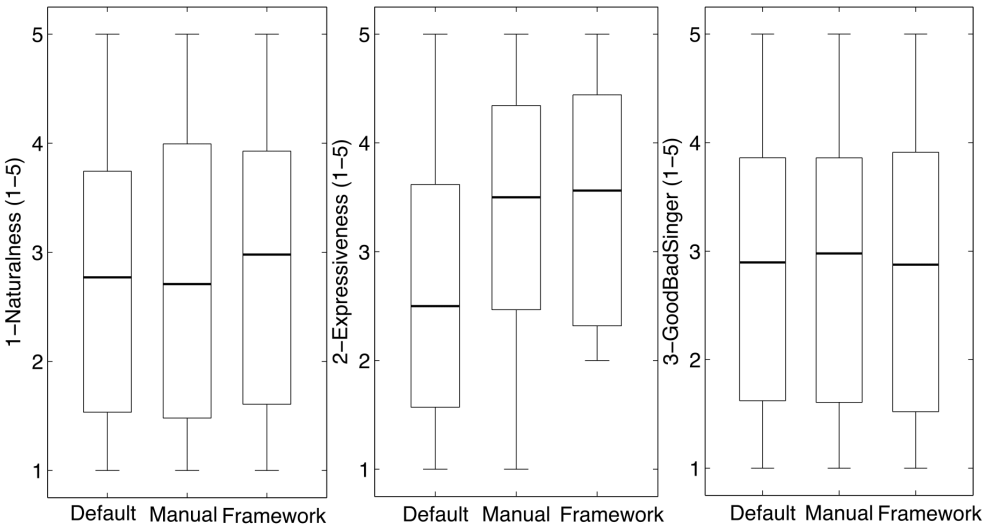


Figure 4.20: Unit Selection: Results of listening tests.

versions were presented in a random order. These songs were synthesized using a Spanish voice bank. The rating task took around 15 minutes.

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 4.20, 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, we can conclude that the proposed system improves expression and

achieves a similar level to the manual 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[1] related to Umbert et al. (2013a).

4.6 Conclusion

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

From the listening tests, we have concluded that this 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.

[1] http://mtg.upf.edu/publications/ExpressionControlinSingingVoiceSynthesis

CHAPTER 5

A statistical-based system for expression control

In the previous chapter we have explained the unit selection approach for expression control of pitch and dynamics. Alternatively, Hidden Markov Models (HMMs) can also be used to statistically model time series. In this chapter we model pitch and dynamics with HMMs in two different ways. The first one is our Note HMM-based system, which model either sequences of notes (as it has been introduced in section 2.4.5 by previous works). Alternatively, HMMs can also be used to model sequences of note transitions and sustains. Both HMM-based systems are explained in this chapter.

5.1 Introduction

In speech, statistical methods like Hidden Markov Models have proven to be flexible and it has also been applied to singing voice synthesis by jointly modeling timbre with pitch and dynamics (Oura & Mase, 2010) where HMMs model phoneme units. Their note pitches and durations are used as contextual data together with the surrounding phonemes and notes amongst others. Thus, as we explain in section 5.2, the same unit concept applies, considering the central note of a unit the item to model, and the previous and succeeding note as contextual data.

In this chapter we explain how we have modified an HMM-based speech synthesis system (also known as HTS[1]) to model pitch and dynamics. First, we have created an HMM-based system to model note sequences (section 5.3). Next, we have modified this framework to model sequences of transitions and sustains (section 5.4). By transition we refer to the pitch contour articulation from one note to the next one (or an attack from a silence to a note, or a release from a note to a silence). On the other hand, a sustain has its pitch

[1] http://hts.sp.nitech.ac.jp

contour around a note (although there might be some deviations like possible detunings or oscillations due to a vibrato).

Apart from the type of sequence that it is being modeled (note vs. transitions and sustains), another difference is that in the first case absolute pitch (and dynamics) values are used in the training and synthesis step, while in the second system pitch values relative to the melody are used, that is to say, the difference between the pitch contour and the theoretical melody based on the score.

5.2 Main concepts

The HTS system for speech synthesis is a complex framework, with many different options concerning its configuration. In this section we only introduce the aspects that we have used to build both HMM-based systems for expression control.

In sections 1.3.1 and 3.3.1 we have introduced and explained the concept of unit. It basically consists of a central note and the corresponding previous and succeeding notes or rests. These three elements are described mainly by their duration and, in the case of notes, also by their pitch. Similarly, a central note and its contextual data is also used in the HTS framework. Although there are some differences, the main idea is basically the same. The contextual data used in the default HMM-based system (section 5.2.1) has been simplified in the proposed HMM-based systems.

The information described in the contextual data is used to distinguish models according to their context and to group the training data into clusters (section 5.2.2) from which its mean, variance, and their fluctuations or dynamic features (delta and delta-delta) are computed and used at synthesis. Finally, the data preprocessing is also described (section 5.2.3).

5.2.1 Contextual data

Since the HTS system for speech and singing voice jointly models timbre, pitch, and dynamics, its original contextual-dependent labels contains information on the phoneme identity, syllables, duration, and pitch. Detailed format on the HTS context-dependent labels can be found in Appendix A. These two pages correspond to the guide provided in the HTS demo.

The idea of such labels is to provide in a single line information of the elements that are being modeled, following the format in Fig. 5.1, where a set of fields are separated by different delimiters. Since the fields' left and right delimiters are different for each field, these are used to identify the fields location and field values. Each new line has a different central phoneme, and therefore the contextual-dependent labels change as well. The main aspects of such format is that first we find five labels specifying the identity of the current

p_1^p_2-p_3+p_4=p_5_p_6%p_7^p_8
/A:a_1-a_2-a_3@a_4 /B:b_1_b_2_b_3@b_4 /C:c_1+c_2+c_3@c_4
/D:d_1!d_2#$d_3$$$d_4$%$d_5$|$d_6$&$d_7$[$d_8$-$d_9$
/E:e_1]e_2^e_3=e_4~e_5!e_6@e_7#e_8+e_9]$e_{10}$$$e_{11}$|$e_{12}$[$e_{13}$&$e_{14}$]$e_{15}$=$e_{16}$^$e_{17}$~$e_{18}$#$e_{19}$@$e_{20}$!$e_{21}$$$e_{22}$&$e_{23}$%$e_{24}$[$e_{25}$|$e_{26}$]$e_{27}$-$e_{28}$^$e_{29}$+
e_{30}~e_{31}=e_{32}@$e_{33}$$$e_{34}$!$e_{35}$%$e_{36}$#$e_{37}$|$e_{38}$|$e_{39}$-$e_{40}$&$e_{41}$&$e_{42}$+$e_{43}$[$e_{44}$;$e_{45}$]$e_{46}$;$e_{47}$~$e_{48}$~$e_{49}$^$e_{50}$^$e_{51}$@$e_{52}$;$e_{53}$=$e_{54}$=$e_{55}$!$e_{56}$~$e_{57}$+$e_{58}$
/F:f_1#f_2#f_3-$f_4$$$f_5$$$f_6$+$f_7$%$f_8$;$f_9$
/G:g_1_g_2 /H:h_1_h_2 /I:i_1_i_2
/J:j_1~j_2@j_3

Figure 5.1: Context-dependent labels line format in HTS framework.

(central) phoneme (*p3*), as well as the identity of the two previous (*p1, p2*) and succeeding phonemes (*p4, p5*).

After the phoneme identities, the contextual label format specifies information on the previous/current/next syllable (*A*, *B*, and *C* labels, respectively), the previous/current/next note (*D*, *E*, and *F* labels), the previous/current/next phrase (*G*, *H*, and *I* labels), and the whole song (*J* label).

The context label format can be simplified since in this thesis we are focusing on pitch and dynamics expression contours. For instance, timbre labels related to phonetics can be erased. The context labels that we have finally used are described in each system section.

5.2.2 Clustering

The clustering is mainly based on a set of yes/no pre-defined questions which separate the data based on the context. These questions define a "tree" with its branches (yes/no answers) and leaves (grouped data with the same answers). The contextual data impacts on how data is clustered together when the clustering tree and its leaves nodes are build.

The set of yes/no questions ask for all possible values in the contextual data. Thus, the original questions in the HTS framework try to split and group the data based on the possible values of the contextual data in the format shown in Fig. 5.1. The original HTS framework generates one tree for each of the 5 emitting states. Within each tree, the questions on the phoneme identity clusters the data, so that in the node leaves there usually are different central phonemes, although close phonemes in similar conditions may be grouped together.

In our HMM-based systems, we model either sequences of notes or sequences of sustains and transitions. Thus, we have introduced some changes regarding the clustering step.

5.2.3 Data preparation

Splitting songs in phrases

Our training data consists of labeled songs or systematic exercises with the corresponding notes and rests. Each audio file, apart from the beginning and

the ending silences, has one or several silences within the melody. Therefore, in each audio file there are at least two phrases.

According to our initial tests, we have seen that training the HMM models with the entire files cause some problems.The beginning and ending silences may be several seconds long, although the target songs that we may want to synthesize do not have such long silences in their score. As a consequence, the contexts reflected in the training files are a bit distant to what is later synthesized. Moreover, the alignment of the model sequence to the training data may be worse when having long silences at the beginning of the training files.

This was solved by splitting the audio files used for training in phrases. Therefore, for each original song or melodic exercise we generated as many files (lf0, vib, dyn, and contextual data) as phrases there are. The silences surrounding each phrase are as long as in the original recorded data, except for the begining and ending silence which where shortened to their original value or shortened to the duration of a measure.

Data format

The data format in the HTS framework is logarithmic when the stream refers to frequencies. In the logarithmic domain the frequencies have a more gaussian distribution. Therefore, the trained and synthesized baseline pitch and vibrato rate values are not directly in the units we might expect, but its logarithmic value. For unvoiced frames, the corresponding value is -10^{10}.

In the HTS framework, data is organized in "streams", which in our case are the expression contours that we want to train and synthesize. Similarly to the unit selection-based system, we have a stream for the baseline dynamics, one for the baseline pitch, and another one which contains both vibrato depth and rate. In the HTS framework, data is organized in "streams", which in our case are the expression contours that we want to train and synthesize. Similarly to the unit selection-based system, we have a stream for the baseline dynamics, one for the baseline pitch, and another one which contains both vibrato depth and rate. Streams of one dimension like dynamics and baseline pitch are called univariate, while the 2 dimensional stream of the vibrato features is a multivariate stream.

5.3 Note HMM-based system

5.3.1 System description

In this section we describe the Note HMM-based system based on the HTS framework for speech synthesis. The main characteristics of this system are summarized in Table 5.1, together with the characteristics of the modified HMM-based system explained in the next section.

Feature	Note HMM-based system	Sustain/transition HMM-based system
Modeled sequence	Note	Sustain and transition
Score change	-	Sustain and transition prediction
States/model	5	5
Dynamics contour	Absolute	Absolute
Pitch contour	Absolute	Difference with score
Depth contour	Absolute	Absolute
Rate contour	Absolute	Absolute
Database modification	Pitch shift	-

Table 5.1: Comparison of the HMM-based systems.

This system is characterized by modeling sequences of notes. Thus, the song score is not changed since the note onsets and duration contain the necessary information. The default HTS framework works with 5 states per model (phonemes). We have used the same number of states per model (notes). We tried modeling notes with 7 states per model but no relevant changes were observed. The input data to the system are absolute pitch and dynamics values from the expression database. The same type of data is predicted for any given target score. In the training section we detail why the database used for training has been modified.

5.3.2 Contextual labels for clustering

In this system, and with the unit concept as a reference, the questions are related to features of the modeled note sequence, like the note pitch, duration, amount of notes in the target song. We are not using timbre related questions to the phonemes for instance. Besides, we want to help the clustering step and ensure that different notes are not clustered together, and this is the reason we have different trees for each model and state.

As an example of contextual labels, in Listing 5.1 we show how the first 3 notes for one song (*Alone together*) may be specified. Each line has the start and end time followed by the contextual labels. These have the central note, with the 2 preceeding and succeding notes (*xx* refers to no context, *sil* refers to silence). The central note is surrounded by the - and + delimiters. Next, we have more contextual information separated by several fields (D, E, F, and J) and unique delimiters (characters like !, ∼, +, /, and #). These labels specify information on the previous note identity and duration (D field), the central note identity, duration, the interval with the previous note, and the interval with the next note (E), the succeeding note identity and duration (F), and the number of notes in this song (J field).

The yes/no questions for clustering the contextual labels are specified in *questions.hed* file. These questions are specified by following the patterns described in the contextual labels. For instance, the set of questions that check if the central note (*C-note*) belongs the the 4th octave are shown in Listing 5.2. Other questions, ask on the note pitch of the left most note, the left note, the

```
0 16892517 xx^sil-F4+D4=E4/D:sil!9/E:F4]17~xx+m3/F:D4#3/J:62
16892517 20375510 sil^F4-D4+E4=sil/D:F4!17/E:D4]3~m3+p2/F:E4#17/J:62
20375510 37732427 F4^D4-E4+sil=D4/D:D4!3/E:E4]17~p2+xx/F:sil#4/J:62
```

Listing 5.1: Note HMM-based system: Contextual labels example

```
QS "C-Note_C4" {*-C4+*}
QS "C-Note_Db4" {*-Db4+*}
QS "C-Note_D4" {*-D4+*}
QS "C-Note_Eb4" {*-Eb4+*}
QS "C-Note_E4" {*-E4+*}
QS "C-Note_F4" {*-F4+*}
QS "C-Note_Gb4" {*-Gb4+*}
QS "C-Note_G4" {*-G4+*}
QS "C-Note_Ab4" {*-Ab4+*}
QS "C-Note_A4" {*-A4+*}
QS "C-Note_Bb4" {*-Bb4+*}
QS "C-Note_B4" {*-B4+*}
```

Listing 5.2: Note HMM-based system: question file example

right note, and the right most note. Also, the start and end note (after/before silence as central note) of the phrase, the upper bound in pitch and duration of the left note, central note, and right note, as well as, the left and right pitch intervals.

5.3.3 Training

We have used the Systematic and the Song expression databases to train the systems. Since the system is modeling note sequences using the absolute pitch values, we have had to pitch shift the expression databases in order to cover a wide tessitura range which contains all possible note values for the target songs. Thus, the used training databases are the original one plus the pitch-shifted versions at ± 1 and ± 6 semitones. Therefore, the training databases are 5 times bigger than the original size.

5.3.4 Synthesis

Vibrato features postprocessing

As we will see in the results section, the system generates depth and rate which are coherent (in the sense that most of the time these are 0, and when there is vibrato both contours are different than 0). However, sometimes a vibrato segment (consecutive non-zero values) are too short to be realistic. In such cases, we are not considering the vibrato to appear in the real output of the system. We have considered that shorter vibratos than 0.1 seconds should be filtered out.

Moreover, for longer vibratos, it usually happens that if vibratos are synthesized from these contours, the vibrato may end at any phase from the last vibrato cycle. A random cycle phase may produce a discontinuity after the last

vibrato frame, since vibratos are added to the baseline pitch, and the values are not continuous in most cases. Therefore, we have enlarged vibrato rates by computing the amount of frames that are needed to finish a vibrato cycle appropriately. The corresponding vibrato depth frame are extrapolated from the predicted values.

Final expression contours

Similarly to the unit selection-based system, the final pitch contour is generated by generating the vibrato contour from the depth and rate contours, and the result is added to the synthesized pitch. The details have been explained in section 4.4. Concerning dynamics, we can directly use the output values from the system.

5.4 Transition and sustain HMM-based system

5.4.1 System description

In this section we describe a modification of the Note HMM-based system. In this case, the system is characterized by modeling sequences of transitions and sustains instead of notes. Then, the first changes that we have had to introduce are related to the yes/no questions to build the corresponding tree and leaf nodes (section 5.4.3).

As described in Chapter 3, the expression databases labeling includes the start and end time of transitions. However, if this information is not available for the target score it has to be estimated for this system (section 5.4.4). We have used the labeled expression databases to learn how transitions deviate from note onsets and this model is applied to the target scores.

Concerning the input data, this system uses the absolute dynamics and the difference between the pitch contour and the reference pitch contour estimated from the nominal score (section 5.4.5). The generated data is of the same type of data, absolute dynamics and pitch difference. Therefore, the final pitch contour has to be reconstructed by estimating the baseline pitch from the score and adding the fluctuation around it, that is to say, the synthesized pitch difference contour.

The advantage of using the pitch difference instead of the absolute value, is that what is being modeled is the fluctuation around the estimated pitch reference from the nominal score. Thus, it is no longer necessary to pitch shift the input data in order to cover a wide tessitura. As a consequence, the training database has a smaller footprint compared to the database used in the Note HMM-based system.

The systems' characteristics are compared in Table 5.1. In this table we summarize the main features: what is being modeled (note vs. sustain and

```
1  s1 e1 xx^sil-attack+sus=tranm/D2:xx/D:9/E:17~xx+xx;p0!m2/F:17/F2:3/J:62
2  s2 e2 sil^attack-sus+tranm=sus/D2:9/D:17/E:17~xx+xx;p3!m2/F:3/F2:3/J:62
3  s3 e3 attack^sus-tranm+sus=tranp/D2:17/D:17/E:3~xx+m3;p0!p0/F:3/F2:17/J:62
4  s4 e4 sus^tranm-sus+tranp=sus/D2:17/D:3/E:3~xx+m3;m2!p0/F:17/F2:17/J:62
5  s5 e5 tranm^sus-tranp+sus=release/D2:3/D:3/E:17~m3+p2;p0!xx/F:17/F2:4/J:62
6  s6 e6 sus^tranp-sus+release=sil/D2:3/D:17/E:17~m3+p2;xx!xx/F:4/F2:4/J:62
7  s7 e7 tranp^sus-release+sil=attack/D2:17/D:17/E:4~p2+xx;xx!m2/F:4/F2:7/J:62
```

Listing 5.3: Transition and sustain HMM-based system: Contextual labels example

transition sequence), the amount of states per model (5 in both cases), and how the expression contours are specified (absolute vs. difference value).

5.4.2 Transition and sustain sequence modeling

The transition and sustain HMM-based system models sequences of sustains and transitions instead of note sequences. In this case, instead of having only these 2 possible models, we have distinguished among different types of transitions. Thus, we have grouped intervals equal or lower than ± 1 semitones in the same cluster (which we call *tran0*), and on the other side ascending intervals of more than 1 semitone (*tranp*), and descending intervals of less than -1 semitone (*tranm*). Besides, we have also distinguished the transitions to the first note, and from the last note, or attack and release, respectively.

According to these categories, we have 1 model for sustains and 5 models for transitions (attack, release, and 3 more models according to the pitch intervals).

5.4.3 Contextual labels for clustering

In this section we describe the changes to the *questions.hed* file which specifies the set of questions used to cluster the data from the yes/no questions. In Listing 5.3 we show an transition-sustain sequence for the same phrase as in Listing 5.1. In this case we are not showing the start and end times of each line due to space constraints. The first label fields contain the central sustain (*sus*) or transition (*attack*, *release*, *tran0*, *tranm*, or *tranp*) information with the 2 previous and the 2 succeeding elements.

Next, we have the fields which specify information on their duration and pitch. These labels specify information on the 2 previous notes durations (D2 and D fields), the central note identity, duration, the interval with the previous note, and the interval with the next note (E), the 2 succeeding notes durations (F and F2), and the number of notes in the song (J field).

The yes/no questions for clustering the contextual labels are specified in a new *questions.hed* file. For instance, the set of questions that check if the central element is one type or another of transition, a sustain, or a silence are shown in Listing 5.4. Similarly, the questions of the Note HMM-based system have been adapted with the new type of sequence that it is being modeled.

```
QS "C-Note_tranp" {*-tranp+*}
QS "C-Note_tranm" {*-tranm+*}
QS "C-Note_tran0" {*-tran0+*}
QS "C-Note_sus" {*-sus+*}
QS "C-Note_attack" {*-attack+*}
QS "C-Note_release" {*-release+*}
QS "C-Note_sil" {*-sil+*}
```

Listing 5.4: Transition and sustain HMM-based system: question file example

5.4.4 Transition prediction

One important aspect in the Transition and Sustain system with respect to the Note HMM-based system is that transitions and sustains are modeled instead of notes. However, in a target score only note onsets and their durations are available. Therefore, transitions and sustains should be predicted from the input score in order to create a new score. This new score is a sequence of transitions and sustains instead of a sequence of notes.

In order to be able to predict the start and end times of transitions from the input score, we have used the Systematic expression database to train and test several algorithms. The Systematic database has been split into the 70% and the 30%to generate the train and test datasets, respectively, and we use one song from the Song expression database as the validation dataset.

We have trained several possible estimators like regression trees, regression with K-Neighbors, and random forests with the *Scikit-learn* python module[2]. For each one, several configurations have been tested (with 10-fold cross-validation) in order to see which one provides the least mean square error. For instance, several regression trees have been trained by varying the minimum number of examples per leaf in the tree from 1 to 100. Several K-nearest neighbour regressors has been trained by varying the number of neighbors from 2 to 80. Finally, several configurations for the random forests have been trained similarly to the regression trees.

For all the tested algorithms we have used the same contextual information as input in order to predict the start and end transition times. This contextual information refers to the central note duration and the pitch interval with the next note, and the same information for the 2 previous and 2 succeeding notes. Besides, the number of notes in the song is also used. From this contextual information, the 2 transition times are trained. These time instants are trained and predicted in their relative value. Concerning the start transition time, the relative value is computed with respect to the duration of the first note of the corresponding interval. The relative end transition time, is computed with respect to the duration of the second note of the corresponding interval.

In order to choose one algorithm for transition times prediction, we have computed several parameters from the predicted time values. The mean square error has been computed for the best algorithm configuration in absolute and

[2] http://scikit-learn.org/

Dataset	Relative or absolute value	MSE value	Kneighbours Regressor	Regression Tree	Random Forest
Test	perc	start time	0.0078	0.0048	**0.0042**
		end time	0.0113	0.0076	**0.0073**
	abs	start time	0.0083	0.0051	**0.0044**
		end time	0.0137	0.0093	**0.0089**
Validation	perc	start time	0.1101	0.1153	**0.1050**
		end time	0.1682	0.1642	**0.1640**
	abs	start time	0.0943	0.0862	**0.0721**
		end time	0.0948	0.0772	**0.0738**

Table 5.2: Mean square error for the transition start and end times (in seconds).

relative values for the start and end times prediction of the test and validation datasets. These values are summarized in Table 5.2. From these figures, we have selected random forests as the algorithm to predict the start and end transition times.

