VOICE PROCESSING AND SYNTHESIS BY PERFORMANCE SAMPLING AND SPECTRAL MODELS

A DISSERTATION SUBMITTED TO THE DEPARTMENT OF INFORMATION AND COMMUNICATION TECHNOLOGIES OF THE UNIVERSITAT POMPEU FABRA OF BARCELONA FOR THE PROGRAM IN COMPUTER SCIENCE AND DIGITAL COMMUNICATIONS IN PARTIAL FULFILMENT OF THE REQUIREMENTS FOR THE DEGREE OF

–

DOCTOR PER LA UNIVERSITAT POMPEU FABRA

–

WITH THE MENTION OF EUROPEAN DOCTOR

Jordi Bonada Sanjaume

2008

DOCTORAL DISSERTATION DIRECTION

Dr. Xavier Serra
Department of Information and Communication Technologies
Universitat Pompeu Fabra, Barcelona

This PhD project has obtained the first Rosina Ribalta prize 2007, granted by the EPSON Foundation (http://www.fundacion-epson.es) to the best PhD Thesis projects within areas of Information Technologies and Communications

This research was performed at the Music Technology Group of the Universitat Pompeu Fabra in Barcelona, Spain. Primary support was provided by the Japanese company Yamaha Corp. This research was partially funded by the EU project IST-507913 and by the European program MOSART (Music Orchestration Systems in Algorithmic Research and Technology) providing an internship at the Music Acoustics Group, Department of Speech, Music and Hearing, KTH in Stockholm, Sweden.

Abstract

Singing voice is one of the most challenging musical instruments to model and imitate. Along several decades much research has been carried out to understand the mechanisms involved in singing voice production. In addition, from the very beginning of the sound synthesis techniques, singing has been one of the main targets to imitate and synthesize, and a large number of synthesizers have been created with that aim.

The goal of this thesis is to build a singing voice synthesizer capable of reproducing the voice of a given singer, both in terms of expression and timbre, sounding natural and realistic, and whose inputs would be just the score and the lyrics of a song. This is a very difficult goal, and in this dissertation we discuss the key aspects of our proposed approach and identify the open issues that still need to be tackled.

This dissertation substantially contributes to the field of singing voice synthesis: a) it critically discusses spectral processing techniques in the context of singing voice modeling, and provides significant improvements to the current state of the art; b) it applies the proposed techniques to other application contexts such as real-time voice transformations, museum installations or video games; c) it develops the concept of synthesis based on performance sampling as a way to model the sonic space produced by a performer with an instrument, focusing on the specific case of the singing voice; d) it proposes and implements a complete framework for singing voice synthesis; e) it explores the sonic space of the singing voice and proposes a procedure to model it; f) it discusses the issues involved in the creation of the synthesizer's database and provide tools to automate its generation; g) it performs a qualitative evaluation of the synthesis results, comparing those to the state of the art and to real singer performance; h) it implements all the research results into an optimized software application for singing voice analysis, modeling, transformation and synthesis, including tools for database creation; i) a significant part of this research has been incorporated to a commercial singing voice software by Yamaha Corp.

Resum

La veu cantada és probablement l'instrument musical més complex i més ric en matisos expressius. Al llarg de varies dècades s'ha dedicat molt d'esforç a investigar i estudiar les seves propietats acústiques i a entendre els mecanismes involucrats en la producció de veu cantada, posant especial èmfasis en les seves particularitats i comparant-les amb les de la parla. A més, des de l'aparició de les primeres tècniques de síntesi de so, s'ha intentat imitar i sintetitzar per mitjà de tècniques de processament del senyal.

El principal objectiu d'aquesta recerca doctoral és construir un sintetitzador de veu cantada capaç de reproduir la veu d'un cantant determinat, que tingui la seva mateixa expressió i timbre, que soni natural, i que tingui com a entrades només la partitura i la lletra de una cançó. Aquest és un objectiu molt ambiciós, i en aquesta tesi discutim els principals aspectes de la nostra proposta i identifiquem les qüestions que encara queden obertes.

Aquesta tesi contribueix substancialment al camp de la síntesi de veu cantada: a) realitza una revisió crítica dels mètodes de processament espectral per al modelat de veu cantada, i aporta importants contribucions a l'estat de l'art; b) aplica les tècniques proposades a altres contextos tals com la transformació de veu a temps real, instal·lacions de museus or videojocs; c) desenvolupa el concepte de síntesi basada en mostreig d'interpretacions com una manera de modelar l'espai sonor produït per un intèrpret amb un instrument determinat, centrant-se en el cas específic de la veu cantada; d) proposa i implementa un sistema complet per a la síntesi de veu cantada; e) explora l'espai sonor de la veu cantada i proposa un procediment general per a modelar-lo; f) discuteix els aspectes involucrats en la creació de la base de dades del cantant i proporciona eines per a automatitzar-ne la creació; g) realitza una avaluació qualitativa de la veu sintètica, comparant-la amb l'estat de l'art i amb cantants reals; h) implementa els resultats de la investigació en un programa informàtic optimitzat dedicat a l'anàlisi, modelat, transformació i síntesi de la veu cantada, incloent eines per a la creació de la base de dades del cantant; i) una part important d'aquesta investigació s'ha incorporat en un sintetitzador comercial de veu cantada desenvolupat per Yamaha Corp.

Resumen

La voz cantada es probablemente el instrumento musical más complejo y el más rico en matices expresivos. A lo largo de varias décadas se ha dedicado mucho esfuerzo de investigación a estudiar sus propiedades acústicas y a entender los mecanismos involucrados en la producción de voz cantada, poniendo especial énfasis en sus particularidades y comparándolas con el habla. Desde la aparición de las primeras técnicas de síntesis de sonido, se ha intentado imitar dichos mecanismos y encontrar maneras de reproducirlos por medio de técnicas de procesado de señal.

El principal objetivo de esta investigación doctoral es construir un sintetizador de voz cantada capaz de reproducir la voz de un cantante determinado, que tenga su misma expresión y timbre, que suene natural, y cuyas entradas sean solamente la partitura y la letra de una canción. Éste es un objetivo muy ambicioso, y en esta tesis discutimos los principales aspectos de nuestra propuesta e identificamos las cuestiones aún sin resolver.

Esta tesis contribuye substancialmente al campo de la síntesis de voz cantada: a) realiza una revisión crítica de los métodos de procesado espectral para modelado de voz cantada, y aporta importantes contribuciones al estado del arte; g) aplica las técnicas propuestas a otros contextos tales transformación de voz a tiempo real, instalaciones de museos o videojuegos; c) desarrolla el concepto de síntesis basada en muestreo de interpretaciones como una manera de modelar el espacio sonoro producido por un intérprete con un instrumento determinado, centrándose en el caso específico de la voz cantada; d) propone e implementa un sistema completo para la síntesis de voz cantada; e) explora el espacio sonora de la voz cantada y propone un procedimiento general para modelarlo; f) discute los aspectos involucrados en la creación de la base de datos del cantante y proporciona herramientas para automatizar su creación; g) realiza una evaluación cualitativa de la voz sintética, comparándola al estado del arte y a cantantes reales; h) implementa los resultados de la investigación en un programa informático optimizado dedicado al análisis, modelado, transformación y síntesis de la voz cantada, incluyendo herramientas para la creación de la base de datos del cantante; i) una parte importante de esta investigación se ha incorporado en un sintetizador comercial de voz cantada desarrollado por Yamaha Corp.

Acknowledgements

I thank Professor Xavier Serra, who introduced me into this magic world of audio digital signal processing and provided funding for me to work at the Music Technology Group. Thanks to Yamaha Corp. for funding most of my research at MTG. Thanks so much to all the people working at the MTG for helping me to spend a really nice time at work, usually a difficult task, and also for being my friends, which is even more impressive. Special thanks to Perfe, Ramon, Àlex, Pedro, Maarten, Xavier, Eduard, Mark, Tue, Lars, Oscar & Oscar, Jaume, Claudia, Enric, Esteban, Alfonso, Merlijn, JJaner , Marc, Graham and Fernando, who have worked next to me and have stood my background music. Finally, thanks to Emilia for supporting my work from the beginning, being my advisor, my friend and the most wonderful wife I never could have thought of.

To Emilia

Contents

Notations

Variables

h	harmonic index
t	continuous time variable in seconds
f	continuous frequency variable in Hz
υ	continuous oscillation frequency
ω	angular frequency in radians/second ($\omega=2\pi\upsilon$)
n	discrete time variable
k	discrete frequency variable
T_s	sampling interval in seconds
F_s	sampling rate in Hz
$F_1, F_2, ...$	Formant frequencies
m	frame index
a_h	amplitude of h^{th} harmonic
f_h	frequency of h^{th} harmonic
f_0	fundamental frequency
$\theta_{0,h}$	phase at time zero ($n=0$) of h^{th} harmonic
$a_{h,m}$	amplitude of the h^{th} harmonic of the m^{th} frame
$f_{h,m}$	frequency of the h^{th} harmonic of the m^{th} frame
$\theta_{0,h,m}$	phase at time zero ($n=0$) of the h^{th} harmonic of the m^{th} frame
$\kappa_{h,m}$	harmonic parameters of the h^{th} harmonic of the m^{th} frame, $\kappa_{h,m}=\left(a_{h,m}, f_{h,m}, \theta_{0,h,m}\right)$

Functions

T_{pitch}	pitch transposition transformation value as a ratio between output and input fundamental frequencies
T_{time}	time-scaling transformation value as a ratio between output and input duration
T_{timbre}	timbre-scaling transformation value as a ratio between output and input frequencies
$\bar{T}_{pitch}(t)$	pitch transposition transformation function as a time-varying ratio between output and input fundamental frequencies
$\bar{T}_{time}(t)$	time-scaling transformation function as a mapping between input and output time
$\bar{T}_{timbre}(f)$	timbre-scaling transformation as a mapping function between input and output frequencies
$\chi(a, f, \theta)$	a function or method which computes the discrete time domain signal of a sinusoid with parameters a , f and θ
$x(t)$	audio signal
$s(t)$	audio signal
$s'(t)$	synthesis audio signal
$w(n)$	analysis temporal window
$w_{ov}(n)$	synthesis overlapping temporal window

Δ_t	hopsize or time distance in seconds between consecutive frames
$\gamma(h)$	harmonic mapping in harmonic-as-spectral-regions synthesis
$\Upsilon_h(k)$	segmentation window associated to h^{th} harmonic
$H_{harm}(f)$	spectral envelope defined by harmonics
$\Im$	imaginary part of a complex number
$\Re$	real part of a complex number

Abbreviations

FT	Fourier Transform
DFT	Discrete Fourier Transform
DTFT	Discrete-Time Fourier Transform
FFT	Fast Fourier Transform
STFT	Short-Time Fourier Transform
MFCC	Mel Frequency Cepstral Coefficients
MFPA	Maximally Flat Phase Alignment
EpR	Excitation plus Resonances
NBVPM	Narrow-Band Voice Pulse Modeling
WBVPM	Wide-Band Voice Pulse Modeling
TTS	Text-to-Speech
SP	Singer Performance
DS	Dissertation Synthesis
OS	Other Synthesizers
VS	Vocaloid Synthesis

Chapter 1

Introduction

1.1 Motivation

The singing voice is probably the most complex musical instrument and the richest one with respect to expressive nuances. Much research along several decades has been devoted to study its acoustical properties and to understand the details of the mechanisms involved in singing voice production, putting special emphasis on its particularities compared to speech.

With the appearance of sound synthesis techniques, special emphasis has been devoted to imitate the processes involved in singing voice production and to find ways to reproduce them by means of signal processing techniques (Kob 2002, Rodet 2002). Among the many existing approaches to the synthesis of musical sounds, the ones that have had the most success are without any doubt the sampling based ones, which sequentially concatenate samples from a corpus database (Schwarz 2007). The success of sampling relies on the simplicity of the approach. It just samples existing sounds, but most importantly it succeeds in capturing the naturalness of the sounds, since the samples are real sounds. However, sound synthesis is far from being a solved problem and sampling is far from being an ideal approach. The lack of flexibility and expressivity are two of the main problems, and there are still many issues to be worked on if we want to reach the level of quality that a professional musician expects to have in a musical instrument. Sampling based techniques have been used to reproduce practically all types of sounds and basically have been used to model the sound space of all musical instruments. They have been particularly successful for instruments that have discrete excitation controls, such as percussion or keyboard instruments. For these instruments it is feasible to reach an acceptable level of quality by using large sample databases, thus by sampling a sufficient portion of the sound space produced by a given instrument. This is much more difficult for the case of continuously excited instruments, such as bowed strings, wind instruments and especially the singing voice, and therefore recent sampling based systems consider a trade-off between performance modeling and sample reproduction (e.g. (Lindemann 2007)). For these instruments there are numerous control parameters and many ways to attack, articulate or play each note. The control parameters are constantly changing and the sonic space covered by a performer could be considered to be much larger than for the discretely excited instruments.

Probably the singing voice is the musical instrument that has been synthesized with the least success in the sound synthesis field, despite the many efforts devoted to achieve a realistic, natural sounding, and expressive result. Nowadays we find singing voice in almost every musical production. Thus, it would be a dream for many composers, musicians and producers to have tools capable of synthesizing real sounding voices anytime anywhere. We share this dream and believe that this dissertation contributes one more step towards this challenging goal.

1.2 Goals

The final goal of this thesis is to reach a singing voice synthesizer capable of reproducing the voice of a given singer, both in terms of expression and timbre, having as inputs simply the score and the lyrics of a song. In addition, the synthesis should sound as natural and realistic as possible. This is a very difficult challenge, and in this dissertation we discuss the key aspects of our proposed approach and identify the open issues that still need to be tackled.

We attempt to use concatenative synthesis as the basis of our synthesizer. This implies transforming and smoothly connecting recorded samples from the target singer. Therefore, we want to explore and discuss different approaches based on spectral processing that aim at transforming those voice samples with the best quality and the highest flexibility. In that sense, we point out the necessity of using spectral voice models that are aware and take advantage of the processes involved in voice production. We want to define an amplitude spectral voice model that distinguishes and considers separately the voice source and the vocal tract, which deals with resonances and is able to reconstruct perfectly the original singer's spectrum with all its details and nuances. This will give us the required quality and flexibility for transforming voice samples. Furthermore, we want to identify a spectral phase model able to predict the harmonic phase relationship at voice pulse onsets without altering the perceptual timbrical characteristics. Such phase model is essential to simplify the concatenation of contiguous samples when synthesizing.

Spectral voice models can be considered as high-level models that work on top of low-level voice processing techniques. Our aim is to combine in a single technique the high temporal resolution typical of time-domain algorithms with the high frequency resolution typical of frequency-domain methods. This way, we attempt to have enough temporal resolution so to transform independently each of the voice pulses while at the same time being able to control independently each harmonic component. This will give us the flexibility to perform most types of voice transformations with high quality.

Another concern is to model the singer sonic space, find its most relevant dimensions, and define recording scripts that cover enough singing contexts as to capture most of the relevant phonetic and expressive aspects of the singer performance. An important aspect to study is how to minimize the length of these scripts so to shorten recording sessions, facilitate the database creation process and reduce the synthesizer's database size to practical levels. In addition, we want to automate the database creation so to reduce the efforts put in such process and facilitate the creation of several voices. One more goal is to explore and develop methods to capture and model relevant aspects of the singer's expressivity. We plan to represent expressive resources with templates obtained from performance examples that can be applied to other samples and produce a similar expression. Furthermore, for each template category we have to identify the controls that offer both natural sounding and flexible transformations.

We attempt to show as well that synthesis results are improved by obtaining samples from actual singing performances instead of mechanical and descontextualized exercises. In addition, we want to put the needed means to ensure the feeling of a continuous synthesis flow. In other words, we expect to minimize the sensation of listening to a sequence of disconnected samples coming from different contexts. More generally, along our research we want to adopt performance actions and physical constraints in order to convert what would be the basic sampling approach to a more flexible and expressive technology while maintaining its inherent naturalness.

Last, but not less important, a significant part of this research has been carried out in the context of a joint research project with Yamaha Corp., with perspectives of developing a prototype to be afterwards turned into a commercial product by this company. It is also a challenge then to combine the requirements of an academic research with the constraints and milestones of an industrial collaboration, which often implies a strict schedule and includes the commitment of generating new intellectual property to be protected by patents.

1.3 Singing Voice Production

We consider singing voice as "the sounds produced by the voice organ and arranged in adequate musical sounding sequences" (Sundberg 1987). The voice organ encloses the different structures that we mobilize when we produce voice: the respiratory system, the vocal folds, and the vocal and nasal tracts. More specifically, as shown in Figure 1.1, voice sounds originate from an airstream from the lungs that is processed by the vocal folds and then modified by the pharynx, the mouth and nose cavities. The sound produced by the vocal folds is called the Voice Source. When the vocal folds vibrate, the airflow is chopped into a set of pulses producing voiced sounds (i.e. harmonic sounds). Otherwise, different parts of the voice organ can work as oscillators to create unvoiced sounds. For example, in whispering, vocal folds are too much tense to vibrate but they form a narrow passage that makes the airstream become turbulent and generate noise. The vocal tract acts as a resonator and shapes acoustically the voice source, especially enhancing certain frequencies called formants[1] (i.e. resonances). The final step is the acoustic radiation though the lips.

The Voice Source

During phonation the respiratory system acts as a compressor and controls the pressure under the glottis or subglottal pressure (P_s). When a certain threshold pressure is exceeded, the vocal folds at the glottis are set into vibration and act as an oscillator. The vibration frequency is mainly determined by the length, tension and mass of the vocal folds, and in a minor grade by P_s. The resulting pulsating air-stream is called the Voice Source. It features a harmonic spectrum whose fundamental frequency (f_0) matches the vocal folds vibration frequency. The voice source is partially described by f_0, amplitude and spectral descriptors, which relate to perceived pitch, loudness and timbre respectively. The voice source spectrum is mainly described by the fundamental amplitude and the spectral tilt. Due to the difficulty of directly recording the voice source, often the inverse filtering method is used to estimate it. It consists on removing the vocal tract transfer function from the radiated sound spectrum by subtracting estimated formants (Lindqvist Gauffin 1964). Figure 1.2 shows a radiated pressure waveform and its corresponding estimated voice source. Several parameters can be derived from such representation, being the most common ones those listed in Table 1.1. These parameters have been widely used to create and control several voice source models such as the well-known Liljencrants-Fant model (Fant 1986).

Voice Processing

In terms of signal processing, voice production is often modeled as a linear *source-filter* system. The voice source signal $s_{vs}(t)$ corresponds to the *source* of the system, whereas the vocal and nasal tracts plus the lips radiation constitute the *filter* of the system. Often the *filter* is modeled as an AutoRegressive Moving Average (ARMA) filter followed by a differentiator filter. In such approach, the AR filter represents the formants of the vocal tract, the MA filter models the antiresonances produced by the coupling between vocal and nasal tracts, and the differentiator filter represents the effect of signal radiation through the lips. Figure 1.3 illustrates the source-filter model, where $p(t)$ would be the sound pressure captured by a microphone. Note that if the whole system is considered to be composed of Linear Time-Invariant (LTI) filters, then the filter order can be changed. This is the case if the filters are considered to vary very slowly respect to the fundamental period, so that they can be regarded as almost constant filters. It is a common practice to replace the ARMA filter by an AR filter. The reason relies in the simplicity of the approach and the reasonably low error obtained in the majority of cases.

[1] There also exist the antiformants or antiresonances, i.e. frequency regions in which the amplitudes of the voice source are attenuated. These are especially present in nasal sounds because nasal cavities absorb energy from the sound wave.

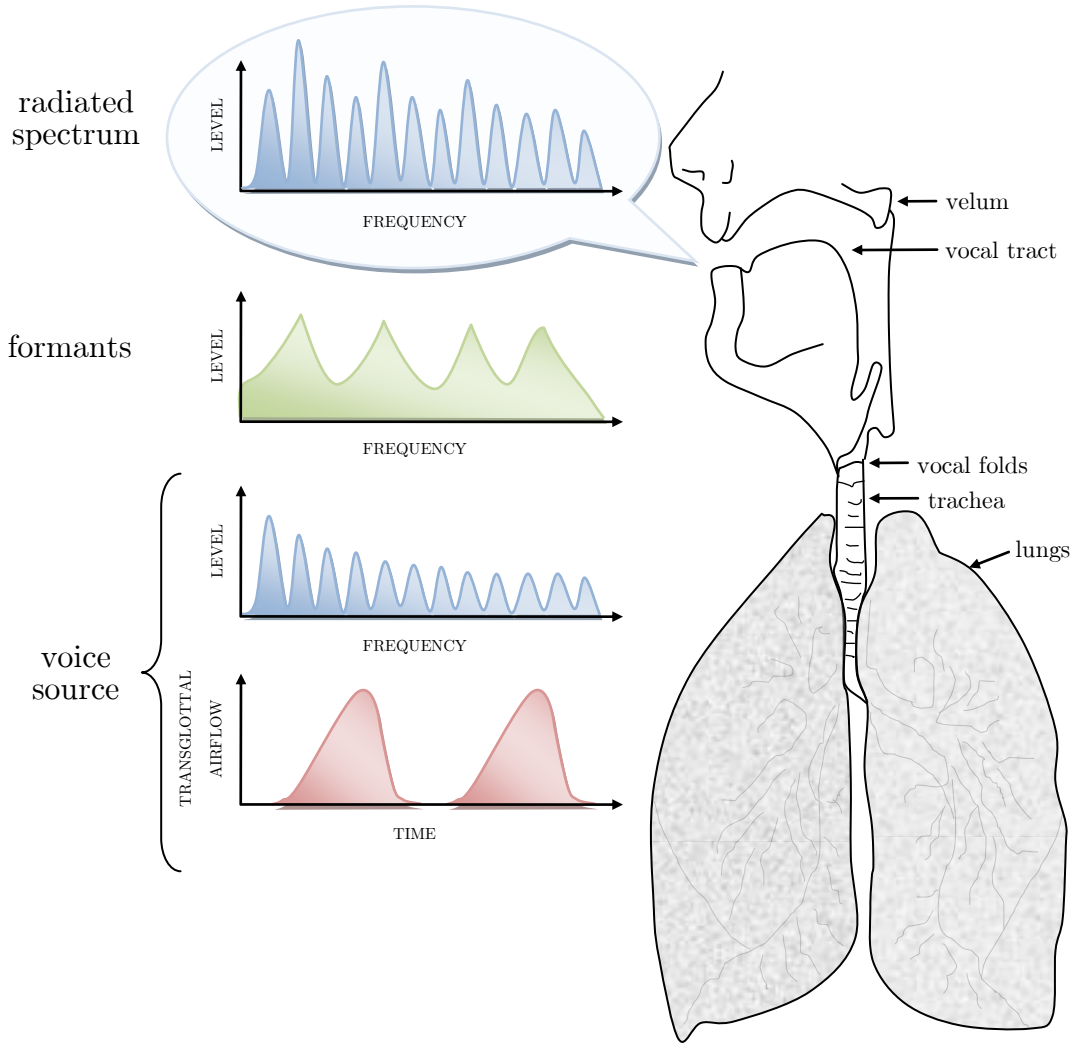


Figure 1.1 The voice organ. During voiced phonation, the airstream coming from the lungs is chopped by the vocal folds into a sequence of voice pulses. This sound produced by the vocal folds is called voice source, and in the frequency domain appears like a slowly decaying (around -6 dB per octave) train of harmonics. The voice source is then modified by the vocal and nasal tracts, which shape it by enhancing certain frequencies (i.e. formants) and attenuating others (i.e. anti-formants). Finally, the voice sound is radiated trough the opened mouth. Adapted from (Sundberg 1987).

Of particular interest for signal processing is the detection of glottal closure instants (GCIs). The AR filter is a good approximation while the vocal folds are closed. When they open the conditions change due to the coupling of the supra and subglottal cavities. The glottal opening produces a subglottal pressure and, due to the Bernoulli effect, the vocal folds close abruptly. This often produces a prominent excitation to the vocal and nasal tracts, and therefore CGIs are strongly correlated with local energy maxima. Particularly, many voice processing methods such as those based on pitch-synchronous overlap-and-add (PSOLA) use the CGI estimations to center analysis frames. Figure 1.4 shows a recorded utterance together with its correspondent Liljencrants-Fant model representation of the transglottal flow and its derivative. Precise CGI estimations can be obtained by measuring the conductivity of the larynx with a laryngograph, which mainly depends on the glottis opening coefficient. Figure 1.5 shows an example of a voice utterance and its corresponding laryngograph signal.