Next, we present several results focusing on the random forests predictions. The configuration with the least mean square error (MSE) with the training data uses at least 2 samples per leaf in the prediction of the start transition time, and at least 1 samples per leaf in the prediction of the end transition time. The evolution of the MSE according to the minimum number of samples per leaf is shown in Fig. 5.2. Besides, a set of histograms on the predicted transition times are presented in Fig. 5.3. First, we show the distribution of the ratio between the predicted and the real transition durations (it should be as close to 1 as possible, the peak is around 1.25), as well as the distribution of the ratio between the duration of the overlaping region and the real transition duration (It should be around 1, where the peak is placed). Next, we show the distribution of the error in the prediction of the start time, which is presented as the ratio with respect to the first interval note duration (it should be placed around 0.0 and the mean is around 0.2). Similarly, we show the distribution of the end transition time prediction error expressed also as a ratio with the second interval note duration (in this case the mean is aournd -0.27, although the peak is placed around 0).

We note that although this is the configuration proposed by the python module we have used to predict note transitions and sustains, there might be over-fitting given both the MSE errors that we get and the low number of samples used in the leaf nodes. This issue should be further studied in future research works.

5.4.5 Pitch difference

As we have introduced, the Transition and Sustain HMM-based system models the fluctuation of the pitch contour around the nominal pitch contour esti-

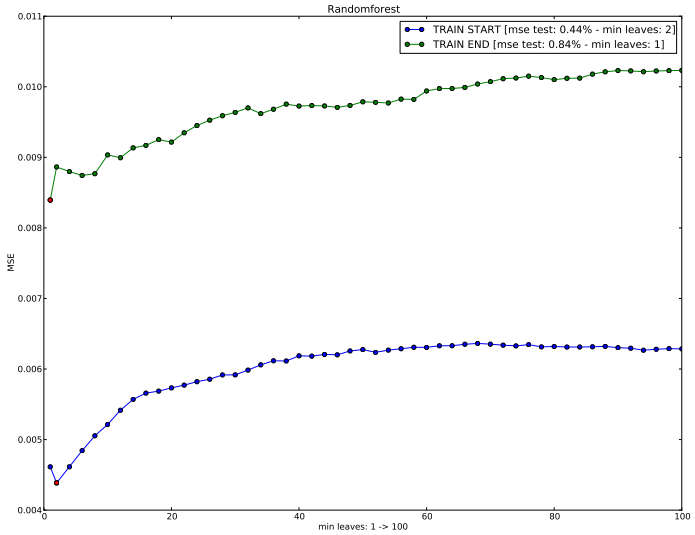


Figure 5.2: Random Forests: MSE vs. minimum number of samples/leaf.

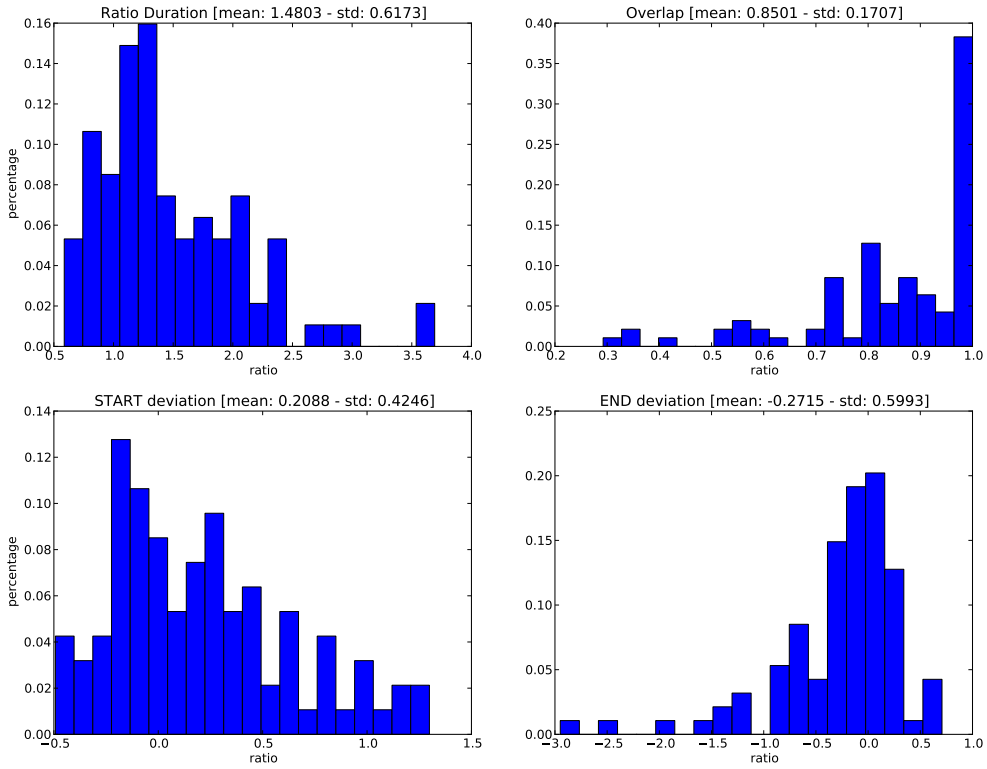


Figure 5.3: Random Forests: histograms on the predictions.

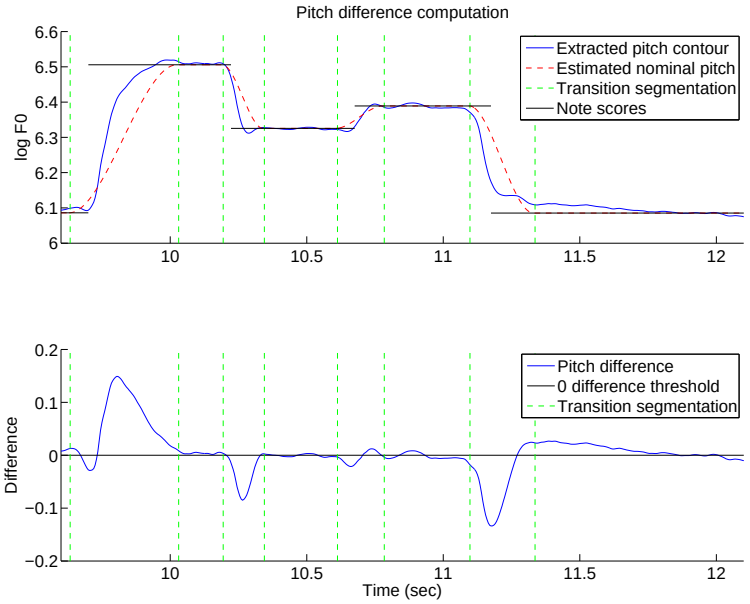


Figure 5.4: Pitch difference computation.

mated from the score. The nominal pitch contour is estimated from the score segmentation into sustains and transitions.

More concretely, the nominal pitch is the cubic interpolation of the transitions segmentation and the pitch values at these time instants according to the score. Therefore, during sustains the interpolated pitch values at the HTS frame rate (5ms) is a flat line. In transitions, the interpolation has a smooth (cubic) shape from the start of a transition time and its note pitch to the corresponding end time and note pitch.

These details can be observed in Fig. 5.4. Note that the contours are in the log scale since this is the format we use for pitch with the HTS framework. In the top figure we show the pitch from the performance (blue), the labeled notes (black) and transitions (green), and the estimated nominal pitch (red dashed line). The cubic shape of the estimated nominal pitch can be observed between the transitions segmentation marks. In the lower figure, we show the difference between the pitch contour and the estimated nominal pitch. During sustains, the difference tends to be around the 0 threshold (black). In transitions, depending on the which pitch contour is greater, the shape of the difference is positive or negative. In the example, we show 4 transitions. While the pitch difference in the first one is positive (the pitch is greater than the nominal pitch), the difference in the second and fourth ones is negative. The third transition is a special case in which the pitch contours cross each other and therefore the difference is partly positive and partly negative.

5.4.6 Training

The training of the Transition and Sustain HMM-based system is done in a similar way as the Note HMM-based model. The most important difference is that in this case there is no need to pitch shift the database to cover a wide note range. Since we are modeling the difference between the pitch and the nominal pitch contour, these differences can be applied to any note pitch. We have trained our system with both the Systematic and the Song expression database to synthesize a set of songs which are evaluated in the Chapter 7.

5.4.7 Synthesis

The synthesis step for this HMM-based system is very similar to the Note HMM-based system. Thus, the vibrato features are also postprocessed before generating the final pitch contour. The only difference is that the synthesized pitch is the fluctuation around the nominal pitch. Hence, the final baseline pitch is generated by computing the nominal pitch from the input score which is then added to the synthesized pitch contour.

5.5 Results

In this section we visualize how the yes/no questions have clustered the training data as well as the synthesized expression contours. We evaluate the synthesized voices in Chapter 7 on the Evaluation.

Clustered data

The yes/no questions in the *questions.hed* file cluster the contours according to the answers to the specified questions. Concerning the F0 feature tree clustering, we show an example in Fig. 5.5, in this case with much less questions, although other trees use more questions to reach the leaf nodes. This tree corresponds to the transition between 2 notes with an interval of ± 1 semitone or less. The first question checks if the pitch interval between the 2 left notes is lower than -2 semitones. The second level question check if the central note (C) length is shorter than 11×0.1 seconds. The last questions check if the pitch interval between the 2 left notes is lower than -3 semitones, and if the right most note (C) length is shorter than 2×0.1 seconds.

On the other hand, for the dynamics feature in Fig. 5.6 we show how the contexts have been grouped for one of states of the transition model of ± 1 semitones or less. Trees from other models contain more questions to reach the leaf nodes, we have chosen these one since it is small enough to be shown. The lines that join the node questions have different colors depending on the answer. The *yes* is marked in blue, while the *no* is marked in red. The first question refers to note interval between the 2 left most notes of the contextual data, and

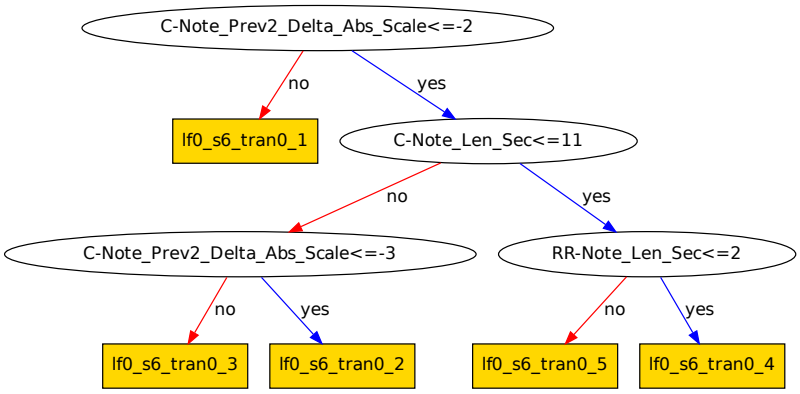


Figure 5.5: Transition and Sustain HMM-based system: Clustered F0 data.

checks if it is lower than 2 semitones. The second level has 2 questions. The first one refers to whether the central note (C) length is shorter than 8×0.1 seconds. The second question checks whether the duration of the right most note (RR) is shorter than 7×0.1 seconds. Other levels have questions related to the other context notes durations, pitch intervals, etc.

We can visualize the expression contours in the different leaf nodes to check if the contours have a similar shape. For instance, in Figs. 5.7, 5.8, 5.9 we show the similar contour shapes within the clustered leaf nodes for the sustain contours, an ascending interval transition, and an attack, respectively.

Synthesized expression contours

In order to visualize the synthesized pitch and dynamics we have synthesized a set subset of songs from the Song database. In Figs. 5.10 and 5.11 we show the different expression contours for the Note and the Transition and Sustain systems, respectively. Both figures have 4 subplots. The first one shows the target score (black horizontal lines) and the synthesized pitches (baseline pitch, pitch with vibrato, and pitch with extrapolated vibrato features). The second subplot shows the predicted vibrato depth, the selected segments (longer than 0.1 seconds) and the extrapolated frames (although the differences may be appreciated). Similarly, the vibrato rate is shown in the third subplot. Finally, the last subplot shows the predicted dynamics.

5.6 Conclusion

In this chapter we have described the Note HMM-based system for expression control as well as the modifications we introduced. The first system models

sequences of notes using absolute pitch and dynamics frame values as training data. The context-dependent data has been related to the unit in the previous chapter, and the different fields that we use have been described (note onsets, note durations, note pitch values, pitch intervals, and number of notes).

The Transition and Sustain HMM-based system models sequences of sustains and transitions and uses absolute dynamics frame values and pitch difference frame values (fluctuation of the pitch around the nominal pitch). Similarly, the context-dependent labels have also been described, as well as the methodology to compute the transition start and end times from the input score.

In the following chapter we combine the ability of modeling time series with comprehensive context-dependent labels from the statistical systems with the synthesis of contours that contain the recordings' fine details from the unit selection-based systems. The combination of these approaches is done by estimating the baseline pitch with the statistical methods and then using this contour as a reference pitch which is considered by extending the unit selection cost functions.

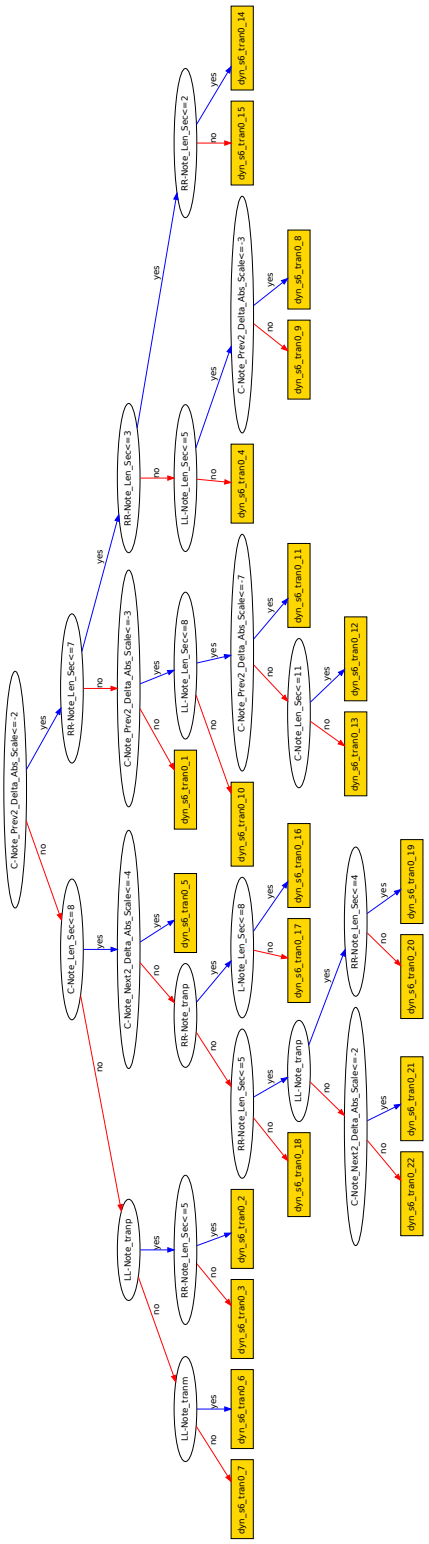


Figure 5.6: Transition and Sustain HMM-based system: Clustered dynamics data.

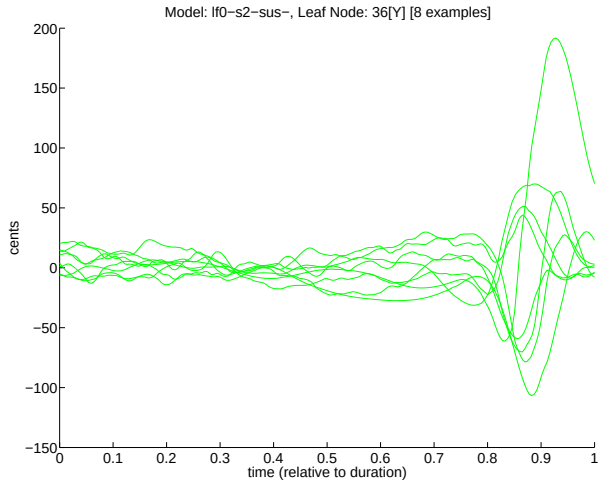


Figure 5.7: Transition and Sustain HMM-based system: sustain clustered contours.

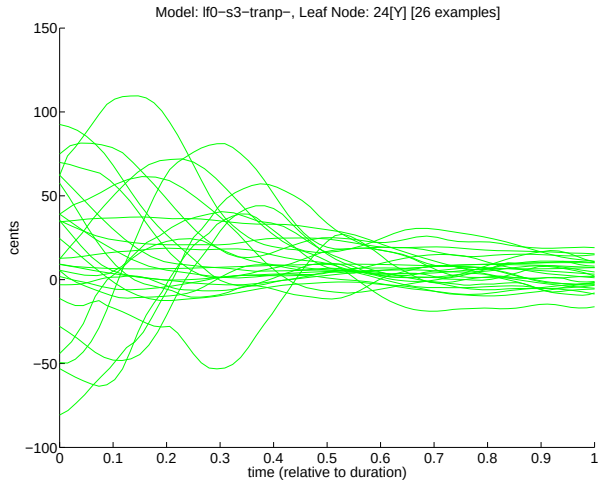


Figure 5.8: Transition and Sustain HMM-based system: ascending transition clustered contours.

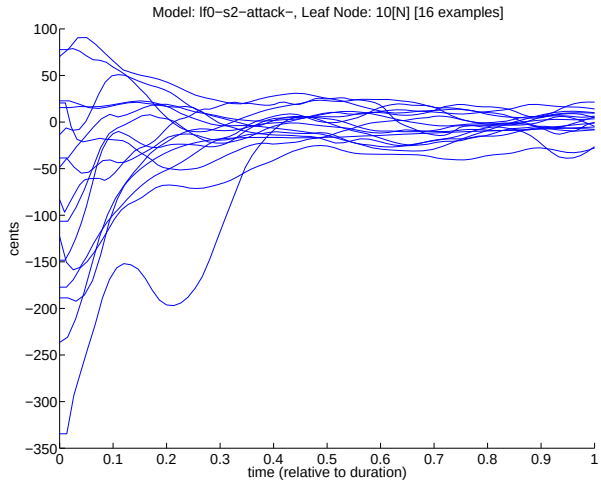


Figure 5.9: Transition and Sustain HMM-based system: attack clustered contours.

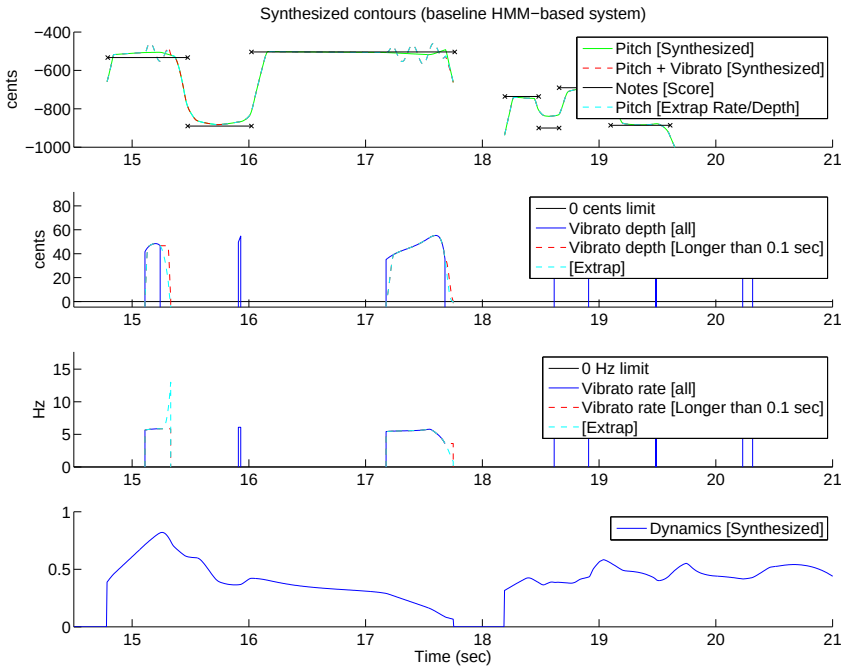


Figure 5.10: Note HMM-based system: synthesized contours.

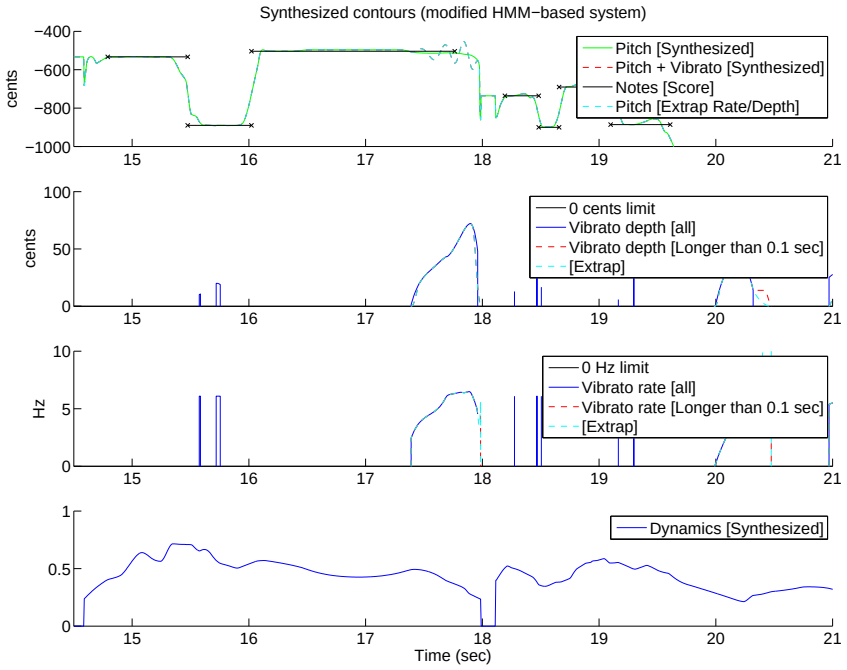


Figure 5.11: Transition and Sustain HMM-based system: synthesized contours.

CHAPTER 6

A hybrid-based system for expression control

The current chapter is devoted to explain how the two previous chapters can be combined in order to have a hybrid system for expression control. The basic idea is that the cost functions of the unit selection-based system can be enriched by adding an initial baseline pitch contour which is obtain by the Hidden Markov Model-based system. First, we start by explaining the new first building block of the hybrid system, which is the generation of the baseline pitch by Hidden Markov Models. Next, we explain how this contour can be used to extend the unit selection cost functions by means of Dynamic Time Warping (DTW) as a distance measure among units.

Although this chapter is shorter than the previous systems' descriptions, we have considered appropriate to describe the hybrid system after the unit selection-based system and the statistical-based system have been presented. Since the hybrid system is based on these other 2 systems, only the system's building block and the DTW cost function are detailed. As we have also done in the previous chapters, we provide a set of figures to complement the description.