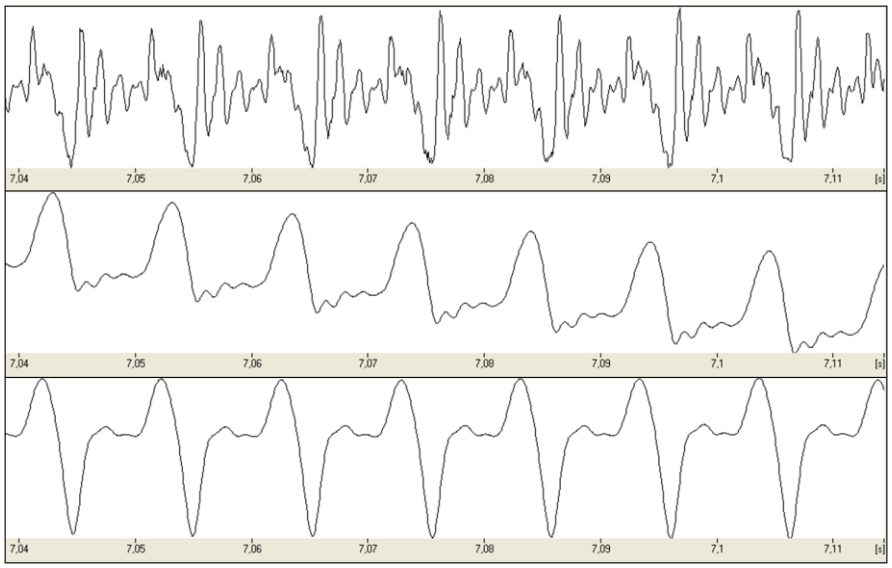


Figure 1.2 Pressure waveform and estimated voice source derivative. On top we see the waveform captured by a microphone of an /a/ Spanish vowel sung by a male. The middle view shows the corresponding estimated voice source obtained by inverse filtering using the Decap software by Svante Granqvist. The bottom view shows the voice source derivative.

period duration	T_0	inverse of vocal fold vibration frequency
flow amplitude	$\hat{U}$	maximum flow amplitude
open phase duration	T_{op}	duration of the phase within a period in which vocal folds are separated
closed phase duration	T_{cl}	duration of the phase within a period in which vocal folds are approximately closed
closed quotient	Q_{closed}	ratio between closed and open durations T_{cl}/T_0
closed phase mean flow	DC flow	mean flow during closed phase, due to glottal leakage
peak-to-peak flow	PtP	flow difference between $\hat{U}$ and DC flow
maximum flow declination rate	$MFDR$	minimum of the flow derivative, i.e. peak speed of flow decrease during closing phase

Table 1.1 Voice source parameters

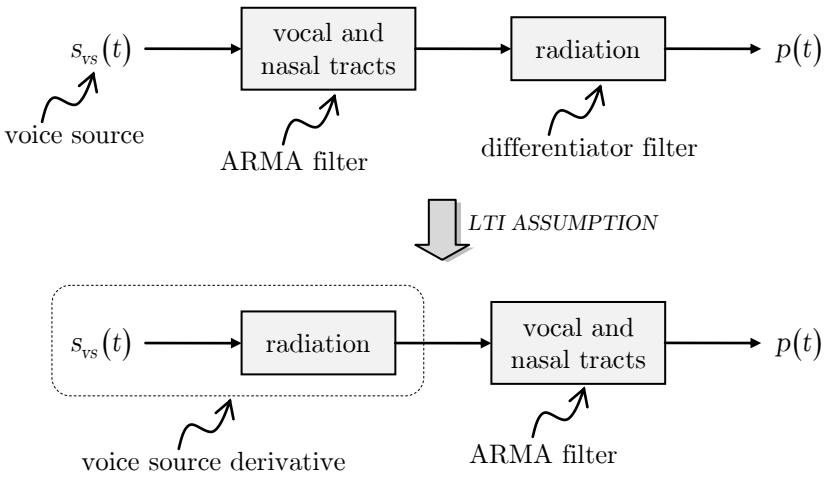


Figure 1.3 The voice-source filter model of the voice production system

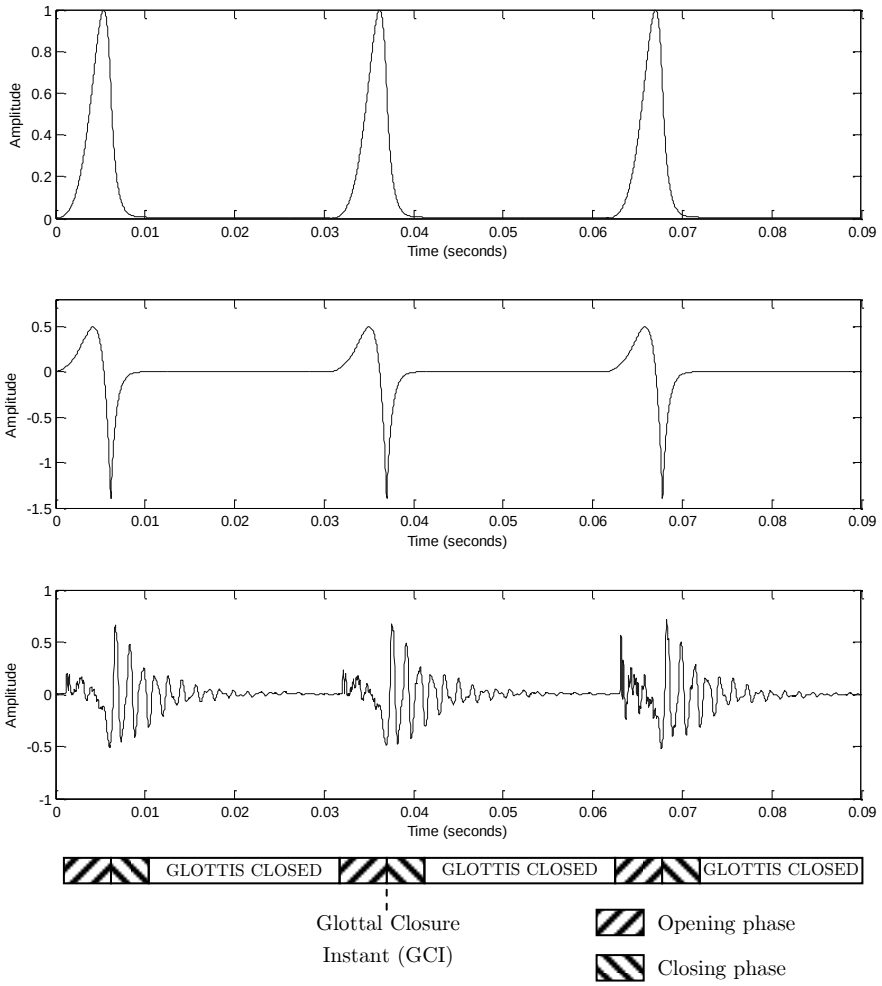


Figure 1.4 Glottis opening/closing cycle. The bottom view shows the recorded waveform corresponding to a fry utterance of an adult male (reference audio [1]). Top and middle views show respectively the corresponding simulated glottal flow and glottal flow derivative using the well-known Liljencrants-Fant model (Fant 1986). Glottal closure instants (GCI) are marked with vertical dashed lines.

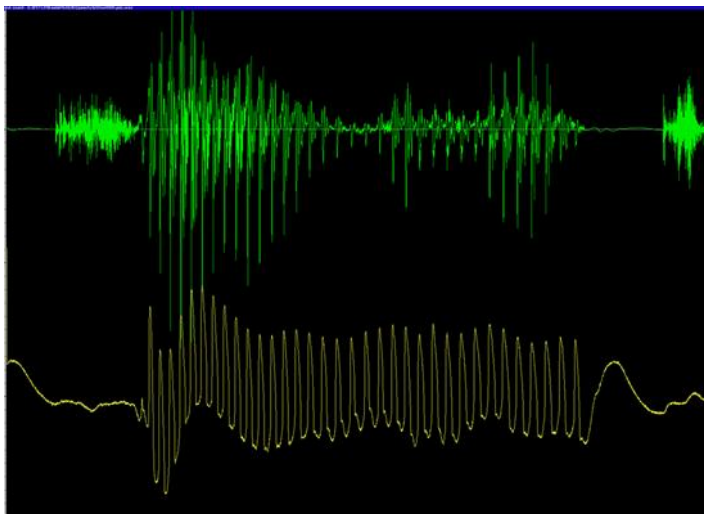


Figure 1.5 The top view shows a voice pressure waveform captured by a microphone. The bottom view shows the corresponding laryngograph signal. Peaks in the laryngograph signal mark the CGIs. Data obtained from the Keele Pitch Database, Keele University (UK).

1.4 Singing Voice Synthesis

In a broad sense, and according to whether the focus is put on the system or its output, synthesis models can be classified into two main groups: spectral models and physical models. Spectral models are mostly based on perceptual mechanisms of the listener while physical models focus on modeling the production mechanisms of the sound source. Any of these two models could be suitable depending on the specific requirements of the application, and they might even be combined for taking advantages of both approaches. Next we overview most common methods used in singing voice synthesis.

Physical Models

Generally speaking, physical models focus on understanding and modeling the mechanisms involved in the sound production process. In other words, some knowledge of the real-world mechanisms must be brought on the design. Typically this implies to make use of differential equations to emulate the way pressure waveforms originate, interact and propagate trough the different elements of the instrument being modeled.

The controls of a physical model are closely related to the ones the performer activates to control the instrument. In the case of the singing voice, the performer is controlling his own vocal system and, therefore, the controls are related to the movements of the vocal apparatus elements: jaw opening, tongue shape, sub-glottal air pressure, tensions of the vocal folds, etc. Although it could seem that the controls of the physical model are quite intuitive, it is not the case for the singing voice. The reason is that the singer is not aware of how he is actually moving the different elements of his voice organ, since most of these movements are unconscious and originate from automated behaviors learnt by speaking and singing over the years. Instead, the singer consciously performs principally melodies and notes, words and phonemes. Moreover, the mapping of those controls of the production mechanism to the final output of the model, and so to the listener's perceived quality, is not a trivial task.

The first digital physical model of the voice was based on simulating the vocal tract as a series of one-dimensional tubes (Kelly y Lochbaum 1962), afterwards extended by means of digital waveguide synthesis (Smith 1992) in SPASM (Cook 1992). Physical models are evolving fast and becoming more and more sophisticated, 2-D models are common nowadays providing increased control and realism (Mullen, et al. 2004)(Mullen, et al. 2006), and the physical configuration of the different organs during voice production is being estimated with great detail combining different approaches. For example, 2D vocal tract shapes can be estimated from Functional magnetic resonance imaging (fMRI) (Story, et al. 1996), X-ray computed tomography (CT) (Story 2003), or even audio recordings and EGG signals by means of genetic algorithms (Cooper, et al. 2006).

Spectral Models

Spectral Models, as their name suggests, deal with the spectral characteristics of the audio, that is, with its frequency domain content. The human auditory system performs a spectral analysis of the air pressure waves getting into the ears and, therefore, we can say that Spectral Models are closely related to the human auditory perception. Indeed, their parameters can be easily mapped to changes of sensations in the listener. Yet, parameter spaces yielded by these systems are not necessarily the most natural ones for manipulation. Typical controls would be pitch, glottal pulse spectral shape, formant frequencies, formant bandwidths, etc. Of particular relevance is the sinusoidal based system in (M. Macon, et al. 1997).

Formant based Synthesis

Formant based synthesizers are essentially spectral models. However, they are typically considered to be pseudo-physical models because they make use of the source-filter decomposition, which considers the voice to be the result of a glottal excitation waveform (i.e., the voice source) filtered by a linear filter (i.e., the vocal/nasal tracts). This assumption is in fact a simplification because it does not take into account the coupling existing between the glottis and the vocal tract during voice production. Formant synthesis is probably the most widely used method during last

decades. Formants are commonly modeled with two-pole resonators providing independent frequency and bandwidth controls. Both parallel and cascade formant filter structures can be used. Cascade structures are more suited for non-nasal voiced sounds, while parallel structures have found to be better for nasals, fricatives and stop-consonants. The most known system is the Klatt Formant Synthesizer (Klatt 1980), which has been incorporated in several TTS systems. It is also worth mentioning the MUSSE DIG system developed at the KTH (Sundberg 2006, Larsson 1977).

Formant Wave Synthesis

Formant wave functions are signal models of a particular resonance. The basic idea of this technique is to combine time-domain waveform models of the impulse response of individual formants to create multi-formant rich spectra. Periodic signals are obtained by repeating in synchrony formant wave functions at the rate determined by the target fundamental frequency. The most known system is the FOF synthesizer in CHANT (Rodet, et al. 1984).

Concatenative Synthesis

Sample based synthesizers transform and concatenate sample units in a database (Schwarz 2007). In those systems, the phonemes and the melody to be sung are used to find the optimal sequence of database samples. This optimization is performed by minimizing a cost function that considers both transformation and concatenation aspects. Then those units are modified in order to match the expected properties: pitch, energy, timbre, expression, etc. Of particular relevance is the system introduced in (Meron 1999).

The approach presented in this dissertation belongs to the category of concatenative synthesis. However, as it will be detailed along the dissertation, our proposed approach to the synthesis of the singing voice makes use as well of spectral models and the source-filter decomposition.

1.5 Organization of the thesis work

This thesis is structured in several chapters. In this Chapter 1, we introduce the motivation of the dissertation, its main goals and the context in which the research has been carried out. As part of the scientific background, we detail the most relevant aspects involved in singing voice production. In addition, we overview the main techniques used to model and synthesize the singing voice.

In Chapter 2, we explore spectral models and voice processing techniques that specifically tackle the characteristics of the singing voice, and we point out the most relevant problems and difficulties we found. We explain that voice utterances can be interpreted as a sequence of filtered time-domain voice pulses or as a set of time-varying frequency components. Each interpretation leads to different processing techniques, which are discussed in depth along the chapter. In addition, at the end we introduce two spectral models specially devised for the human voice.

Chapter 3 introduces the concept of performance based sampling synthesis, putting special emphasis in the fact that we model the sonic space of a performer-instrument combination, and discuss the different modules required to build a synthesizer. In section §3.1, we discuss the key aspects of the proposed synthesizer and describe its components. Next, in §3.2 we detail the issues involved in the creation of the synthesizer's database, starting with the definition of the singing voice sonic space and ending with our efforts in automating the creation process. Section §3.3 focuses on the different aspects involved in the performer modeling and details the most common approaches. In our proposed synthesizer, we have distinguished two main processes. The former, covered in section §3.4, consists of transforming an input score into a performance trajectory within the sonic space of the target instrument, i.e. the singing voice. The latter, detailed in section 3.5, actually generates the output sound by concatenating a sequence of transformed samples that approximate the target performance trajectory. Finally, in section 3.6, we evaluate the synthesis results in terms of singer identity, naturalness, intelligibility and expressivity, and briefly point out the difficulties of performing an objective evaluation.

In Chapter 4 we summarize our contributions, detail our conclusions and discuss about future perspectives of the dissertation research.

In the annexes section, we outline Yamaha's singing voice synthesizer Vocaloid, detail the Spanish recording script, and list the publications and patents by the author relevant to the thesis research.

Finally, we include the bibliography referenced along the dissertation.

Chapter 2

Voice Processing by Spectral Models

Speech and singing voice are signals that concatenate voiced (i.e. pitched) and unvoiced segments. If we focus our attention on voiced sections, they can be considered as the result of a glottal pulses sequence filtered by the vocal tract, as seen in previous chapters, once we assume the simplification of a linear system. As pointed out in the literature (e.g. (Peeters 2001)), the pulse periodicity is never constant due to the fact that the voice organ is a complex mechanical system with many physical variables constantly evolving. In this sense, the periodicity might be stable up to a point where we can talk of quasi-sinusoidal signals, but never of pure sinusoidal signals. It is therefore not straightforward to model voice signals with just sinusoids, even when the phonation has the least possible aspirated air.

In order to obtain a frequency domain resolution good enough as to distinguish the different frequency components (i.e. harmonics) and reliably estimate their parameters, we need a window covering at few periods of the signal. However, since the period is not constant neither the vocal tract immobile, we are then analyzing together periods with different durations, and the magnitude spectrum is not a perfect train of pulses located at the fundamental period and its multiples. Indeed, covering several periods with the analysis window typically results in smearing and smoothing of the synthesized signal. This is probably the main difficulty of spectral harmonic based models: the input signal contains non-stationary sinusoidal components and the accurate estimation of their parameters in real world signals becomes a challenge. Moreover, it is difficult to preserve the inner differences between consecutive pulses, especially during attacks and transitions (i.e. unstable segments). As it will be shown, several techniques have been developed in the last decade with the aim of improving the accuracy in the estimation of each harmonic component, and to achieve the best possible trade-off between temporal and frequency resolution.

Voiced utterances can be interpreted as (1) a set of time-varying quasi-sinusoidal signals looking at the frequency axis, or (2) as a sequence of voice pulses looking at the temporal axis. This duality is illustrated in Figure 2.1. In the first case (1), we can think of a set of oscillators whose frequency relation is fixed to natural numbers (multiples of the fundamental frequency, inverse of the period), and whose amplitude is set by the target spectral envelope. Conversely, in the second case (2), we would filter a train of time domain pulses at the target periodicity.

A common problem of the first interpretation (1) is the one of *shape variance:* the loss of the intrinsic phase synchronization between harmonics found in voice signals, also called *phase-coherence*. This is typical of frequency domain based techniques (e.g. sinusoidal models, phase vocoder), although several ways of improving this issue have been proposed during the last years, as it will be detailed afterwards in section 2.1.

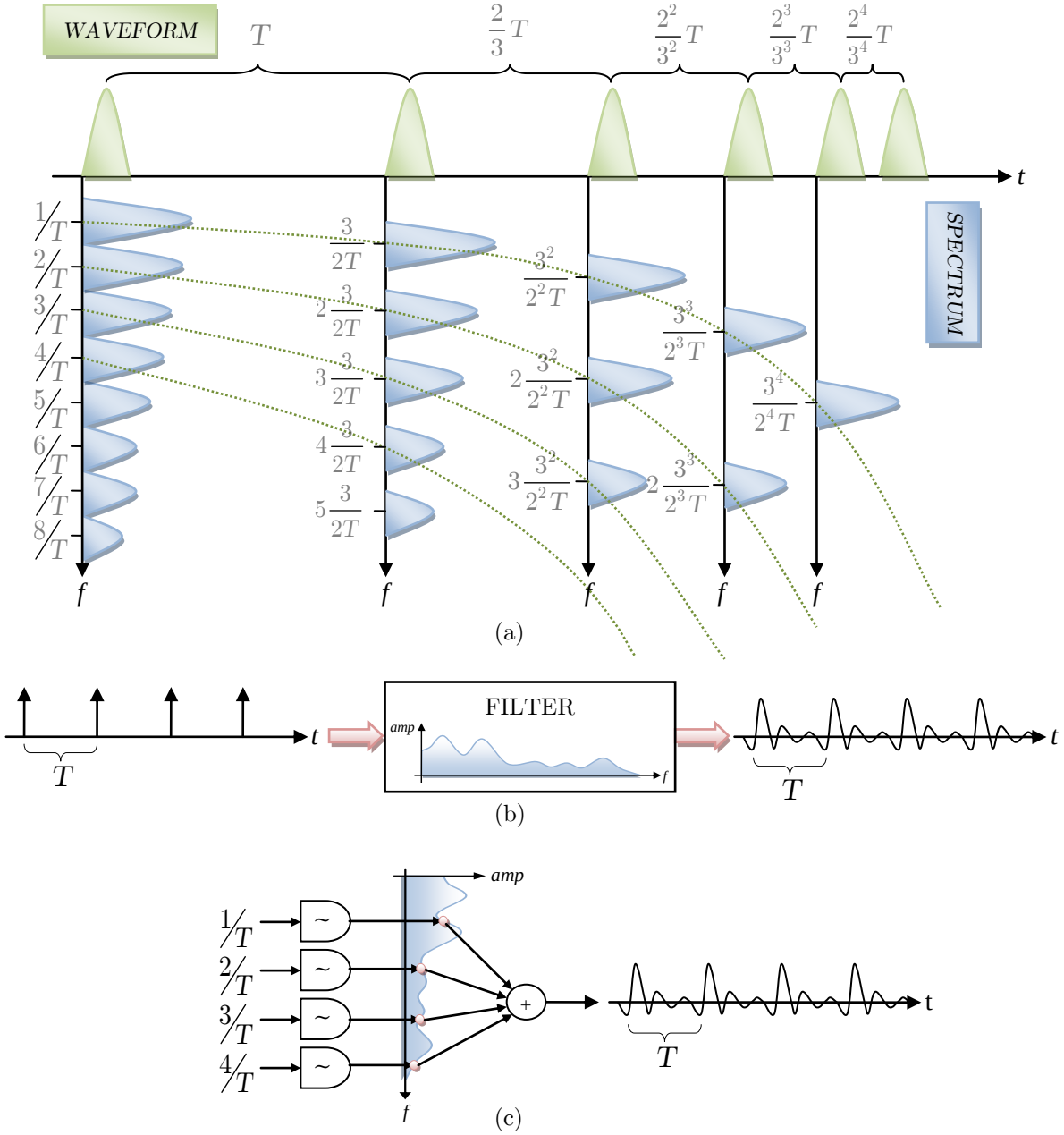


Figure 2.1 Harmonic signals can be seen as a set of time-varying quasi-sinusoidal signals (looking at the frequency axis), or as a sequence of voice pulses (looking at the time axis). The top figure (a) shows a train of time domain pulses (horizontal) with a period decreasing 2/3 each time, and its magnitude spectrum (vertical) at different time positions. It becomes obvious the inverse behavior of time and frequency, producing a given temporal periodicity an inverse value of frequency periodicity (e.g. T seconds → 1/T Hertz). In the same way, while temporal pulses get closer, frequency components get more distant. The middle figure (b) shows the horizontal interpretation as a train of filtered pulses. In (c) we see the vertical interpretation as a group of oscillators whose amplitude follows the target spectral envelope. Here we haven't considered phase for simplification.

In the following section we will see how both interpretations lead to different processing techniques, and discuss in depth the pros and drawbacks of each one. First, we will explore the most common methods for estimating harmonic parameters. Then we will see how following the first interpretation (1) those estimated harmonics are incorporated into a set of harmonic trajectories, transformed and synthesized. Next, we will see how the second interpretation (2) leads us to estimate and model voice pulses out of the harmonic parameters. Finally, we will propose a hybrid method that combines both interpretations, modeling each individual pulse with a set of sinusoids.

2.1 Harmonic estimation

The Fourier transform (FT) of a stationary sinusoid $x(t)$ with constant amplitude and frequency is a pair or deltas located at its positive and negative frequency, of amplitude equal to half the sinusoid's amplitude.

$$x(t) = A\cos(2\pi f_0 t)$$
$$X(f) = FT[x](t) = \int_{-\infty}^{\infty} A\frac{e^{j2\pi f_0 t} + e^{-j2\pi f_0 t}}{2} e^{-j2\pi ft} dt = \frac{A}{2}\left[\delta(f - f_0) + \delta(f + f_0)\right] \tag{2.1}$$

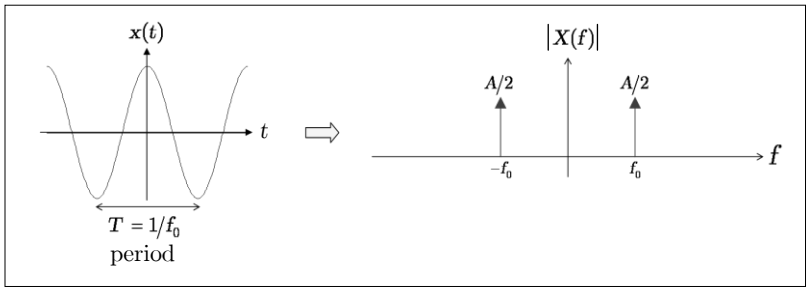


Figure 2.2 Fourier transform (FT) of a stationary sinusoid.

If the sinusoid $x(t)$ is sampled each T_s seconds we obtain a discrete signal whose Fourier transform is equal to the two previous deltas repeated around multiple periods of the sampling frequency $f_s = 1/T_s$.

$$x(t) = \begin{cases} A\cos(2\pi f_0 t) & if\ t = nT_s\ \ \forall n \in \mathbb{Z} \\ 0 & otherwise \end{cases} \Longrightarrow \quad x(n) = A\cos(2\pi f_0 n T_s) \tag{2.2}$$

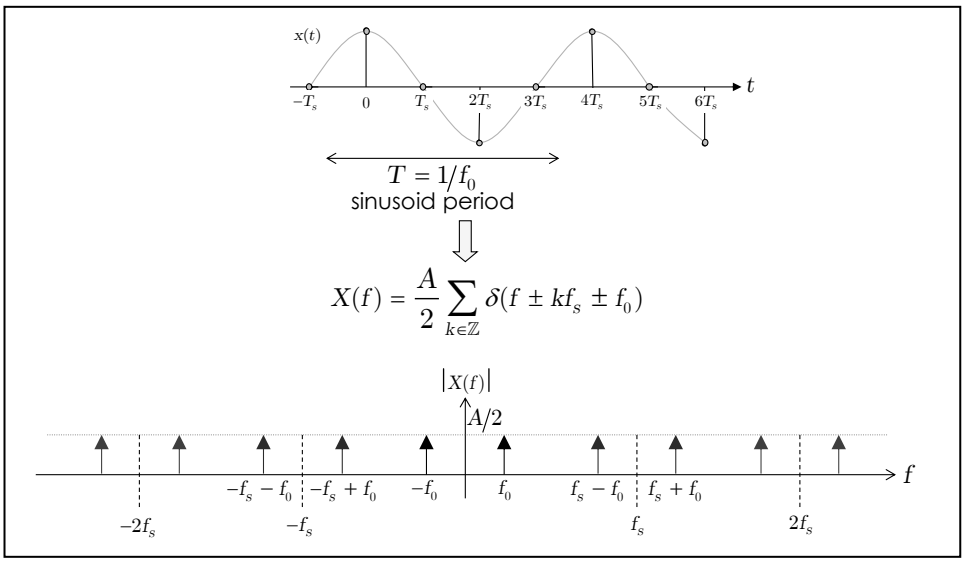


Figure 2.3 Discrete-Time Fourier Transform (DTFT) of a sinusoid.

If we then multiply the discrete signal $x(n)$ by an arbitrary window $w(n)$, we can compute its forward and inverse discrete-time STFT (Short Time Fourier Transform) as

$$\begin{aligned} X_w(k) &= \sum_{n=0}^{N-1} w(n)x(n)e^{-j2\pi nk/N} \qquad k = 0,1,...,N-1 \\ w(n)x(n) &= \frac{1}{N}\sum_{k=0}^{N-1} X_w(k)e^{j2\pi nk/N} \qquad n = 0,1,...,N-1. \end{aligned} \tag{2.3}$$

The spectrum we obtain is equal to the convolution of the window transform by the signal transform; it consists of shifted versions of the window transform at each delta position[2]. This is illustrated in Figure 2.6 and the following drawing

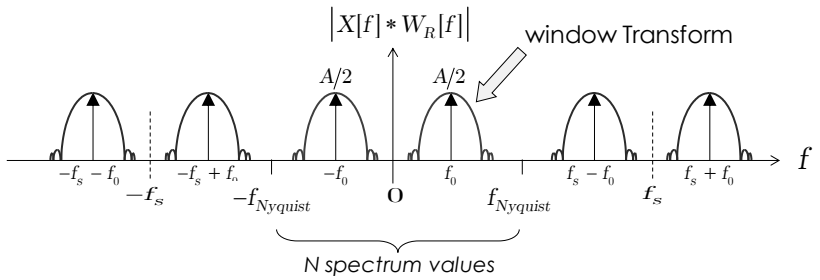


Figure 2.4 Convolution of the window transform by the signal transform.

However, when we deal with real world signals we find more than one single sinusoid and also other non-sinusoidal frequency components. Thus, we have to face a few issues that complicate the estimation of those sinusoidal components:

- ❖ Time-frequency resolution tradeoff

  The width B of the main lobe of the window transform is inversely proportional to the length L of the analysis window $w(n)$, i.e. $B \approx C/L$, where C is a constant that depends on each window type[3]. If we use a long window the main lobes of its transform are narrow and the different harmonic components are clearly distinguishable, thus the frequency resolution is high. However, the window covers a long time extension and the temporal variations are smoothed. Therefore, the temporal resolution is low and fast transitions are not properly represented. On the other hand, if we apply short analysis windows, the width of the main lobes is increased. This causes main lobes of consecutive harmonics to overlap and the spectral peaks are not distinguishable anymore (see Figure 2.7); consequently the frequency resolution becomes poor. However, as the analysis window is short, fast temporal changes are better handled and the temporal resolution is high. Summarizing, long windows suppose

---

[2] This can be demonstrated by computing the STFT of the complex exponential $x(n) = Ae^{j\frac{2\pi k_x n}{N}}$, where $k_x \in \mathbb{R}$

$$X_w(k) = \sum_{n=0}^{N-1} w(n)x(n)e^{-j\frac{2\pi kn}{N}}\Big|_{k=0,1,..,N-1} = \sum_{n=0}^{N-1} w(n)Ae^{j\frac{2\pi k_x n}{N}}e^{-j\frac{2\pi kn}{N}} = A\sum_{n=0}^{N-1} w(n)e^{-j\frac{2\pi(k-k_x)n}{N}} = AW(k-k_x)$$

[3] Let's consider the case of the rectangular window $w_R(n)=1\ \forall n\in[0,L-1]$. Its N-point DFT is written as $W_R(k) = \sum_{n=0}^{N-1} w(n)e^{-j\frac{2\pi kn}{N}} = \sum_{n=0}^{L-1} e^{-j\frac{2\pi kn}{N}} = \frac{1-e^{-j\frac{2\pi kL}{N}}}{1-e^{-j\frac{2\pi k}{N}}} = e^{-j\frac{2\pi k}{N}\frac{L-1}{2}}\frac{\sin\left(\frac{\pi kL}{N}\right)}{\sin\left(\frac{\pi k}{N}\right)}$ and its module $|W_R(k)| = \frac{\sin\left(\frac{\pi kL}{N}\right)}{\sin\left(\frac{\pi k}{N}\right)}$.

The first zeros of $|W_R(k)|$ are located at $k=\pm N/L$ and therefore the bandwidth of the main-lobe is $B=2N/L$.

good frequency but poor temporal resolution; alternatively, short windows imply poor frequency but good temporal resolution.

- Close frequency components

  In the case of polyphony, it happens that harmonic components of different sounds often have close frequencies that force the main lobes of the analysis window to overlap. We find then a similar effect in the frequency domain to the beating effect in the time domain. Depending on the relative relationship of the sinusoidal parameters, the overlapping creates in the spectrum a predominant peak with time-varying amplitude or two peaks. This effect is illustrated in Figure 2.8.