6.1 Introduction

In this chapter proposing a third new system for expression control of pitch and dynamics we wanted to explore whether the best characteristics of the unit selection system (chapter 4) and the statistical system (section 5.4) could be combined. The resulting hybrid system would benefit from two aspects in which each system is best at. For instance, the unit selection-based system has the advantage of capturing the fine details of the transformed units, while the generated expression contours with statistical systems are smooth. On the other hand, the statistical systems use more complete contextual information than the unit selection-based system.

Considering these advantages and disadvantages of each method we have combined both systems in the following manner (section 6.2). First, given a target score the statistical system can be used to generate the dynamics and baseline pitch expression contours (vibrato features are not considered at this stage). Then, these generated contours are used as a reference in the unit selection-based system by including a new subcost function which measures the distance between them and the candidate source unit expression contours. In short, the first step takes into account the richer contextual information of the statistical systems (indirectly through the generated contours), and the second steps tackles the generation of expression contours with finer details and without the oversmoothing problems.

The distance measure between the target unit and the candidate source unit is done by computing the Dynamic Time Warping (DTW) of both the baseline pitch and dynamics expression contours (section 6.3). The lower the DTW values are, the more similar the compared contours are. Again, the advantage of preferring the unit contours over the statistical contours is that the latter ones are smoother than the former.

6.2 Building blocks

In order to visualize how the unit selection-based system and the statistical system are combined, in this section we show a clearer figure than the one introduced in section 1.4. In Fig. 6.1 we can see the order in which the steps of both systems are organized. First, as in the statistical-based system, contextual data is prepared, the sustain and transition models are trained, and the contours are synthesized. We are only using the baseline pitch without rendering vibratos, and the sound synthesis step is done at the end of the unit selection-based system.

In the hybrid system, the unit selection step in section 4.2 is extended by including a distance measure between expression contours based on DTW. As we have already introduced, this distance is computed to find source units that have a similar baseline expression contours to the contours generated by the statistical-based system.

After the source units have been selected with this new subcost measure based on DTW (and also the other subcost functions), the next steps are the unit transformation and concatenation, contour generation, and the sound synthesis as explained in chapter 4 for the unit selection-based system.

6.3 Hybrid unit selection

The unit selection step in the hybrid system adds one more subcost function to the set of cost functions in the unit selection in chapter 4. Therefore, the complete list of subcost functions is shown in Table 6.1 where the last row is the

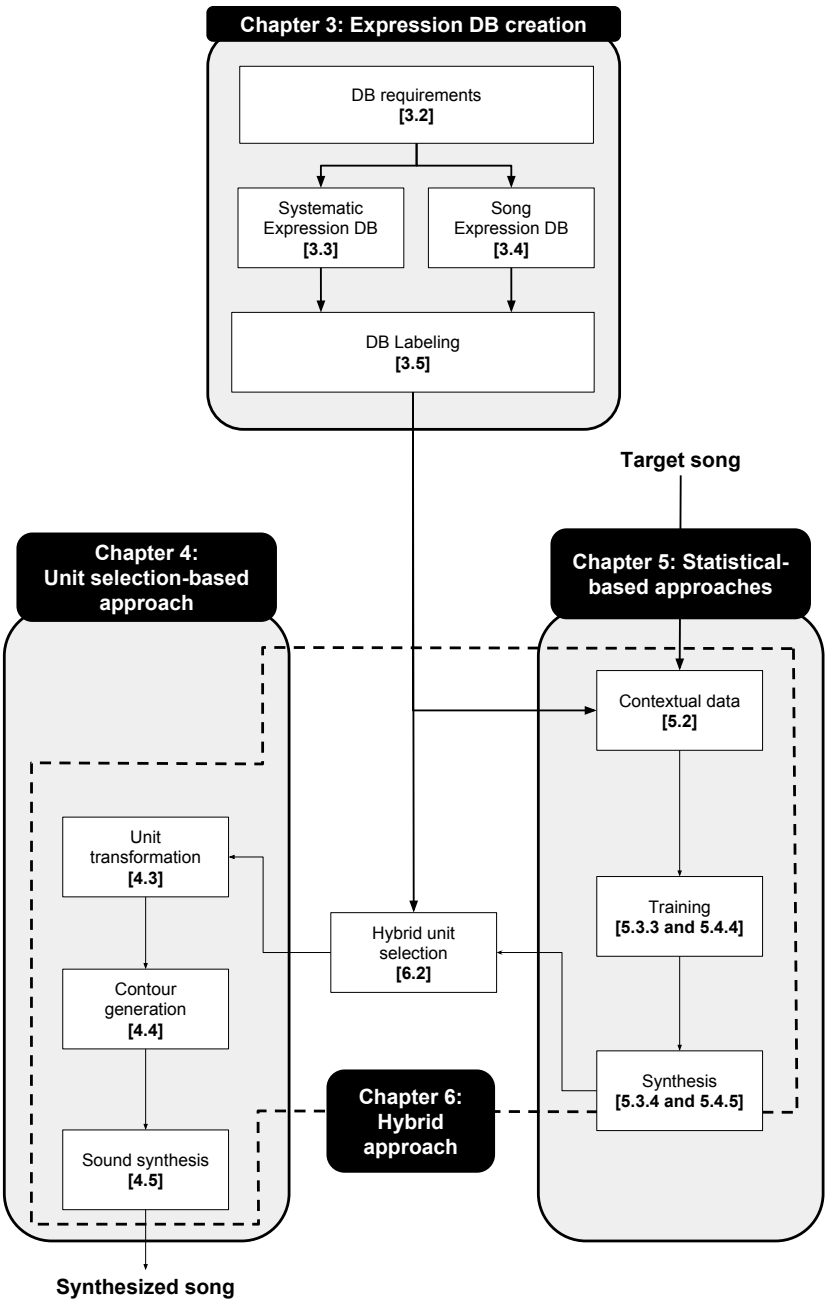


Figure 6.1: Block diagram of the hybrid system.

Cost	Description	Computation
Time-scaling	Compare source and target unit durations	Octave ratio (source/target unit notes)
Pitch shift	Compare source and target unit intervals	Octave ratio (source/target unit intervals)
Note strength	Compare source and target unit note strength	Octave ratio (source/target note strength)
Concatenation	Favor compatible units from the DB	Zero if consecutive units, or depends on transition times
Phrasing	Favor selection of groups of consecutive units	Penalize selection of nonconsecutive units
Similarity	Favor selection of pitch contours close to reference	Dynamic Time Warping cost

Table 6.1: Hybrid system: subcost functions.

new subcost that we add. This similarity cost measures the distance between the pitch contour (in cents) between two units: the transformed candidate source unit and the pitch contour of the target unit which has been obtained with the HMM-based system.

In 6.2 and 6.3 we show an example of the computation of the DTW cost from 2 pitch contours. The former figure shows the 2 pitch contours from which the distance measure is computed. The latter shows the accumulated distance matrix and optimal path to align the two signals.

In eq. 6.1 we show the computation of the DTW as the normalized distance between the two unit pitch contours (t_i and u_i), so that we divide the unnormalized cost ($DTW(t_i, u_i)$) by the length of the optimal path ($DTWlen$). Thus, the DTW cost is independent of the signals length. Finally, since the normalized cost tends to have to higher values compared to the other subcost functions, we compute its $log2$ value to obtain the final $C^t_{DTW}(t_i, u_i)$ cost, which again introduces the idea of octave-based costs explained in section 4.2.2.

$$C^t_{DTW}(t_i, u_i) = log2\left(\frac{DTW(t_i, u_i)}{DTWlen}\right) \tag{6.1}$$

Hence, the $C^t_{DTW}(t_i, u_i)$ cost is added to the transformation cost in eq. 6.2 and is now computed as:

$$C^t(t_i, u_i) = \frac{1}{3}\left(C^t_{ts}(t_i, u_i) + C^t_{ns}(t_i, u_i) + C^t_{ps}(t_i, u_i)\right) + C^t_{DTW}(t_i, u_i) \tag{6.2}$$

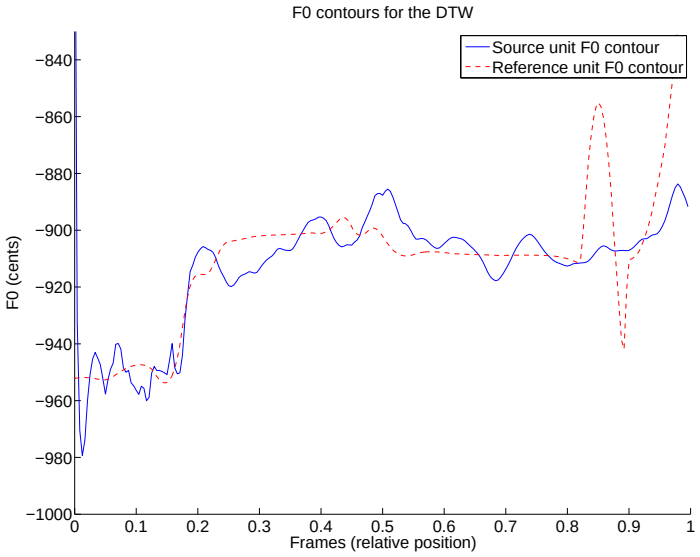


Figure 6.2: Hybrid system: DTW for pitch.

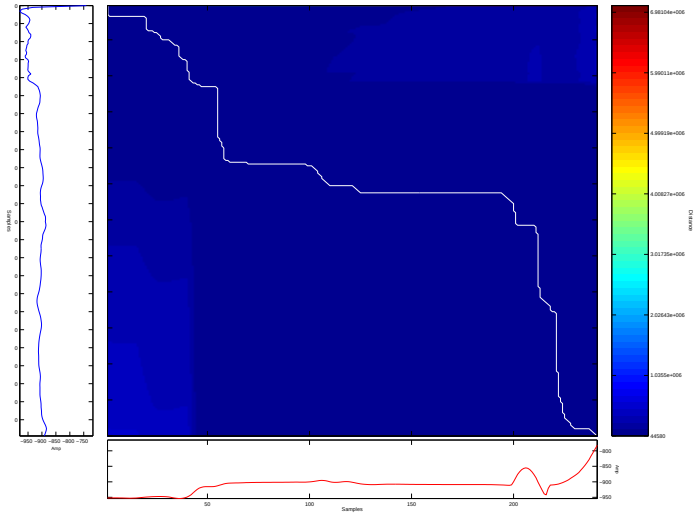


Figure 6.3: Dynamic Time Warping path example.

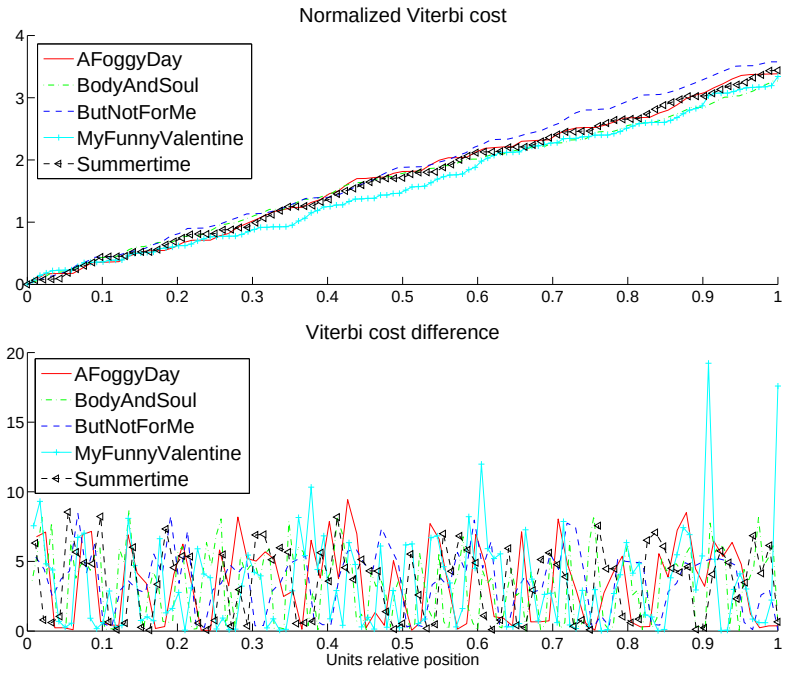


Figure 6.4: Cumulated Viterbi cost.

6.4 Results

Similarly to Chapter 4, we present some figures on the described costs for the hybrid system. First, we present the time evolution of the overall Viterbi cost (the cumulated cost in each node of the Trellis). We have again computed it for the 5 songs that we have evaluated in Chapter 7. More details on these songs can be found in this chapter. We also present the distribution of costs for this system, the length of the selected sequences of consecutive units in the expression database, the time-scaling and pitch interval factors, and a comparison of the reference and generated pitch contours.

Cumulated Viterbi cost

In Fig. 6.4 we present the time evolution of the cumulated Viterbi cost for the 5 songs. The same methodology as in the unit selection system has been followed, we have normalized the cost by the total amount of units in each song in order to be able to compare them. The time axis is referred to the relative position of the units, so that all of them are placed between 0 and 1. On the bottom figure we show the cost increment among consecutive nodes, which we

	Unit selection		Hybrid	
Cost	mean	std	mean	std
Time-scaling	0.89	3.06	0.88	3.03
Pitch shift	0.58	0.86	0.61	1.10
Note strength	0.44	0.62	0.46	0.75
Concatenation	0.20	0.28	0.21	0.28
Phrasing	0.45	0.84	0.58	0.91
Similarity	-	-	3.87	0.85

Table 6.2: Mean and standard deviation of the subcost functions.

can see that have a wider range than in the unit selection system (below 5 in general) since the DTW cost has been included (below 10 in general).

As a difference with respect to Chapter 4, in this figure the normalized Viterbi cost seems to have a more similar evolution in time between songs than in the unit selection system. This may be because the Similarity cost has a greater range of values than the rest of subcost functions in eq. 6.2 as we will see in the next section.

Distribution of the subcost functions

In Figs. 6.5, 6.6, we have plotted the duration and note strength costs to see if the introduction of the DTW cost had some side effect on these other costs. Since the corresponding cost functions have not changed we can see that the distributions of values are very similar to the distributions in the unit selection system. Similary, in Figs. 6.7, 6.8, and 6.9 we show the pitch interval, concatenation, and phrasing costs distributions which behave similarly to the unit selection system.

Besides visual inspection of the distributions in these figures, in Table 6.2 we confirm the low variability in the mean and standard deviation of each of these subcosts when have been applied to the same 5 songs with the Song expression database. Regarding the DTW cost (Similarity), its distribution is shown in Fig. 6.10, with most values within 2 and 6, a mean of 3.87, and a standard deviation of 0.85, therefore having much more relevance than the other cost functions.

Consecutive source units sequence length

In Figs. 6.11 and 6.12 we show the histograms concerning the length of the selected sequences of consecutive units in the expression database. Although there are some variations, the percentages shown in these distributions are very similar to the ones in section 4.2.2 for the unit selection system.

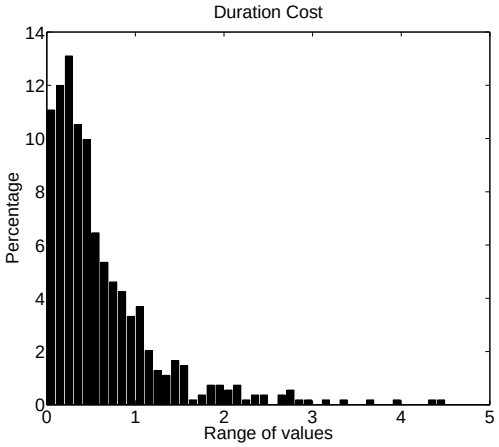


Figure 6.5: Duration cost.

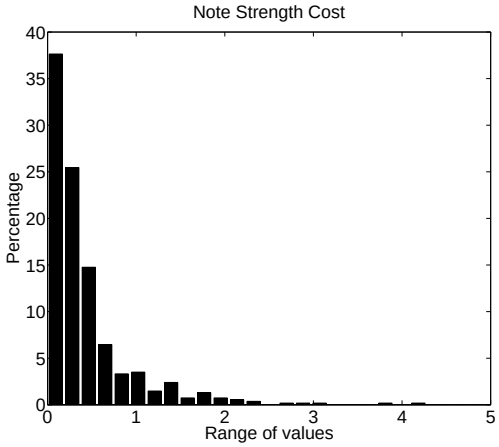


Figure 6.6: Note strength cost.

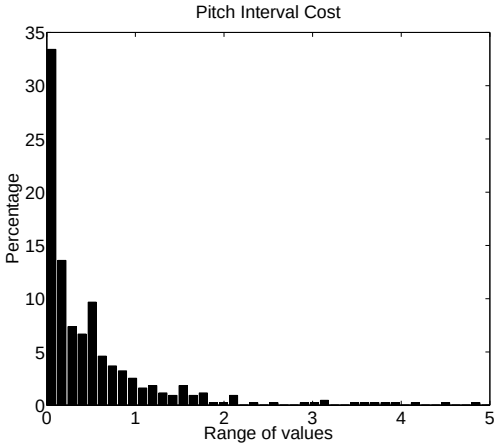


Figure 6.7: Pitch interval cost.

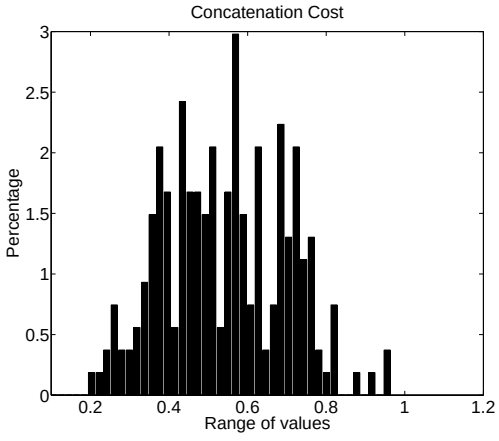


Figure 6.8: Concatenation cost.

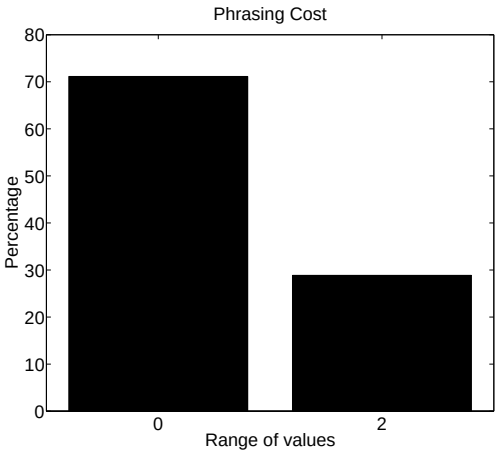


Figure 6.9: Phrasing cost.

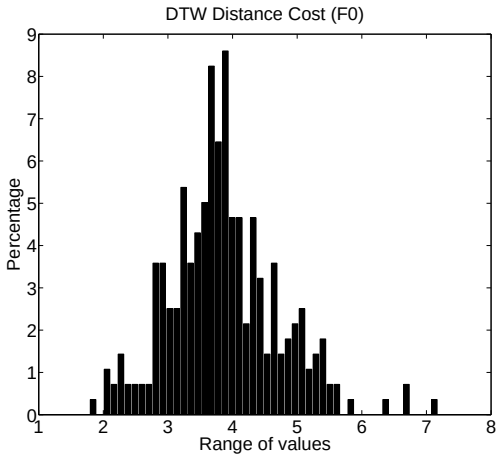
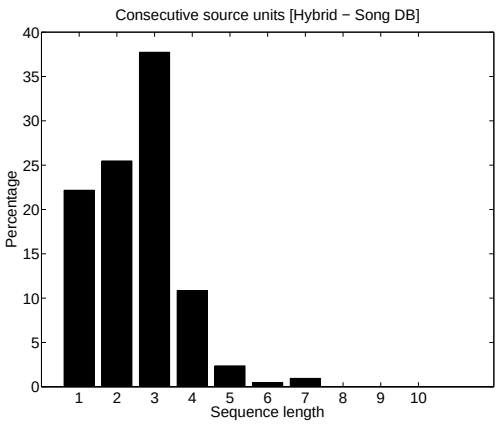


Figure 6.10: DTW pitch cost.



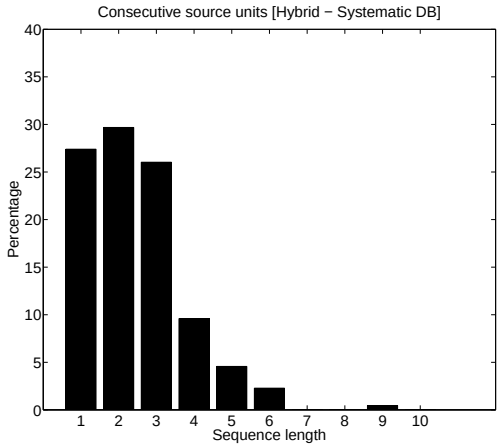


Figure 6.11: Unit sequences (Song DB). **Figure 6.12:** Unit sequences (Syst. DB).

Time-scaling and pitch interval factors

Concerning the degree of transformations actually applied to the selected units we have computed the time-scaling (ratio of note duration between the central note of the source and target units) and pitch interval factors (semitone difference between the selected units and the target units in the central note attack interval). The time-scaling factor is represented in Figs. 6.13 and 6.14 for the Song database and to the Systematic database, respectively. Simlarly, the pitch interval factor is shown in Figs. 6.15 and 6.16 for both databases as well.

The experiment has been done for the same 5 target songs. Similarly to the unit selection system, in both databases source units have been time-scaled with a factor between 0 and 2. The average time-scaling factor are 1.16 and 1.16, and the histogram peaks are placed at 0.87 and 1.05 (almost no change in note duration) for the Song and Systematic databases, respectively.

Concerning the pitch interval, the average factors are -0.01 (nearly no difference in the pitch intervals) and -0.12, and the histogram peaks are placed at 1.02 and -1.11 for the Song and Systematic databases, respectively. As in the unit selection system, in both cases, most of the semitones difference between source and target units is less than 2.5 semitones.

Baseline pitch comparison

Finally, given that the Similarity cost (based on DTW) measures the distance between the candidate source units pitch contour and a reference pitch contour (generated by the HMM-based system in 5.4) we have considered worth visualizing the two pitch contours .

In Fig. 6.17 we show the expression pitch contours generated by the hybrid system and the HMM-based system (in this case both include vibratos as well). The red line (hybrid system) follows the blue dashed line (reference), although there might be some differences like the second note attack, which in the reference pitch it is flatter than in the generated pitch.

6.5 Conclusion

In this chapter we have explained the hybrid system for the generation of pitch and dynamics expression contours. The hybrid systems aims at combining the baseline pitch generated by rich contextual data (used in the modified HMM-based system(with the ability to capture the fine details (used in the unit selection system).