- Non-stationary sinusoids

  In singing or speech spectra we expect to see trains of analysis window transforms located at the harmonic frequencies. However, along voiced utterances neither harmonic amplitudes nor frequencies are constant, not just during vibratos or note transitions, but even in the case of sustained notes. Such amplitude and frequency variations affect the shape of the spectrum around each harmonic frequency in different ways, distorting the shape of the window transform. Figure 2.9 and Figure 2.12 show such distortions in the case of linear amplitude and frequency modulations in the form $s(t) = (a_0 + a_1 t)\cos\left(2\pi(f_0 + f_1 t)t\right)$, with $a_0 = 1$ and $f_0 = 200$ Hz. On the other hand, Figure 2.10 shows the spectrum of a more general case $s(t) = (a_0 + a_1 t)^{p_a} \cos\left(2\pi f_0 (1 + f_1 t)^{p_f} t\right)$ for different values of p_a and p_f. Observing them we can conclude that if $a_1 = 0$ or $f_1 = 0$ the shape of the amplitude spectrum is even symmetric with respect to the frequency of the sinusoid at the center of the analysis window (i.e. f_0). This is not true, however, if both a_1 and f_1 are different than 0, or in the case of non linear amplitude and frequency variations in the form $s(t) = \left(a_0 + a_1 |t|^{p_a}\right)\cos\left(2\pi\left(f_0 + f_1 |t|^{p_f}\right)t\right)$ as the ones in Figure 2.11. Table 2.1 shows the most relevant relationships between spectral shapes and sinusoidal amplitude and frequency functions.

- Sinusoids close to frequency boundaries

  The STFT of a stationary sinusoid has window transforms located at $f_0 + g f_s$ and $-f_0 + g f_s$, $\forall g \in \mathbb{Z}$. If the frequency is very low, close to the 0 Hz boundary, then the two window transforms at f_0 and $-f_0$ probably overlap and distort the shape of the spectrum, so that even the spectral peak could be missing around the frequency of the sinusoid. The same happens for high frequencies close to $f_s/2$, where two window transforms are placed at f_0 and $f_s - f_0$. This effect is similar to that one of *close frequency components*, with the difference that in this case the two frequency components have exactly the same amplitude and opposite phases. In addition, estimation errors are often more relevant at low than at high frequencies because energy usually decays along frequency in musical signals.

- Noise

  Another issue that degrades the accuracy of sinusoid estimators is the presence of noise. In voice utterances we typically perceive the presence of harmonics together with aspirated noise. Besides, during a recording other types of noise are added to the audio, such as ambient noise, room reverberation or noise coming from the sensor devices and circuitry. All these types of noises degrade the estimator performance, especially when their energy is similar or higher than the harmonic one.

- Signals with limited time-domain support

  Another aspect to consider is the case of sinusoids that partially exist in the analysis window, i.e. that start or end within the analysis window. More generally, we could speak of attacks releases, or transients. This case could be considered as a particular case of non-stationary sinusoids, as a sinusoid with an amplitude modulation so deep that gets almost to zero. The main effect in frequency domain is that the spectral peaks increase their wideness significantly, so that close partials overlap.

Nevertheless, for voiced utterances we generally expect to see clear spectral peaks close to each harmonic frequency. An example of a male voice excerpt is shown in Figure 2.13. In consequence, we can estimate the harmonic parameters (i.e. amplitude, frequency and phase) in frequency domain by identifying the spectral peaks and deriving their amplitude, frequency and phase values. Many spectral models dealing with harmonics rely on this assumption, and therefore several techniques have come out for estimating those parameters with high accuracy. In the following we detail the most common and relevant ones. A comprehensive comparison of most of these methods in the presence of stationary sinusoids is found in (Keiler and Marchand 2002).

- Quadratic interpolation

  This is probably the most common technique due to its simplicity. A spectral peak is defined as a local maximum in the magnitude spectrum. Due to the sampled nature of the spectrum, each peak is accurate only to within half a sample. A spectral sample represents a frequency interval of f_S / N Hz, where f_s is the sampling rate and N the size of the FFT. Zero-padding in the time domain increases the number of spectral samples per Hz and therefore the accuracy of the peak estimation. However, as pointed out in (Serra 1989), a zero-padding factor of 1000 is required to obtain frequency accuracy on the level of 0.1 percent of the distance from the top of an ideal peak to its first zero crossing (in the case of a Rectangular window). A more efficient way to increase the frequency accuracy is to zero-pad enough so that a quadratic interpolation of the log amplitude spectrum refines the estimation to a 0.1 percent frequency accuracy, using only the bins surrounding the spectral peak. Since, as we have seen, the amplitude shape is mostly symmetric with respect to the sinusoid frequency at the analysis window center, this approximation seems to be a good choice. Frequency and magnitude parameters of the sinusoid are estimated as the ones of the maximum of the parabola whereas phase is calculated by interpolating the unwrapped phase 2nd order polynomial function at the estimated frequency. Thus, for each spectral peak located at bin k_p, let's define $a = X_w^{dB}(k_p - 1)$, $b = X_w^{dB}(k_p)$ and $c = X_w^{dB}(k_p + 1)$. The estimated sinusoid frequency in bins is $\hat{k}_p = k_p + (a-c)/2(a-2b+c)$, and the estimated amplitude $\hat{a}_p = b - (a-c)^2/8(a-2b+c)$. Nevertheless, this is not an ideal solution, since deviations in the order of 20 dB in amplitude and 1 radian in phase can be found even in the case of simple linear frequency modulations (see Figure 2.9).

- Derivative algorithm

  (Desainte-Catherine and Marchand 2002, Marchand 1998, Desainte-Catherine and Marchand 1998) improved the Fourier analysis precision using a k^{th}-order Fourier transform. Considering a sinusoidal model, the input signal can be expressed as

$$s(t) = \sum_{h=0}^{H-1} a_h(t)\cos\left(\theta_h(t)\right). \tag{2.4}$$

  The relationship between frequency and phase is given by

$$\frac{\partial \theta_h}{\partial t} = 2\pi f_h(t) \qquad \theta_h(t) = \theta_h(0) + 2\pi \int_0^t f_h(\tau)\, d\tau \tag{2.5}$$

  where t is the time in seconds, h is the partial index, H the number of partials, f_h, a_h and θ_h respectively the frequency, amplitude and phase of the h^{th} partial. Assuming that frequency and amplitude are slow time-varying parameters, we can suppose that frequency and amplitude derivatives are close to zero for a single analysis window of the STFT. In other words, a sinusoid derivative is another sinusoid with a different phase but the same frequency.

$$\frac{\partial s}{\partial t}(t) = \sum_{h=1}^{H-1} 2\pi f_h(t) a_h(t) \cos\left(\theta_h(t) - \frac{\pi}{2}\right) \tag{2.6}$$

  Let DFT^p be the amplitude spectrum of the Discrete Fourier Transform of the p^{th} signal derivative

$$DFT^p(k) = \frac{1}{N}\left|\sum_{n=0}^{N-1} w(n) \frac{\partial^p s}{\partial t^p}(n) e^{-j\frac{2\pi kn}{N}}\right| \tag{2.7}$$

where w is the N-point analysis window. For each partial h we expect to find a maximum in both DFT^0 and DFT^1 spectra for a certain index k_h. Approximate frequency and amplitude values for the partial p^{th} can be computed from DFT^0 by

$$f_h^0 = k_h \frac{f_s}{N} \qquad a_h^0 = DFT^0(k_h) \tag{2.8}$$

where f_s is the sampling frequency. From equation (2.7) we get more accurate frequency and amplitude values by

$$f_h^1 = \frac{1}{2\pi} \frac{DFT^1(k_h)}{DFT^0(k_h)} \qquad a_h^1 = \frac{a_h^0}{W\left(\left|f_h^1 - f_h^0\right|\right)} \tag{2.9}$$

where $W(f)$ is the amplitude of the continuous spectrum of the analysis window w at frequency f. The window should be chosen as short as possible, with the only restriction that partials must lie in two different Fourier transform bins. Thus, with the DFT^1 method the window can be smaller than with the standard STFT, and a better time-resolution can be achieved, especially important for instance when dealing with vibratos.

❖ Triangle algorithm

Althoff et al (Althoff, et al. 1999) proposed a method to improve the estimation of sinusoid parameters by using an analysis window function with a triangular transform. The absolute value of the desired zero-phase frequency response would be

$$A(e^{j\Omega}) = \begin{cases} 1 - \left|\dfrac{\Omega}{\Omega_c}\right| & , \quad |\Omega| < \Omega_c \\ 0 & , \quad \text{otherwise} \end{cases} \tag{2.10}$$

for $|\Omega| < \pi$. Figure 2.5 shows $A(e^{j\Omega})$ and the corresponding causal window. Frequency resolution is higher for small values of Ω_c while noisy sinusoids can be better detected with greater values of Ω_c. A good compromise is to choose $\Omega_c = 8\pi / N$, which results into a triangle length of 8 bins without zero-padding. If D is the length of the triangle, then it can be described by two lines, $h_1(k)$ on the left and $h_2(k)$ on the right side,

$$h_1(k) = ak + b \qquad h_2(k) = -a(k - D) + b \tag{2.11}$$

with $a > 0$. If k_m is a local maximum, the six closest spectral values surrounding $X_m(k_m)$ are used to calculate the parameters a and b by minimizing the squared error. And finally the peak is determined by $k_0 = 2 - b/a$. The authors compared this method with the previous algorithm based on signal derivatives. They found out that the triangle algorithm performs better in noisy situations while the derivative algorithm is superior at low noise levels.

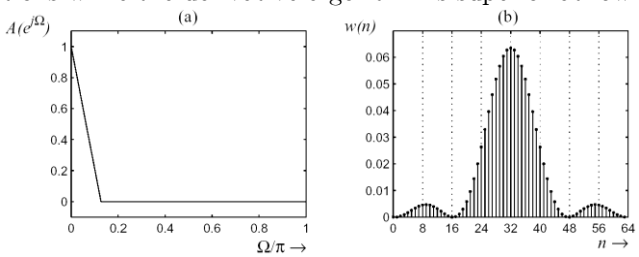


Figure 2.5 Triangular amplitude response and corresponding causal window for N=64. Illustration from (Althoff, et al. 1999)

❖ Time-frequency reassignment

In common time-frequency representations, the signal is decomposed in time-frequency atoms that are assigned to the geometrical center of the cells, i.e. the bins of the Fourier transform and the center of the analysis window. In this approach, instead, the sinusoidal estimations are non-uniformly distributed in both time and frequency, according to the center of gravity of the cell's energy. The reassigned frequency $\hat{f}_k$ corresponding to $k^{\hat{th}}$ bin is obtained by

$$\hat{f}_k = k\frac{f_s}{N} - \Im\left\{\frac{X_{dw}(k)X_w^*(k)}{\left|X_w(k)\right|^2}\right\}\cdot\frac{f_s}{2\pi} \tag{2.12}$$

where X_w and X_{dw} are the discrete-time STFT of $x(n)$ using the window $w(n)$ and its derivative $dw \triangleq \partial w(n)/\partial n$ respectively. The reassigned time $\hat{t}_m$ corresponding to the center time t_m of the m^{th} frame analysis window is computed by

$$\hat{t}_m = t_m + \Re\left\{\frac{X_{dw}(k)X_w^*(k)}{\left|X_w(k)\right|^2}\right\}. \tag{2.13}$$

Therefore, for each local maximum in X_w, the sinusoidal parameters are obtained by reassignment using the previous equations. This method was proposed in (Auger and Flandrin 1995), and has been used by several authors (e.g. (Fitz 1999, Peeters 2001)).

❖ FFT with phase vocoder approach

Another common method is based on computing the frequency estimation of local maxima from the phase difference between two STFT of the analyzed signal $x(n)$ using the same window but delayed by R samples (Arfib, et al. 2002, p.337). The instantaneous frequency of a sinusoid is given by the derivative of the phase after the time. Thus, for a local maximum located at bin k, the estimated frequency is obtained by

$$\hat{f}_k = \frac{f_s}{2\pi}\cdot\frac{\text{princarg}\left\{\measuredangle X_w^{m+R}(k) - \measuredangle X_w^{m}(k)\right\}}{R} \tag{2.14}$$

where X_w^{m+R} and X_w^m are the STFT of $x(n)$ centered at samples $m+R$ and m respectively. Some refinements have been recently added that allow estimating amplitude, frequency and phase parameters of the sinusoids with great accuracy, even for low-frequency tones where the interference from the negative frequencies is unavoidable (Garcia and Short 2006).

❖ Linear amplitude and frequency modulated sinusoids

Several authors have explored the estimation of sinusoidal parameters for non-stationary sinusoids with linear amplitude and frequency modulation. Let us consider the complex discrete time sinusoid

$$s(n) = (a_0 + a_1 n)e^{j\left(\theta_0 + 2\pi\omega_0 n + \pi D n^2\right)}. \tag{2.15}$$

The DFT of the $s(n)$ when using a window $w(n)$ is

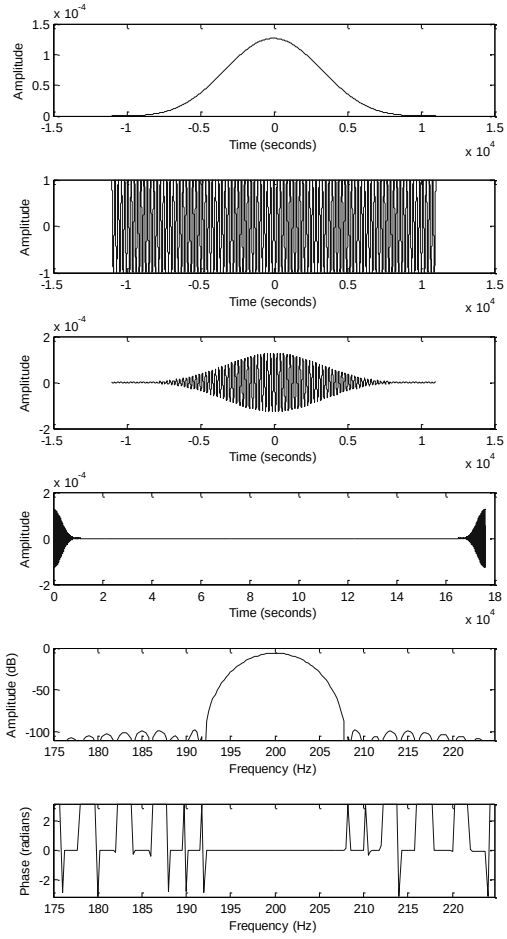
$$S(\omega) = \sum_{n=-\infty}^{\infty} w(n)(a_0 + a_1 n)e^{j\left(\theta_0 + 2\pi\omega_0 n + \pi D n^2\right)}e^{-j2\pi\omega n}. \tag{2.16}$$

Assuming an even symmetric window and removing the parts of the sum that are odd symmetric in n, the previous equation can be rewritten as

$$\begin{aligned} S(\omega) &= S_1(\omega) + S_2(\omega) \\ S_1(\omega) &= a_0 e^{j\theta_0}\sum_{n=-\infty}^{\infty} w(n)\cos\left(2\pi(\omega_0 - \omega)n\right)e^{j\pi D n^2} \\ S_2(\omega) &= a_1 e^{j\theta_0}\sum_{n=-\infty}^{\infty} w(n)nj\sin\left(2\pi(\omega_0 - \omega)n\right)e^{j\pi D n^2}. \end{aligned} \tag{2.17}$$

According to (Röbel 2007) only the frequency modulation creates additional estimation bias for the standard sinusoidal parameter estimators. This means that for example in the case of vibratos the standard methods would systematically fail to accurately estimate the harmonic parameters. One of the proposed methods (Abatzoglou 1986) is based on time domain signal demodulation employing an initial search over a grid of frequencies and frequency slopes, for afterwards refining the parameters using an iterative maximization of the amplitude of the demodulated signal. Other methods model the complex spectrum using Gaussian analysis windows so that mathematic equations become tractable (e.g. (Peeters 2001). In (Abe and

Smith 2005) a set of linear bias correction functions are proposed to extend the method to other windows. Recently (Röbel 2007) proposed a general method for any analysis window and shows how it greatly reduces the energy of the residual [4] signal compared to the quadratic interpolation method.



$w(n)$ = Blackman-Harris window

$s(n) = \cos\left(2\pi \frac{200}{f_s} n\right)$
stationary sinusoid of amplitude 1, frequency 200 Hz, phase 0 radians at $n=0$

$x(n) = w(n) \cdot s(n)$ windowed signal

windowed signal zero-padded by a factor of 8 and centered at n=0

$20\log_{10}\left(|X(\omega)|\right)$ amplitude spectrum in dB

$\measuredangle\left(X(\omega)\right)$ phase spectrum

Figure 2.6 Short-time Fourier Transform (STFT) of a stationary sinusoid with parameters: amplitude=1, frequency=200Hz, phase 0 at $n=0$. The sinusoid is multiplied by a Blackman-Harris 92 dB window, zero-padded by a factor of 8, and centered at the first sample of the FFT circular buffer. The two bottom views show the amplitude and phase spectrum of the computed STFT. The maximum of the spectrum is found as expected at 200Hz, being its amplitude -6dB (i.e. $20\log_{10}(0.5)$). The phase under the main lobe is constant and equal to 0 radians. The side lobes are found at -92dB from the maximum value.

[4] obtained by subtracting the estimated linearly modulated sinusoids from the input signal

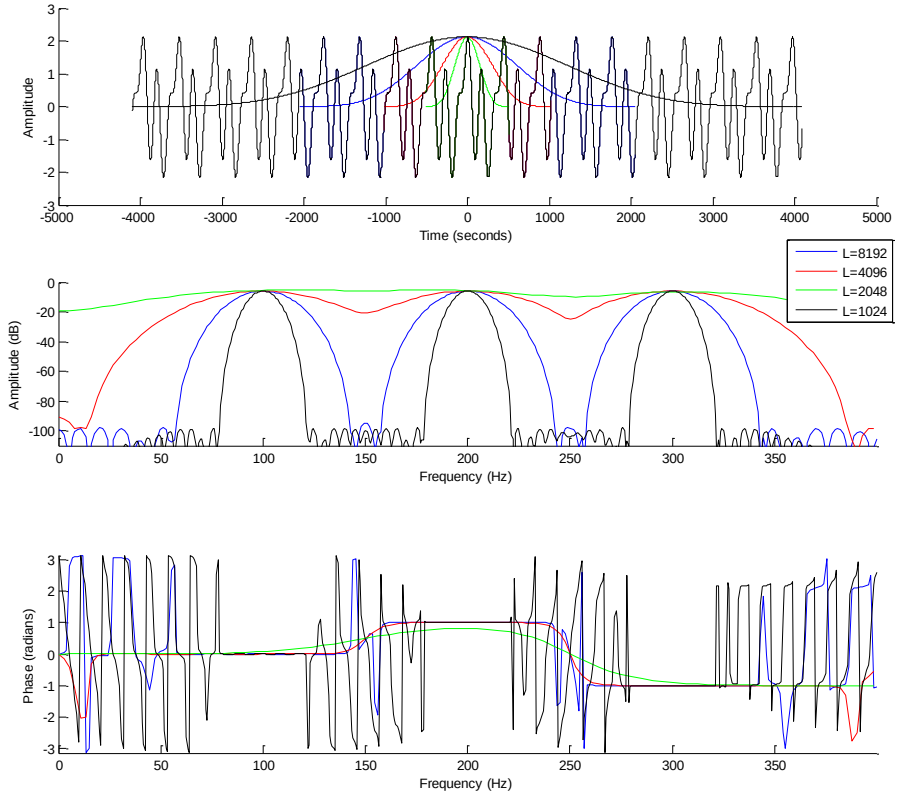


Figure 2.7 Long windows provide high frequency resolution and harmonics are distinguishable. Conversely, short windows make the main lobes of the window transforms overlap and harmonic peaks are not visible anymore. The analysis window in this example is a Blackman-Harris 92dB.

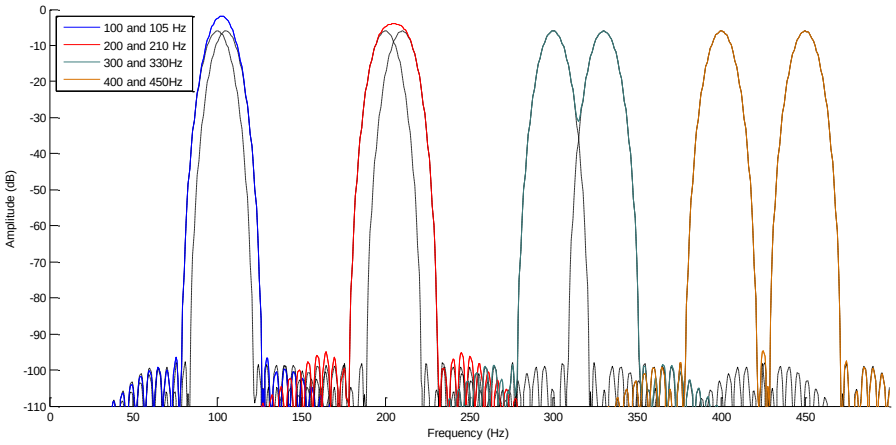


Figure 2.8 When sinusoidal components are close in frequency, the main lobes of the window transforms overlap and appear as a single component. The analysis window is a Blackman-Harris 92dB.

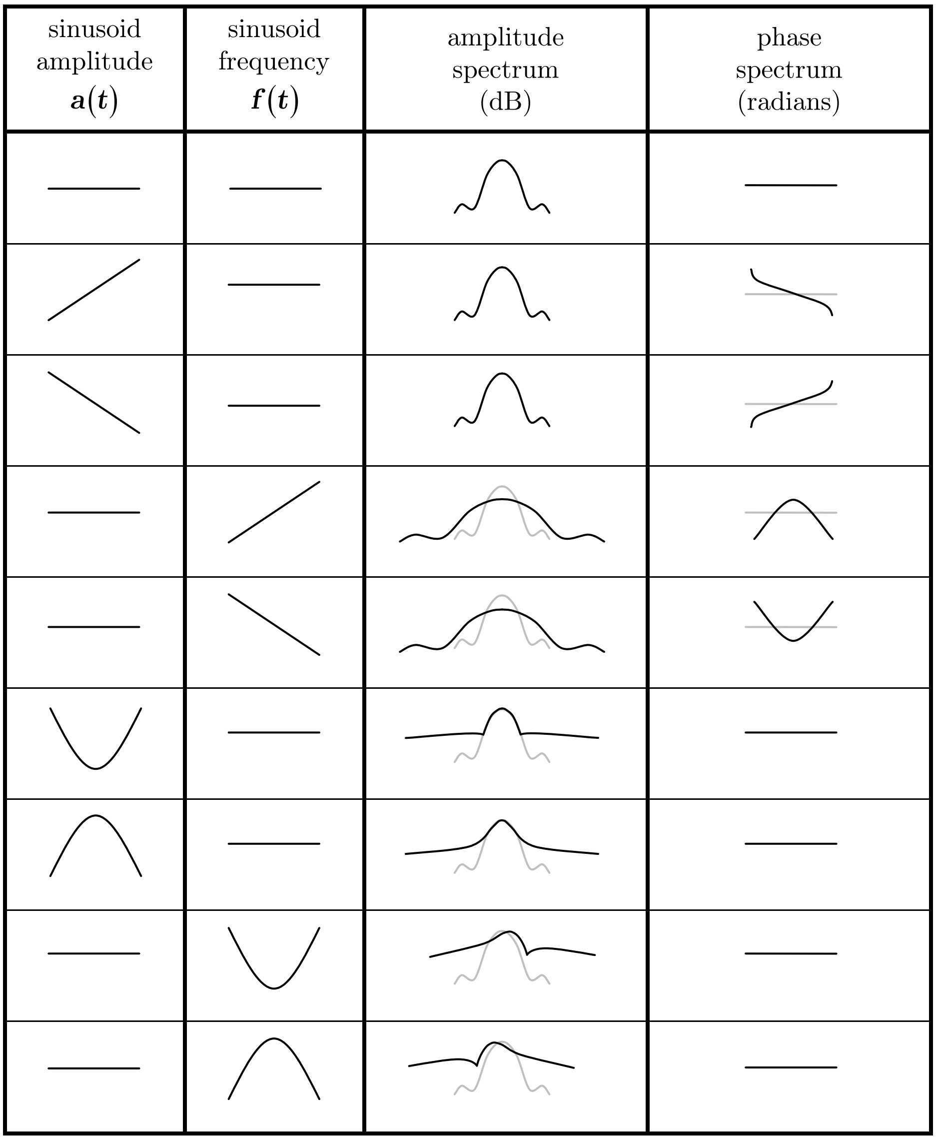


Table 2.1 Relationship between spectral shape and most relevant first and second order sinusoidal amplitude and frequency functions.

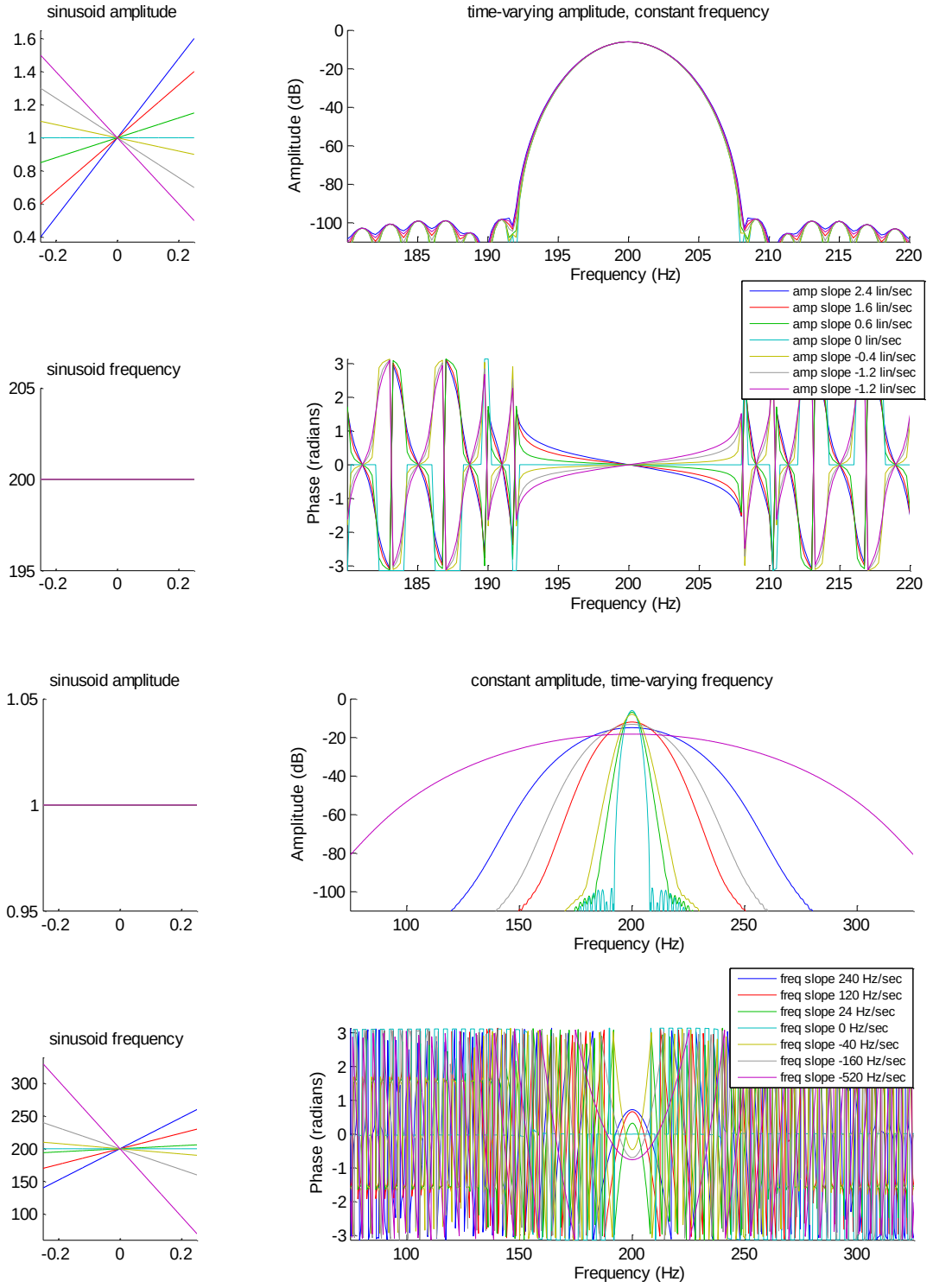


Figure 2.9 Short-time Fourier Transforms (STFT) of non-stationary sinusoids with linearly time varying frequency and amplitude parameters, and phase 0 radians at time 0 seconds: $s(t) = (a_0 + a_1 t)\cos(2\pi(f_0 + f_1 t)t)$. The frequency of the sinusoid at the center of the window is 200Hz in all cases. The window is a 0.5 seconds long Blackman-Harris 92dB. Each color is associated to the amplitude and frequency slopes indicated in the respective legends. Both amplitude and phase shapes greatly differ from the window transform. Amplitude variations affect mostly the phase shape whereas frequency modulations affect both amplitude and phase shapes. In the latter case the amplitude can decrease up to 20 dB, and the phase shift by up to 1 radian.