We have described that the unit selection step includes one more subcost function based on Dynamic Time Warping which measures the distance between the reference unit baseline pitch and the candidate source units. The

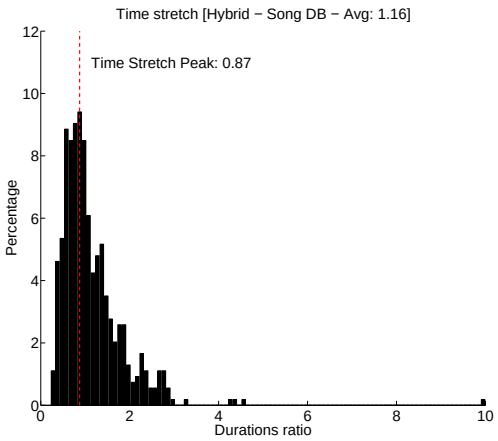


Figure 6.13: Time-scaling (Song DB).

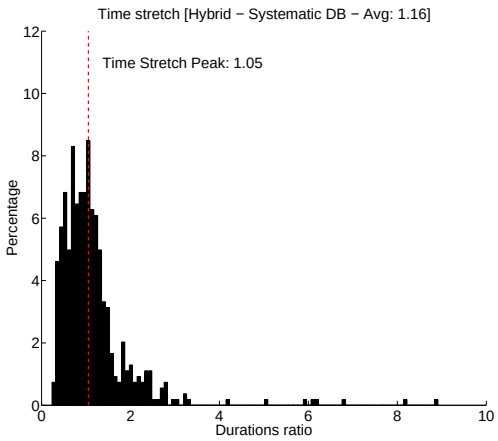


Figure 6.14: Time-scaling (Syst. DB).

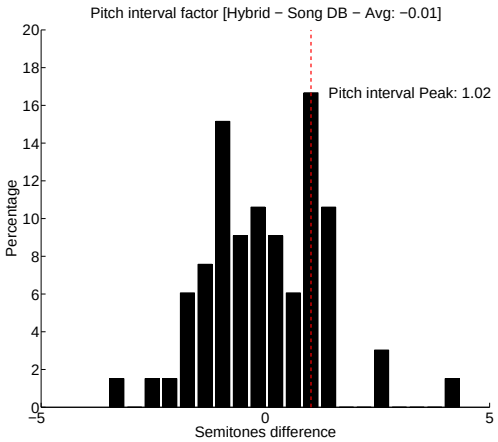


Figure 6.15: Pitch interval (Song DB).

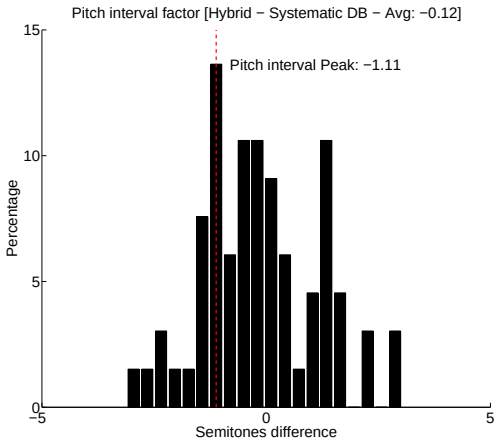


Figure 6.16: Pitch interval (Syst. DB).

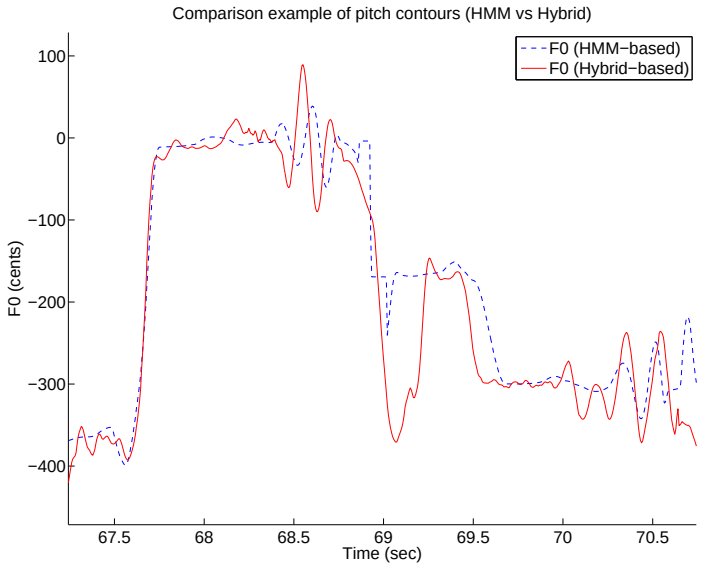


Figure 6.17: Hybrid system: Comparison example of pitch contours

DTW cost is normalized and its $log2$ is computed in order to have a kind of octave-based cost value.

In the results section we have visualized the time evolution of the overall Viterbi cost, the distribution of costs for this system, the length of the selected sequences of consecutive units in the expression database, the time-scaling and pitch interval factors, and a comparison of the reference and generated pitch contours.

This chapter concludes the presentation of the 3 systems in which we have worked in this thesis. The next chapter is devoted evaluate with a perceptual evaluation how the naturalness of the expression is perceived by a set of subjects. Also, we evaluate the computational efficiency of the systems.

CHAPTER 7

Evaluation

In the previous chapters we have described a set of systems that generate expression contours for pitch and dynamics. In this chapter we evaluate them both subjectively and an objectively. First, with the perceptual evaluation we want to know whether the performance of the described systems is perceived natural and at the same time we compare them to other existing systems. Regarding the objective evaluation, we focus on the algorithms efficiency. Besides, we also present a different use case in which the expression contours could be applied, which is to transform a real singing voice recording in order to improve the naturalness of expression. Finally, we discuss on a couple of topics related to the evaluation of the singing voice synthesis systems.

7.1 Introduction

As we have defined in section 1.2.1, although expression is an intuitive aspect of music, it is actually a difficult term to define. Thus, its evaluation is neither an easy task. Nevertheless, either an objective or subjective evaluation of singing voice synthesis systems is necessary in order to gather some knowledge on the task these are asked to perform. As we have presented in in table 2.10, previous works choose one of these two strategies for the evaluation. The evaluation is also adapted to the task being evaluated. For instance, in subjective tests, the group of participants (ranging from 14 to 50 subjects) is asked to rate some aspects of the synthesized voices like voice quality, naturalness, or singing style. In objective tests, an error is computed by comparing the synthesized features with a reference one (F0, intensity, or timbre representation).

In our case, we have envisaged both a subjective and an objective evaluation. Many aspects could probably be evaluated from both perspectives. From the subjective point of view, in section 7.2 we describe the online test. We wanted to know how well different methods perform and whether there is an influence from the training database.

Although the original website of the online test is no longer active, we have collected the same information in the *PhD Evaluation* tab in the thesis site[1].

Next, in section 7.3 we compare the efficiency of each proposed system. Although we cannot compare unit selection methods vs. HMM-based methods due to implementation details, we compare different configurations within each type of method. Then, in section 7.4 we present another use case in which expression control is applied to improving expression in singing voice recordings. This example has been developed at the very end of this thesis and it has not been properly evaluated. Nevertheless, we consider it is worth mentioning it.

Finally, in section 7.5 we discuss on a couple of topics related to the evaluation of the singing voice synthesis systems. On the one hand, we consider that the field would benefit from going towards a common evaluation framework to easily evaluate and compare different singing synthesis systems. On the other hand, adopting perceptually-motivated objective measures would contribute to do comprehensive objective evaluations correlated to subjective measures.

7.2 Perceptual evaluation

The aim of the subjective evaluation is to test up to what point the systems described in this thesis provide naturalness to the expression control of a singing voice synthesizer concerning pitch and dynamics. More details on the aim of the evaluation are detailed in section 7.2.1. In section 7.2.2 we explain the criteria behind the selection of methods, databases, songs, and participants. Next, the conditions that constraint the design of the perceptual evaluation are explained in section 7.2.3. Then, in section 7.2.4 we explain the details of the experiment that we have finally carried out. Finally, the participants demographics is summarized in section 7.2.5, and in section 7.2.7 we provide the analysis based on the provided ratings.

7.2.1 Aim of the evaluation

The aim of the perceptual evaluation is to compare how the naturalness of expression is perceived by a group of participants given a set of systems which use several expression databases. The participants are presented with a set of song excerpts which have been generated by the combination of one method and one expression database.

The questions we want to answer with the evaluation are the following:

1. Are there perceptual differences due to the methods?
2. Are there differences due to the databases? And among songs?
3. Do subjects show differences in the perceived naturalness of expression?

[1] http://www.mtg.upf.edu/publications/ExpressionControlinSingingVoiceSynthesis

Type	Method	Description
Baseline	Performance driven	Expression from real singing voice
	Vocaloid baseline	Default Vocaloid expression control
	HMM-based (1)	Note models, absolute pitch
Contributions	Unit selection	Unit selection based system
	HMM-based (2)	Sustain/transition models, relative pitch
	Hybrid	Unit selection and HMM-based (2)

Table 7.1: Baseline and new methods tested in the evaluation.

In the next section we describe the criteria by which we have selected the different factors that may have an effect on the perceived naturalness of expression.

7.2.2 Selection of methods, databases, songs, and participants

Methods

The methods that we want to evaluate are presented in Table 7.1, which are divided between methods that we use a baseline, and the methods that are a contribution of this thesis. The baseline methods are performance driven (section 2.4.3), the built-in expression in the Vocaloid synthesizer (heuristic rules in section 2.4.4), and the baseline HMM-based method (section 5.3) which models notes in absolute pitch values (*HTSnote*). We expect the performance driven and the Vocaloid baseline methods to be rated as the most and less naturally expressive, respectively.

The other evaluated methods are contributions of this thesis. We have the unit selection-based method described in Chapter 4. Then, there is the modified HMM-based system (section 5.4) which models sustains and transitions in relative values (*HTSsustran*). Finally, there is the hybrid system (Chapter 6).

Databases

The two databases that the methods use in the perceptual evaluation are the Systematic and the Song expression databases (Chapter 3). While the unit selection methods use these database to select, transform, and concatenate units, the statistical methods train models based on sequences of notes or sustains and transitions.

The performance driven and the Vocaloid baseline methods do not use them. The performance driven takes the expression controls directly from the original recording, and the Vocaloid baseline is already built-in in the synthesizer.

Song name	Excerpt duration
But not for me	11.0
Body and soul	14.8
My funny valentine	13.2
My funny valentine	6.9
Summertime	6.3

Table 7.2: Songs names and duration (in seconds) used for the evaluation (2 excerpts where extracted from 'My funny valentine').

Songs

We have selected 5 songs for which to generate the pitch and dynamics contours. The songs in Table 7.2 are jazz standards, the same style in which the expression databases songs and melodic exercises were recorded. Actually, the 5 songs are a subset of the Song expression database, so that the remaining 12 song were used to train our systems. The mean duration of the selected excerpts is 10 seconds, which we consider a long enough musical context to be rated.

The idea is to generate the expression contours for each song with the combination of one method and one database, and then synthesize it.

Participants

Participants are one variable to take into account in the perceptual evaluation. We encouraged people to participate in this perceptual evaluation through several mailing lists from the Music Technology Group as well as external mailing lists from the field (ISMIR, Music-dsp, SMC network, and Music-IR). Two emails were sent to each mailing list as a reminder.

7.2.3 Evaluation constraints

The first constraint is related to the time limitations. We have considered that the perceptual should take less than 30 minutes in order keep the participants attention and avoid fatigue. The organization of the files to compare has to be taken into account as well. Given one configuration, i.e. one song and one database, we generate 6 different excerpts (one per method). Depending on the type of test, all 5 songs may be used for the evaluation or not.

Another constraint is that we want to measure the participants' consistency. This can be done by repeating one configuration, i.e. to repeat 6 files, and comparing the evaluation results. The repeated questions should take place during the same 30 minutes time limit. Note that with up to 5 songs, 2 expression databases we have 10 possible configurations, and a total of 11 including the repeated configuration for the consistency issues.

Configuration	A/B testing					Group testing				
number of songs	1	2	3	4	5	1	2	3	4	5
n. of ratings	15	30	45	60	75	6	12	18	24	30
n. of files/rating	2	2	2	2	2	1	1	1	1	1
n. of files to rate	30	60	90	120	150	6	12	18	24	30
material (min.)	5	10	15	20	25	1	2	3	4	5
1 DB (min.)	12.5	25	37.5	50	62.5	2.5	5	7.5	10	12.5
2 DB (min.)	25	50	75	100	125	5	10	15	20	25
consistency (min.)	37.5	62.5	87.5	112.5	137.5	7.5	12.5	17.5	22.5	**27.5**

Table 7.3: Evaluation duration for A/B and group testings.

With these constraints, we have to decide which test may be the most adequate. We have considered that there are 2 possible tests that might be adequate for what we want to evaluate, A/B testing and a test asking to compare and rate the 6 audio excerpts. From now on we name this second type of testing as group testing as opposed to the pair-wise comparisons of A/B testing. The main criteria to decide which one we should carry out is the one that allows us to evaluate as much audio excerpts as possible.

In Table 7.3 we summarize this criteria for both tests. The first row represents the amount of songs that we may evaluate (from 1 to 5 songs in both tests). Note that for each song to evaluate we want to compare the 6 methods in Table 7.1. Therefore, in the case of the A/B testing we have 15 paire-wise comparisons for 1 song (first column), 30 for 2 songs and so forth. Each comparison involves listening to 2 files to provide a single rating. Thus, 30 files need to be listened to for these 15 ratings, which in average last 5 minutes (counting 10 seconds as the average excerpt duration). However, in a real situation each file may be listened two or three times. Therefore, taking 2.5 as the ratio a file is listened to, it would take 12.5 minutes to rate the 5 minutes audio material of 1 songs pair-wise ratings. Since we want to compare the results for 2 expression database, the estimated perceptual evaluation duration would be around 25 minutes. Finally, adding the consistency question (1 song, 1 database, 12.5 minutes), the perceptual evaluation would last 37.5 minutes. The estimation for the other amount of songs is similarly computed. The closest estimation duration is 37.5, far beyond the 30 minutes limit. Besides, it would allow us to extract conclusions from a single song, which is probably not enough.

Similarly, we can estimate the perceptual evaluation duration for the group testing evaluation. The difference is that for a song, the 6 audio excerpts are listened to one after the other (6 ratings for 1 song), and participants should rate 6 audio files per song. Therefore, there is less audio material to rate for 1 song (1 minute), which becomes 2.5, 5, and 7.5 minutes to rate the files for 1 database, 2 database, and the same with the consistency question, respectively. As we can see in the last column, even rating 5 songs, the 27.5 minutes (highlighted in bold font) estimated perceptual duration is acceptable.

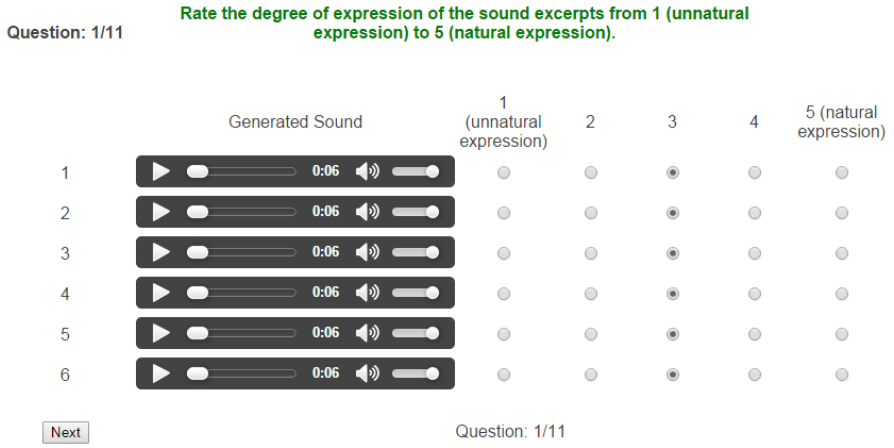


Figure 7.1: Screenshot of the perceptual evaluation website.

Thus, we have finally carried out the group testing with 5 songs. In the next section we explain how we have finally designed the experiment and the online website.

7.2.4 The experiment

Given the constraints explained in the previous section, we have decided not to do pair-wise comparison but to compare all files within one configuration at a time. Since the 6 configuration excerpts corresponding to the same song are evaluated together, from now on we will refer to it as a *question*. An example of a question is shown in 7.1 as shown to the participants in the online website that we prepared.

For each question, participants have been asked to first listen to all of the 6 sound files and then to rate them from 1-5 according to the perceived naturalness of singing expression (1 meaning unnatural expression, and 5 natural expression). We have randomized the order in which questions are presented in order to avoid any learning effect in the participants. Within each question, the order in which the audio files are presented is also randomized. Therefore, there is a low probability that a pair of participants rates the audio files in the same order.

Given the number of songs (5) and databases (2), we have a total of 10 questions. We added one more question which was selected from the 10 previous questions. This repeated question as the *consistency* question as we previously introduced. This question can be used it to check how consistent participants are in the rating process doing the *Spearman* correlation between the 2 sets of rated values. The lower the correlation value the less consistent the participant is with the answers, and therefore his or her answers are less re-

liable. The Spearman correlation is used because we have small samples (each vector has 6 values, 1 per method) and the values are from the ordinal scale (1-5). The consistency measure can then be used to see if there differences in the results between 2 groups of participant: all participants vs. the consistent ones.

The perceptual evaluation has provided four types of feedback. The main ones are the actual rating values from which we can extract some statistics and conclusions on the naturalness of expression. Before the questions, we introduced the task to the participant and we asked some demographics (like gender, age, and familiarity with the field). More details on the task introduction and demographic questions are explained in Appendix B, like what to focus on when listening, the ratings values and their meaninings, or the experiment duration. Besides, the website automatically annotated the time a participant spent to answer each question. This measure, together with the consistency question provide an idea of the difficulty of the task. Finally, we asked participants to voluntarily provide some comments on the task they had been asked to rate. The participants' comments and our observations are detailed on Appendix C.

7.2.5 Participants' demographics

In this section we briefly summarize the results of the first part of the perceptual evaluation. The demographics of the average participant is a 25-34 male, who listens to music every day but does not sing in a choir or band. He has played an instrument for more than 8 years and he is familiarized either with speech/singing voice synthesis or music technology. The fact that there is such a clear participant profile means that in some aspects the histograms are not balanced. However, we do not expect any bias coming from unbalanced gender and age distribution, and in other cases it might be rather positive, like the fact that many participants have played an instrument for several years and that are familiar with the field.

We present the complete picture of the participants diversity in the following figures. In Fig. 7.2 we show the participants' distribution with respect to their age (in 6 the groups described in the previous section) and gender. In Fig. 7.3 we show the answers corresponding to the participants' listening habits and whether they sing in a band or choir or not. Almost 80% of participants listen to music at least "nearly every day" and around 40% of them sing in a choir or band. In Fig. 7.4 we show the time participants have been playing an instrument and the participants' relationship with the topic. More than 45% of participants have played an instrument for at more than 8 years and around 80% of them are familiar either to speech or singing voice synthesis or to music technology in general.

Concerning the time devoted to do the task, in Fig. 7.5 we represent a histogram of the durations of the perceptual evaluation sessions each partici-

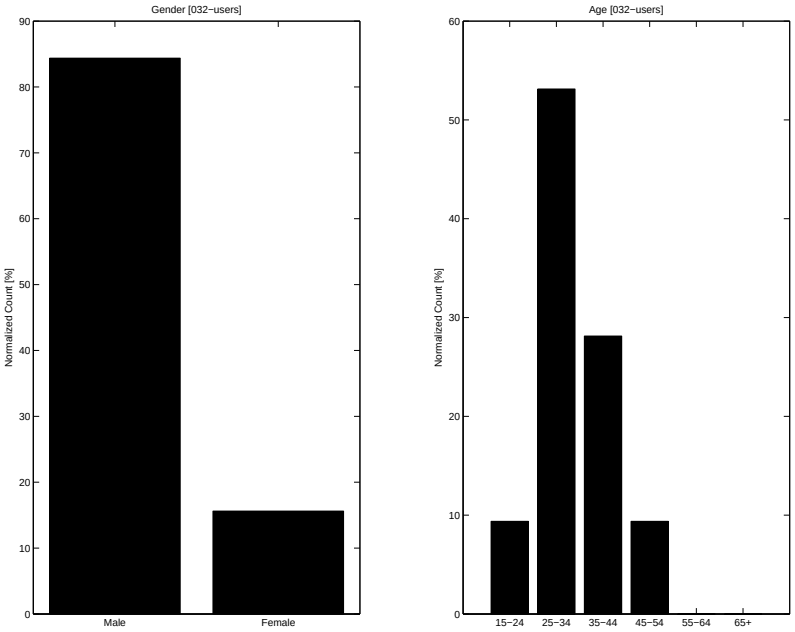


Figure 7.2: Age and gender of the participants.

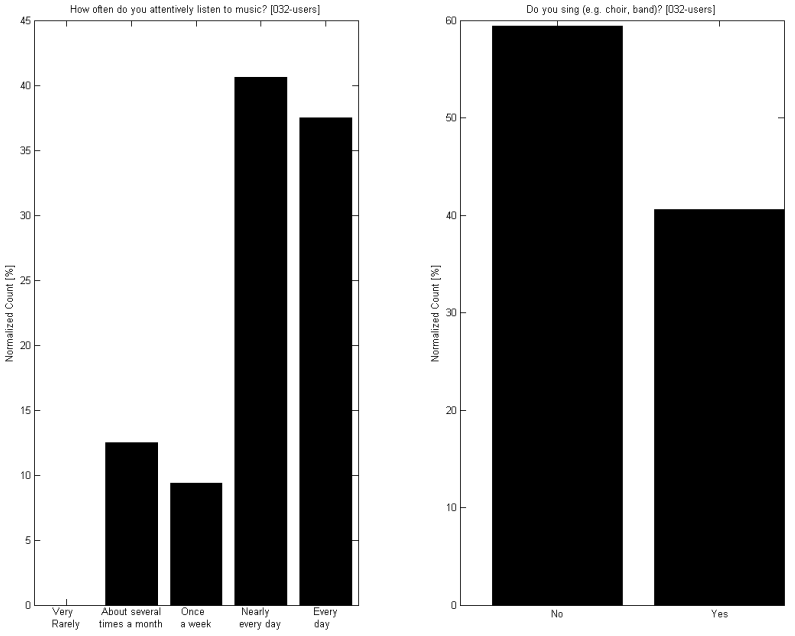


Figure 7.3: Listening and singing characteristics of the participants.