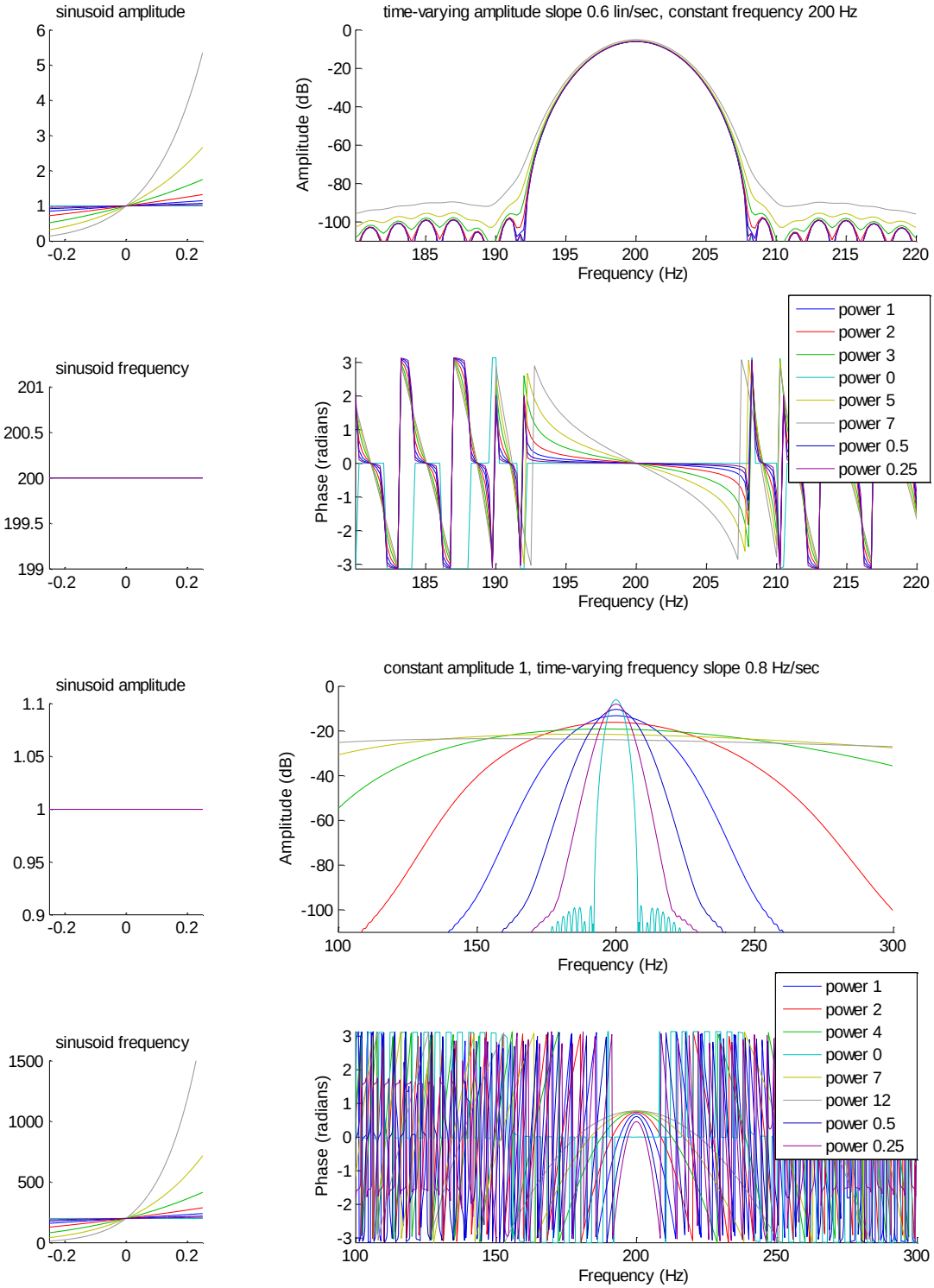


Figure 2.10 Short-time Fourier Transforms (STFT) of non-stationary sinusoids in the form $s(t) = (a_0 + a_1 t)^{p_a} \cos\left(2\pi f_0 (1 + f_1 t)^{p_f} t\right)$. The frequency of the sinusoid at the center of the window is 200Hz in all cases. The window is a 0.5 seconds long Blackman-Harris 92dB. Each color is associated to the power factors p_a and p_f indicated in the respective legends. Both amplitude and phase shapes greatly differ from the window transform. Amplitude variations affect mostly the phase shape whereas frequency modulations affect both amplitude and phase shapes.

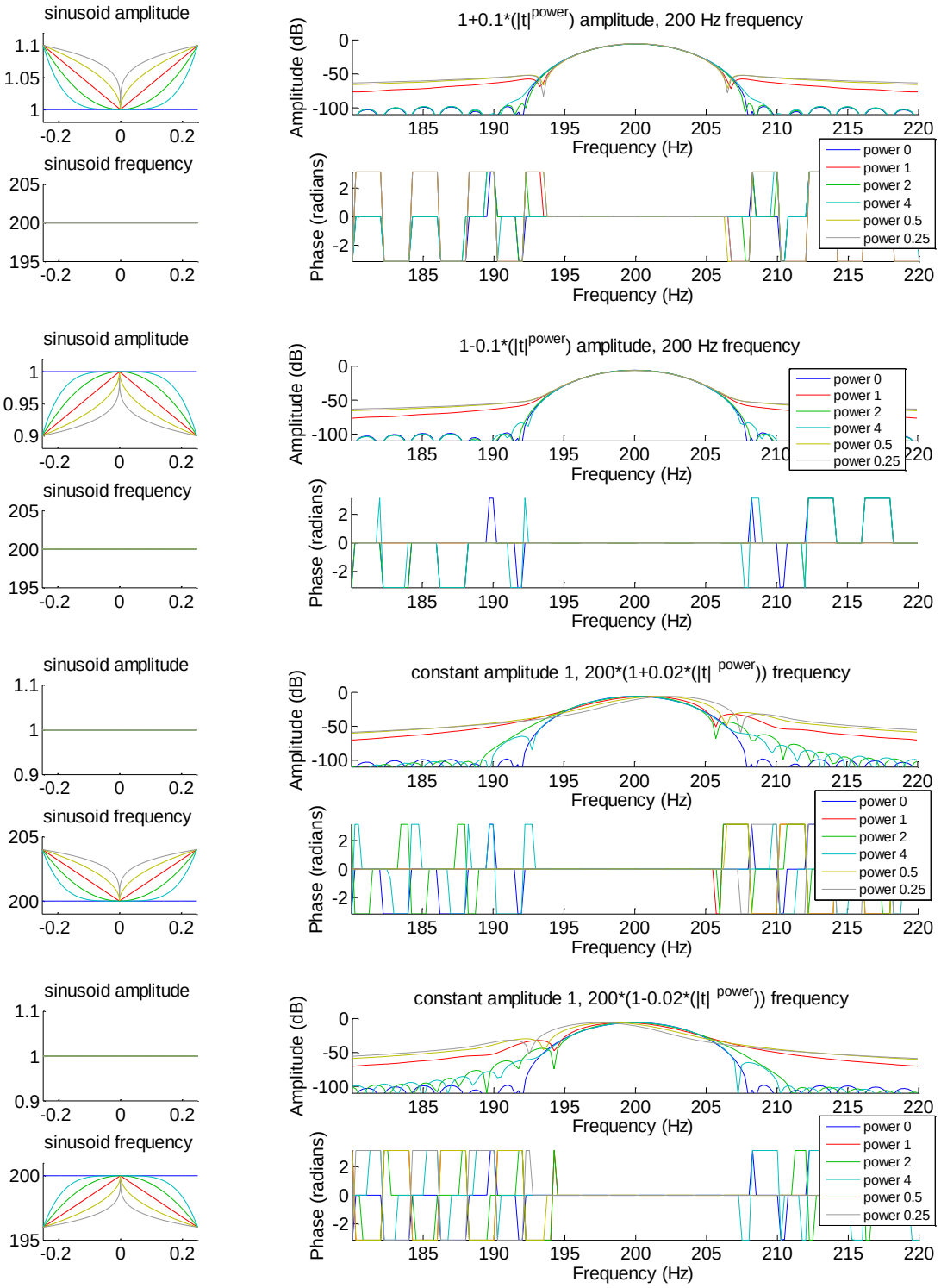


Figure 2.11 Short-time Fourier Transforms (STFT) of non-stationary sinusoids in the form $s(t) = \left(a_0 + a_1 |t|^{p_a}\right) \cos\left(2\pi \left(f_0 + f_1 |t|^{p_f}\right) t\right)$. The frequency of the sinusoid at the center of the window is 200Hz in all cases. The window is a 0.5 seconds long Blackman-Harris 92dB. Each color is associated to the power factors p_a and p_f indicated in the respective legends. Both amplitude and phase shapes greatly differ from the window transform. Frequency modulations greatly affect the amplitude shape so that the spectral peak frequency shifts. This would be the case for example of vibrato utterances around fundamental frequency maxima and minima, where the fundamental can be approximated with a second order polynomial.

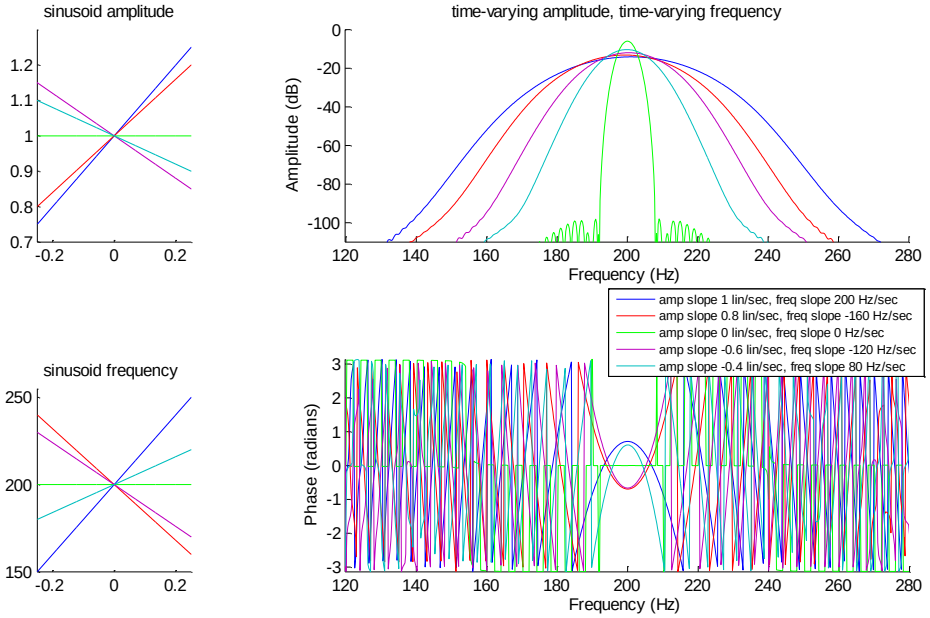


Figure 2.12 Short-time Fourier Transforms (STFT) of non-stationary sinusoids with simultaneous variations of frequency and amplitude parameters, $s(t) = (a_0 + a_1 t)\cos(2\pi(f_0 + f_1 t)t)$. The window used is a Blackman-Harris 92dB with a duration of 0.5 seconds. Each color is associated to the amplitude a_1 and frequency f_1 slopes indicated in the legend. Both amplitude and phase shapes greatly differ from the window transform. Phase is still even with respect to f_0, but the amplitude peak moves away from it.

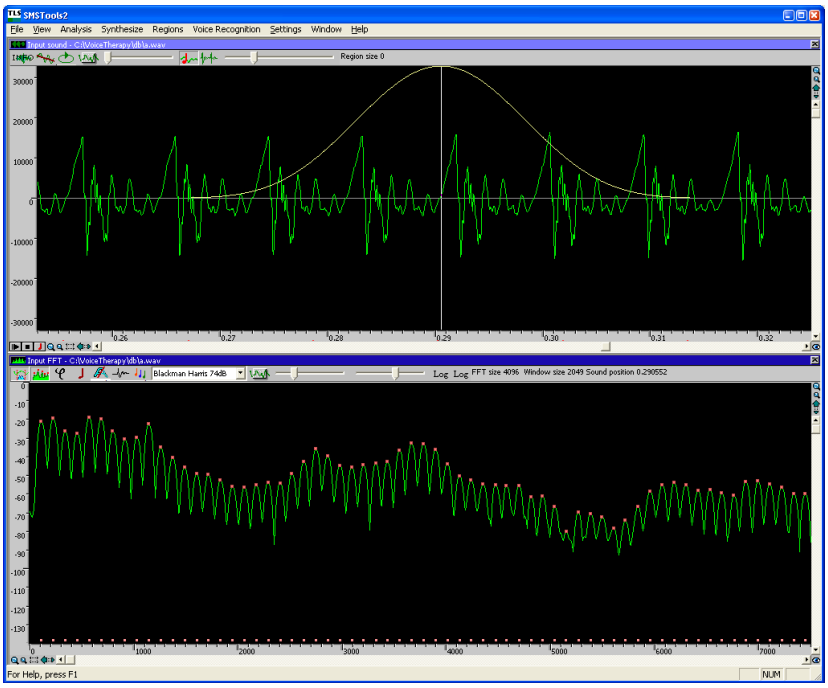


Figure 2.13 The top view represents the waveform of a voice utterance of a sung /a/ vowel by a male singer, whereas the bottom view presents the amplitude spectrum (in dB) of a windowed portion of the same utterance. The window used is a Blackman Harris 74 dB, drawn in yellow color, and covers over 5 periods of the audio signal. The harmonics are marked with a red dot and clearly correspond to spectral peaks.

2.2 Harmonic Trajectories (sinusoidal models)

We have seen that voice utterances can be interpreted as a set of time-varying quasi-sinusoidal signals corresponding to harmonics, parameterized by its time-varying values of frequency, amplitude and phase. The voice signal $s(n)$ can be consequently modeled by a set of sinusoids (i.e. harmonics) as

$$s(t) = \sum_{h=0}^{H-1} a_h(t)\cos\left(\phi_h(t)\right). \tag{2.18}$$

Using the instantaneous frequency definition $f(t) = \frac{1}{2\pi}\frac{\partial\theta(t)}{\partial t}$, we can rewrite the previous equation as

$$s(t) = \sum_{h=0}^{H-1} a_h(t)\cos\left(2\pi\int_0^t f_h(\tau)\,d\tau + \phi_{0,h}\right) \tag{2.19}$$

and its discrete domain version as

$$s(n) = \sum_{h=0}^{H-1} a_h(nT_s)\cos\left(2\pi\sum_{k=0}^{n-1} f_h(kT_s)T_s + \phi_{0,h}\right). \tag{2.20}$$

Typically, analysis tools estimate those parameters applying a sliding square window $w_{square}(n)$ of length $L+1$ at equidistant time instants $t_m = m\Delta_t$. The obtained segments are often called *frames*.

$$\begin{aligned} s_m(n) &= s\left(n + m\frac{\Delta_t}{T_s}\right)\cdot w_{square}(n) \\ w_{square}(n) &= \begin{cases} 1 & if \quad |n| \le L/2 \\ 0 & elsewhere \end{cases} \end{aligned} \tag{2.21}$$

If the sinusoidal components are supposed to be stationary along the m^{th} time window (i.e. frame), equation (2.20) can be rewritten[5] as

$$s_m(n) = \sum_{h=0}^{H-1} a_{h,m}\cos\left(2\pi f_{h,m} nT_s + \phi_{0,h,m}\right) \tag{2.22}$$

and the original signal $s(n)$ can be approximated by overlapping all the intermediate signals $s_m(n)$ with an appropriate[6] overlapping window $w_{ov}(n)$.

$$s(n) \approx \sum_{m=0}^{M-1} s_m\left(n - m\frac{\Delta_t}{T_s}\right)\cdot w_{ov}\left(n - m\frac{\Delta_t}{T_s}\right) \tag{2.23}$$

Since the sinusoidal components have been estimated independently at diverse discrete times, there is a need to connect the various detected harmonics at consecutive time instants, building up a set of continuous[7] trajectories for each parameter. This is what we call *harmonic trajectories*[8]. Figure 2.14

[5] Assuming frequency to be constant, we get $\sum_{k=0}^{n-1} f_h(kT_s)T_s = \sum_{k=0}^{n-1} f_{h,m}T_s = f_{h,m}nT_s$

[6] The condition that $w_{ov}(n)$ of length $L+1$ must fulfill is that overlapped with itself each Δ_t/T_s samples, it must add to constant 1, so $\sum_{m=0}^{M-1} w_{ov}\left(n - m\Delta_t/T_s\right) = 1$, $\forall n \in \left[-L/2, (M-1)\Delta_t/T_s + L/2\right]$

[7] Here *continuous* means a value for each sample n instead of a value for each frame m. Even if that *continuous* parameter function is not explicitly computed, there is always an underlying continuous model such as for instance linear frequency assumption when propagating phases of harmonics in consecutive frames, or amplitude interpolation of consecutive harmonics.

[8] Also often referred to as partial tracks.

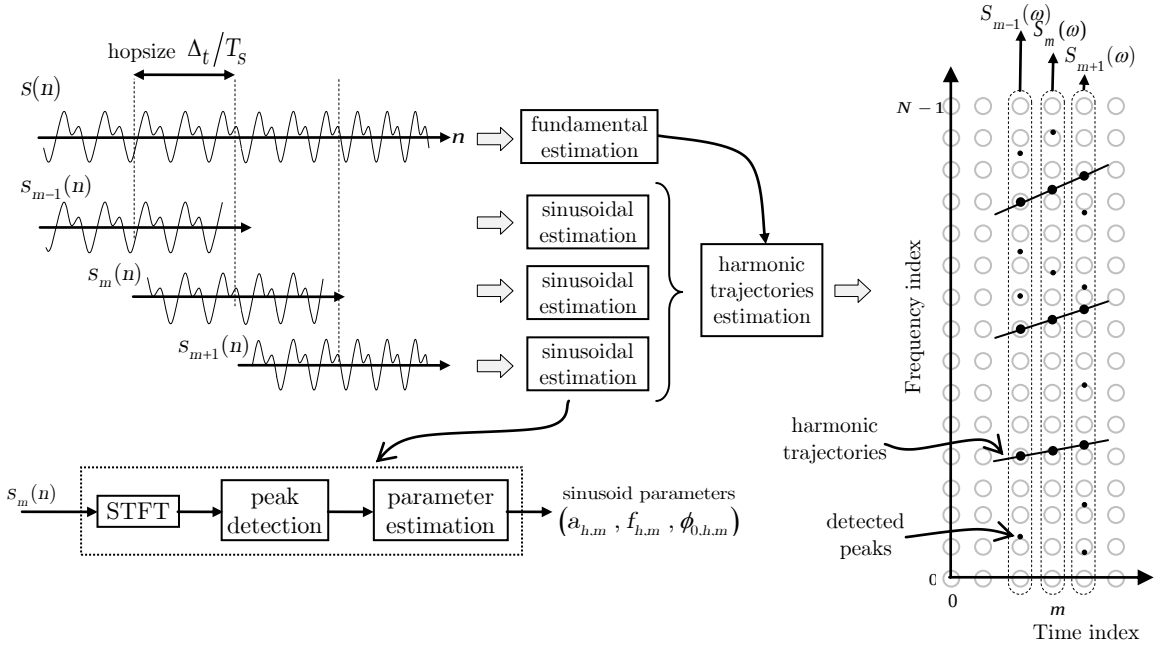


Figure 2.14 Harmonic trajectory estimation. The input signal $s(n)$ is segmented into overlapping excerpts $s_m(n)$. The sinusoidal components are estimated for each segment and afterwards combined into the harmonic trajectories. A fundamental estimation algorithm is often used to help the trajectories estimation. On the right side, there is the discrete time-frequency plane with the estimated trajectories.

shows a representation of a typical framework for estimating harmonic trajectories. The input signal $s(n)$ (top-left) is segmented into a sequence of frames $s_m(n)$ centered at equidistant time indices $m\Delta_t / T_s$. For each of the frames $s_m(n)$, a *sinusoidal estimation* module detects the sinusoidal peaks and estimates their parameters. This is often done by first computing the STFT using an analysis window with good frequency characteristics, then detecting the local maxima and finally estimating the sinusoidal parameters. Any of the estimators reported in the previous section can be applied. On the other hand, given that we are analyzing voice signals and we know in advance that there is a single voice, it makes sense to use a fundamental frequency estimator to help the estimation of harmonic trajectories. A typical way of doing so is to favor peak candidates and trajectories that fall close to multiples of the estimated fundamental at each frame. The fundamental estimation can be integrated into the *sinusoidal estimation* modules or be apart, running at different frame rate for example. Besides, some variants of the framework are also frequent; for instance, pitch synchronous frameworks where frames are not equidistant but separated by periods of the estimated fundamental frequency. Another example of an interesting variation would be the addition of a *peak classification* module to discriminate between sines, noise and sidelobes of the window transform. Figure 2.15 shows a representation of those trajectories for a singing utterance.

2.2.1 Trajectories estimation

Several strategies have been explored during the last decades with the aim of connecting the harmonic components in the best possible way. McAulay and Quatieri (1986) proposed a simple sinusoidal continuation algorithm based on finding, for each spectral peak, the closest one in frequency in the next frame. Serra (1989) added to the continuation algorithm a set of *frequency guides* used to create sinusoidal trajectories. The frequency guide values were obtained from the peak values and their context, such as surrounding peaks and fundamental frequency. In the case of harmonic sounds, these guides were initialized according to the harmonic series of the estimated fundamental frequency. The trajectories were computed by assigning to each guide the closest peak in frequency.

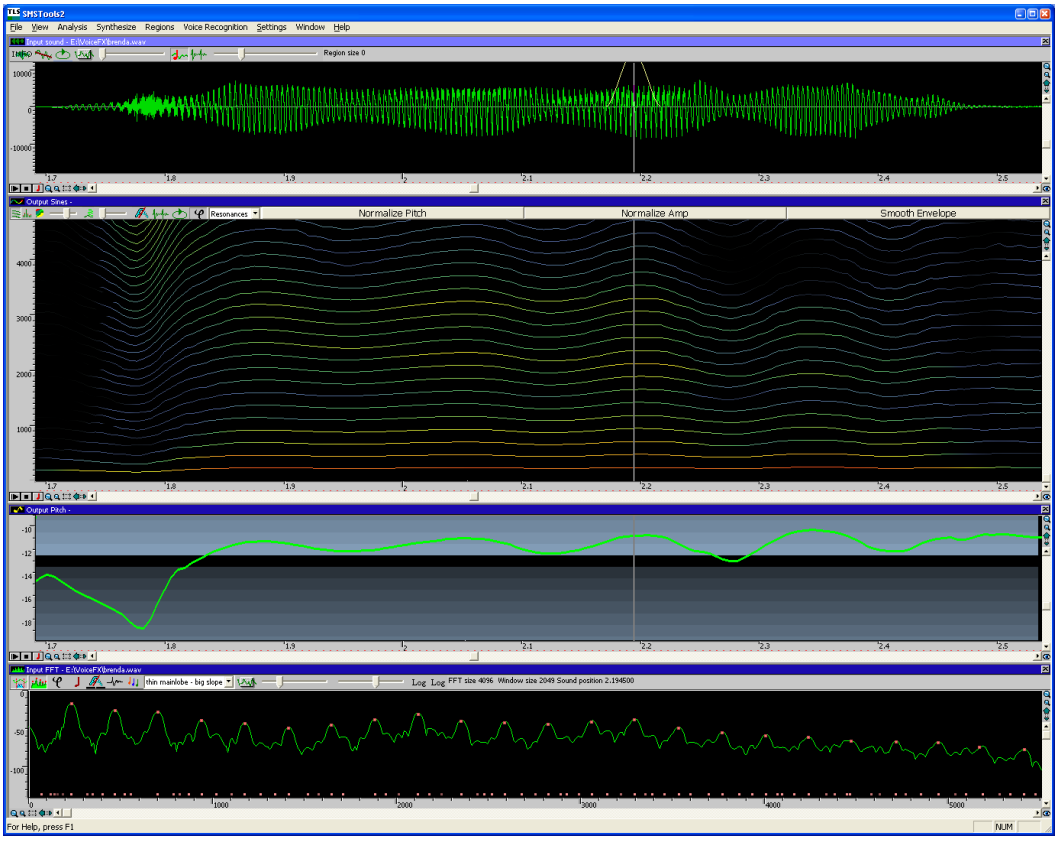


Figure 2.15 Harmonic trajectories representation. The top view represents the waveform of the word *gin* sung by a female singer. The second view shows the harmonic components in a frequency versus time representation, where the color is related to the amplitude. The third view corresponds to the fundamental frequency. Finally, the bottom view shows the amplitude spectrum at the time instant specified in the other views (the vertical line), being the harmonics marked with a red rectangle. The word was sung with vibrato; therefore a strong frequency modulation is appreciated in harmonic and fundamental frequency curves.

There are other continuation methods based on Hidden Markov models, which seem to be very valuable for tracking partials in polyphonic signals and complex inharmonic tones. For instance, García (1992) considers as *state* of the model pairs of parameter estimations at consecutive frames. If $\boldsymbol{\kappa}_{l,m}=\left(a_{l,m},f_{l,m},\theta_{0,l,m}\right)$ denotes the estimated sinusoid parameters of the l^{th} detected peak of the m^{th} frame, then a state would be the pair $\left[\boldsymbol{\kappa}_{l,m},\boldsymbol{\kappa}_{l',m+1}\right]$. The transition probabilities between states are then computed so to maximize the continuity of the frequency and amplitude differentiations. Thus, being Δ the difference between consecutive values,

$$prob_{trans}\left(\left[\boldsymbol{\kappa}_{l',m+1},\boldsymbol{\kappa}_{l'',m+2}\right]\middle|\left[\boldsymbol{\kappa}_{l,m},\boldsymbol{\kappa}_{l',m+1}\right]\right)=e^{-\frac{\left(\Delta f_{l',m+1}-\Delta f_{l,m}\right)^2}{\sigma_{\Delta f}^2}}e^{-\frac{\left(\Delta a_{l',m+1}-\Delta a_{l,m}\right)^2}{\sigma_{\Delta a}^2}}. \tag{2.24}$$

Peeters (2001) proposes to use a non-stationary sinusoid model, with linearly varying amplitude and frequencies: $a_{h,m}(t)=a_{0,h,m}+a_{1,h,m}t$ and $f_{h,m}(t)=f_{0,h,m}+f_{1,h,m}t$. Within that framework, the continuation algorithm focuses on the continuation of value and first derivative for both amplitude and frequency polynomial parameters, combined with a measure of sinusoidality. Observation and transitions probabilities are defined, and a Viterbi algorithm is proposed for computing the sinusoidal trajectories.

However, we are unaware of any trajectory estimator strongly relying on a proper classification of spectral peaks between sines, noise and sidelobes, such as the one proposed in (Zivanovic, et al. 2007). We believe this would help to reduce significantly both the number of candidates (therefore the computational cost) and the number of incorrect connections.