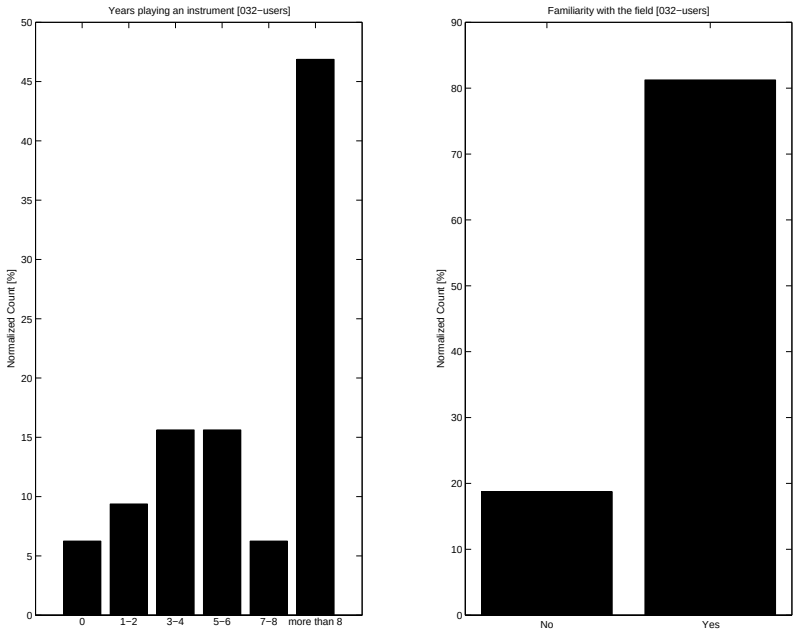


Figure 7.4: Time having played an instrument and familiarity with the topic.

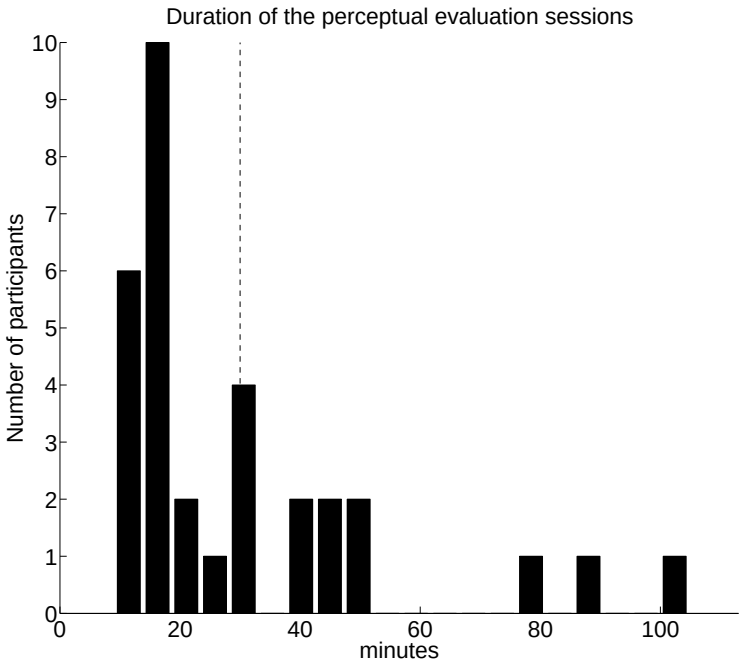


Figure 7.5: Perceptual evaluation session duration.

pant has devoted. The dashed line represents the 30 minutes we considered to be the maximum time a participant would devote to do the perceptual evaluation. Some participants (3) have spent between 40 and 50 minutes, and 3 other participants have spent between 80 and 100 minutes. These might have probably answered the test questions not in a row but with pauses in between.

7.2.6 Statistical analysis of all participants' ratings

The aim of the perceptual evaluation is to visualize the ratings' distribution and analyze if any statistically significant difference depends on the methods, or any other first degree interaction due to the expression databases, the selected songs, and participants. We also want to study second degree interactions like database::song, song::method, or database::method. In this section we do this analysis for all participants' ratings, and in the next one we will focus on the consistent ones to check whether there is any difference or not. The R statistical computing software[2] is adequate for studying these kind of dependencies.

Descriptive statistics

In this section we graphically describe the ratings' distribution in the perceptual evaluation as the basis for the subsequent quantitative analysis of our data. While boxplots are centered around the median, a red cross is be included showing the mean to help see the tendency of the rating values across methods. Boxplots are ordered from left to right by ascending mean value.

First, we compare the ratings' distribution for each database in Fig. 7.6. The variances and median seem to be very similar, showing that there is not a significant difference based on the expression database. Next, in Fig. 7.7 we compare the ratings' distribution for each song (the databases are mixed). We observe similar variances and median, with slightly different means, which are analyzed in the next section. The last two songs (*Body and Soul* and *Summertime*) seem to present a slightly higher mean value than the rest. In addition, the last one has a different variance range. In this case, we are not showing the separate boxplots for the Song and the Systematic databases because we didn't observe differences from the previous one.

In Figs. 7.8, 7.9, and 7.10 we compare the ratings's distribution for each method, first without distinguishing the databases, next for the Song database, and then for the Systematic database, respectively. In this case we observe some differences with respect to the median and mean values. Concerning the Song DB, we may observe that the default and the performance driven systems appear differentiated from the other 4 in the middle. Regarding the Systematic DB, there seem to be 2 groups, the lower three (default and HMM-based systems), and the upper three systems (performance driven and unit selection-based systems).

[2] http://R-project.org/

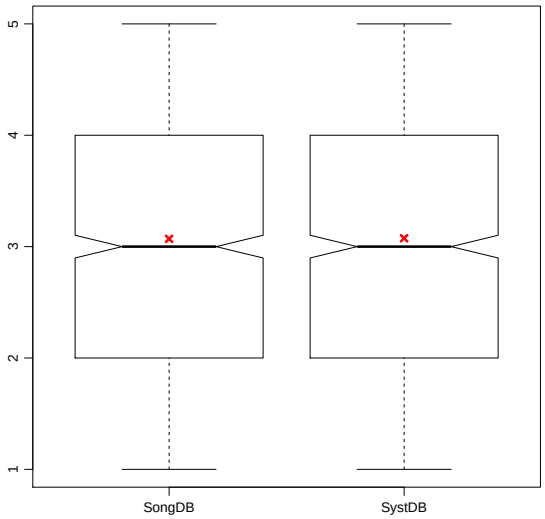


Figure 7.6: Ratings' distribution per database.

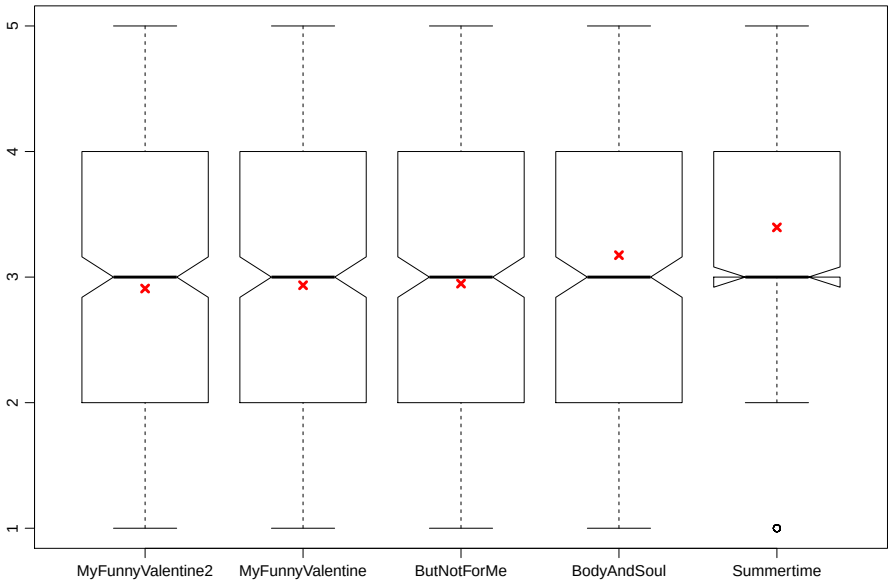


Figure 7.7: Ratings' distribution song.

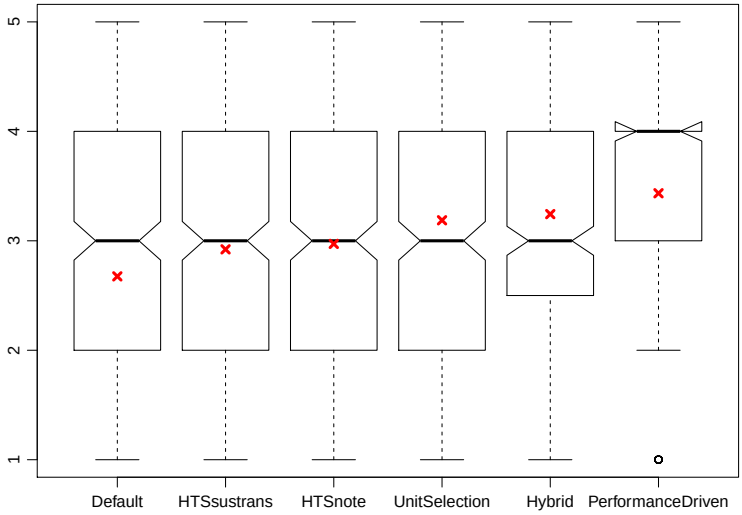


Figure 7.8: Ratings' distribution per method (All DBs).

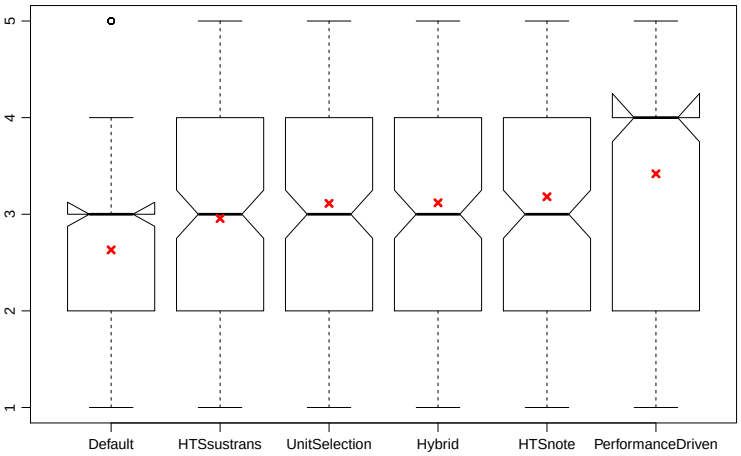


Figure 7.9: Ratings' distribution per method (Song DB).

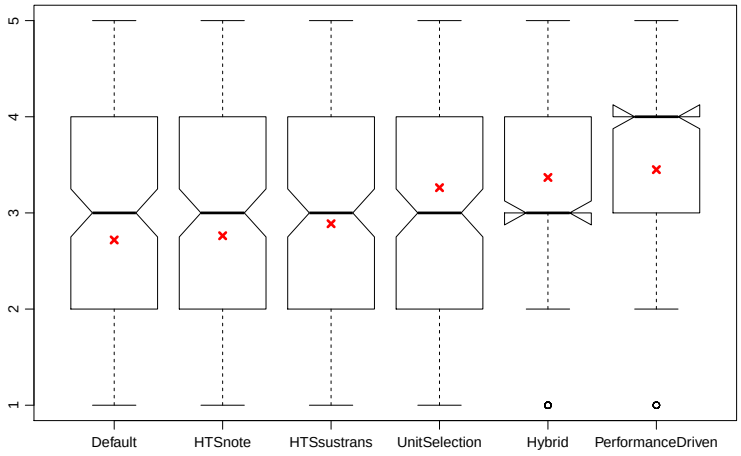


Figure 7.10: Ratings' distribution per method (Systematic DB).

	Df	Sum Sq	Mean Sq	F value	p value
participant	31	510.05	16.45	15.86	***<0.001***
database	1	0.01	0.01	0.01	0.91
song	4	67.65	16.91	16.30	***<0.001***
method	5	116.58	23.31	22.47	***<0.001***
database::song	4	4.08	1.01	0.98	0.41
song::method	20	148.78	7.43	7.17	***<0.001***
database::method	5	21.88	4.37	4.21	***<0.001***
Residuals	1849	1917.89	1.03		

Table 7.4: ANOVA test with all participants.

Inferential statistics

In this section we try to reach some general conclusions on the perceptual evaluation ratings. Basically, we want to check if the differences that we observe in the previous section are statistically significant. In the following tables, we highlight in bold font the p-values which are lower than $\alpha = 0.05$, which is a typical significance level used to reject the null hypothesis.

As we have previously introduced, we want to study the factors that have a significant effect on the ratings. To this purpose, in Table 7.4 we show the results of an ANOVA test. The factors that have a significant effect on the rating ($p<\alpha$) are participants, songs, and methods. The interactions song::method and database::method are also significant. Other factors do not have statistically significant effect on the ratings, like the database and the interactions database::song. Note that we are not including more interactions of level 2 and 3 because we do not expect these to have any effect. Actually, the current analysis is probably enough since we have low residual values in the *Mean Sq* column (1.0373).

The first row of results corresponds to the participants. Given that there too many participants to present the data clearly in a plot, we point out that there are differences among participants. Ideally, the differences should be only due to the expression control methods. This may be an indirect measure of the difficulty of the task we have been asking for in the perceptual evaluation, together with the low number of highly correlated participants, as we will see in the next section.

In the previous section, we have seen there are some differences in the mean and variance of the last 2 songs in Fig. 7.7. The p-value confirms that these differences are statistically significant, and it seems to be due to the ratings of hte Summertime song.

Next, we want to know if the differences that we have observed concerning the methods perception are statistically significant and how this relates to the database::method interaction. To this purpose, the Tukey analysis is shown in Table 7.5. We can conclude that the methods are clustered into several

	Default	HTSnote	HTSsustrans	Hybrid	PerformanceDriven
HTSnote	**0.0031**	-	-	-	-
HTSsustrans	**0.0267**	0.9895	-	-	-
Hybrid	***p<0.001***	**0.0097**	***p<0.001***	-	-
PerformanceDriven	***p<0.001***	***p<0.001***	***p<0.001***	0.1682	-
UnitSelection	***p<0.001***	0.0801	**0.0126**	0.9821	**0.0267**

Table 7.5: Tukey pair wise comparison of methods (p-value for all participants).

groups. First, the Default system is clustered alone since we see that there are differences with all methods. Next, the HTSnote system is clustered together with the HTSsustrans and Unit selection systems. However, the p-value with respect to the second one shows that they might be different since the value is close to α. A third group might be Hybrid system with the Performance driven and the Unit selection systems. However, the Performance driven approach might also clustered alone since the differences with Unit selection are significant. As we can see, these clusters are not homogeneous, but in general the HMM-based systems tend to be in different clusters than the unit selection-based ones.

These results support the fact that the naturalness of the expression synthesized by the unit selection-based methods is closer to a real singer than our HMM-based approaches. However, this has to be limited to our implementation of the HMM-based approaches, since these could probably be improved.

7.2.7 Statistical analysis of consistent participants' ratings

In this section focus on the consistent subset of participants and we reproduce the same steps as in the previous section. First, we identify and filter the subset of consistent participants. Then, we do a descriptive analysis of the ratings. Finally, we extract some conclusions from the inferential analysis.

Consistent participants

As we have previously introduced, the Spearman correlation can be used to filter the most reliable participants and take some conclusions based only on this subset. To visualize this information, we can order all participants by their correlation value. Typically, the empirical distribution function (ecdf) is shown as a function of the correlation values, as presented in Fig. 7.11. For a specific correlation value, it indicates the probability of finding lower correlation values.

If we set 0.2 as a minimum required correlation value (vertical line), it turns out that 17 out of 32 subjects should be considered as the most consistent ones, and therefore more reliable (which corresponds to nearly the 50% ecdf value). The discarded participants are the contractory (negative correlation) and the random (correlation around 0) ones.

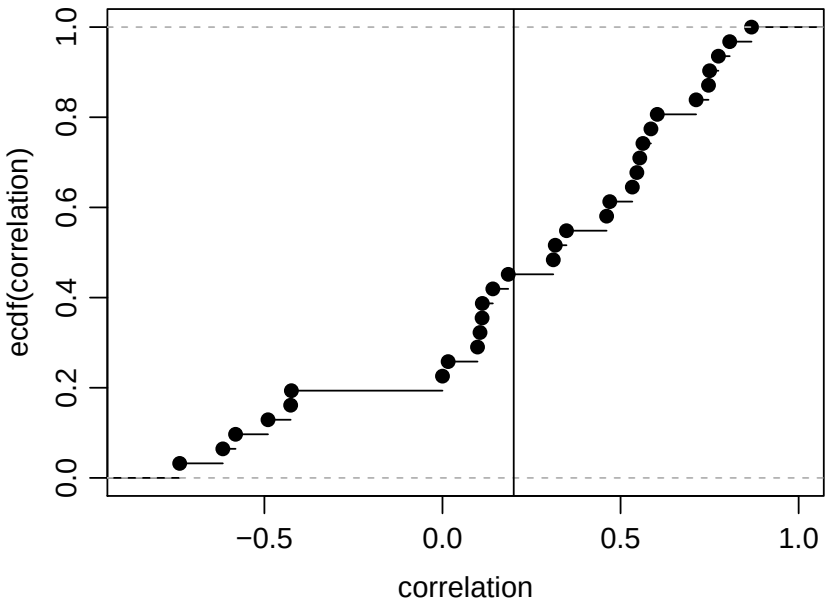


Figure 7.11: Participants' consistency distribution.

Descriptive statistics

Similarly to what we have done with all participants, we show the rating's distribution for per song in Fig. 7.12. Again, there are no differences looking at the 2 boxplots. The rating's distribution per song is shown in Fig. 7.13, and we can see a similar pattern to the previous section, with a higher mean for the Summertime song.

Next, in Figs. 7.14, 7.15, and 7.16 we compare the ratings' distribution for each method with all ratings, the ones from the Song DB database, and the Systematic database, respectively. In this case we observe more differences among methods than with the whole participants set of values with respect to the median, variances, and mean values. Looking at these last two figures, methods can be similarly clustered as in the previous section.

Inferential statistics

The corresponding ANOVA test with the consistent participants' ratings is presented in Table 7.6. Similarly to the ANOVA test of the whole set of participants, the same factors have an effect on the ratings. That is to say, the factors that have a significant effect are participants, songs, and methods, and the interactions song::method and database::method are also significant. The database and the interactions database::song do not have statistically significant effect on the ratings

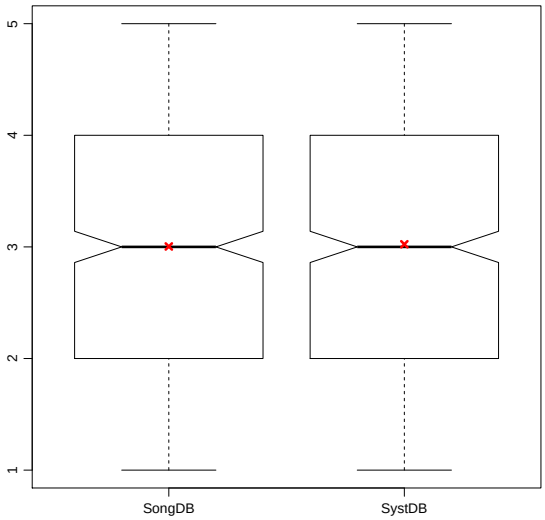


Figure 7.12: Consistent ratings' distribution per database.

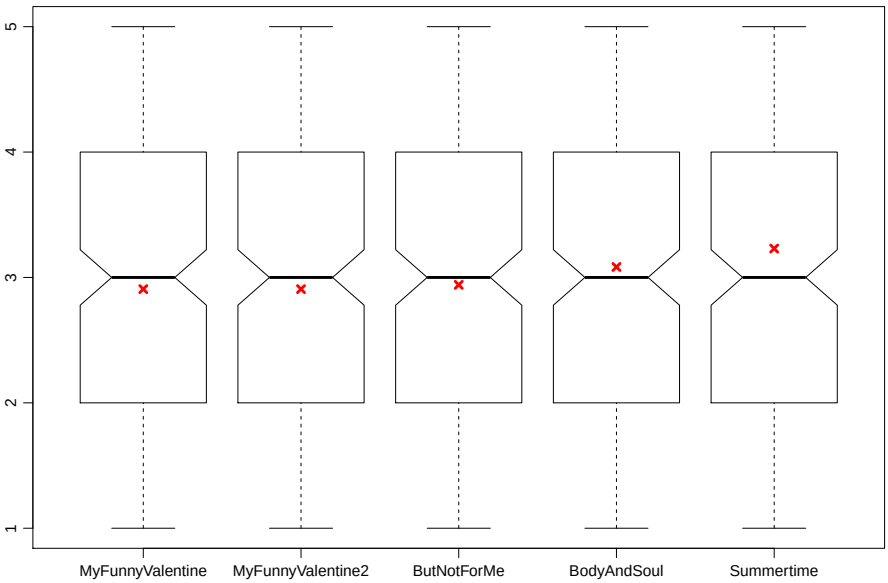


Figure 7.13: Consistent ratings' distribution per song.

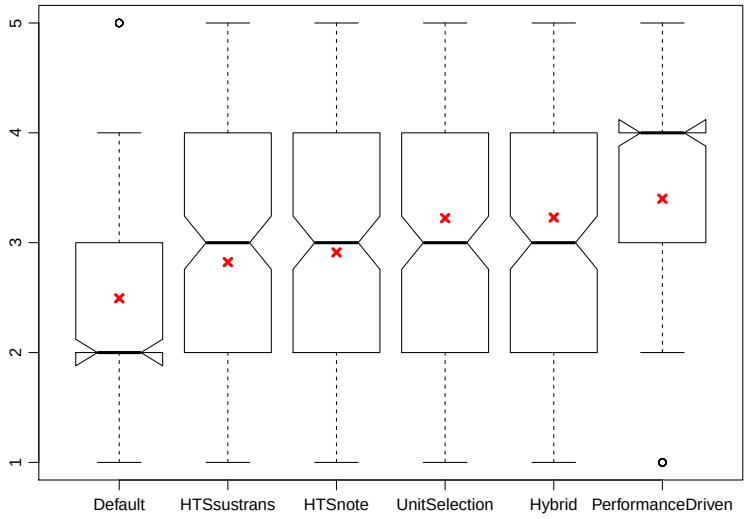


Figure 7.14: Consistent ratings' distribution per method (All DBs).