2.2.2 Trajectories transformation

Once the harmonic trajectories have been estimated, it is possible to modify them for generating interesting sound transformations. The fact that we have an independent control of each of the harmonics at each frame offers many possibilities such as temporal transformations (e.g. time-scaling), frequency transformations (e.g. transposition, frequency shifting, vibrato), and timbre transformations (e.g. formant-shifting). In this section we focus on the most relevant transformations using harmonic trajectories: time-scaling, frequency transposition and timbre scaling.

❖ TIME-SCALING

The time-scaling transformation $\bar{T}_{time}(t) \Rightarrow t \rightarrow t'$ is defined as the mapping function between input signal time t and output synthesis time t'. Let's denote T_{time} as the local derivative of $\bar{T}_{timbre}(t)$. When $\bar{T}_{time}$ is constant, it corresponds to the ratio between output and input signal durations, and it is often called time-scaling ratio.

In a pitch-synchronous framework the transformation modifies the interval between consecutive analysis frames so that $t'_{m+1} - t'_m = T_{time} \cdot (t_{m+1} - t_m)$. On the other hand, in a constant frame-rate framework, things are different and the transformation does not modify the hop size, so $t'_m = t_m$. Instead, the inverse of the mapping function is typically used so that the synthesis parameters κ' are computed out of the interpolation between the two closest input frames to the mapped time. The following Figure 2.16 illustrates this.

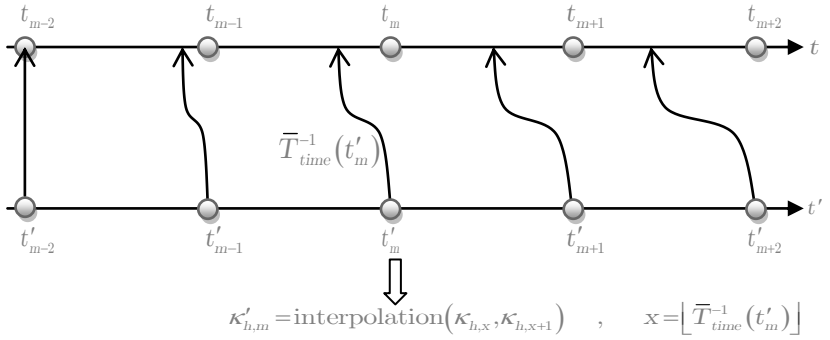


Figure 2.16 Time-scaling in a constant frame-rate framework ($t'_m = t_m$). For each output frame time t'_m, the sinusoid parameters are computed as the interpolation of the two frames around the mapped time $\bar{T}^{-1}_m(t'_m)$.

❖ TIMBRE SCALING

Here we use a simplified definition of timbre as the spectral amplitude envelope $H_{harm}(f)$ defined by the significant frequency components of the voice signal, i.e. harmonic amplitude envelope in voiced utterances and smooth spectral envelope in unvoiced sections. Following such definition, the timbre scaling transformation consists of modifying such amplitude envelope according to the mapping function $\bar{T}_{timbre}(f) \Rightarrow f \rightarrow f'$, which sets the scaling factor applied to each frequency position. Therefore,

$$H'_{harm}(f) = H_{harm}\left(\bar{T}^{-1}_{timbre}(f)\right). \tag{2.25}$$

The effect of a linear scaling $\bar{T}_{timbre}(f) = T_{timbre}$ is equivalent to a physical enlargement ($T_{timbre} > 1$) or reduction ($T_{timbre} < 1$) of the vocal tract length, and in terms of frequency domain to a stretching ($T_{timbre} > 1$) or compression ($T_{timbre} < 1$) of the spectral envelope. In the case of voiced utterances modeled with harmonics, this transformation applies uniquely to the harmonic amplitude as follows

$$a'_h = H'_{harm}(f_h) = H_{harm}\left(\bar{T}^{-1}_{timbre}(f_h)\right). \tag{2.26}$$

❖ PITCH TRANSPOSITION

Also often referred to as *pitch shifting*, this transformation formally corresponds to a linear scaling of the frequency axis by the transposition factor ($f'=T_{pitch}\cdot f$), and in terms of perception to a modification of the perceived pitch without altering other perceptually relevant features. When the transposition factor is time-varying, the function $T_{pitch}(t)$ determines the factor at each time. This would be the case, for instance, of adding vibrato to a sustained utterance by modulating the fundamental frequency.

The transformation cannot be simplified to modify only the harmonic frequency values, i.e. $f'_h=T_{pitch}\cdot f_h$, since then the timbre would be scaled by the same factor. In other words, we would get two transformations simultaneously: T_{pitch} and $T_{timbre}=T_{pitch}$. Therefore, the transposition transformation requires also the modification of harmonic amplitudes to correct this effect and preserve the timbre, which leads to

$$\begin{aligned} f'_h &= T_{pitch}\cdot f_h \\ a'_h &= H_{harm}\left(T_{pitch}\cdot f_h\right). \end{aligned} \tag{2.27}$$

The following figure illustrates these concepts. The top view shows the input signal, the middle one the transposition modifying only harmonic frequencies, and the bottom the same transposition but modifying both frequency and amplitude values.

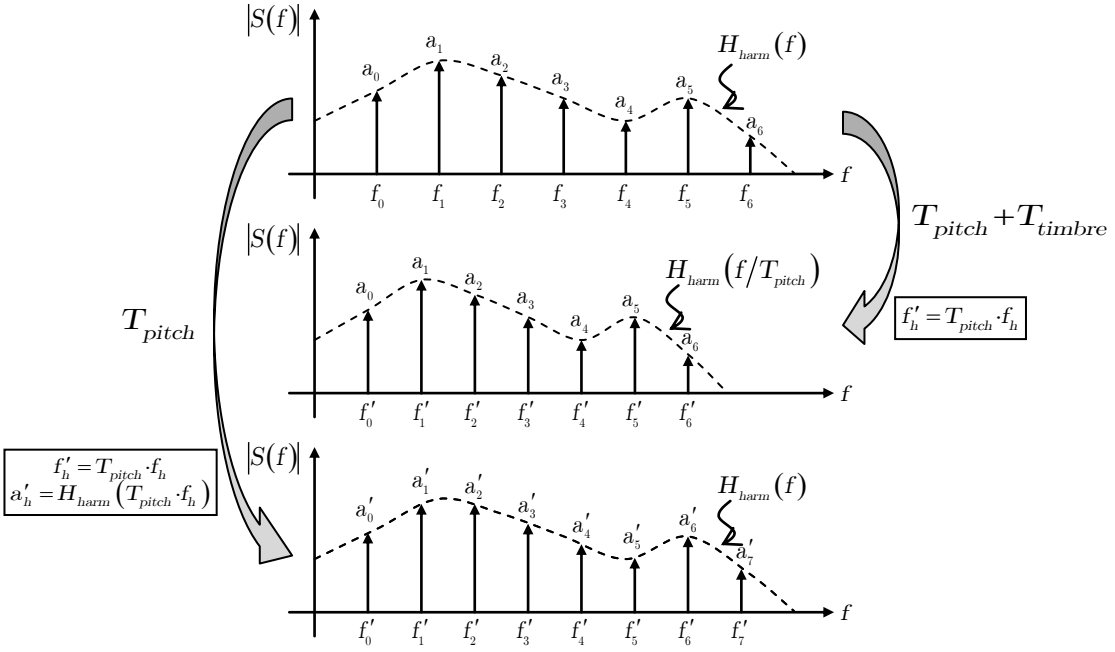


Figure 2.17 Pitch transposition. The signal whose transform is represented in the top view is transposed to a lower pitch. The middle view shows the result when only harmonic frequencies are modified, whereas in the bottom view representation both harmonic frequencies and amplitudes have been modified.

We have intentionally obviated any reference to harmonic phases in the previous descriptions, since phase requires some in-depth considerations that we will now tackle. Let us start with the case of an ideal sinusoid where phase is computed from the transformed frequency function as

$$\phi'(t) = \phi'(0) + 2\pi\int_0^t f'(\tau)d\tau\,. \tag{2.28}$$

However, when dealing with real signals, the estimation of the frequency parameter function might not be accurate enough and introduce significant bias in the integration computation. In addition, the analysis provides estimations of the harmonic parameters κ_m at certain time instants t_m, therefore a

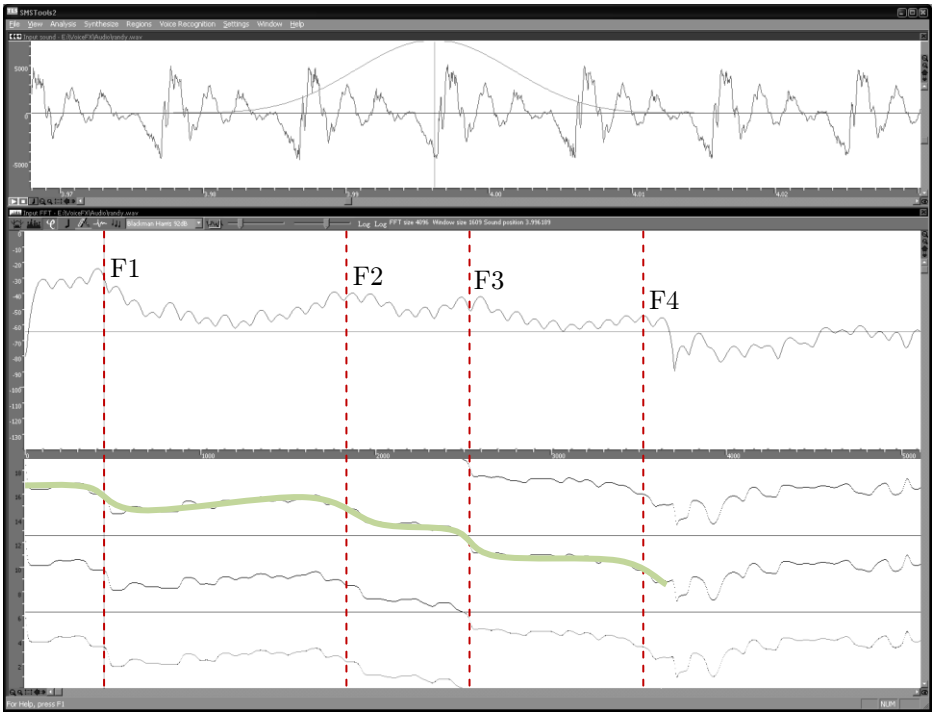


Figure 2.18 Formant to spectral-phase relation when the analysis window is centered at a voice pulse onset. The top view shows the waveform of a /e/ utterance by a male singer. The fundamental frequency is around 100Hz. The middle view corresponds to the magnitude spectrum (in dB). The bottom view shows the phase values in the vertical axis which covers three periods, i.e. $[0, 6\pi)$. The frequencies of the first four formants are marked with red dashed lines. Clearly, the phase appears to be mostly flat with phase shifts under each formant resonance.

certain underlying model of the harmonic frequency is required to obtain a continuous function. For instance, the simplest model is linear interpolation where phase is propagated by

$$\phi'_m(t) = \phi'_{m-1}(t_{m-1}) + 2\pi \frac{f_{m-1} + f_m}{2} \Delta_t \quad , \quad t_{m-1} \le t \le t_m \, . \tag{2.29}$$

This and more complex propagation methods successfully avoid discontinuities after transformations while at the same time are able to approximate quite well the behavior of the input sinusoid. While this might be acceptable for a single sinusoid, it is not the case when dealing independently with the simultaneous harmonics of a voiced utterance. Phase is perceptually significant and requires special attention for producing natural sounding transformations (e.g. see (Peeters 2001)). There is an inherent phase relationship between harmonics in voiced utterances, very much related to the formants or resonances of the vocal tract, which is perceptually relevant. Figure 2.18 illustrates such strong relationship existing between formant frequencies and harmonic phases when the analysis window is centered at a pulse onset. Not preserving such relationship results not only in a loss of presence and definition of the voice, but also makes it sound artificial, especially in the case of low pitches and also during transitions. This issue is intrinsically related to the concept of shape invariance and it is next discussed in detail.

2.2.3 Shape Invariance

In a simplified model of voice production, a train of impulses (i.e. glottal pulses) at the pitch rate excites a resonant filter (i.e. the vocal tract). According to this model, a speaker or singer changes the pitch of his voice by modifying the rate at which those impulses occur. An interesting observation is that the shape of the time-domain waveform signal around the impulse onsets is roughly independent

of the pitch, but it depends mostly on the impulse response of the vocal tract (see Figure 2.19). This characteristic is called *shape invariance*. In terms of frequency domain, this shape is related to the amplitude, frequency and phase values of the harmonics at the impulse onset times (see Figure 2.20).

A given processing technique will be *shape invariant* if it preserves the phase coherence between the various harmonics at estimated pulse onsets. This is illustrated in Figure 2.21. Thus, in order to obtain the best sound quality, it is desirable that those detected onsets match the actual glottal pulse onsets.

Several algorithms have been proposed in the literature regarding the harmonic phase-coherence for both phase-vocoder and sinusoidal modeling (for example (Laroche 2003) and (DiFederico 1998)). Most of them are based on the idea of defining pitch-synchronous input and output onset times and reproducing at the output onset times the phase relationship existing in the original signal at the input onset times. However, the results are not good enough because the onset times are not synchronized to the voice pulse onsets, but assigned to an arbitrary position within the pulse period. This causes unexpected phase alignments at voice pulse onsets that do not reproduce the formant to phase relations, adding an unnatural 'roughness' characteristic to the timbre (see Figure 2.22 and Figure 2.23).

Therefore, it is desirable to detect the glottal voice pulse onsets and use them as the algorithm's frame onsets. Different methods detect glottal onsets relying on the minimal phase characteristics of the voice source (i.e. glottal signal) (e.g. (Smits and Yegnanarayana 1995)(Yegnanarayana and Veldhuis 1998)). Peeters (2001) discusses and compares such methods. Following this same idea we proposed in (Bonada 2004) a method to estimate the voice pulse onsets out of the harmonic phases that fits very well and efficiently in a constant frame-rate spectral analysis framework. It is based on the property that when the analysis window is properly centered, the unwrapped phase envelope is nearly flat with shifts under each formant, thus being close to a Maximally Flat Phase Alignment (MFPA) condition. By means of this technique, both phasiness and roughness can be greatly reduced to be almost inaudible, even in the case of a constant frame-rate framework. The following subsection discusses this technique in depth.

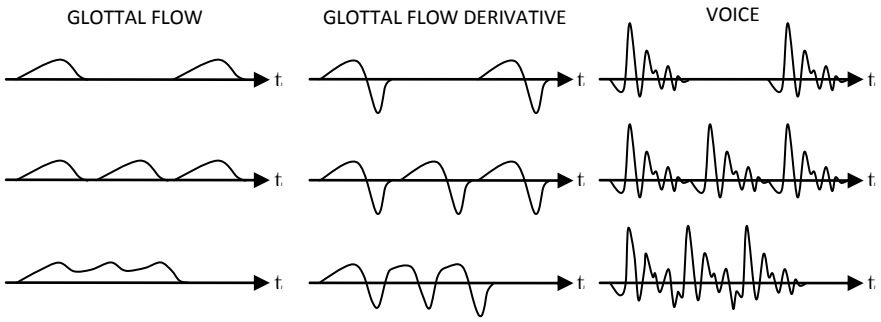


Figure 2.19 Shape invariance: the shape of the time-domain waveform signal around the impulse onsets is roughly independent of the pitch, but it is dependant mainly on the impulse response of the vocal tract. We see in the left column glottal flow pulses (*air flow*), in the middle column its derivative (*air pressure*), and in the right column voice pulses (*air pressure after the vocal tract*). The vocal tract filters the glottal pulses, adds some resonances and antiresonances (*i.e. formants and antiformants*), thus shaping the glottal flow derivative pulses. Each row has a different pitch. Clearly, the shape of the waveform signal around each voice pulse is the same independently of the pitch.

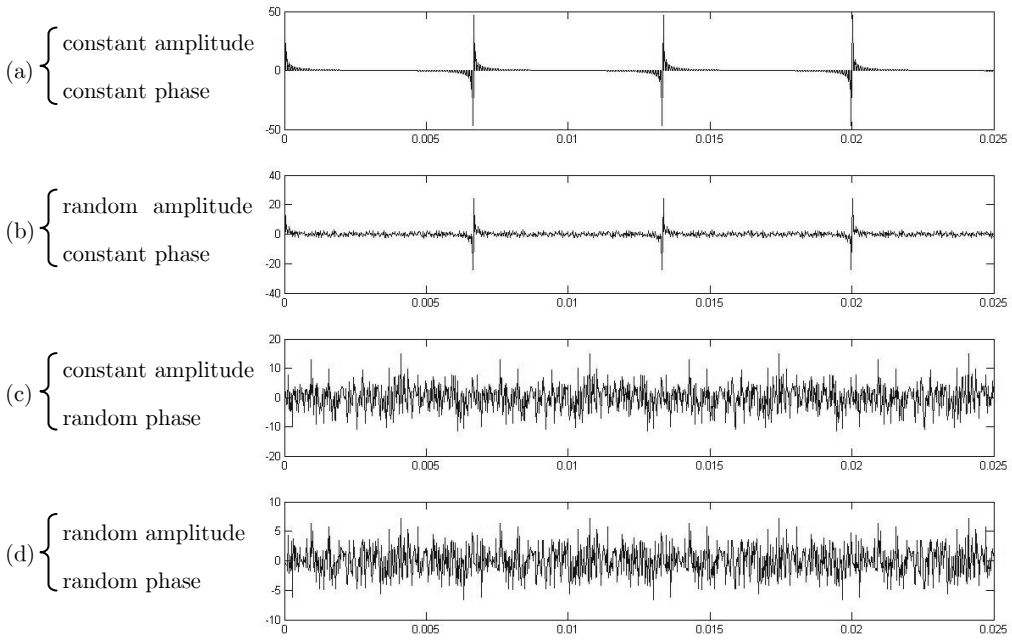
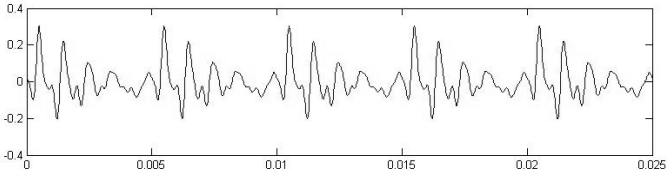


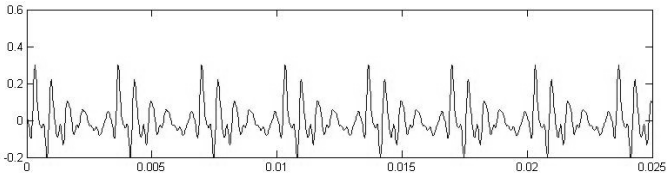
Figure 2.20 The shape of the waveform around each voice pulse is related to the amplitude, frequency and phase values of the harmonics at the pulse onset times. In the diagrams we see the time domain waveform of four pulses. The pitch rate is 150Hz. In (a) both the amplitude and phase of all harmonics are constant at each pulse onset. In (b) the amplitude is random. In (c) the phase is random. And in (d) both amplitude and phase are random. We can notice how the waveform shape is significantly affected by these changes.

(1) Original audio



(2) Shape invariant pitch transposition.

Harmonic phases at pulse onsets are preserved.



(3) Shape variant pitch transposition.

Harmonic phase at pulse onsets are not preserved

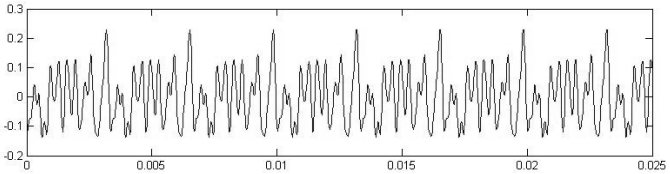


Figure 2.21 Shape variant and invariant processes. (1) is the original audio. In (2) and (3) we see a pitch transposition transformation: the pitch rate is arisen by a 1.5 factor. (2) can be considered a shape invariant transformation because the waveform shape at each pulse onset is preserved, although time-compressed. Conversely, (3) is a shape variant transformation because the waveform shape is radically different than in the original signal.

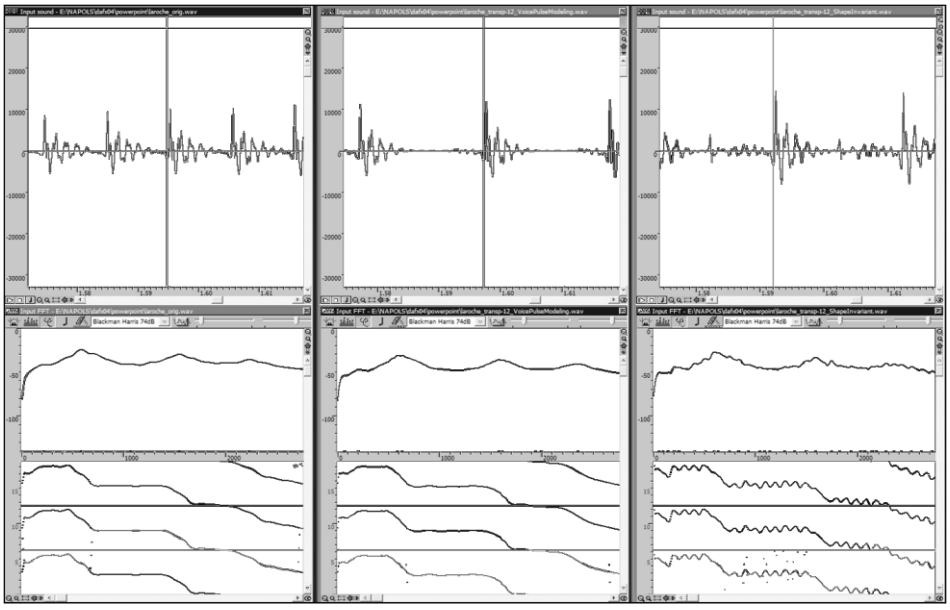


Figure 2.22 In the left, waveform and spectrum of a speech utterance (audio [100]). In the middle, octave down transposition generated using the voice pulse onsets (audio [101]). On the right, octave down transposition performed without taking into account the voice pulse onset locations (audio [102]).

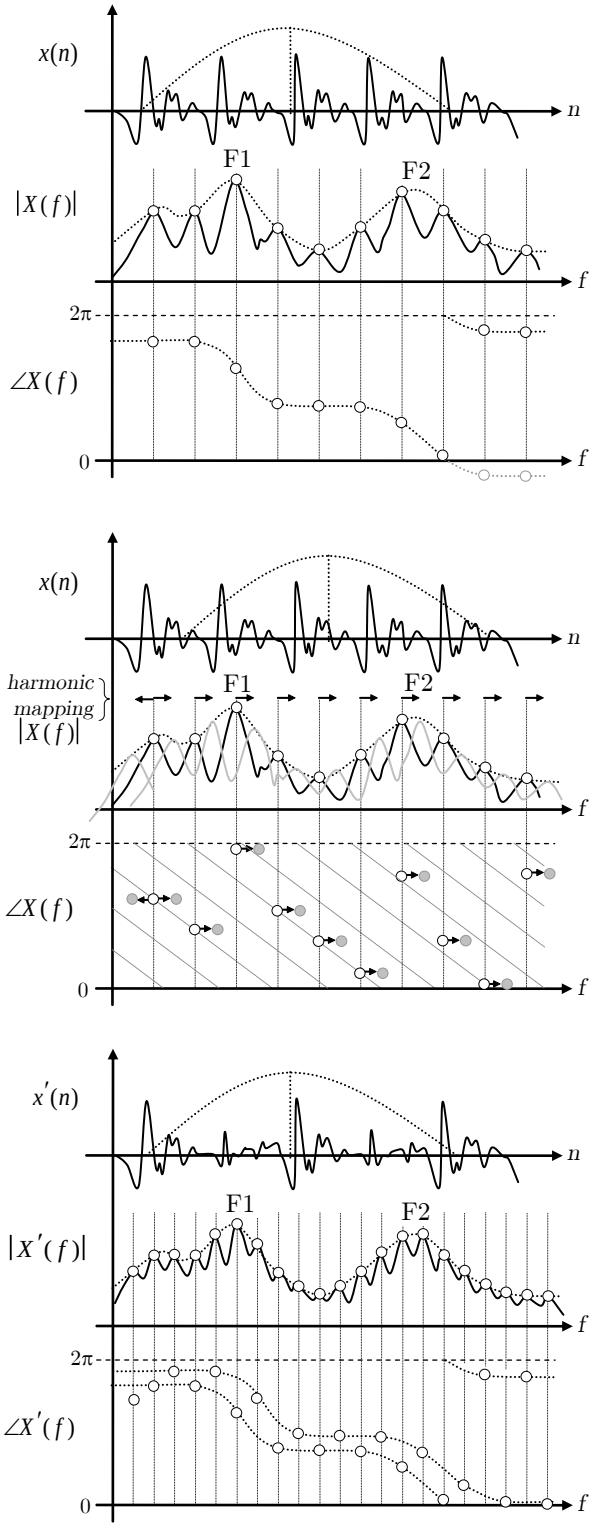


Figure 2.23 Spectrums obtained when the window is centered at the voice pulse onset (top figure), and between two pulse onsets (middle figure). In the middle figure harmonics are mapped and shifted in frequency to perform one octave down transposition. In the bottom figure, spectrum of the transformed signal with the window centered at the voice pulse onset. The 'doubled' phase alignment adds an undesired 'roughness' characteristic to the voice signal. Besides, we do not see only one voice pulse per period as expected, but two with strong amplitude modulation.

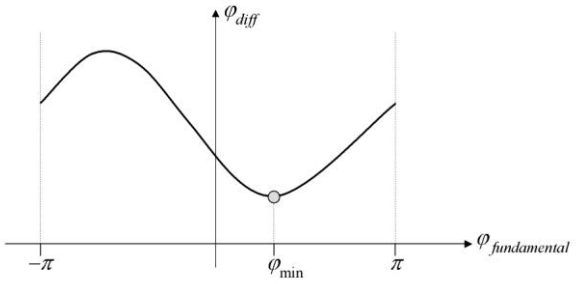


Figure 2.24 MFPA phase difference function. The minimum value corresponds to the phase $\varphi_{\min}$ of the fundamental when the analysis window is approximately centered at a voice pulse onset.

◊ *MFPA ALGORITHM*

As illustrated in Figure 2.18, when the analysis window is almost centered to a voice pulse onset, the harmonics are synchronized in a way that the phase spectrum is nearly flat with phase shifts under each resonance (i.e. formant) area. Whenever we move such window, the corresponding time shift adds a phase shift that varies linearly along frequency ($e^{-j\omega\Delta t}$). Thus, one way to estimate the pulse location is to estimate the slope of this linear phase shift. However, phase wrapping complicates the problem because all phase values are located in the range $[-\pi,\pi)$.

The MFPA algorithm attempts to find the time-shift Δt that minimizes the phase differences between harmonics, therefore obtaining a maximally flat phase alignment. Such procedure involves the following steps:

a) Define several fundamental phase candidates $\tilde{\phi}_0$ in the interval $[-\pi,\pi)$

b) For each candidate, apply the corresponding time shift $\widetilde{\Delta t}$ to each harmonic peak. The phase of each harmonic $\phi_{0,h}$ will be rotated as $\tilde{\phi}_{0,h} = \phi_{0,h} + 2\pi f_h \widetilde{\Delta t}$, where f_h is the frequency of the h^{th} harmonic, which is assumed to be constant.

c) Compute the sum of rotated phase differences as $\tilde{\phi}_{diff} = \sum \left| \text{princarg}\left(\tilde{\phi}_{0,h+1} - \tilde{\phi}_{0,h}\right)\right|$, where *princarg*[9] is a function that puts an arbitrary radian phase value into the interval $[-\pi,\pi)$ by adding an integer number of 2π periods.

d) After estimating the phase difference sum of all phase candidates, a function is obtained that is similar to a sinusoid and whose minimum sets the desired fundamental phase $\phi_{\min}$ that approximately centers the voice pulse onset. This is illustrated in Figure 2.24.

e) Finally, the closest pulse onset t_{MFPA} to the frame center time t_{frame} is estimated as $t_{MFPA} = t_{frame} + \text{princarg}\left(\phi_{\min} - \phi_{0,0}\right)/2\pi f_0$

The method can be illustrated with a synthetic example. Let us define $s(n)$ as an audio signal composed of a set of harmonics with amplitude and initial phase values computed from the spectral envelopes shown in the next figure.

[9] This is mathematically expressed as $\text{princarg}(x) = x - 2\pi\left\lfloor \frac{x}{2\pi} + 0.5 \right\rfloor$

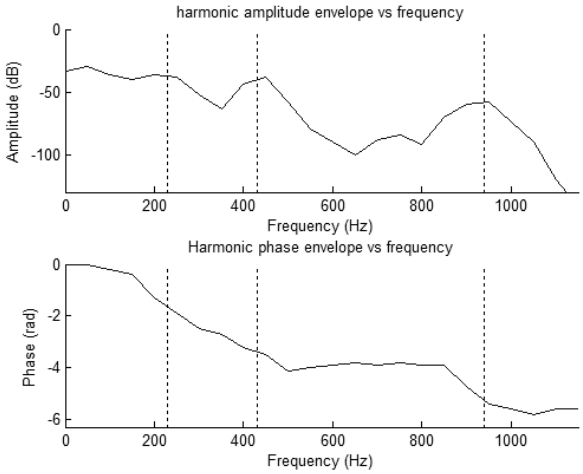


Figure 2.25 Spectral envelopes of a synthetic signal used to illustrate the MFPA algorithm.

Note that vertical dashed lines indicate formant frequencies and that phase values decay around each formant frequency. The fundamental frequency is set to vary linearly from 55 to 95 Hz. Harmonic phases are initialized using the previous phase envelope, and then propagated by integration of their frequency value at each sample. The signal is sampled at 44.1Khz and analyzed with a sliding 2049 samples Blackman-Harris window with a constant hop size of 249 samples. In Figure 2.27 left views show the estimated harmonic amplitude and phase values at frame onsets drawn with different colors for each frame. As expected, the estimated harmonic amplitudes follow the proposed envelope. However, harmonic phases do not follow the proposed phase envelope because hop size and fundamental frequency are not synchronized. In the same figure, the top-right view shows the superposition of the obtained MFPA error functions for each frame. All the estimated $\phi_{\min}$ values are close to 0.25 radians. The bottom-right view shows the harmonic phases time-shifted to the MFPA onsets, which as expected look much more flat then harmonic phases at frame onsets. Figure 2.26 shows the audio signal together with the analysis frame onsets and the estimated MFPA onsets. Note that all pulses have been correctly detected though the analysis hop size is constant. In certain cases we observe slightly different estimations (e.g. the second pulse) probably due to small errors in the estimation of the harmonic parameters and to the fact that the pitch is supposed to be constant around each frame, which is not true.

In the case of voice signals, there are several issues to take into account. In our experiments we have seen that it is better to consider only low frequency harmonics (up to around 3 Khz) since usually most energy is located at low frequencies and higher frequency components are often more unstable or noisy. In addition, 80 seems to be a reasonable size for the set candidates $\tilde{\theta}_0$ in the interval $[-\pi,\pi)$. Besides, noisy peaks such as the ones in amplitude valleys might introduce significant errors in the estimation. However, results seem to improve by weighting the phase differences values by a function computed out of the harmonic amplitudes, such as for example the ratio between harmonic and fundamental amplitude. Moreover, a continuation algorithm for smoothing $\phi_{\min}$ along time is desired in order to better handle transitions and noisy parts.

Figure 2.28 shows the results of the analysis of a low pitch male utterance pronouncing the words "I can be". We can see in the bottom view the evolution of $\phi_{\min}$ with values along the whole period $[0,2\pi)$. The corresponding estimated pulse onsets appear in the top view as a sequence of marks below the voice waveform. Actually, the two closest pulse onset estimations have been drawn for each frame, so we can see that the estimation of a certain pulse sometimes differ significantly for consecutive frames. This is the case of the pulses between 3.29 and 3.35 seconds, and the one around 3.41 seconds, where both fundamental frequency and $\phi_{\min}$ are varying considerably.

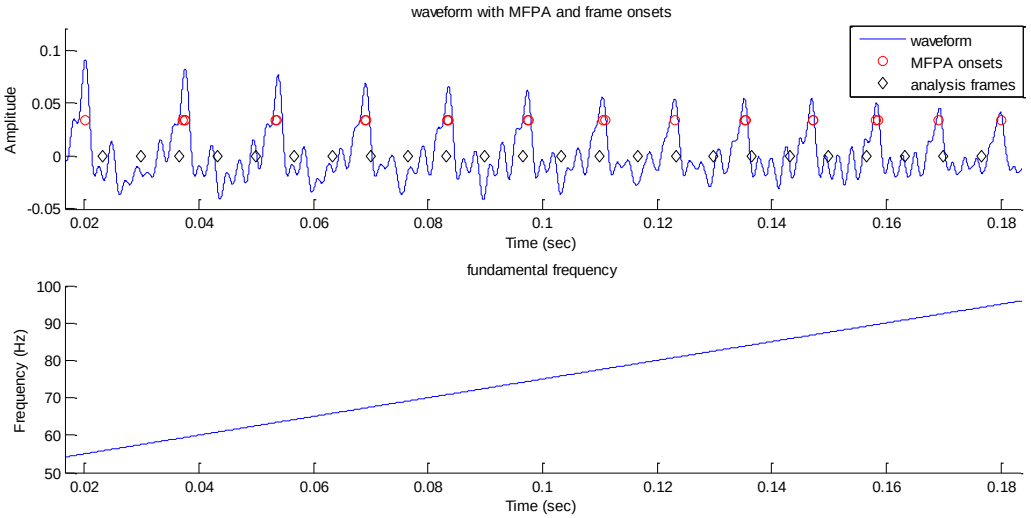


Figure 2.26 MFPA estimation. The top view shows the audio signal together with the analysis frame center times and the estimated MFPA onsets, whereas the bottom view shows the fundamental frequency. Although the analysis hop size is constant and therefore not synchronized to the fundamental frequency, the MFPA onsets correctly indicate the pulses.

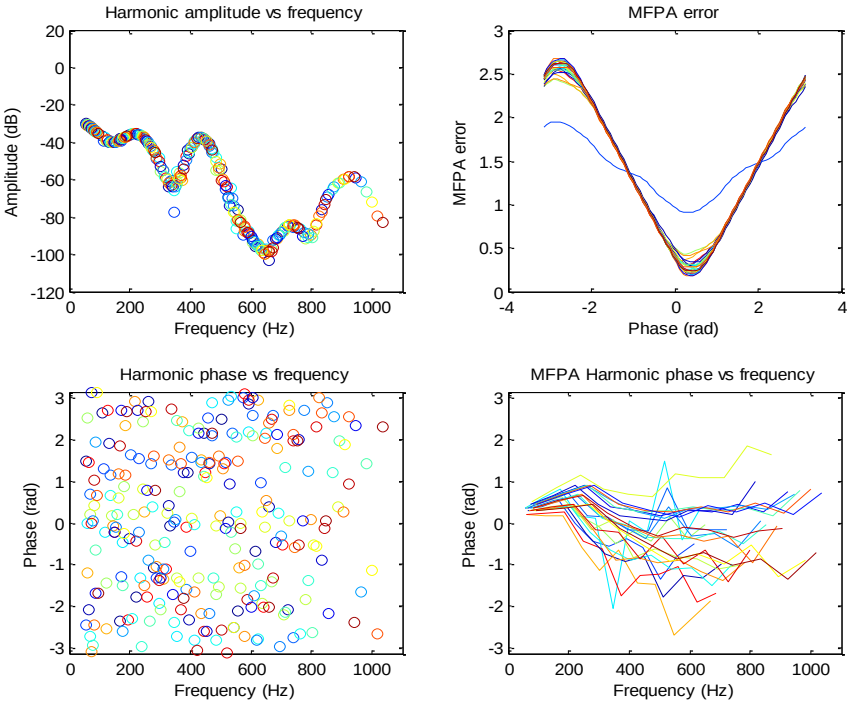


Figure 2.27 These four figures show harmonic parameters and MFPA results for each frame. In the left views harmonic amplitude and phases are drawn using different colors for each frame. In the right views MFPA error and corresponding harmonic phases are drawn.

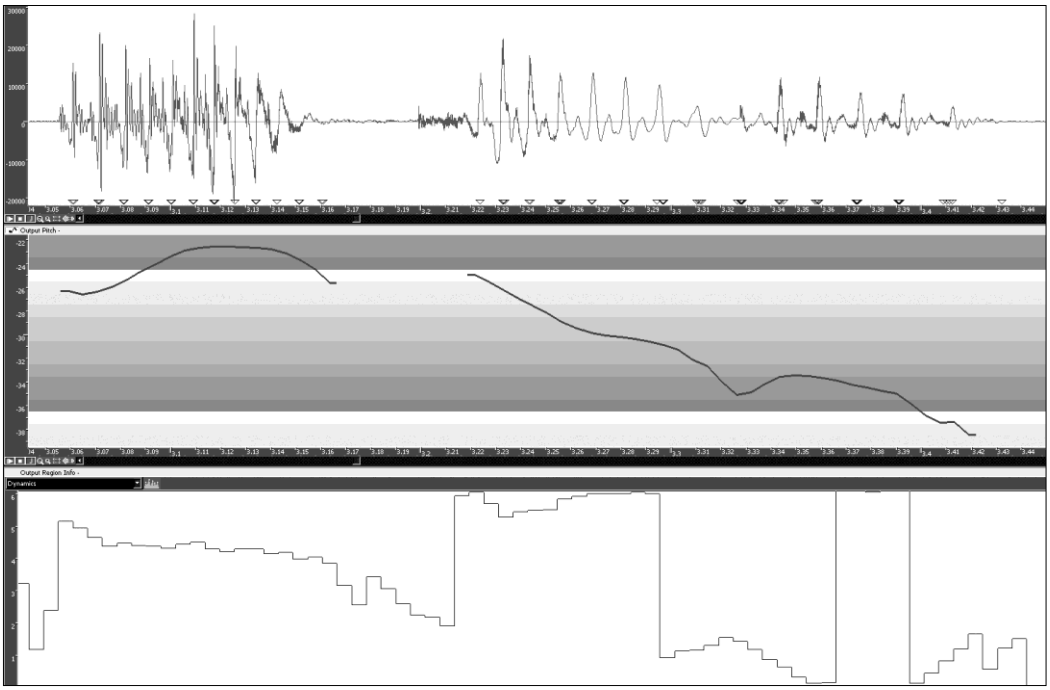


Figure 2.28 MFPA analysis of a low pitch male utterance of the words "I can be". The top view shows the voice waveform together with the estimated pulse onsets, the middle view the estimated fundamental frequency (from 48 to 120Hz), and the bottom view the estimated $\phi_{\min}$ in radians.

SEQUENCE OF PULSE ONSETS

Up to this point, we have obtained a set of probable pulse onsets, but not the actual sequence of pulse onsets. We have just seen that since the method uses a constant frame rate analysis, sometimes we obtain different onset estimations for a given pulse that might differ significantly. Therefore, assuming that the fundamental frequency estimation is accurate, we propose a method to find the best match between the estimated pulse candidates and the fundamental frequency function. Let's define P as the set of the p possible pulse onsets $\hat{t}_i$ obtained from the MFPA analysis, $P=\{\hat{t}_0,\hat{t}_1,\dots\hat{t}_{p-1}\}$. What we want to find is $Q_K=\{\hat{t}_{k_0},\hat{t}_{k_1},\dots\hat{t}_{k_{q-1}}\}$, satisfying $Q_K\subseteq P$ $K=\{k_0,k_1,\dots k_{q-1}\}\subseteq\{0,1,\dots p-1\}$, which is the subset of q pulses that best explains the estimated fundamental frequency function. Therefore, we establish the following error measure to be minimized:

- FUNDAMENTAL FREQUENCY ERROR

  The fundamental frequency error of a certain sequence Q_K might be defined in the continuous case as the relative difference between pulse duration and average period duration.

$$E_{f_0}(Q_K)=\sum_{i=1}^{q-1}e_{f_0}(k_i,k_{i-1})=\sum_{i=1}^{q-1}\frac{\left|\left(\hat{t}_{k_i}-\hat{t}_{k_{i-1}}\right)-\bar{T}_i\right|}{\bar{T}_i}\quad,\quad \bar{T}_i=\frac{1}{\left(\hat{t}_{k_i}-\hat{t}_{k_{i-1}}\right)}\int_{\hat{t}_{k_{i-1}}}^{\hat{t}_{k_i}}f_0^{-1}(t)dt \qquad (2.30)$$

  However, the analysis is performed in discrete steps and what we obtain are the frame estimations of the fundamental frequency $f_{0,m}$. The average period duration is then computed as

$$\bar{T}_i=\frac{1}{(m_e-m_b+1)}\sum_{m=m_b}^{m_e}\frac{1}{f_{0,m}} \qquad (2.31)$$

  where m_b and m_e are the indices of the closest voiced frames to $\hat{t}_{k_{i-1}}$ and $\hat{t}_{k_i}$ respectively.

We should consider as well that the best pulse sequence Q_K might not be a subset of P but a sequence of onsets probably close to the ones contained in P. This is likely to be the general case for P contains just a few number of predicted onsets that could be theoretically located anywhere in the continuous temporal axis. An obvious way of improve the chances to find a better solution is to increase the number of pulse candidates, with the drawback of increasing as well the computational cost of the algorithm. We propose to do this systematically by adding not just one but N_p candidates per frame as follows

$$P_{\exp anded} = \left\{\hat{t}_{i,m}\right\} \qquad \forall i \in \left\{0,1,\ldots N_p - 1\right\}, \forall m \text{ having } f_{0,m} > 0$$

$$\hat{t}_m = t_m + \frac{\text{princarg}\left(\theta_{\min} - \theta_{0,0}\right)}{2\pi f_{0,m}} \tag{2.32}$$

$$\hat{t}_{i,m} = \begin{cases} \hat{t}_m + \dfrac{\gamma\left(i/(N_p - 1)\right) - 1}{2\pi f_{0,m}} & if \quad \hat{t}_m \geq t_m \\ \hat{t}_m + \dfrac{\gamma\left(i/(N_p - 1)\right)}{2\pi f_{0,m}} & if \quad \hat{t}_m < t_m \end{cases} \qquad \forall i \in \left\{0,1,\ldots N_p - 1\right\}$$

being $\gamma(x)$ a function that sets the density of candidates along one period, ideally higher close to the predicted pulse onsets, such as the one shown in the following figure.

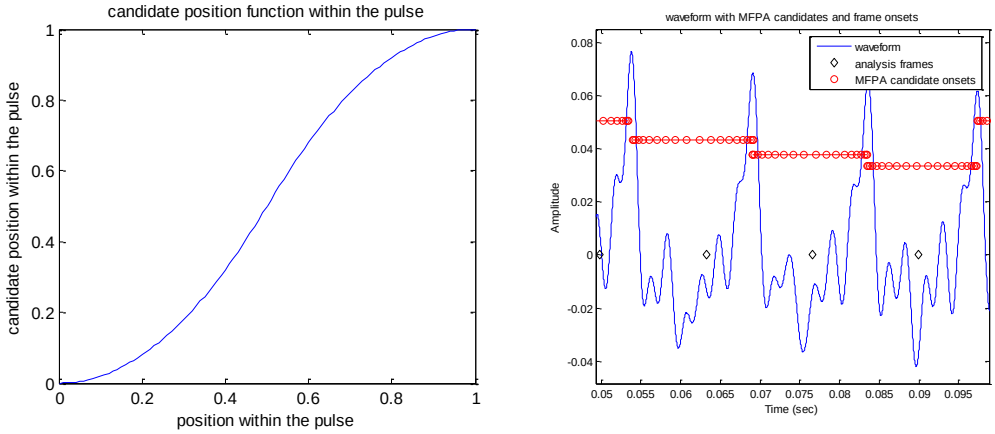


Figure 2.29 The left view shows an example of a possible candidate position function, whereas the right view shows the corresponding candidates drawn on top of the waveform.

The addition of these other pulse candidates requires considering another error to be minimized:

- MFPA ONSET CANDIDATE ERROR
  The onsets predicted by the MFPA algorithm have a greater likelihood to be the actual voice pulse onsets than the other ones. Therefore, we can define the candidate error as the relative distance between each candidate and its closest MFPA prediction

$$e_{MFPA}(k_i) = \left|\hat{t}_{k_i} - \hat{t}_j\right| f_{0,m} \quad \text{where} \quad \begin{cases} m \text{ is the frame index corresponding to the candidate } \hat{t}_{k_i} \\ j \text{ is the closest MFPA prediction of the } m^{th} \text{ frame} \end{cases}$$

  Therefore the total error would be the sum of each onset error.

$$E_{MFPA}(Q_K) = \sum_{i=0}^{q-1} e_{MFPA}(k_i) \tag{2.33}$$

A dynamic programming approach seems to be a good choice for finding the best pulse subset Q_K that minimizes both fundamental frequency and MFPA errors. In our specific case, we could define

$e_{MFPA}(k_i)$ as the state error and $e_{f_0}(k_i)$ as the transition error. The *observation events* are the set P of possible pulse onsets, whereas the *path events* are the onsets corresponding to a certain path defined by K. The following figure shows an example of this.

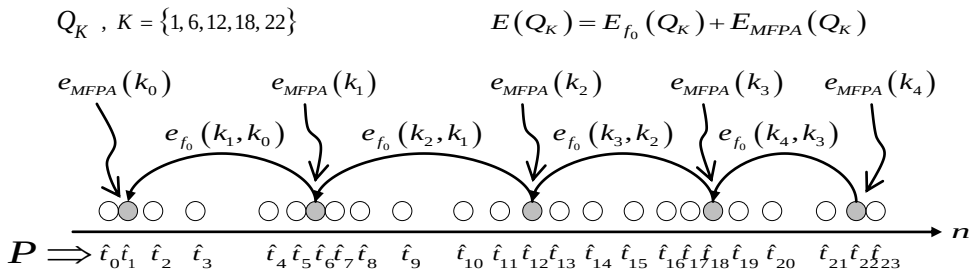


Figure 2.30 Illustration of a dynamic programming approach for finding the optimal pulse subset Q_K.

This approach is valid for voiced sections but does not work for unvoiced ones, because then $\bar{T}_i$ in Equation (2.31) would be infinite. We could think of setting the pitch error $e_{f_0}(k_i, k_{i-1})$ to zero whenever there is an unvoiced frame between $t_{k_{i-1}}$ and t_{k_i} to allow any unvoiced section length, but this would not penalize connecting the first voiced frame to the last voiced frame of the whole signal. Adding a penalty for having voiced frames within a jump would not improve the results either. What we propose is to add each unvoiced frame center time to P and to modify the definition of $e_{f_0}(k_i, k_{i-1})$ as

$$e_{f_0}(k_i, k_{i-1}) = \begin{cases} \dfrac{\left|\left(\hat{t}_{k_i} - \hat{t}_{k_{i-1}}\right) - \bar{T}_i\right|}{\bar{T}_i} & \text{if } \hat{t}_{k_i} \text{ and } \hat{t}_{k_{i-1}} \text{ correspond to voiced frames} \\ \dfrac{\left|MAX\left(\Delta_t, \hat{t}_{k_i} - \hat{t}_{k_{i-1}}\right) - \Delta_t\right|}{\Delta_t} & \text{if } \hat{t}_{k_i} \text{ or } \hat{t}_{k_{i-1}} \text{ correspond to unvoiced frames.} \end{cases} \tag{2.34}$$

This redefinition of the fundamental frequency error avoids infinite error values as well as forces Q_K to connect consecutive frames center times during unvoiced sections. Besides, in voiced-to-unvoiced (VU) transitions it would connect a voiced frame onset to the next unvoiced frame. Conversely, in unvoiced-to-voiced transitions (UV) it would connect a voiced frame onset to the previous closest unvoiced frame.

In general, computing all possible paths would mean to evaluate so many combinations that it would make the proposed method impractical. However, inspired by the well-known Viterbi algorithm, an interesting optimization is to force each onset to choose the best previous onset considering the accumulated error and the transition error as follows

$$\begin{gathered} e_{acum}(k_i) = e_{acum}(k_{i-1}) + e_{MFPA}(k_i, k_{i-1}) + e_{f_0}(k_i) \\ \text{for } k_i \Rightarrow \text{choose } k_{i-1} \text{ which minimizes } e_{acum}(k_{i-1}) + e_{MFPA}(k_i, k_{i-1}). \end{gathered} \tag{2.35}$$

In addition, in order to avoid missing onsets at the beginning of the signal, we set $e_{acum}(0) = 0$, $k_0 = 0$ and an we add an onset at the beginning ($\hat{t}_0 = 0$) if it does not already exists. Finally, in a similar way, we force $k_{q-1} = p - 1$ and we add an onset at the end of the signal ($\hat{t}_{p-1} = \text{duration}$).

We have implemented the proposed approach and the results are very promising. An example of an excerpt of the best found sequence for a recorded speech signal is shown in Figure 2.31, where the estimated pulse onsets are drawn as vertical dashed lines on top of the waveform. Note that during unvoiced sections the pulse rate is constant as expected (equal to the analysis frame rate), and also that the connection between voiced and unvoiced onsets is smaller or equal than the analysis hopsize in VU and UV transitions. Another example is shown in Figure 2.32, where the voice signal features a strong growl characteristic and therefore the pulse sequence is affected by strong amplitude modulation and several subharmonics appear in the spectrum. Nevertheless, MFPA correctly predicts voice pulse onsets. One interesting improvement of the proposed method would be to run it

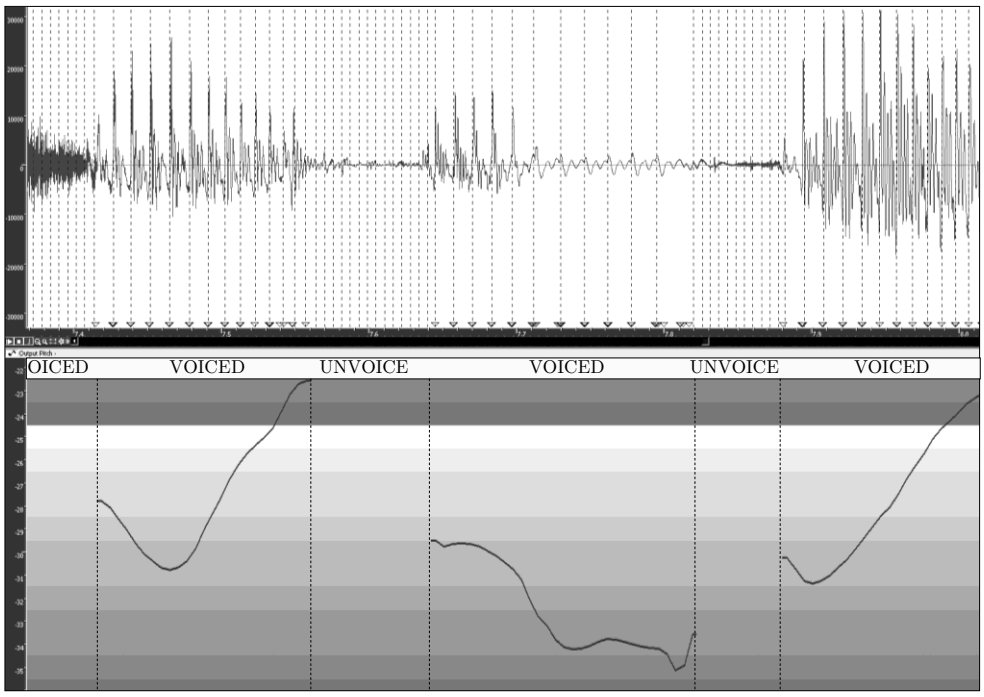


Figure 2.31 MFPA analysis of a speech recording. In the top view the onsets of the best path Q_K are marked as vertical dashed lines on top of the utterance waveform. The bottom view shows the estimated fundamental frequency and the voiced/unvoiced segmentation.

iteratively so to converge to the best possible solution. Although we have not implemented this idea yet, we have actually thought of how to do it. Basically, after each n^{th} iteration we would get a pulse sequence $Q_{K,n}$ that would be used to define the new set of possible pulses P_{n+1} . This set would explore the temporal space around each predicted pulse $\hat{t}_{k_i,n}$ with higher temporal resolution at each iteration. This way, each step would decrease the best path error and the convergence criterion could be defined by a threshold of the percent of relative error decrement.

TRANSFORMED HARMONIC PHASES

The initial motivation of the MFPA algorithm was to estimate the glottal pulse onsets and to use them to achieve natural sounding shape invariant transformations. In that sense, both methods detailed in (Laroche 2003) and (DiFederico 1998) for constant frame-rate processing frameworks are a good solution once adapted to reproduce at synthesis frame times the phase relationship existing in the original signal at the glottal pulse onsets found by the MFPA algorithm (Q_K). The idea is to propagate only the phase of the fundamental by any appropriate method (e.g. linear frequency model, linear frequency modulation estimation, etc), and afterwards compute the rest of harmonic phases as indicated by the following procedure, illustrated in Figure 2.33.

a) SYNTHESIS FUNDAMENTAL PHASE
For each frame m to process, propagate the synthesis fundamental phase $\phi'_{0,0,m}$ by any appropriate method (e.g. assuming linear frequency)

b) MFPA ALIGNMENT
Calculate the closest pulse onset $\hat{t}_{k_i}$ and compute the harmonic phases $\phi^{mfpa}_{0,h,m}$ resulting from the time-shift corresponding to the time difference $\hat{t}_{k_i} - t_m$. Assuming frequency to be constant we would get

$$\phi^{mfpa}_{0,m}(h) = \phi^{mfpa}_{0,h,m} = \phi_{0,h,m} + 2\pi f_{h,m}\left(\hat{t}_{k_i} - t_m\right). \tag{2.36}$$

c) TRANSFORMED MFPA ALIGNMENT
Compute the transformed MFPA phase alignment $\phi'^{\,mfpa}_{0,h,m}$ by any appropriate method $\gamma\left(\phi^{mfpa}_{0,m}, h\right)$, taking care of the fact that phase is wrapped into a single

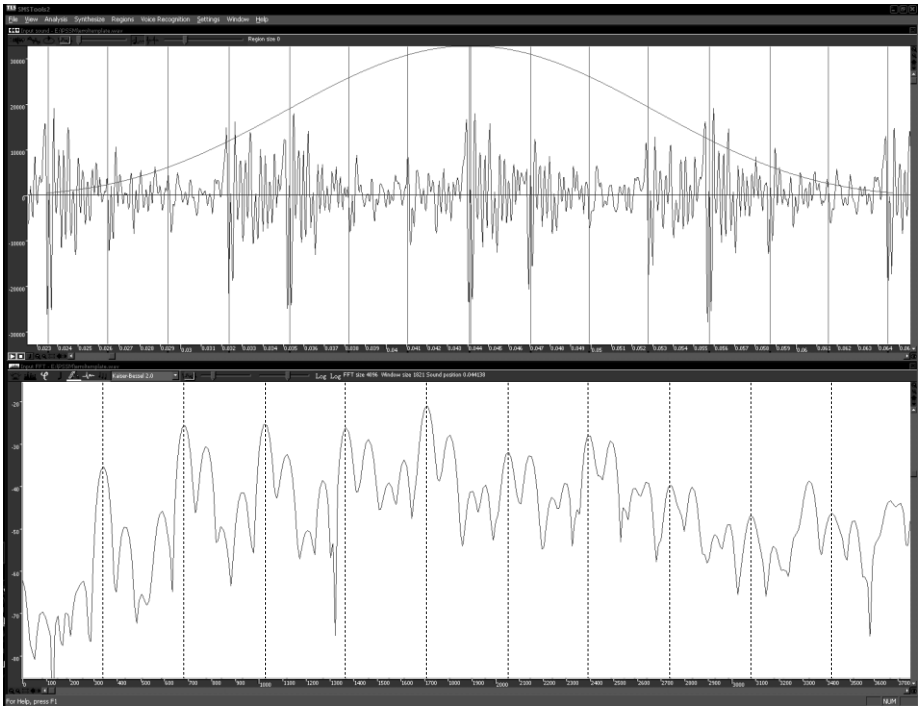


Figure 2.32 Voice pulse onsets (vertical solid lines) detected by the MFPA algorithm in a growl utterance. Dashed lines are located at the frequencies of the harmonics. Several subharmonics appear due to the strong amplitude modulation present along the growl.

period. The simplest option is a discrete mapping between harmonics, whereas other possibilities involve interpolation methods of the unwrapped phase envelope. In the case of transposition and time-scaling transformations the transformed MFPA alignment would be ideally an interpolation of the MFPA alignment. Instead, a timbre-scaling transformation would result into a scaling of the MFPA alignment.

d) SYNTHESIS HARMONIC PHASES
Compute the synthesis phases by applying the time-shift corresponding to the difference between the transformed MFPA fundamental phase $\phi'^{\,mfpa}_{0,0,m}$ and the synthesis one $\phi'_{0,0,m}$,

$$\phi'_{0,h,m}=\phi'^{\,mfpa}_{0,h,m}+\frac{f'_{h,m}}{f'_{0,m}}\text{princarg}\left(\phi'_{0,0,m}-\phi'^{\,mfpa}_{0,0,m}\right). \tag{2.37}$$

PHASE UNWRAPPING

One potential source of artifacts relies in the fact that estimated harmonic phases are wrapped into a single period $[-\pi,\pi)$. Whenever we want to interpolate harmonic phases, such as for instance when computing the transformed MFPA alignment, we have to consider this issue and therefore unwrap the phase sequence so that we get the shortest distance between consecutives harmonic phases. The unwrapping procedure sets the initial phase to be $\phi_0^{uw}=\phi_0$. The next unwrapped phases ϕ_h^{uw} can be obtained iteratively by

$$\phi_h^{uw}=\phi_{h-1}^{uw}+\text{princarg}\left(\phi_h-\phi_{h-1}\right). \tag{2.38}$$

Note that the princarg function basically adds an integer number of 2π periods to its argument so to get a phase between $-\pi$ and π. The transformed MFPA alignment is then computed by interpolation of the unwrapped phases ϕ_h^{uw}. It may happen however that, for a given harmonic, during the phase unwrapping procedure different numbers of periods are added in consecutive frames.