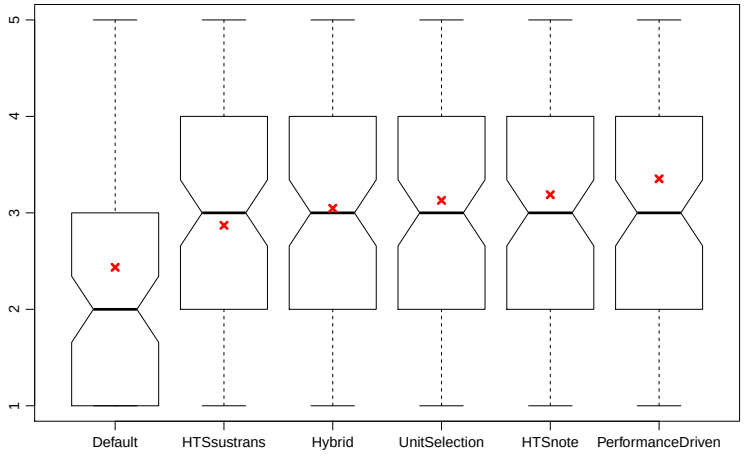


Figure 7.15: Consistent ratings' distribution per method (Song DB).

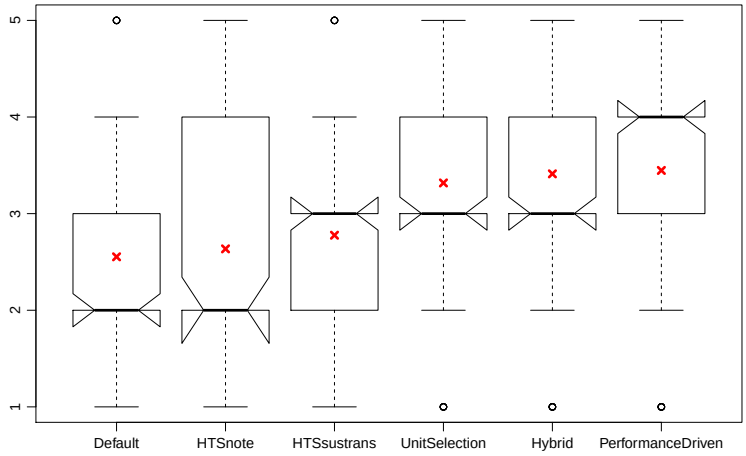


Figure 7.16: Consistent ratings' distribution per method (Systematic DB).

	Df	Sum Sq	Mean Sq	F value	p value
participant	16	153.01	9.56	8.44	***p<0.001***
database	1	0.10	0.09	0.08	0.76
song	4	16.30	4.07	3.59	**0.006**
method	5	94.57	18.91	16.70	***p<0.001***
database::song	4	3.12	0.77	0.68	0.60
song::method	20	103.70	5.18	4.57	***p<0.001***
database::method	5	21.40	4.27	3.77	**0.002**
Residuals	964	1091.62	1.13		

Table 7.6: ANOVA test with consistent participants.

The Tukey analysis in Fig. 7.7, shows in which pair-wise comparisons there are statistically significant differences on how the methods are perceived. From these p-values we can extract nearly the same conclusions as for all participants. With the consistent participants the HTSnote and the Hybrid systems would be clustered together. However, the p-value is close to α. On the other hand, the unit selection and the performance driven systems are now clustered together.

7.3 Efficiency evaluation

In the previous section we have explained the perceptual test that evaluates the methods for expression control of pitch and dynamics. As described in Table 2.10, the subjective perspective is the most common type of evaluation in the analyzed works. Nevertheless, we have considered interesting to compare the efficiency of the described methods. Thus, in this section we provide some insights on the computational cost of the different methods.

7.3.1 Constraints and methodology

Ideally, we would like to compare the time it takes the different systems to generate the expression contours. However, there are 2 expression databases with different sizes and, more importantly, the systems' differ on the implementation. Unit selection-based systems are implemented in MATLAB, while the HMM-based systems are implemented in C. The machine used for this computation has a Windows 7 Professional (32 bits) operating system with 2 Intel Core CPUs at 2.4 GHz.

The implementation constraints makes it difficult to compare unit selection-based systems versus HMM-based systems. Therefore, we only compare the different configurations for the same type of system. Besides, we are not providing data for the Vocaloid baseline or the performance-driven systems because these are straightforward. In the first one the Vocaloid synthesizer is in charge of generating the expression contours according to its internal implementation.

	Default	HTSnote	HTSsustrans	Hybrid	PerformanceDriven
HTSnote	**0.0042**	-	-	-	-
HTSsustrans	**0.0502**	0.9733	-	-	-
Hybrid	***p<0.001***	0.0663	**0.0060**	-	-
PerformanceDriven	***p<0.001***	***p<0.001***	***p<0.001***	0.6785	-
UnitSelection	***p<0.001***	0.0759	**0.0072**	1.00	0.6456

Table 7.7: Tukey pair-wise comparison of methods (consistent participants).

In the second one, pitch and dynamics contours are directly computed from the recorded singing voice performance.

The efficiency is computed differently for each type of system. For each type of system and expression database, we compute the efficiency from the time it takes to process each target song of the perceptual evaluation (the whole song, not the short excerpts). Next, we explain how we compute the efficiency for the two types of systems.

7.3.2 Unit selection-based systems efficiency

In Table 7.9 we show the computation of the efficiency for the unit selection and hybrid systems. First, we have run the systems for the 2 expression databases (*DB name* column), which have different sizes (M), to generate the expression contours for the same songs (*Song name*) in the perceptual evaluation. Each song has a different number of units (N). We compute the duration (*step duration*) of each step: unit selection (US), unit transformation and concatenation (TC), and generation of the contours (G). The duration values of each step is normalized. The unit selection cost depends on both the size of the expression database and the number of units of the target song. Therefore, we normalize this durations as in equation 7.1.

$$C_{US} = \frac{dur_{US}}{M \times N} \tag{7.1}$$

The transformation and concatenation cost depends on the size of the target song. Therefore, we normalize the duration as in equation 7.2. We normalize the time of the contour generation by the same factor.

$$C_{TC} = \frac{dur_{TC}}{N} \tag{7.2}$$

Then, the steps are normalized. Each column is normalized by the minimum values. Next, for a given song, we can sum the 3 normalized costs (*Cost sum*). Then, the efficiency for a given database and system (*DB and system*) is computed as the mean of the 4 songs costs. Finally, the efficiency for the whole system (*system*) is computed as the mean of the two efficiencies for the 2 databases.

System	DB	DB duration	Training time
HTSsustrans	Song	18:29	4519
	Systematic	11:59	5742
HTSnote	Song	18:29 × 5 shifts	48838
	Systematic	11:59 × 5 shifts	31486

Table 7.8: HMM-based systems efficiency.

With the figures in Table 7.9 we can quantify the cost of each step. We can conclude that the Hybrid system is around 15 times more costly than the unit selection-based system given the ratio of the values in the last column. The increment on the cost comes basically from the unit selection step of the hybrid system, since it has to compute the DTW cost between all the candidate source units and the target units. If we look at the normalized costs, we can see the other costs are more or less similar given that the most values are between 1 and 1.5.

7.3.3 HMM-based systems efficiency

The efficiency computation for the HMM-based systems has to be tackled in a different way than in the unit selection. First, the synthesis is a quick steps that takes less than a second for the 5 target songs. Therefore, we should focus on the training step. However, we cannot consider the HMM-based systems to be as linear as the unit selection systems, for example due to the clustering step, which may change the amount of data to process depending on the contexts and how these are clustered.

In Table 7.8 we just show the duration of the expression databases and duration of the training step. Note that the databases for the *HTSnote* system have been extended to cover a wide pitch range by pitch shifting the original database 4 times (±1 semitones, ±6 semitones).

7.4 Improving singing voice recordings expression

We have introduced in section 1.1.1 that beyond singing voice synthesis, software like Melodyne [3] improve the recorded expression of a real singing performance by changing some singing voice features (timing, note durations, tuning, vibrato depth, erasing artifacts, etc). However, a singing voice performance could be improved by adding other aspects which are not present in the recording. For instance, changing the expression at different scopes (from just a note to a whole phrase or song) could provide a significant improvement.

We think that research in this direction could be welcomed in the field, and that more research should be devoted with this respect. Although applying

[3] http://www.celemony.com/

Configuration				step duration (sec)			step cost (sec/unit)			step cost (norm.)			Cost	efficiency	
System	DB name	M	Song name	N	US	TC	G	US	TC	G	US	TC	G	sum	DB and system	System
Unit selection	Song	1254	But not for me	90	52.95	19.66	4.78	0.0005	0.2184	0.0531	1.31	1.03	1.00	3.34	4.32	4.32
			Body and soul	156	93.10	34.34	12.86	0.0005	0.2201	0.0824	1.33	1.04	1.55	3.92		
			My funny valentine	120	60.97	27.77	21.77	0.0004	0.2314	0.1814	1.13	1.10	3.42	5.64		
			Summertime	93	58.98	21.22	9.35	0.0005	0.2282	0.1005	1.41	1.08	1.89	4.38		
	Systematic	982	But not for me	90	40.83	20.35	7.20	0.0005	0.2261	0.0799	1.29	1.07	1.51	3.86	4.04	
			Body and soul	156	66.29	35.19	11.36	0.0004	0.2256	0.0728	1.21	1.07	1.37	3.65		
			My funny valentine	120	42.24	28.50	17.36	0.0004	0.2375	0.1446	1.00	1.12	2.72	4.85		
			Summertime	93	43.39	21.25	6.89	0.0005	0.2285	0.0740	1.33	1.08	1.39	3.80		
Hybrid	Song	1254	But not for me	90	2645.71	20.36	5.28	0.0234	0.2262	0.0587	65.40	1.07	1.10	67.58	61.09	61.09
			Body and soul	156	3965.40	36.12	10.48	0.0203	0.2315	0.0672	56.55	1.10	1.26	58.91		
			My funny valentine	120	2756.81	27.76	22.30	0.0183	0.2313	0.1858	51.11	1.10	3.50	55.70		
			Summertime	93	2484.04	20.99	8.29	0.0213	0.2257	0.0891	59.42	1.07	1.68	62.17		
	Systematic	982	But not for me	90	2104.45	19.03	5.86	0.0238	0.2114	0.0651	66.43	1.00	1.23	68.66	59.95	
			Body and soul	156	3324.53	33.75	10.87	0.0217	0.2163	0.0697	60.54	1.02	1.31	62.88		
			My funny valentine	120	2008.05	26.70	15.64	0.0170	0.2225	0.1304	47.54	1.05	2.45	51.05		
			Summertime	93	1796.29	19.64	6.61	0.0197	0.2112	0.0711	54.87	1.00	1.34	57.21		

Table 7.9: Unit selection-based systems' efficiency.

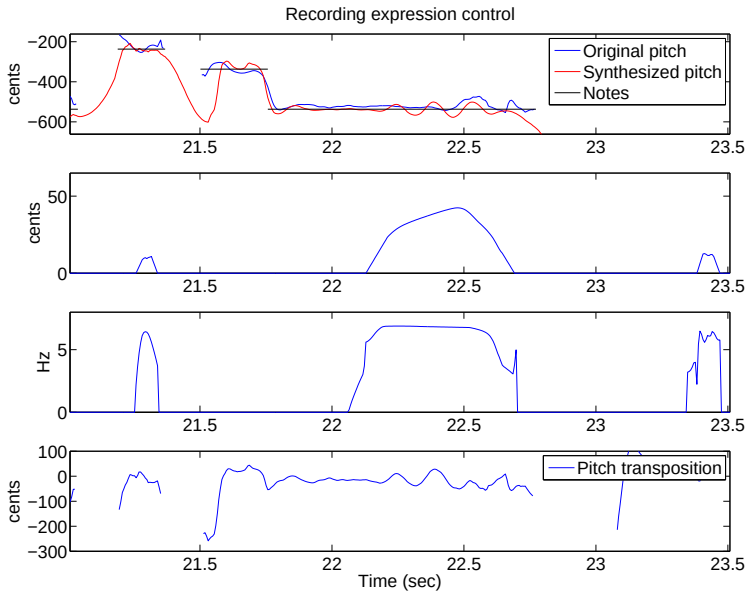


Figure 7.17: Improved expression contours of a real singing voice recording.

the methods explained in this thesis to transform a real recording is probably out of the scope of this thesis, we have tried a prove of concept experiment to show the applicability of the same methods to a recorded singing voice. The results have not properly been evaluated with participants out of our research group, we have just shown to some researchers to know their opinion.

Experiment description

The experiment consists of recording a singer in the studio and then to transform the song in pitch to obtain a performance which is more expressive performance since we asked the singer not to sing the song very expressively on purpose. The recorded song has been processed following these steps:

1. Extract pitch,
2. Segment the notes (onset time, duration, and pitch value),
3. Generate the score from the segmented notes with the unit selection based system using the Song DB.

The extracted score is the target song for which we want to generate expression contours for pitch and dynamics. We have used the unit selection-based system as an example. The extracted expression contours and the generated ones are shown in Fig. 7.17. The top figure shows the pitch contours (both

the extracted and the generated contours) with the estimated notes. The second and third figures correspond to the generated depth and rate contours, which is helpful to locate to generate vibratos. The bottom figure shows the difference between the 2 pitch contours, since this sequence of pitch values (in cents) indicates the pitch shift applied to the original recording.

We have used an in-house tool called Kaleivoicecope (Mayor et al., 2009) to transform the original recording so that the output sound has the generated pitch contour. The original and the transformed excerpts are accessible online[4].

Experiment evaluation

In this case we have not done a comprehensive evaluation due to time limitations. However, we have observed that the outcome is more expressive while keeping the naturalness of the voice at the same time.

This new use case could be significantly improved given that only pitch and has been transformed. Dynamics, timing, and voice quality are not modified. The modification of these features would definitely help to obtain more expressive results. One drawback of the current implementation is that microprosody (see section 2.3.2) is not taken into account, and therefore the pitch is not following the expected shape in some voiced consonants.

On the other hand, there is mainly one positive aspect of this transformation. It is similar to the work in Saitou et al. (2007), which is using speech to generate singing voice. Similarly, in this case, we start from singing voice to generate singing voice. The timbre quality is already natural since it is human-like instead of synthesized, which helps to obtain a more natural result than a synthesized singing voice.

7.5 Discussion

This section, based on Umbert et al. (2015), discusses a couple of topics related to the evaluation of the singing voice synthesis systems. First, we consider that the field would benefit from going towards a common evaluation framework to easily evaluate and compare the singing synthesis systems. Then, we highlight the importance of adopting perceptually-motivated objective measures and how this would also help the field since such measures would allow for comprehensive objective evaluations correlated to subjective measures.

7.5.1 Towards a common evaluation framework

In this thesis we have focused on the naturalness of expression control with respect to pitch and dynamics. However, a comprehensive system for expression control should include all features related to singing voice, as explained

[4] http://www.mtg.upf.edu/publications/ExpressionControlinSingingVoiceSynthesis

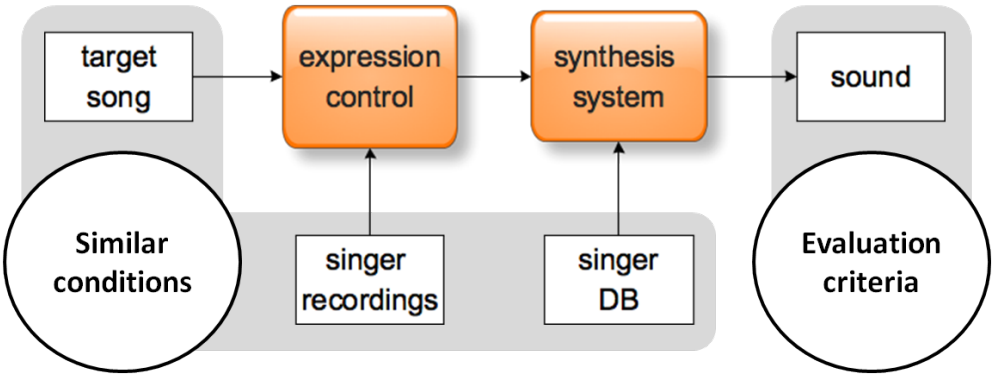


Figure 7.18: Proposed common evaluation framework.

in section 2.3. Depending on the system, comprehensive expression control may involve all the building blocks of singing voice synthesis in Fig. 2.3. As a consequence, if we want to compare different systems, there are too many aspects that differ among systems which make the comparison a difficult task. In this thesis, since we have focused on pitch and dynamics, we have only used a common singing voice synthesizer for all methods, avoiding differences due to other aspects.

Having this in mind, the evaluation methodology could be improved by building the systems under similar conditions to reduce the differences among performances and by sharing the evaluation criteria. Building a common framework would help to easily evaluate and compare the singing synthesis systems.

The main blocks of the reviewed works are summarized in Fig. 7.18. For a given target song, the expression parameters are generated to control the synthesis system. In order to share as many commonalities as possible amongst systems, these could be built under similar conditions and tested by a shared evaluation criterion. Thus, the comparison would benefit from focusing on the technological differences and not on other aspects like the target song and singer databases.

Concerning the conditions, several aspects could be shared amongst approaches. Currently, there are differences in the target songs synthesized by each approach, the set of controlled expression features, and the singer recordings (e.g. singer gender, style, or emotion) used to derive rules, to train models, to build expression databases, and to build the singer voice models.

A publicly available dataset of songs, with both scores (e.g. in MusicXML format) and reference recordings, could be helpful if used as target songs in order to evaluate how expression is controlled by each approach. In addition, deriving the expression controls and building the voice models from a common set of recordings would have a great impact on developing this evaluation

framework. If all approaches shared such a database, it would be possible to compare how each one captures expression and generates the control parameters, since the starting point would be the same for all them. Besides, both sample-based and HMM-based synthesis systems would derive from the same voice. Thus, it would be possible to test a single expression control method with several singing voice synthesis technologies. The main problem we envisage is that some approaches are initially conceived for a particular synthesis system. This might not be a major problem for the pitch contour control, but it would be more difficult to apply the voice timbre modeling of HMM-based systems to sample-based systems.

The subjective evaluation process is worthy of particular note. Listening tests are a time consuming task and several aspects need to be considered in their design. The different backgrounds related to singing voice synthesis, speech synthesis, technical skills, and the wide range of musical skills of the selected participants can be taken into consideration by grouping the results according to such expertise, and clear instructions have to be provided on what to rate like to focus on specific acoustic features of the singing voice, and how to rate using pair-wise comparisons or MOS. Moreover, uncontrolled biases in the rating of stimuli due to the order in which these are listened can be avoided by presenting them randomly, and the session duration has to be short enough to not decrease the participant's level of attention. However, often the reviewed evaluations have been designed differently and are not directly comparable. In the next section, we introduce a proposal to overcome this issue.

7.5.2 Perceptually-motivated objective measures

The constraints in Section 7.5.1 make unaffordable to extensively evaluate different configurations of systems by listening to many synthesized performances. This could be solved if objective measures that correlate with perception were established. Such perceptually-motivated objective measures could be computed by learning the relationship between MOS and extracted features at a local or global scope. The measure should be ideally independent from the style and the singer, and it should provide ratings for particular features like timing, vibratos, tuning, voice quality, or the overall performance expression. These measures, besides helping to improve the systems' performance, would represent a standard for evaluation and allow for scalability.

The development of perceptually-motivated objective measures could benefit from approaches in the speech and audio processing fields. Psychoacoustic and cognitive models have been used to build objective metrics for assessing audio quality and speech intelligibility (Campbell et al., 2009) and its effectiveness has been measured by its correlation to MOS ratings. Interestingly, method specific measures have been computed in unit selection cost functions for speech synthesis (Chu et al., 2001). Other approaches for speech quality prediction are based on a log-likelihood measure as a distance between a syn-

thesized utterance and an HMM model built from features based on MFCCs and F0 of natural recordings (Möller et al., 2010). This gender-dependent measure is correlated to subjective ratings like naturalness. For male data, it can be improved by linearly combining it with parameters typically used in narrow-band telephony applications, like noise or robotization effects. For female data, it can be improved by linearly combining it with parameters related to signal like duration, formants, or pitch. The research on automatic evaluation of expressive performances is considered an area to exploit, although it is still not mature enough (Katayose et al., 2012), for example, it could be applied to develop better models and training tools for both systems and students.

Similarly to the speech and instrumental music performance communities, the progress in the singing voice community could be incentivized through evaluation campaigns. These types of evaluations help to identify the aspects that need to be improved and can be used to validate perceptually-motivated objective measures. Examples of past evaluation campaigns are the Synthesis Singing Challenge[5] and the Performance Rendering Contest[6] (Rencon) (Katayose et al., 2012). In the first competition, one of the target songs was compulsory and the same for each team. Performances were rated by 60 participants with a five-point scale involving quality of the voice source, quality of the articulation, expressive quality, and the overall judgment. The organizers concluded "the audience had a difficult task, since not all systems produced both a baritone and a soprano version, while the quality of the voices used could be quite different (weaker results for the female voice)"[5]. The Rencon's methodology is also interesting. Expressive performances are generated from the same Disklavier grand piano, so that the differences among approaches are only due to the performance and subjectively evaluated by an audience and experts. In 2004, voice synthesizers were also invited. Favorable reviews were received but not included in the ranking.

Correlation between cost functions and the evaluation ratings

Inspired by the work in Chu et al. (2001) for speech synthesis based on unit selection, we have done a similar experiment in order to see if we could find a relationship between the participants' mean rating value and the cumulated cost of the unit selection approach. If the participants' ratings could be clearly determined as a function of the cost values of the unit selection-based systems, it would be a possible way of predicting the average participant perception of the naturalness of expression of an audio excerpt in a scalable manner as explained in the beginning of this section.

The audio excerpts used in the perceptual evaluation were originally much longer, and shorter segments were selected in order to be able to do the eval-

[5] http://www.interspeech2007.org/Technical/synthesis_of_singing_challenge.php

[6] http://renconmusic.org/

Method	DBname	Song	# units	cost	norm. cost	rating
Unit selection	Song	But not for me	19	17.74	0.93	2.97
		Body and soul	25	26.14	1.05	3.03
		My funny valentine	6	10.91	1.82	3.06
		My funny valentine	7	18.13	2.59	3.06
		Summertime	8	7.42	0.93	3.44
	Systematic	But not for me	19	30.53	1.61	3.44
		Body and soul	25	40.1	1.60	3.22
		My funny valentine	6	12.83	2.14	3.16
		My funny valentine	7	21.02	3.00	3.16
		Summertime	8	10.47	1.31	3.34
Hybrid	Song	But not for me	19	82.10	4.32	2.93
		Body and soul	25	68.79	2.75	2.87
		My funny valentine	6	23.25	3.88	3.23
		My funny valentine	7	35.45	5.06	2.97
		Summertime	8	38.35	4.79	3.73
	Systematic	But not for me	19	89.16	4.69	3.30
		Body and soul	25	79.78	3.19	3.37
		My funny valentine	6	22.49	3.75	3.30
		My funny valentine	7	35.28	5.04	3.33
		Summertime	8	43.09	5.39	3.57

Table 7.10: Values used to find relationship between ratings and cumulated costs.

uation within a reasonable amount of time. Thus, the unit selection-based systems were run for the whole song scores. However, we cannot use the whole song cumulated cost values, since these refer to the complete song, but only a part of it was evaluated. Therefore, we have taken the cumulated cost of only the part that was finally evaluated by using only the cost increment between the first and last excerpt unit. The final value has been obtained by dividing the cumulated cost by the amount of units. This value has been computed per song and placed in the x axis.