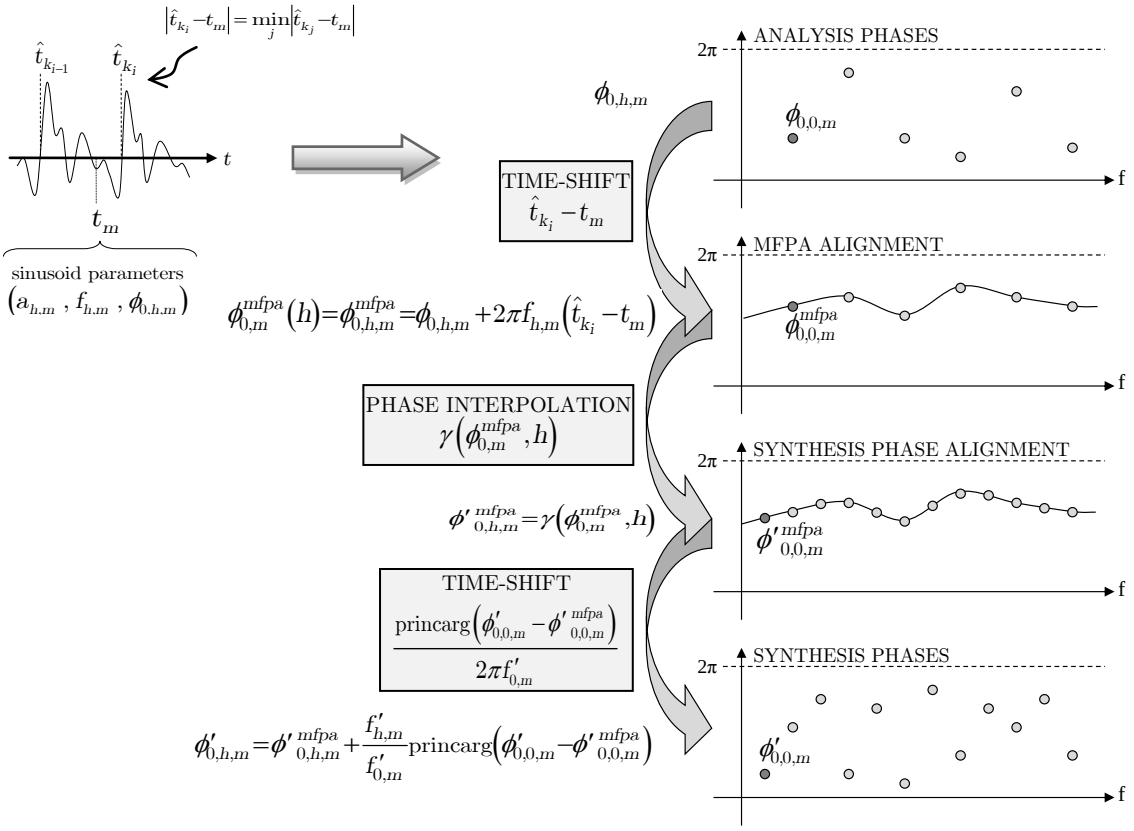


Figure 2.33 MFPA preservation in a constant frame-rate framework. In this example a downwards transposition is applied and the unwrapped MFPA phase alignment is smoothly interpolated.

In that case we will get discontinuities in the unwrapped phase envelopes such as the ones illustrated in Figure 2.34. In order to avoid this artifact a phase correction $\theta_{h,m}$ can be added to each harmonic so to compensate the period differences:

$$\theta_{h,m} = \begin{cases} -2\pi & if \quad x \geq \pi \\ 0 & if \quad -\pi \leq x < \pi \\ 2\pi & if \quad x < -\pi \end{cases} \qquad (2.39)$$

$$x = \text{princarg}(\phi_{h,m} - \phi_{h-1,m}) - \text{princarg}(\phi_{h,m-1} - \phi_{h-1,m-1}).$$

In addition, in order to minimize and smooth the corrections, we propose accumulating frame corrections $\theta_{h,m}$ and slowly decay the resulting correction to zero by adding or subtracting θ_{correc} radians at each frame. The resulting phase correction $\tilde{\theta}_{h,m}$ for the h^{th} of the m^{th} is then computed as

$$\tilde{\theta}_{h,m} = \tilde{\theta}_{h,m-1} - \text{sign}(\tilde{\theta}_{h,m-1}) \cdot \min(|\tilde{\theta}_{h,m-1}|, \theta_{correc}) + \theta_{h,m}. \qquad (2.40)$$

In our experiments we have used $\theta_{correc} = 0.1$ radians. This way, if for frame $m-1$ two consecutive harmonics ($h-1$ and h) had a phase difference of -0.9π and in the next frame m the difference became 0.9π, the phase correction would be $\tilde{\theta}_{h,m} = -2\pi$. If in the next frame $m+1$ the harmonic phase difference became again -0.9π, then frame correction would be $\theta_{h,m+1} = 2\pi$ and the resulting phase correction $\tilde{\theta}_{h,m+1} = \tilde{\theta}_{h,m} + 0.1 + \theta_{h,m+1} = 0.1$ radians. The final sequence of unwrapped phases ϕ_h^{uw} would be $\phi_{h-1}^{uw} - 0.9\pi$, $\phi_{h-1}^{uw} - 1.1\pi$, and $\phi_{h-1}^{uw} - 0.8\pi$, with no unwrapping artifacts. Instead, if no correction was applied, then the final sequence of unwrapped phases would have been $\phi_{h-1}^{uw} - 0.9\pi$,

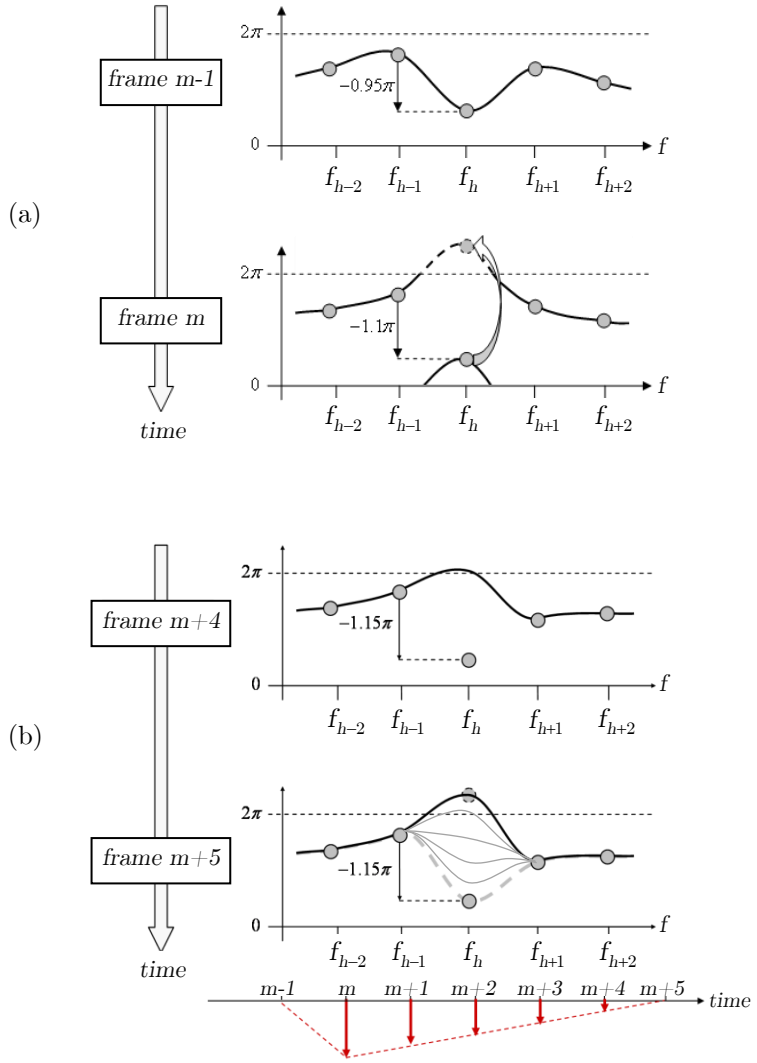


Figure 2.34 Phase unwrapping problem: different numbers of 2π periods are added to a harmonic in consecutive frames. In (a) the wrapped phase difference between harmonics *h-1* and *h* is -0.95π and -1.1π respectively for frames $m-1$ and m. The interpolated phase envelopes differ a lot and introduce significant discontinuities when used to compute transformed harmonic phases. In (b), we see how the proposed correction algorithm deals with a similar situation. The bottom figure shows the phase correction applied to the h^{th} harmonic for several consecutive frames, whereas the middle figure shows in gray the successive interpolated phase envelopes and in black the interpolated phase envelope of the last frame.

$\phi_{h-1}^{uw}+.9\pi$, and $\phi_{h-1}^{uw}-0.9\pi$ producing discontinuities for transformed harmonics whose frequency was found between f_{h-1} and f_h.

REAL-TIME ADAPTATION

Thinking of a real-time implementation, a possible simplification of the previous method consists on using the predicted MFPA fundamental phase $\phi_{0,0,m}^{mfpa}$ at each frame instead of the best pulse sequence Q_K. In that case, the second step (b) would use the wrapped phase difference as a replacement for the pulse-frame time difference, giving $\Delta t=\text{princarg}\left(\phi_{0,0,m}^{mfpa}-\phi_{0,0,h}\right)\Big/2\pi f_{0,m}$. This approach definitely reduces the latency and the complexity of the algorithm avoiding the pulse sequence prediction, although at the same time we lose the global optimizations introduced by the dynamic programming algorithm and get a lower quality signal. Figure 2.35 shows an example of the

results obtained from a male utterance with and without the real-time adaptation, compared to a laryngograph signal.

An alternative is to force a decision of the dynamic programming algorithm with a certain latency $T_{latency}$. This way, instead of waiting to the end of the signal for computing the best path, we would compute intermediate best paths at each step and take as good the pulse onset sequence up to the current time minus the given latency. Although this procedure might introduce some sequence discontinuities when different paths are chosen in consecutive steps, a simple continuation algorithm would be enough to minimize them. For instance, if the algorithm steps follow a constant frame-rate approach with hopsize Δ_t, given the current output path $Q_K = \left\{\hat{t}_{k_0}, \hat{t}_{k_1}, ...\hat{t}_{k_{q-1}}\right\}$ and the current step best path $Q_G^m = \left\{\hat{t}_{g_0}, \hat{t}_{g_1}, ...\hat{t}_{g_{y-1}}\right\}$ of the m^{th} frame, the predicted pulse onsets of Q_G^m falling between $\hat{t}_{k_{q-1}} + \alpha/f_{0,m}$ and $m\Delta_t - T_{latency}$ would be added to Q_K. In this case, α sets the minimum ratio of fundamental period allowed for consecutive pulses at the path connection. In our experiments $\alpha = 0.75$ has shown to be a reasonable choice.

EVALUATION

Figure 2.36 to Figure 2.39 show several examples of results obtained with the MFPA algorithm. Detected voice pulse onsets (dashed lines) are displayed together with the audio waveform (in green), the laryngograph signal (in yellow) and the fundamental frequency (in green). Those examples include sections with rapidly varying pitch and with irregular pulse sequences. Their main characteristics are displayed in Table 2.2. Results indicate that if the fundamental frequency is correctly detected, the MFPA algorithm outputs onsets mostly match very well the laryngograph signal, even in the case of highly irregular pulse sequences such as the one in Figure 2.39b.

In addition, we have performed a more formal evaluation by computing histograms of pulse onset detection errors for the Keele Pitch Database[10]. This database contains audio recordings at a 22Khz sampling rate of ten speakers, five male and five female, reading the "North Wind story", a phonetically balanced text. In addition, laryngograph signal is also available, providing means for a robust estimation of GCIs and fundamental frequency. We have analyzed all files in this database and computed the corresponding pulse onsets with the MFPA algorithm. Laryngograph peaks indicate CGIs. However, laryngograph signals are noisy and have a strong low frequency component. For estimating appropriate peaks we have first smoothed the laryngograph signal, and then looked for local relative maxima above a minimum value in segments that feature a signal range above a certain threshold. In order to measure the detection performance we propose to consider a two-way mismatch approach, considering errors from laryngograph to MFPA and vice versa. Considering that L laryngograph peaks were detected, errors are defined as

- laryngograph to MFPA (e_{LM})
  temporal distance in seconds between a laryngograph peak t_l^{LG} and the closest MFPA predicted onset $\hat{t}_{k_c}$, divided by the fundamental period, i.e.

$$e_{LM}(l) = \frac{t_l^{LG} - \hat{t}_{k_c}}{f_0^{-1}\left(t_l^{LG}\right)} \qquad where \ \left|t_l^{LG} - \hat{t}_{k_c}\right| \le \left|t_l^{LG} - \hat{t}_k\right| \ \forall k \in \{0,1,...q-1\}. \tag{2.41}$$

- MFPA to laryngograph (e_{ML})
  temporal distance in seconds between a predicted MFPA onset and the closest laryngograph peak, divided by the fundamental period, i.e.

$$e_{ML}(k) = \frac{\hat{t}_k - t_{l_c}^{LG}}{f_0^{-1}\left(\hat{t}_k\right)} \qquad where \ \left|\hat{t}_k - t_{l_c}^{LG}\right| \le \left|\hat{t}_k - t_l^{LG}\right| \ \forall l \in \{0,1,...L-1\}. \tag{2.42}$$

Note that if one MFPA onset is predicted for each fundamental period, than both errors e_{LM} and e_{ML} are expected to be in the range $[-0.5,0.5]$. It is important to consider both errors in order to avoid giving low errors to MFPA onset estimations much denser than the pitch rate. For instance,

[10] http://www.liv.ac.uk/Psychology/hmp/projects/pitch.html

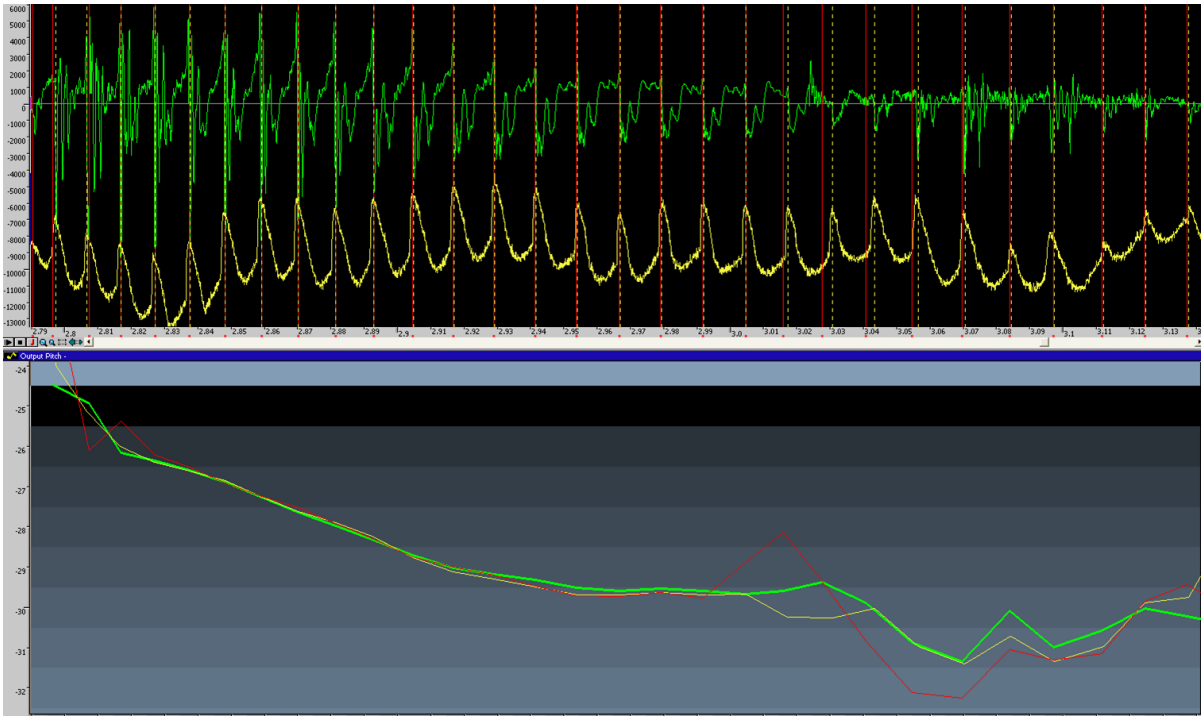


Figure 2.35 MFPA voice pulse onset estimations with and without real-time adaptation compared to a laryngograph signal. The top view shows a male voice utterance in green, the corresponding laryngograph signal in yellow, the estimated pulse onsets obtained with the proposed method as vertical yellow dashed lines, and the estimated onsets obtained with the real-time adaptation as vertical red solid lines. The main differences between pulse onsets happen around 3.04 seconds, where the real-time adaptation departs significantly from the laryngograph peaks. In the bottom view the corresponding fundamental frequency envelopes are shown: from the original signal in green, from the proposed method in yellow, from the real-time adapted method in red. Clearly, the global optimization generates a pitch envelope more similar to the ideal one, especially around 3.04 seconds.

FIGURE	DESCRIPTION
Figure 2.36a	decreasing pitch, unvoiced plosive
Figure 2.36b	increasing pitch
Figure 2.37a	two-pulse amplitude pattern (in the middle), unvoiced fricative
Figure 2.37b	decreasing pitch, stop in the middle
Figure 2.38a	increasing pitch, two-pulse amplitude pattern (right section)
Figure 2.38b	fry phonation at the beginning of a vowel
Figure 2.39a	decreasing pitch, voiced plosive
Figure 2.39b	very irregular sequence

Table 2.2 Reference to figures showing MFPA results together with the laryngograph signal

predicting MFPA onsets every ten percent of the fundamental period would systematically give absolute e_{LM} errors below 0.05. By contrast, e_{ML} values would be spread in the $[-0.5,0.5]$ range.

Figure 2.40 shows the histogram of e_{LM} errors for each speaker. Most errors fall in the range between -0.1 and 0.1. This means that estimated CGI deviations are mostly lower than a ten percent of the fundamental period. The error mean is specified in the title of each histogram. Error means tend to be positive, what means that the estimated onset tends to be slightly delayed with respect to the laryngograph peak. In the worst case (m3nw0000) the error mean gets close to 0.1, so estimated

CGIs are systematically delayed by about a ten percent of the fundamental period. Nevertheless, this deviation is low enough as to achieve good sound quality and a reasonable shape preservation. In turn, Figure 2.41 shows the histograms of e_{ML} errors. Note that these histograms are essentially a mirrored version of e_{LM} histograms, as expected if there is approximately a one-to-one correspondence between MFPA onsets and laryngographs peaks. Global histograms for the whole database are shown in Figure 2.42. Mean errors are 0.032 and -0.028 for e_{LM} and e_{ML} respectively. For the 76.27 percent of laryngograph peaks, the closest MFPA onset is closer than a ten percent of the period. Moreover, for the 89.11 percent the closest MFPA prediction is within a fifteen percent of the period.

DISCUSSION

MFPA algorithm generally obtains good enough approximations of the glottal closure instants (GCIs) as to effectively minimize smearing and roughness in processed voiced signals. Compared to other methods, it has the advantage of being able to estimate the GCI out of the harmonic parameters, and transform appropriately those without requiring performing another analysis with a window centered at the estimated GCI. In this sense, it fits very well a constant hop-size framework with sinusoidal analysis adding just a minimum extra computational cost.

MFPA relies on the existence of the fundamental frequency. In some voice utterances, however, the fundamental frequency might have a very low energy compared to other harmonics, and be greatly affected by the interference of those. In such cases, the GCI estimations are not reliable anymore and may jump significantly in consecutive frames. Future work should focus on improving the MFPA performance in those contexts. One idea would be to drop the fundamental and only use the harmonics whose parameters can be reliable estimated. If the second harmonic was one of those, then we would obtain two GCI candidates instead of a single one per MFPA analysis, and some post-processing should be added to generate a consistent GCI sequence.

(a)

(b)

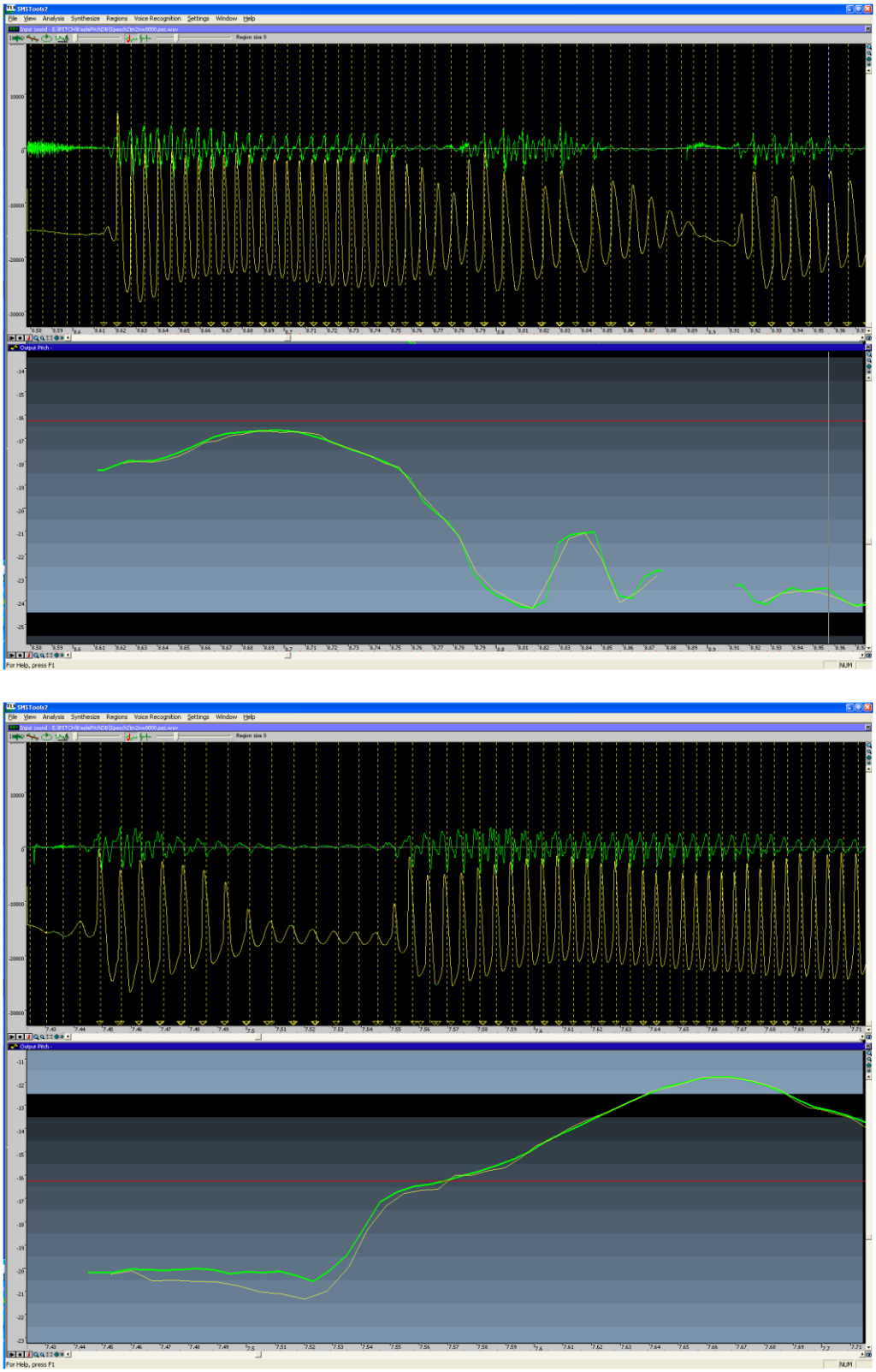
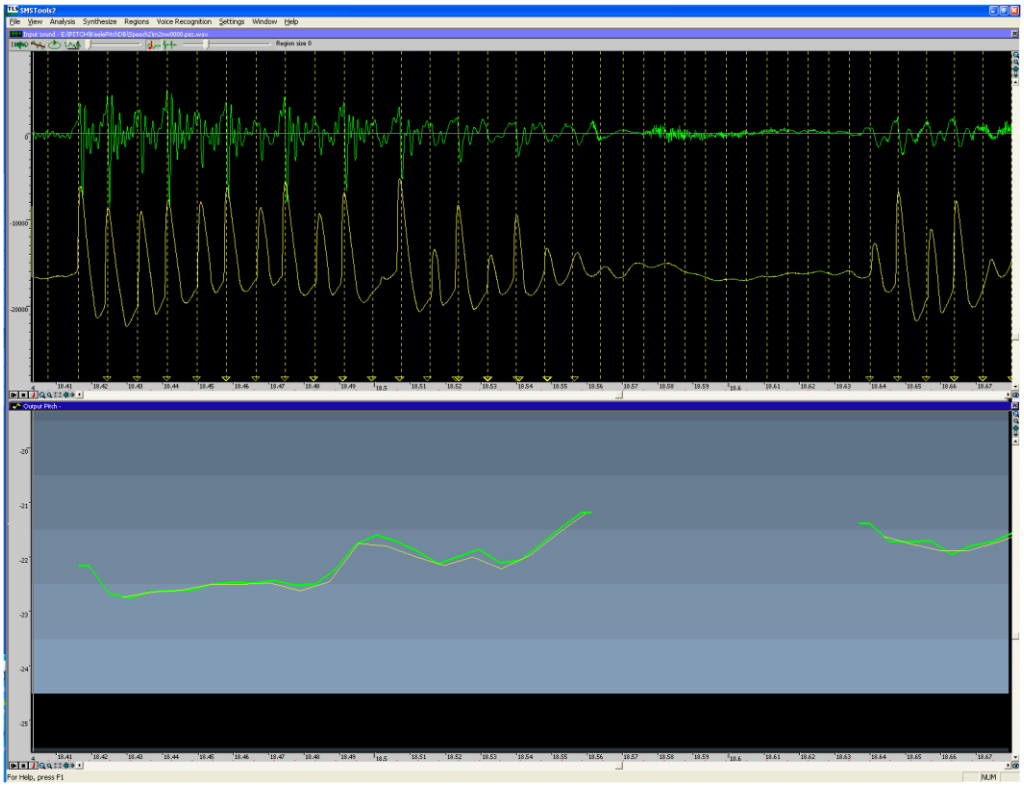


Figure 2.36 Voice pulse onsets computed by the MFPA algorithm in several utterances. Each estimated onset is marked with a vertical line.