The values involved in this computation are shown in Table 7.10. The first column indicates method, which can be the normal unit selection-based or the hybrid system. Then, the expression database used to extract the contours (Song or Systematic). Next, the song name the following figures are related to. These figures are the number of units, the cumulated cost, the normalized cost (*cost/units*), and finally the mean value of the participants' ratings.

We show this information in Fig. 7.19, with the cumulated cost value of the unit selection-based systems placed in the x axis, and the mean of the participants' ratings in the y axis. We want to approximate the 5 points for each method and DB combination. Polynomials of degree 1 and 2 have been used to approximate the dots for each group. Although we only have 5 points per method and DB combination, the groups of points seem to be more or less organized in their respective clouds and that can be approximated by the

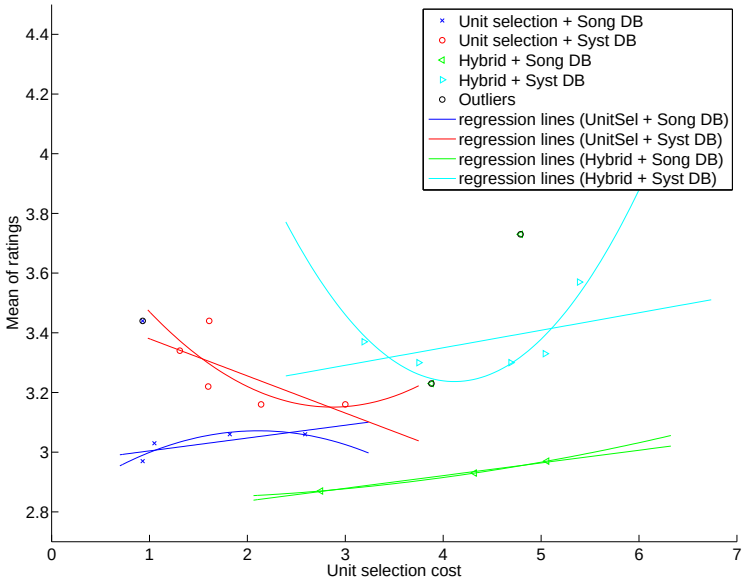


Figure 7.19: Participants mean ratings vs. unit selection normalized cost.

polynomials. We have dropped out some points which seemed to be "outliers". However, it remains unclear why the points of the unit selection method with the Systematic DB (red points) have a negative slope compared to the other combinations. Besides, we would probably need more points to extract more conclusions on the type of regression line (linear or polynomial) is appropriate as a perceptually-motivated measure.

7.6 Conclusion

In this chapter we have evaluated a set of systems that generate expression contours for pitch and dynamics with a perceptual evaluation and an efficiency comparison.

In section 7.2 we have done an online subjective evaluation with 32 participants, in which during less than 30 minutes they had to rate from 1 to 5 the perceived naturalness of expression of 6 audio excerpts in 11 randomly presented questions. The 6 audio excerpts per question have been generated by 6 methods: the Vocaloid baseline system, performance driven from the original recording, the proposed unit selection-based methods (hybrid and non-hybrid), and the proposed HMM-based methods.

In section 7.3, after analyzing the participants' demographics, we have shown that the differences that we observed in several boxplots are statistically significant. The ANOVA and Tukey tests show that the analysis of all participants and the consistent ones provide similar results. First, methods

have a significant effect on the ratings and that these are clustered into non homogeneous groups. On the one side, the HMM-based methods do not differ from the default Vocaloid method. On the other hand, the unit selection methods cluster together with the performance driven approach. Secondly, the databases seem not to have an effect on the perceived naturalness. However, if the interaction database::method is significant as we can see in the boxplots.

Next, in section 7.4 another use case in which the expression contours could be applied has also been analyzed. It consists on the transformation of a real singing voice recording in order to improve the naturalness of expression. The preliminary results show that the naturalness of expression is not degraded and even improved when the original recording does not contain specific expression resources like vibrato.

Finally, in section 7.5 we have discussed on a couple of topics related to the evaluation of the singing voice synthesis systems. We have explained that, in order to easily evaluate and compare several singing synthesis systems, the field would benefit from going towards a common evaluation framework. We have also highlighted the importance of adopting perceptually-motivated objective measures. Such measures would allow for comprehensive objective evaluations correlated to subjective measures.

CHAPTER 8

Conclusions

In this dissertation, we have addressed expression in singing voice and how it can be used to control singing voice synthesizers in order to achieve natural performances. From the wide variety of features that are related to the naturalness of expression, we have focused on the generation of pitch and dynamics expression contours by proposing 3 systems: a unit selection-based system, a statistical system based on Hidden Markov Models, and a hybrid system. In the conducted perceptual evaluation we have compared these methods to each other, to a performance-driven method and the synthesizer baseline performance.

8.1 Introduction

This chapter aims to summarize the contributions this dissertation makes to the field of singing voice synthesis (section 8.2). We highlight the contributions of each chapter from different perspectives: the discussion on the topic, the datasets, the methodologies, the proposed expression control systems, the evaluation and the proposal for its improvement, and the thesis impact through the publications.

Following the summary, we present areas of future work that have arisen through the course of the research. (section 8.3). Some of these ideas have not been explored before due to time limitations, and some others are proposed now thanks to the perspective and experience that these last years working on this topic provide to us. Thus, several improvements are proposed related to the expression databases, the proposed systems, and the evaluation. Furthermore, we discuss other use cases not explored in this thesis where expression control can be applied. Finally, we describe the challenges that we currently foresee in the field of singing voice synthesis (section 8.4).

8.2 Summary of contributions

Discussion, definitions, and analysis of expression

Expression is a complex term to define, and natural expression is a complex task in the music technology fields and, more concretely, when applied to the singing voice synthesis, as we have seen in Chapter 1. We have discussed several musical and psychological definitions that have been attached to the "expression" term, both from a general and a singing voice perspective. A voice excerpt has been analyzed to illustrate the topic of research.

An in-depth review of expression control in singing voice synthesis

As humans, we are completely familiarized with the singing voice instrument, and can easily detect whether synthesis results are similar to a real singer or not. A wide variety of contributing features make achieving a natural expression control a complex task. Hence, in Chapter 2 we have provided a summary of the state of the art background on the field. These involve an explanation of the singing voice production mechanism and how it may be emulated algorithmically with computers. We have also presented an in-depth description of the features related to the singing voice expression. We have classified, described, and compared several systems for expression control, covering performance driven, rule-based, and statistical-based approaches. The strategies to evaluate the naturalness achieved by these expression control systems have also been studied.

A compilation of sound excerpts from different works

To our knowledge, the sound examples from previous works on the same topic of research had not been gathered before, and thus there lacks a repository with sound excerpts. In the state of the art we have compiled several sound examples from the reviewed works for ease of comparison and made the compilation available online[1]. This is probably best summarized with the feedback provided by one of the anonymous reviewers of Umbert et al. (2015) who reported that "*Hearing is believing*", pointing out that accompanying research with sound excerpts helps to better understand the details of the topic being described.

A methodology for expression database creation

The singing voice databases (for jazz style) that we have used for expression control have some specific requirements. In Chapter 3 we have defined a methodology for their design, recording, and labeling. The Systematic expression database covers a set of note figures, note pitches, and note strengths combinations. The methodology to obtain the melodic exercises based on the

[1] http://www.mtg.upf.edu/publications/ExpressionControlinSingingVoiceSynthesis

Viterbi algorithm has been described. The Song expression database is easier to create since it is a compilation of jazz standard songs.

Microprosody effects in the extracted features from the recordings has also been considered. Both expression databases have been recorded with interleaved vowels at every note instead of lyrics to remove microprosody due to phonetics. Regarding the Systematic database, since the corresponding melodies have no lyrics, it is an appropriate decision to record vowels instead.

Concerning the database labeling, we have proposed to extract note onsets and durations in a semiautomatic procedure based on GMM. Note transitions are also automatically estimated and manually corrected. We have also proposed an iterative procedure for vibrato features estimation to generate the corresponding depth, rate, and baseline pitch contours.

A unit selection-based system for expression control

In Chapter 4 we have introduced a novel unit selection-based system for expression control. Typically, unit selection approaches are used as synthesizers, thus considering timbre information as well. In contrast, in the proposed system the output consists of pitch and dynamics contours used to control a synthesizer. For this system, the strategies for unit selection (cost functions), unit transformation and concatenation (pitch interval modifications, time-scaling, and crossfading masks), and contour generation (pitch tuning and vibrato generation) have been described. The proposed system is able to generate expression control contours with fine details similar to the expression recordings.

A statistical-based system for expression control

In Chapter 5 we have proposed two HMM-based systems. The first one models note sequences using absolute pitch and dynamics as observations. The second system models sequences of sustains and transitions, where pitch observations correspond to the difference between the pitch and the estimated nominal pitch from the score. Within this statistical system we have also proposed the prediction of the note transitions using random forests.

A hybrid system for expression control

In Chapter 6 we have proposed a system that combines the positive aspects from the unit selection and the HMM-based system. The hybrid system extends the unit selection cost function by adding a reference pitch contour. In our case, the reference pitch contour is generated by the modified HMM-based system which handles richer contextual data than the unit selection system. The cost function is a distance measure between pitch contours based on the Dynamics Time Warping cost.

A comparison with state of the art systems for expression control

The perceptual evaluation carried out has raised three key points. The first one is that singing voice expression can be generated artificially. Secondly, the best rated systems regarding naturalness of expression (concerning pitch and dynamics) are hybrid and performance driven ones, with no significant statistical difference between them. Finally, all proposed methods are rated equally or better than the default expression control found in the Vocaloid singing voice synthesizer.

Another use case for expression control

We have also shown another use case in which a singing voice recording is transformed to change its pitch contour. The preliminary results show that the naturalness of expression is not degraded and even improved when the original recording does not contain specific expression resources like vibrato.

Proposals for evaluation improvement

In Chapter 7 we have contributed to debate on the problems that make the comparisons between systems a difficult task. We consider that the research in this field would benefit from building a common evaluation framework. We also identify weaknesses in the current evaluation of singing voice performances. The lack of perceptually motivated objective measures prevents from evaluating singing voice synthesis systems in a scalable way. Furthermore, we have studied whether the cost of the unit selection systems is related to the perceptual evaluation ratings.

Impact

With regards to the publications, in Appendix D we have summarized the published work during this thesis as well as contributions to workshops. The publication with most impact is Umbert et al. (2015), which is the core part of this thesis state of the art (Chapter 2). Moreover, we plan to make publicly available the systematic and song expression databases used in this thesis.

8.3 Future perspectives

In this section we outline several future research directions that this thesis could follow. These are mainly related to the expression databases, improvements on the proposed systems, and the evaluation.

Expression database

The methodology for creating expression databases can be improved in several ways. The current Systematic database covers combinations of pitch intervals, note figures, and note strength. However, we designed it for a single tempo. Although the recording and labeling of singing voice databases are time consuming tasks, replicating the same systematic score at different tempos would ensure having more variety in the coverage, and therefore units may benefit from less transformation given the tempo is included in the subcost functions.

Voice quality could also be considered in the expression databases. Expression databases could be recorded with different voice qualities (for instance with modal and growl voices). Voice quality feature contours (presence of subharmonics, or noise level) could also be extracted and be used together with the pitch and dynamics expression contours.

Finally, the current labeling process is semi-automatic. However, some steps could be automated, like the detection of the first/last peak/valley of vibratos, note onsets and durations, or note transition start and end times.

Unit selection-based systems

We have identified at least two aspects in which the unit selection-based systems could be improved. First, the expression contours could be represented with a parametric model (for instance, by Bézier curves). This would allow clustering contour shapes, providing a better understanding of singer and style particularities. On the other hand, the unit selection cost in the hybrid system could be improved in several directions. First, other functions than DTW could be tested as distance measures. Secondly, dynamics added to the DTW cost function. Finally, a distance measure considering transition and sustain segmentation of the HMM-based system and the source units could be included.

HMM-based systems

The statistical systems could be improved by adding more context-dependent labels. For instance, features regarding the presence of a vibrato in the previous, current, or succeeding notes. The presence of vibratos could be directly indicated by some value related to the depth and rate.

Regarding the transition and sustain prediction, we highlighted that there might be over-fitting. This issue should be further studied in future research works.

Recurrent neural networks systems

Other systems could be used to model pitch and dynamics. In the last master thesis I co-supervised, pitch was modeled using long-short term memory (LSTM) recurrent neural networks. The contextual data was very similar to

the contextual data of the HMM-based systems. In this work we compared the synthesized excerpts to the unit selection-based system and the HMM-based system. Although no statistical differences were observed with the proposed implementation, this method may provide state of the art results with some improvements. First, it could be studied which other contextual data could be added. More importantly, the distribution of the output values could also be modeled with gaussian distributions parameters. This has proven successfully for instance to model handwriting in Graves (2013).

Concerning the evaluation, expression databases related to other singing styles could be used. Then, further perceptual tests would help to evaluate the impact of the database on the target song depending on the target style. For instance, is it perceived as more naturally expressive a song when the database is from the same style as the target song? Up to which point is it important?

Application to other use cases

As we have already mentioned, singing voice expression is related not only to pitch and dynamics but also to the voice quality or timing. An environment to evaluate this labeled information on the expression database on recorded singing voice would be a way to avoid the imperfections of the synthesis itself. The use case experiment on the expression contours of a singing voice performance goes in this direction.

Expression control could also be applied in online repositories of scores or score editors. As we have introduced in Chapter 1, singing voice synthesis with natural expression would improve significantly the current status of these applications, in which scores with vocal tracks are simply rendered with another instrument or a single vowel.

Finally, the methodology we have described in this thesis could be adapted to model expression for other instruments. In some cases it may be easier to adapt than others. For instance, it may be easier for wind instruments which are monophonic than for polyphonic instruments like the piano.

8.4 Challenges

While expression control has advanced in recent years, there are still many open challenges. First, we discuss some specific challenges and consider the advantages of hybrid approaches. Next, we discuss important challenges in approaching a more human-like naturalness in the synthesis. This section is based on Umbert et al. (2015).

Towards hybrid approaches

Several challenges have been identified in the described approaches. Only one of the performance-driven approaches deals with timbre, and it depends on the

available voice quality databases. This approach would benefit from techniques for the analysis of the target voice quality, its evolution over time, and techniques for voice quality transformations so to be able to synthesize several voice qualities. The same analysis and transformation techniques would be useful for the unit selection approaches. Rule-based approaches would benefit from machine learning techniques that learn rules from singing voice recordings in order to characterize a particular singer and to explore how these are combined. Statistical modeling approaches are currently not dealing with comprehensive databases that cover a broad range of styles, emotions, and voice qualities. If we could take databases that efficiently cover different characteristics of a singer it would lead to interesting results using model interpolation.

We consider the combination of existing approaches to have great potential. Rule-based techniques could be used as a pre-preprocessing step to modify the nominal target score so that it contains variations such as ornamentations and timing changes related to the target style or emotion. The resulting score could be used as the target score for statistical and unit selection approaches, or a combination of both, where the expression parameters would be generated.

Towards human-like singing synthesis

One of the ultimate goals of singing synthesis technologies is to synthesize human-like singing voices that cannot be distinguished from human singing voices. Although the naturalness of synthesized singing voices has been increasing, perfect human-like naturalness has not yet been achieved. Singing synthesis technologies will require more dynamic, complex, and expressive changes in pitch, loudness, and timbre. For example, voice quality modifications could be related to emotions, style, or lyrics.

Moreover, automatic context-dependent control of those changes will also be another important challenge. The current technologies synthesize words in the lyrics without knowing their meanings. In the future, the meanings of the lyrics could be reflected in singing expressions as human singers do. Human-like singing synthesis and realistic expression control may be a highly challenging goal, given how complex this has been proven for speech.

In Umbert et al. (2015) we mention other aspects that could be improved, like interfaces for singing synthesis which avoid time-consuming manual adjustments and that work in real-time. Besides, multimodality is also discussed with respect to the other aspects that surround a virtual singer like its as voice, face, and body. The simultaneous generation of some of these singer attributes (voice and face) has also started to be tackled in some projects like VocaWatcher (Goto et al., 2012).

Martí Umbert, Barcelona, Tuesday 6$^{\text{th}}$ October, 2015.

Bibliography

Alonso, M. (2004). *Model dâĂŹExpressivitat Emocional per a un Sintetitzador de Veu Cantada*. Ph.D. thesis, Universitat Pompeu Fabra.

Arcos, J. L., de Mántaras, R. L., & Serra, X. (1998). Saxex: A case-based reasoning system for generating expressive musical performances. *Journal of New Music Research*, *27*(3), 194–210.

Bonada, J. (2008). *Voice Processing and synthesis by performance sampling and spectral models*. Ph.D. thesis, University Pompey Fabra.

Bonada, J. & Serra, X. (2007). Synthesis of the Singing Voice by Performance Sampling and Spectral Models. *IEEE Signal Processing Magazine*, *24*(2), 67–79.

Bresin, R. & Friberg, A. (2000). Emotional Coloring of Computer-Controlled Music Performances. *Computer Music Journal*, *24*(4), 44–63.

Campbell, D., Jones, E., & Glavin, M. (2009). Audio quality assessment techniques - A review, and recent developments. *Signal Processing*, *89*(8), 1489–1500.

Canazza, S., De Poli, G., Drioli, C., Rodà, A., & Vidolin, A. (2004). Modeling and control of expressiveness in music performance. *Proceedings of the IEEE*, *92*(4), 686–701.

Chu, M., Chu, M., Peng, H., & Peng, H. (2001). An objective measure for estimating MOS of synthesized speech. In *Proc. 7th European Conf. on Speech Communication and Technology (Eurospeech)*, pp. 2087–2090. Aalborg.

Cook, P. R. (1998). Toward the perfect audio morph? singing voice synthesis and processing. *Proceedings of the 1st. International Conference on Digital Audio Effects (DAFX), Barcelona*.

Cuthbert, M. S. & Ariza, C. (2010). Music21 A Toolkit for Computer-Aided Musicology and Symbolic Music Data. *11th International Society for Music Information Retrieval Conference(ISMIR 2010)*, (Ismir), 637–642.

Doi, H., Toda, T., Nakano, T., Goto, M., & Nakamura, S. (2012). Singing voice conversion method based on many-to-many eigenvoice conversion and training data generation using a singing-to-singing synthesis system. *Signal Information Processing Association Annual Summit and Conference (APSIPA ASC), 2012 Asia-Pacific*, pp. 1–6.

Floría, H. (2013). Expressive speech synthesis for a Radio DJ using Vocaloid and HMM's.

Friberg, A., Bresin, R., & Sundberg, J. (2009). Overview of the KTH rule system for musical performance. *Advances in Cognitive Psychology*, *2*(2), 145–161.

Gabrielsson, A. & Juslin, P. N. (1996). Emotional Expression in Music Performance: Between the Performer's Intention and the Listener's Experience. *Psychology of Music*, *24*(1), 68–91.

Gómez, E. & Bonada, J. (2013). Towards computer-assisted flamenco transcription: An experimental comparison of automatic transcription algorithms as applied to a cappella singing. *Computer Music Journal*, *37*(2), 73–90.

Goto, M. (2012). Grand Challenges in Music Information Research. *Multimodal Music Processing*, *3*, 217–226.

Goto, M., Nakano, T., Kajita, S., & Matsusaka, Y. (2012). VocaListener and VocaWatcher: Imitating a Human Singer by Using Signal Processing. In *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 5393–5396. Kyoto.

Graves, A. (2013). Generating Sequences With Recurrent Neural Networks.

Ilie, G. & Thompson, W. (2006). A comparison of acoustic cues in music and speech for three dimensions of affect. *Music Perception*, *23*(4), 319–330.

Iserte Agut, J. (2014). Síntesis de voz cantada y canto coral: criterios musicales y estadísticos.

Janer, J., Bonada, J., & Blaauw, M. (2006). Performance-driven control for sample-based singing voice synthesis. In *Digital Audio Effects (DAFx)*, pp. 41–44. Montreal, Canada.

Juslin, P. N. (2003). Five Facets of Musical Expression: A Psychologist's Perspective on Music Performance. *Psychology of Music*, *31*(3), 273–302.

Juslin, P. N. & Laukka, P. (2003). Communication of emotions in vocal expression and music performance: different channels, same code? *Psychological bulletin*, *129*(5), 770–814.

Justel Pizarro, L. M. (2014). Síntesis de voz cantada y canto coral: "Herramienta de ensayo para integrantes de coros clásicos".

Katayose, H., Hashida, M., De Poli, G., & Hirata, K. (2012). On Evaluating Systems for Generating Expressive Music Performance: the Rencon Experience. *Journal of New Music Research*, *41*(4), 299–310.

Kawahara, H., Nisimura, R., Irino, T., Morise, M., Takahashi, T., & Banno, H. (2009). Temporally variable multi-aspect auditory morphing enabling extrapolation without objective and perceptual breakdown. *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings*, (2), 3905–3908.

Kenmochi, H. & Ohshita, H. (2007). VOCALOID-commercial singing synthesizer based on sample concatenation. In *Interspeech*, August, pp. 4009–4010. Antwerpen.

Kirke, Alexis, Miranda, E. R. (2013). An Overview of Computer Systems for Expressive Music Performance. In E. R. Kirke, Alexis, Miranda (Ed.) *Guide to Computing for Expressive Music Performance*, chap. 1, pp. 1–47. Springer.

Kob, M. (2002). *Physical modeling of the singing voice*. Ph.D. thesis, RWTH Aachen.

Kob, M. (2003). Singing voice modeling as we know it today. In *Stockholm Music Acoustics Conference (SMAC'03)*, vol. 90, pp. 431–434. Stockholm.

Lesaffre, M. (2006). Music information retrieval: conceptuel framework, annotation and user behaviour.

Lindemann, E. (2007). Music synthesis with reconstructive phrase modeling. *IEEE Signal Processing Magazine*, *24*(2), 80–91.

Loscos, A. & Bonada, J. (2004). Emulating rough and growl voice in spectral domain. In *Proceedings of the 7th International Conference on Digital Audio Effects (DAFx'04)*, pp. 49–52. Naples.

Maestre, E. (2009). Modeling Instrumental Gestures : An Analysis / Synthesis Framework for Violin Bowing. *Tesisenxarxanet*.

Mantaras, R. D. & Arcos, J. (2002). AI and music: From composition to expressive performance. *AI magazine*, *23*(3), 43–58.

Marinescu, M.-C. & Ramirez, R. (2011). A Machine Learning Approach to Expression Modeling for the Singing Voice. *International Conference on Computer and Computer Intelligence (ICCCI)*, *31*(12), 311–316.

Mayor, O., Bonada, J., & Janer, J. (2009). KaleiVoiceCope: Voice Transformation from Interactive Installations to Video-Games. *AES 35th International Conference: Audio for Games*.

Meron, Y. (1999). *High Quality Singing Synthesis using the Selection-based Synthesis Scheme*. Ph.D. thesis, University of Tokyo.

Mion, L., Poli, G. D., & Rapanà, E. (2010). Perceptual organization of affective and sensorial expressive intentions in music performance. *ACM Transactions on Applied Perception*, *7*(2), 1–21.

Möller, S., Hinterleitner, F., Falk, T. H., & Polzehl, T. (2010). Comparison of Approaches for Instrumentally Predicting the Quality of Text-To-Speech Systems. In *Proc. Interspeech*, September, pp. 1325–1328. Makuhari, Japan.

Nakano, T. & Goto, M. (2009). Vocalistener: a Singing-To-Singing Synthesis System Based on Iterative Parameter Estimation. In *Proceedings of the 6th Sound and Music Computing Conference (SMC)*, July, pp. 343–348. Porto.

Nakano, T. & Goto, M. (2011). Vocalistener2: A singing synthesis system able to mimic a user's singing in terms of voice timbre changes as well as pitch and dynamics. In *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings*, pp. 453–456. Prague.

Narmour, E. (1990). *The analysis and cognition of basic melodic structures: The implication-realization model.* Chicago: Chicago, IL, US: University of Chicago Press.

Narmour, E. (1992). *The Analysis and Cognition of Melodic Complexity: The Implication-Realization Model.* Chicago: Chicago, IL, US: University of Chicago Press.

Obin, N. (2011). *MeLos : Analysis and Modelling of Speech Prosody and Speaking Style.* Ph.D. thesis, Université Pierre et Marie Curie-Paris VI.

Oura, K. & Mase, A. (2010). Recent development of the HMM-based singing voice synthesis system - Sinsy. In *Proc. Int. Speech Communication Association (ISCA), 7th Speech Synthesis Workshop (SSW7)*, pp. 211–216. Tokyo.

Plack, C. J. & Oxenham, A. J. (2005). Overview: The Present and Future of Pitch. In C. J. Plack, R. R. Fay, A. J. Oxenham, & A. N. Popper (Eds.) *Pitch*, *Springer Handbook of Auditory Research*, vol. 24, chap. 1, pp. 1–6. New York, NY: Springer New York.

Posner, J., Russell, J. a., & Peterson, B. S. (2005). The circumplex model of affect: an integrative approach to affective neuroscience, cognitive development, and psychopathology. *Development and psychopathology*, *17*(3), 715–734.

Rodet, X. (2002). Synthesis and processing of the singing voice. *Proc.1st IEEE Benelux Workshop on Model based Processing and Coding of Audio (MPCA-2002)*, pp. 99–108.

Russell, J. (1980). A circumplex model of affect. *Journal of personality and social psychology*, *39*(6), 1161–1178.

Saino, K., Tachibana, M., & Kenmochi, H. (2010). A Singing Style Modeling System for Singing Voice Synthesizers. In *Training*, September, pp. 2894–2897. Makuhari, Japan.

Saino, K., Zen, H., Nankaku, Y., Lee, A., & Tokuda, K. (2006). An HMM-based Singing Voice Synthesis System. In *Interspeech2006*, pp. 1141–1144. Pittsburgh, USA.

Saitou, T., Goto, M., Unoki, M., & Akagi, M. (2007). Speech-to-singing synthesis: Converting speaking voices to singing voices by controlling acoustic features unique to singing voices. In *IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*, pp. 215–218. New Paltz, NY.

Salamon, J., Gómez, E., Ellis, D. P., & Richard, G. (2014). Melody Extraction from Polyphonic Music Signals: Approaches, Applications and Challenges. *IEEE Signal Processing Magazine*, *31*(2), 118–134.

Scheirer, E. D. (1998). Tempo and beat analysis of acoustic musical signals. *The Journal of the Acoustical Society of America*, *103*(1), 588–601.

Schröder, M. (2001). Emotional speech synthesis: A review. In *Proceedings of Eurospeech*, vol. 1, pp. 561–564. Aalborg.

Schröder, M. (2009). Expressive Speech Synthesis: Past, Present, and Possible Futures. In J. Tao & T. Tan (Eds.) *Affective Information Processing*, chap. 7. London: Springer London.

Schwarz, D. (2007). Corpus-based concatenative synthesis. *IEEE Signal Processing Magazine*, *24*(2), 92–104.

Sundberg, J. (1981). Synthesis of singing. In *Musica e tecnologia: Industria e cultura per lo sviluppo del Mezzogiorno*, pp. 145–162. Venice: Quaderni di Musica/Realta (Italian Edition).

Sundberg, J. (1987). The Science of the Singing Voice.

Sundberg, J. (2006). The KTH synthesis of singing. *Advances in Cognitive Psychology*, *2*(2), 131–143.

Sundberg, J. & Bauer-Huppmann, J. (2007). When Does a Sung Tone Start? *Journal of Voice*, *21*(3), 285–293.

Tachibana, M., Yamagishi, J., Masuko, T., & Kobayashi, T. (2005). Speech Synthesis with Various Emotional Expressions and Speaking Styles by Style Interpolation and Morphing.

Ternström, S. (2002). Session on naturalness in synthesized speech and music.

Thalén, M. & Sundberg, J. (2001). Describing different styles of singing: a comparison of a female singer's voice source in "Classical", "Pop", "Jazz" and "Blues". *Logopedics, phoniatrics, vocology*, *26*(2), 82–93.

Umbert, M., Bonada, J., & Blaauw, M. (2013a). Generating singing voice expression contours based on unit selection. In *Stockholm Music Acoustics Conference (SMAC)*, pp. 315–320. Stockholm.

Umbert, M., Bonada, J., & Blaauw, M. (2013b). Systematic database creation for expressive singing voice synthesis control. In *Proc. Int. Speech Communication Association (ISCA), 8th Speech Synthesis Workshop (SSW8)*, pp. 213–216. Barcelona.

Umbert, M., Bonada, J., Goto, M., Nakano, T., & Sundberg, J. (2015). Expression Control in Singing Voice Synthesis: Features, Approaches, Evaluation, and Challenges. *IEEE Signal Processing Magazine*, *32*(6), 55–73.

Umbert, M., Bonada, J., & Janer, J. (2010). *Emotional speech synthesis for a Radio DJ: corpus design and expression modeling.* Master's thesis, Univeristat Pompeu Fabra.

Widmer, G. (2001). Using AI and machine learning to study expressive music performance : project survey and first report. *AI Communications*, *14*, 149–162.

Widmer, G. & Goebl, W. (2004). Computational Models of Expressive Music Performance: The State of the Art. *Journal of New Music Research*, *33*(3), 203–216.

Yoshimura, T., Tokuda, K., Masuko, T., Kobayashi, T., & Kitamura, T. (1999). Simultaneous modeling of spectrum, pitch and duration in HMM-based speech synthesis. In *Proc. 6th European Conf. on Speech Communication and Technology (Eurospeech)*, pp. 2347–2350. Budapest.

Appendix A: Context-dependent labels

An example of context-dependent label format for HMM-based singing voice synthesis in Japanese

HTS Working Group

December 25, 2012

p_1^p_2-p_3+p_4=p_5-p_6%p_7^p_8
/A:a_1-a_2-a_3@a_4 /B:b_1-b_2-b_3@b_4 /C:c_1+c_2+c_3@c_4
/D:d_1!d_2#$d_3$$$d_4$%$d_5$|$d_6$&$d_7$[$d_8$-$d_9$
/E:e_1]e_2^e_3=e_4~e_5!e_6@e_7#e_8+e_9]$e_{10}$$$e_{11}$|$e_{12}$[$e_{13}$&$e_{14}$]$e_{15}$=$e_{16}$^$e_{17}$~$e_{18}$#$e_{19}$@$e_{20}$!$e_{21}$$$e_{22}$&$e_{23}$%$e_{24}$[$e_{25}$|$e_{26}$]$e_{27}$-$e_{28}$^$e_{29}$+
e_{30}~e_{31}=e_{32}@$e_{33}$$$e_{34}$!$e_{35}$%$e_{36}$#$e_{37}$|$e_{38}$|$e_{39}$-$e_{40}$&$e_{41}$&$e_{42}$+$e_{43}$[$e_{44}$;$e_{45}$]$e_{46}$;$e_{47}$~$e_{48}$~$e_{49}$^$e_{50}$^$e_{51}$@$e_{52}$;$e_{53}$=$e_{54}$=$e_{55}$!$e_{56}$~$e_{57}$+$e_{58}$
/F:f_1#f_2#f_3-$f_4$$$f_5$$$f_6$+$f_7$%$f_8$;$f_9$
/G:g_1_g_2 /H:h_1_h_2 /I:i_1_i_2
/J:j_1~j_2@j_3

p_1	the phoneme identity before the previous phoneme
p_2	the previous phoneme identity
p_3	the current phoneme identity
p_4	the next phoneme identity
p_5	the phoneme idendity after the next phoneme
p_6	falsetto flag
p_7	training flag
p_8	pitch-shift
a_1	the number of phonemes in the previous syllable/mora
a_2	position of the previous syllable/mora identity in the note (forward)
a_3	position of the previous syllable/mora identity in the note (backward)
a_4	the language dependent context of the previous syllable/mora
b_1	the number of phonemes in the current syllable/mora
b_2	position of the current syllable/mora identity in the note (forward)
b_3	position of the current syllable/mora identity in the note (backward)
b_4	the language dependent context of the current syllable/mora
c_1	the number of phonemes in the next syllable/mora
c_2	position of the next syllable/mora identity in the note (forward)
c_3	position of the next syllable/mora identity in the note (backward)
c_4	the language dependent context of the next syllable/mora
d_1	the absolute pitch of the previous note (C0-G9)
d_2	the relative pitch of the previous note (0-11)
d_3	the key of the previous note (the number of sharp)
d_4	the beat of the previous note
d_5	the tempo of the previous note (SS: 1-75 SM: 76-90 SH: 91-105 MS: 106-120 MM: 121-135 MH 136-150 HS: 151-165 HM: 166-180 HH: 181-)
d_6	the length of the previous note by the syllable/mora
d_7	the length of the previous note by 0.1 second (1-99)
d_8	the length of the previous note by three thirty-second note (1-199)
d_9	breath mark of the previous note
e_1	the absolute pitch of the current note (C0-G9)
e_2	the relative pitch of the current note (0-11)
e_3	the key of the current note (the number of sharp)
e_4	the beat of the current note
e_5	the tempo of the current note (SS: 1-75 SM: 76-90 SH: 91-105 MS: 106-120 MM: 121-135 MH 136-150 HS: 151-165 HM: 166-180 HH: 181-)
e_6	the length of the current note by the syllable/mora
e_7	the length of the current note by 0.1 second (1-99)
e_8	the length of the current note by three thirty-second note (1-199)
e_9	breath mark of the current note
e_{10}	position of the current note identity in the current measure by the note (forward, 1-49)
e_{11}	position of the current note identity in the current measure by the note (backword, 1-49)
e_{12}	position of the current note identity in the current measure by 0.1 second (forward, 1-49)
e_{13}	position of the current note identity in the current measure by 0.1 second (backward, 1-49)
e_{14}	position of the current note identity in the current measure by three thirty-second note (forward, 1-99)
e_{15}	position of the current note identity in the current measure by three thirty-second note (backward, 1-99)
e_{16}	position of the current note identity in the current measure by % (forward)
e_{17}	position of the current note identity in the current measure by % (backward)

e_{18}	position of the current note identity in the current phrase by the note (forward, 1-99)
e_{19}	position of the current note identity in the current phrase by the note (backward, 1-99)
e_{20}	position of the current note identity in the current phrase by 0.1 second (forward, 1-199)
e_{21}	position of the current note identity in the current phrase by 0.1 second (backward, 1-199)
e_{22}	position of the current note identity in the current phrase by three thirty-second note (forward, 1-499)
e_{23}	position of the current note identity in the current phrase by three thirty-second note (backward, 1-499)
e_{24}	position of the current note identity in the current phrase by % (forward)
e_{25}	position of the current note identity in the current phrase by % (backward)
e_{26}	whether tied (slur) or not in between the current note and the previous note (0: not tied, 1: tied)
e_{27}	whether tied (slur) or not in between the current note and the previous note (0: not tied, 1: tied)
e_{28}	dynamic mark of the current note
e_{29}	the distance between the current note and the next accent by the note (1-9)
e_{30}	the distance between the current note and the previous accent by the note (1-9)
e_{31}	the distance between the current note and the next accent by 0.1 second (1-99)
e_{32}	the distance between the current note and the previous accent by 0.1 second (1-99)
e_{33}	the distance between the current note and the next accent by three thirty-second note (1-99)
e_{34}	the distance between the current note and the previous accent by three thirty-second note (1-99)
e_{35}	the distance between the current note and the next staccato by the note (1-9)
e_{36}	the distance between the current note and the previous staccato by the note (1-9)
e_{37}	the distance between the current note and the next staccato by 0.1 second (1-99)
e_{38}	the distance between the current note and the previous staccato by 0.1 second (1-99)
e_{39}	the distance between the current note and the next staccato by three thirty-second note (1-99)
e_{40}	the distance between the current note and the previous staccato by three thirty-second note (1-99)
e_{41}	position of the current note in the current crescendo by the note (forward, 1-49)
e_{42}	position of the current note in the current crescendo by the note (backward, 1-49)
e_{43}	position of the current note in the current crescendo by 1.0 second (forward, 1-99)
e_{44}	position of the current note in the current crescendo by 1.0 second (backward, 1-99)
e_{45}	position of the current note in the current crescendo by three thirty-second note (forward, 1-499)
e_{46}	position of the current note in the current crescendo by three thirty-second note (backward, 1-499)
e_{47}	position of the current note in the current crescendo by % (forward)
e_{48}	position of the current note in the current crescendo by % (backward)
e_{49}	position of the current note in the current decrescendo by the note (forward, 1-49)
e_{50}	position of the current note in the current decrescendo by the note (backward, 1-49)
e_{51}	position of the current note in the current decrescendo by 1.0 second (forward, 1-99)
e_{52}	position of the current note in the current decrescendo by 1.0 second (backward, 1-99)
e_{53}	position of the current note in the current decrescendo by three thirty-second note (forward, 1-499)
e_{54}	position of the current note in the current decrescendo by three thirty-second note (backward, 1-499)
e_{55}	position of the current note in the current decrescendo by % (forward)
e_{56}	position of the current note in the current decrescendo by % (backward)
e_{57}	pitch difference between the current and previous notes
e_{58}	pitch difference between the current and next notes
f_1	the absolute pitch of the next note (C0-G9)
f_2	the relative pitch of the next note (0-11)
f_3	the key of the next note (the number of sharp)
f_4	the beat of the next note
f_5	the tempo of the next note (SS: 1-75 SM: 76-90 SH: 91-105 MS: 106-120 MM: 121-135 MH 136-150 HS: 151-165 HM: 166-180 HH: 181-)
f_6	the length of the next note by the syllable/mora
f_7	the length of the next note by 0.1 second (1-99)
f_8	the length of the next note by three thirty-second note (1-199)
f_9	breath mark of the next note
g_1	the number of syllables/moras in the previous phrase (1-99)
g_2	the number of phonemes in the previous phrase (1-99)
h_1	the number of syllables/moras in the current phrase (1-99)
h_2	the number of phonemes in the current phrase (1-99)
i_1	the number of syllables/moras in the next phrase (1-99)
i_2	the number of phonemes in the next phrase (1-99)
j_1	the number of syllables/moras in this song / the number of measures in this song (1-99)
j_2	the number of phonemes in this song / the number of measures in this song (1-99)
j_3	the number of phrases in this song (1-99)

Appendix B: Perceptual evaluation instructions

The perceptual evaluation despcribed on Chapter 7 was presented as a two steps task. In the first task, we provided the necessary information to do the task with the basic instructions (in bold font the relevant ones) in the following form:

1. First, listen to all audio files for a given question in order to have a general idea.
2. Then, compare them and rate the perceived naturalness in the expression of the singing voice in each file.
3. You can focus your attention mainly on **pitch or melody** (for instance, note articulations, vibratos, etc) and **dynamics evolution over time** (is the energy always the same or are there fluctuations that make sense depending on the part of the song).
4. You should **NOT** focus on other aspects like timing, the timbre of the voice, and to how similar to a real singing voice the excerpts are.
5. You will listen to 6 files (10 seconds each) for each of the 11 questions. The test will take less than 30 mins.
6. Use **headphones** to better appreciate the differences amongst the audio files.
7. You can listen to the sounds as many times as you want.
8. You are allowed to review your ratings at any time (until you hit "Next")
9. You are allowed to rate different audio files equally.
10. If possible, **try to use the whole range of ratings from 1-5**.

After the instructions were presented to the participants, an example of 6 audio excerpts were presented so that the participant could hear them and start familiarizing with the task. No ratings were asked at this point. Next, a set of demographic information was asked so that we could have a profile of the participants. The information we asked (and the possible values) was:

- Age (15-24, 25-34, 35-44, 45-54, 55-64, more than 65)

- Gender (male, female)
- How often do you attentively listen to music? (Very rarely, About several times a month, About once a week, Nearly every day, Every day)
- Do you sing (e.g. in a choir, in a band, etc)? (Yes, No)
- If you play an instrument, how many years have you been playing it? (0, 1-2, 3-4, 5-6, 7-8, more than 8)
- Are you familiar with speech/singing voice synthesis or music technology? (Yes, No)

The second part of the evaluation was to answer the 11 questions. The instructions introduced in the first part of the evaluation were reminded to the participants.

Appendix C: Participants' feedback

This appendix aims to comment on the feedback provided by some participants on the perceptual evaluation described on Chapter 7. Only a few of them commented on the task since this part was optional. We have summarized their comments here below grouped according to the aspect it is related to (difficulty of the test, the instructions, and the organization of the audio excerpts). We also provide our comment within each topic.

Difficulty of the test

Participants' comments:

1. *It might be difficult test to do for people not used to listen to synthetic singing to appreciate differences.*

2. *As well I find the slow sample "you'll make me smile with my heart" very hard to distinguish between dynamics.*

3. *As well note that some samples sound on average unnatural (the japanese voice) and other on average very natural ("with daddy and mommy"). I think it is not biasing results, just noticed.*

4. *Your synthesis of the male voice is really good! I first thought you just sang it directly. But than I was mentioned that these are synthesized versions.*

This test might be difficult for people not used to listen to synthetic voices. Fortunately, most people was related to the field according to the demographics information (see Fig. 7.4). The fact that in some excerpt the expression features are more difficult to distinguish is normal, it depends especially on the melody. Besides, we are used to listen to all features as a whole instead of separating dynamics from pitch. Finally, it is good that subjects noticed differences amongst songs, and that some others reported a positive feedback on the excerpts quality.

Instructions

Participants' comments:

1. *There are no guidelines about what to look at. Maybe this is intentionally, but for people not used to synthetic singing, they might be lost on how to evaluate "expression".*

2. *In several examples, there are timbre abrupt changes that affect the overall sensation. For example: look at the naturalness of this vibrato in example X; or look at the note transition <my>-<heart> in example Y, etc.*

3. *I think the concept of singing badly and unnatural synthesis is quite different. But since some of the singing components (vibrato, depth and rate) are parameters for singing synthesis, it becomes little fuzzy. In the sense that several times you feel that the synthesis is good and this is precisely how a person would make a mistake (like a person singing). So I am not sure if these two things were same for the evaluation but maybe this distinction or at least a comment on this aspect should be made in the instructions.*

In the instructions we provided the guidelines on what should participants focus when rating the songs (pitch and dyanmics in point 3), as described in the "Website online" description. However, this may have not been read by the participant, or maybe not clear enough in the instructions. In the same section we explain that in the description of the task we point out not to focus on the timbre changes (point 4). Besides, we are not looking to get ratings on a particular vibrato for instance, but on the overall performance of the excerpts. Finally, it is true that the quality of singing and the naturalness of the expression are different concepts. A bad singer is natural although the expression is natural since it comes from a real human voice. We asked to rate the naturalness of the expression, so maybe it should have been made clear that this concept involves all kind of singer qualities, whether these may be good or bad.

Organization of the audio excerpts

1. *I find it not optimal that the same audio sample comes twice in the 11 questions.*

2. *I realize it is not the same output because it has different parameters in the 6 version and different in the next 6 versions.*

3. *I feel that the ears are tired of the same sounds and makes then less succinct to the subtle differences.*

4. *I feel that the ears remember the configuration of the first listen and tend to compare the second same to the first one.*

5. *At least try to make them not repeat immediately after each other.*

The fact that audio excerpts come twice is because we are using two expression databases. The fact that this participant feels tired indicates that the test may have been a little bit to long in this case and that for some participants it may have been better to do it 5-10 minutes shorter and to have less song repetitions with the two databases. Songs were presented randomly and in some participant it may have been the case that the same song appeared twice in a row.

Appendix D: Publications by the author

Submitted

Umbert, M., Bonada, J., Goto, M., Nakano, T., & Sundberg, J. (Nov. 2015). Expression Control in Singing Voice Synthesis: Features, Approaches, Evaluation, and Challenges. IEEE Signal Processing Magazine, 32(6), pp. 55-73.

Article contributions to peer-reviewed conferences

Umbert, M., Bonada, J., & Blaauw, M. (2013). Generating singing voice expression contours based on unit selection. In Stockholm Music Acoustics Conference (SMAC), pp. 315-320, Stockholm, Sweden.

Pratyush, Umbert M., & Serra X. (2010) A look into the past: Analysis of trends and topics in the Sound & Music Computing Conference. Sound and Music Computing Conference (SMC), Barcelona, Spain.

Workshops

Umbert, M., Invited Workshop on the Synthesis of Singing. (2014) 40th International Computer Music Conference (ICMC) and the 11th Sound & Music Computing conference (SMC), Athens, Greece, September.

Umbert, M., Bonada, J., & Blaauw, M. (2013). Systematic database creation for expressive singing voice synthesis control. In Proc. Int. Speech Communication Association (ISCA), 8th Speech Synthesis Workshop (SSW8), pp. 213-216, Barcelona.

Theses

Umbert, M., Bonada, J., & Janer, J. (2010). Emotional speech synthesis for a Radio DJ: corpus design and expression modeling. Master's thesis, Universitat Pompeu Fabra, Barcelona, Spain.

Additional and up-to-date information about the author may be found at the author's web page[2].

[2] http://martiumbert.weebly.com