(a)



(b)

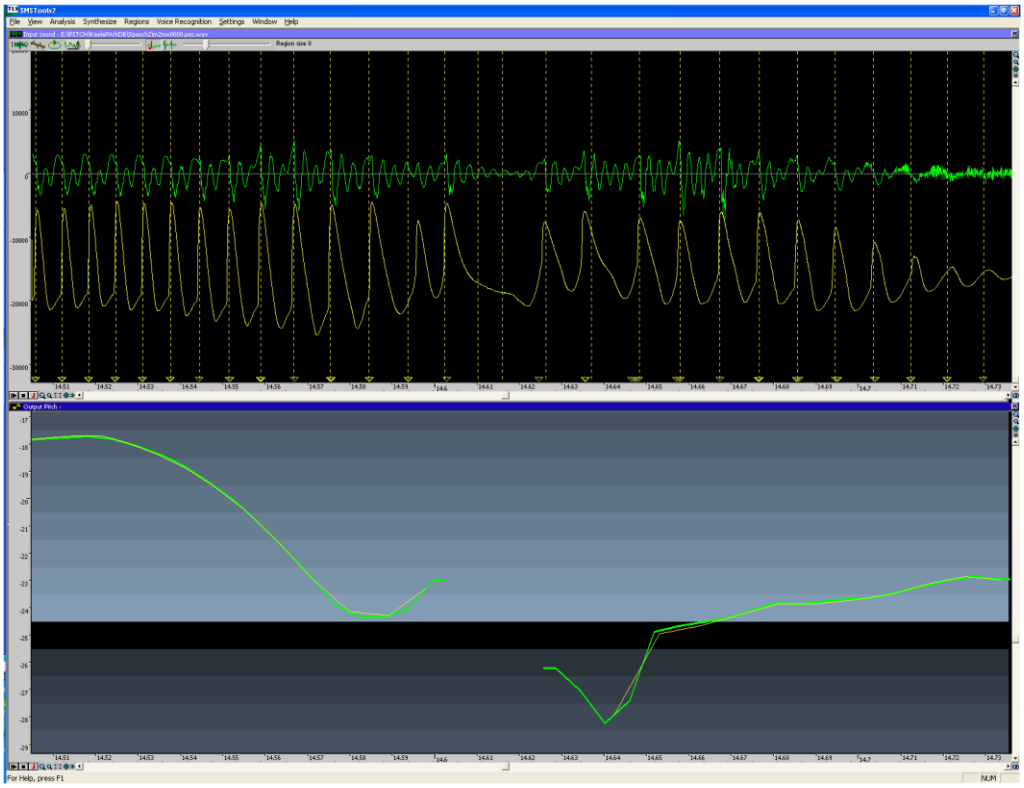


Figure 2.37 Voice pulse onsets computed by the MFPA algorithm in several utterances. Each estimated onset is marked with a vertical line.

(a)

(b)

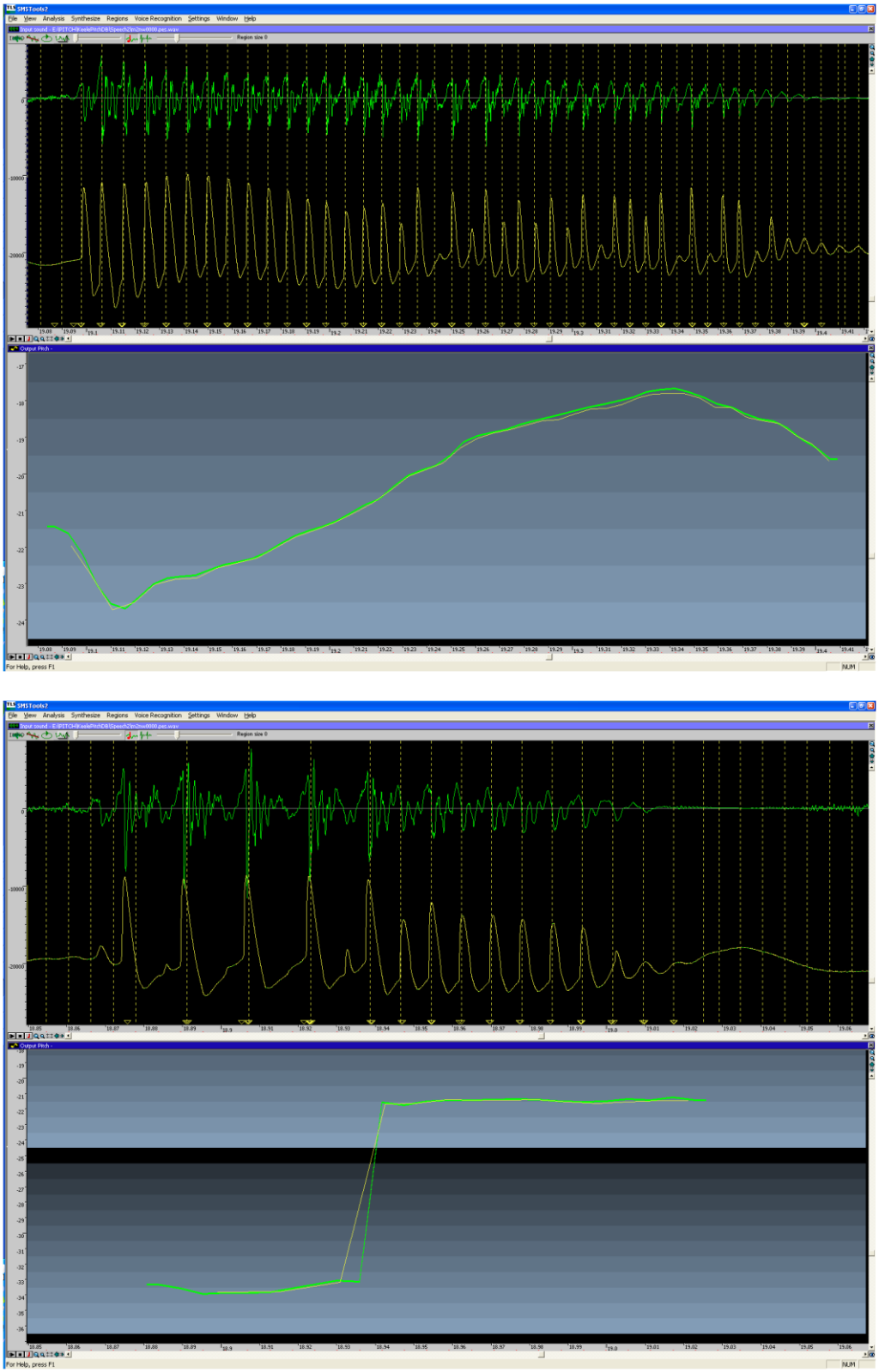


Figure 2.38 Voice pulse onsets computed by the MFPA algorithm in several utterances. Each estimated onset is marked with a vertical line.

(a)

(b)

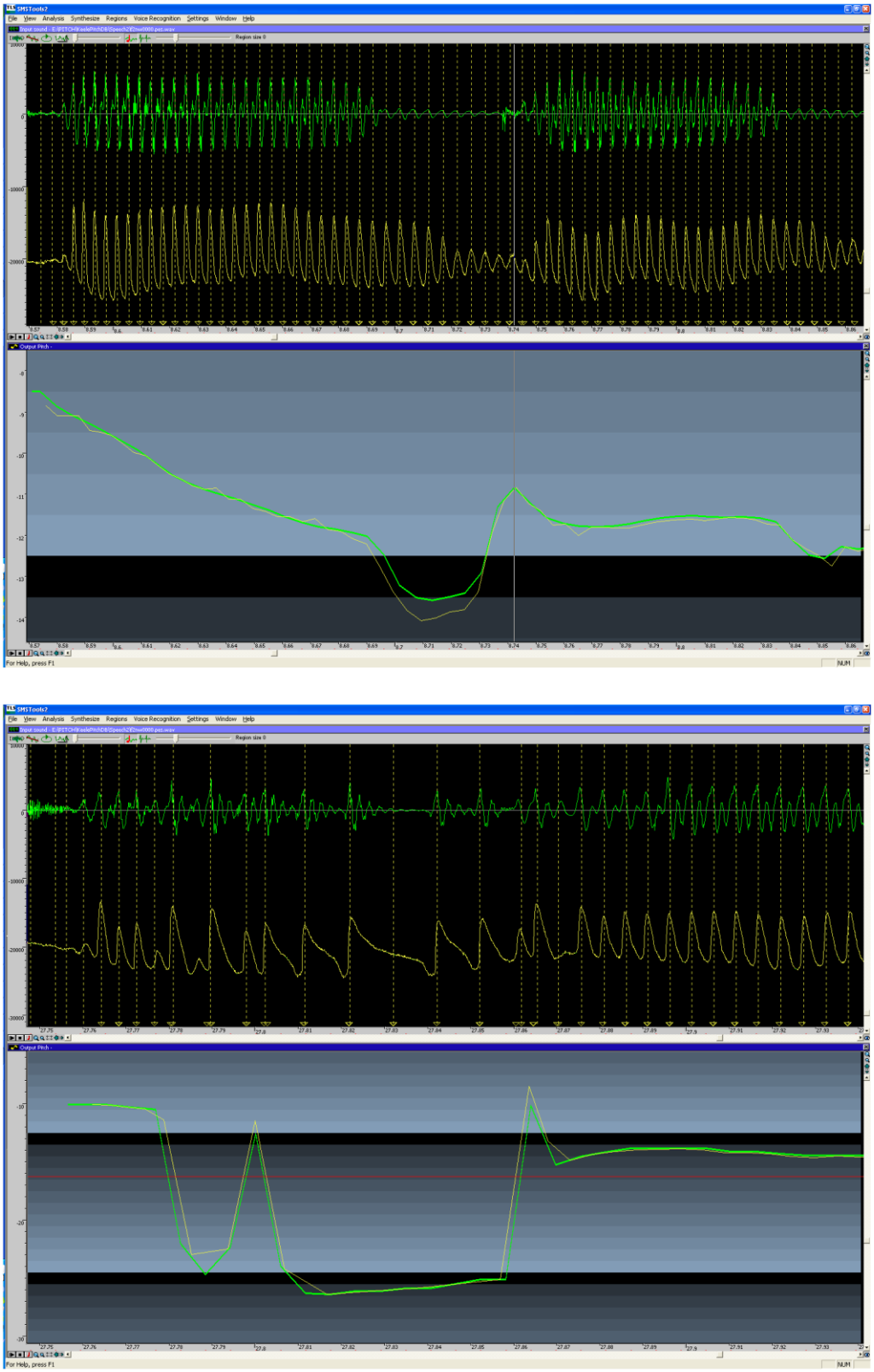


Figure 2.39 Voice pulse onsets computed by the MFPA algorithm in several utterances. Each estimated onset is marked with a vertical line.

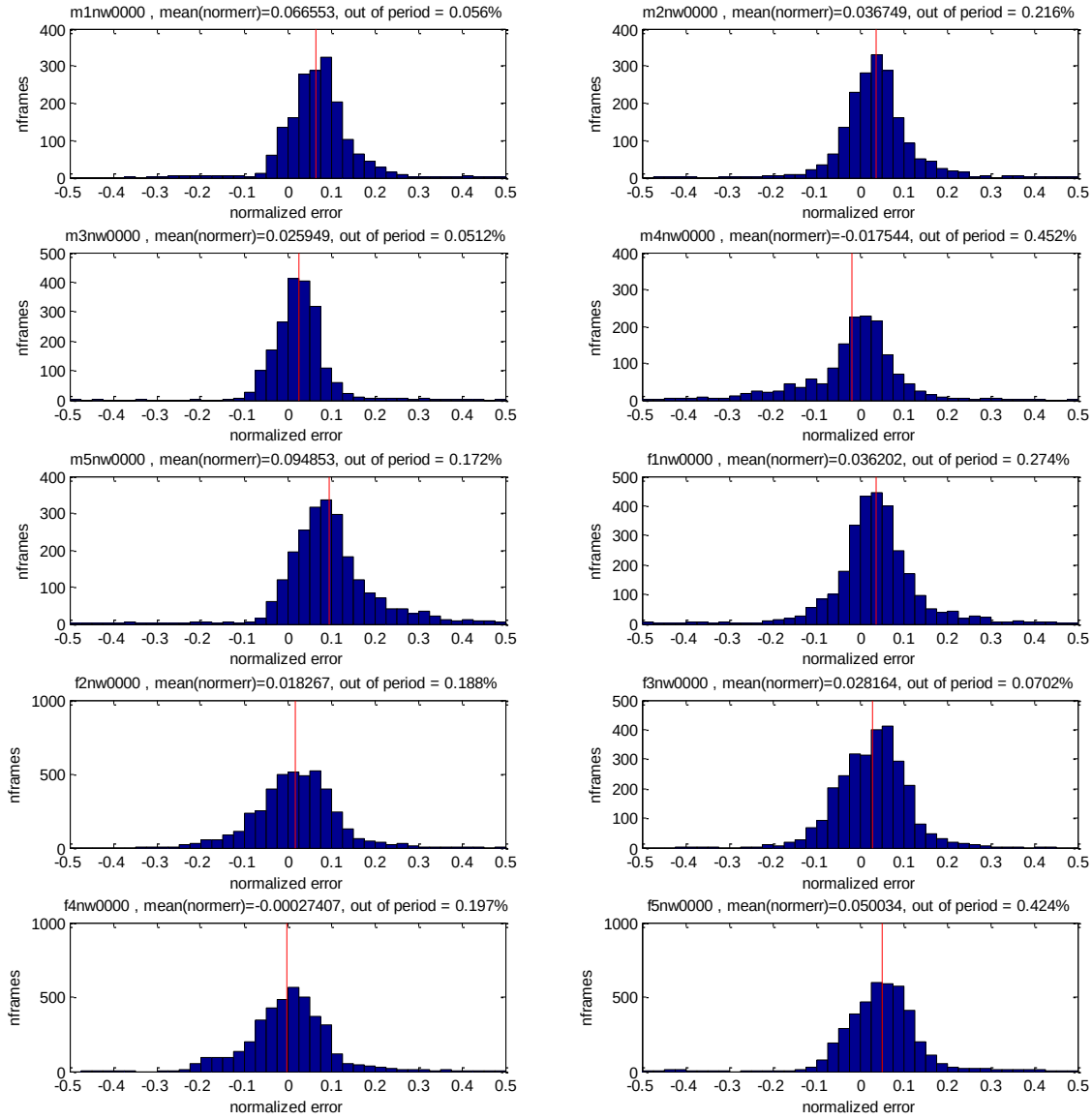


Figure 2.40 Histograms of e_{LM} errors for each file in the Keele Pitch Database.

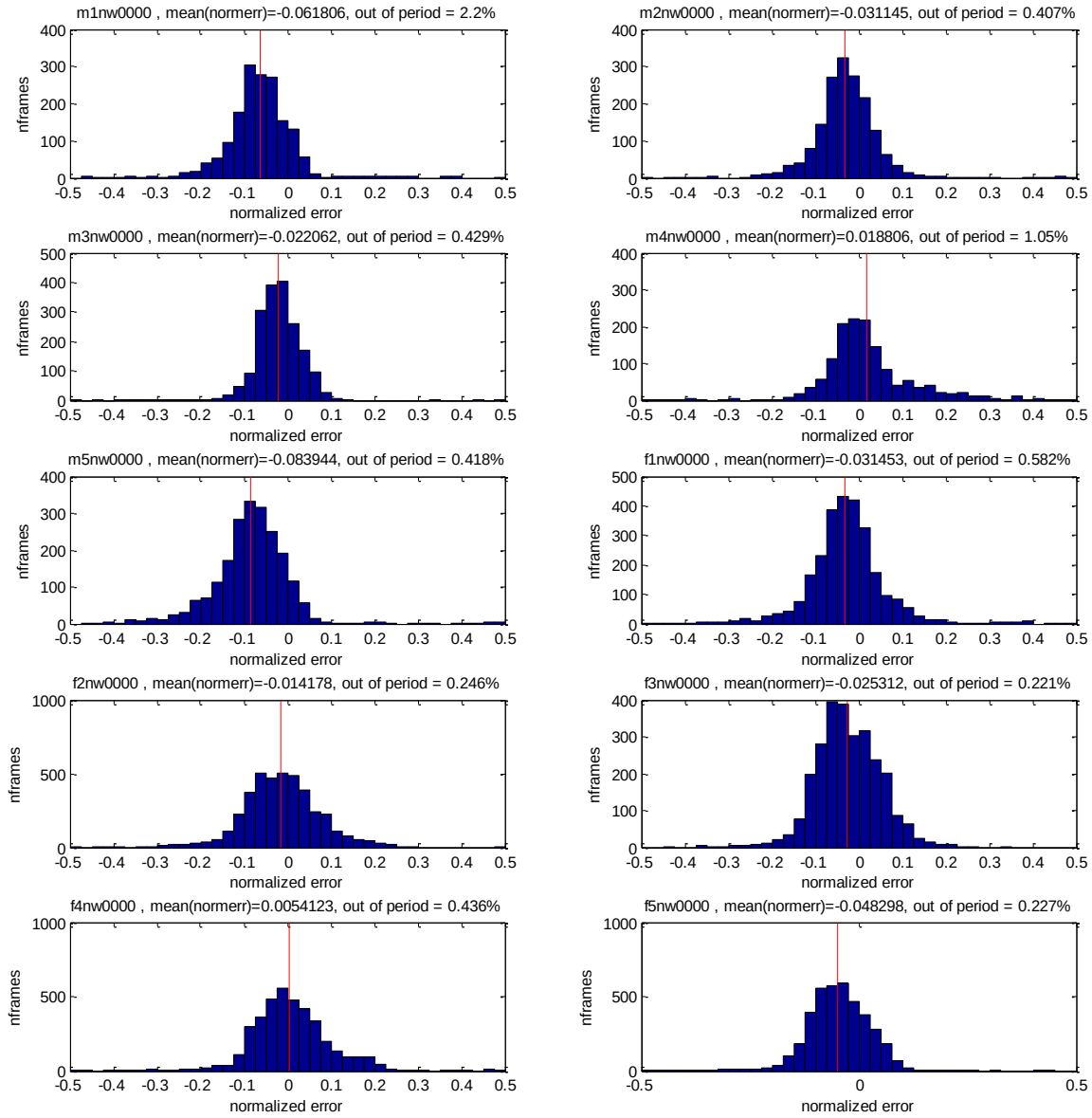


Figure 2.41 Histograms of e_{ML} errors for each file in the Keele Pitch Database

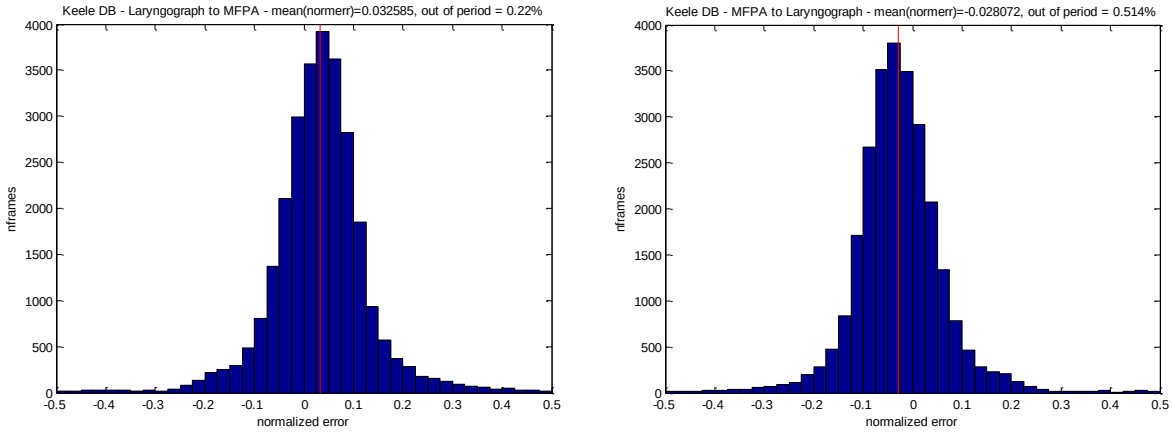


Figure 2.42 Histograms of e_{LM} and e_{ML} errors for the whole Keele Pitch Database. For e_{LM} , 76.27% of cases fall within -0.1 and 0.1, 89.11% within -0.15 and 0.15, and 94.17% between -0.2 and 0.2. For e_{ML} , 78.31% of cases fall within -0.1 and 0.1, 90.85% within -0.15 and 0.15, and 95.39% between -0.2 and 0.2.

2.2.4 Pulse Sequence Irregularities

Several interesting voice transformations relate to irregularities in the pulse sequence or vocal disorders, which have been largely studied as pathology in the field of phoniatrics. However, in the context of popular singing, vocal disorders not always come from pathologies but often healthy voices use them as an expressive resource. In this section, we present some algorithms that focus on two of the most common singing expressions related to this topic: rough and growl (Loscos and Bonada 2004). Our aim is to achieve natural rough and growl effects in order to enhance the singing voice.

Unlike many of the studies concerning vocal disorders, the algorithms presented here arise from spectral models and work with frequency domain techniques instead of working with physical models and time domain techniques. More concretely, both rough and growl algorithms make use of the sinusoidal model by modifying existing harmonic trajectories and adding new ones.

- ROUGHNESS TRANSFORMATION

  Roughness in voice can come from different pathologies such as biphonia, or diplophonia, and can combine with many other voice tags such as hoarse or creaky (Titze 1994). Here we do not stick to the rigorous rough voice definition but we refer to rough voice as the one due to cycle-to-cycle variations of the fundamental frequency (jitter), and the period amplitude (shimmer). Being aware of such nomenclature, we can say the most common techniques used to synthesize rough voices work with the source-filter model and reproduce both jitter and shimmer aperiodicities in time domain (D. Childers 1990). These aperiodicities can be applied to the voiced pulse train excitation by taking real patterns that have been extracted from rough voices recordings or by using statistical models (Schoentgen 2001).

  The main idea behind our algorithm for turning a normal phonation voice into a rough voice is to take the original input signal, transpose it down by an integer factor $T_{pitch}=1/N$, take then the transposed signal, and overlap it with randomly delayed versions of it to resynthesize the original voice with its new rough character. The delay Δ_i applied to each of the N shifted versions of the transposed signal is

$$\Delta_i = iT_0 + X_i \tag{2.43}$$

  where T_0 is the period duration and X_i is a zero mean random variable. These differently shifted N versions of the transposed signal are then scaled by a unity mean random variable Y_i and finally overlapped, as illustrated in Figure 2.43A. The concrete case of $N=2$ is

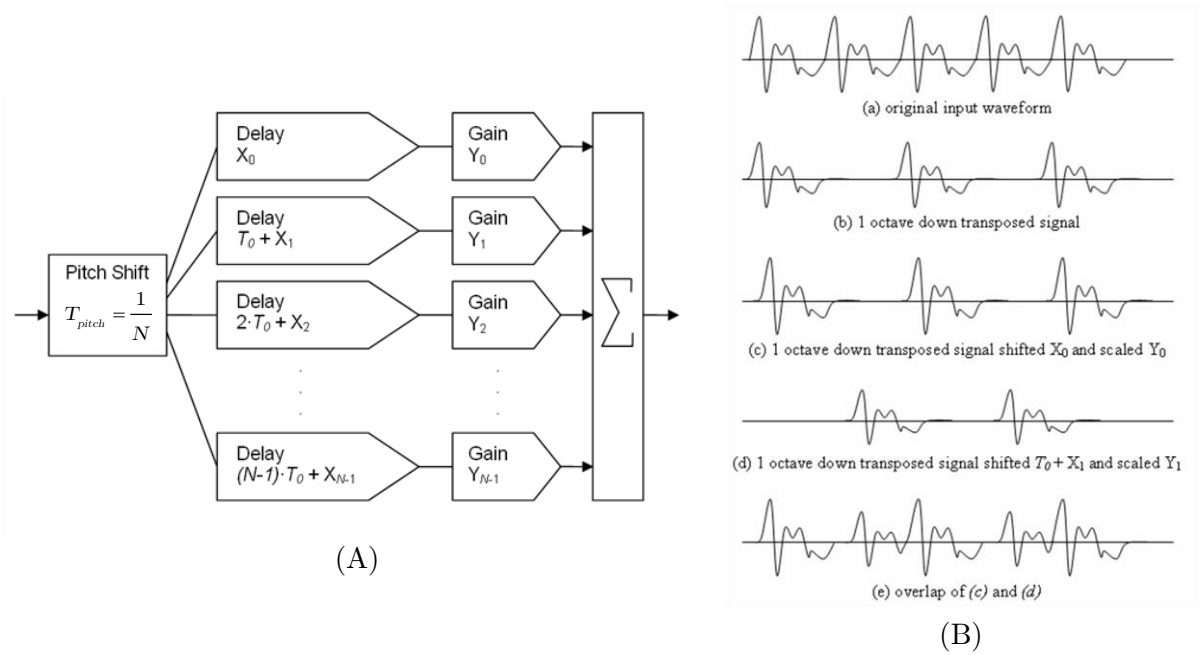


Figure 2.43 Rough transformation algorithm. (A) contains the block diagram of the rough emulator. (B) is a figurative representations of the waveforms at different steps of the algorithm for N=2

illustrated in Figure 2.43B. The input signal is transposed one octave down in (b). Then scaled and delayed versions of it (c and d) are overlapped to generate an irregular pattern (e) with a period of 2 pulses. Note that producing such patterns in a constant hop-size processing framework would require taking into account the relationship between the hop-size and fundamental period. However, our current implementation does not consider yet such relationship.

With the aim of applying this transformation in real-time, we have designed a greatly simplified version of the rough algorithm with very little extra computational cost. All operations are performed directly in the spectrum avoiding the need of overlapping several time domain signals. The one octave down transposition is accomplished by adding pure sinusoids to the spectrum at middle frequencies between harmonics (i.e. $1.5f_0$, $2.5f_0 \ldots$). We have consciously omitted the lower frequency $0.5f_0$, since in rough utterances it has very low energy and can be neglected. The amplitude of each sinusoid is obtained by interpolation of the harmonic spectral envelope. The phase is computed as the one of the closest upper or lower harmonic with the corresponding offset correction, i.e.

$$\phi_r = \phi_g + 2\pi f_g \left(\frac{f_r}{f_g} - 1 \right) \Delta_t \tag{2.44}$$

where r is the sinusoid index, and g is the index of the corresponding harmonic. The jitter and shimmer stochastic variables of the first channel are set to its mean value $X_0 = 0$ and $Y_0 = 1$. For the second channel X_1 and Y_1 are set to have a normal distribution with variances 3% of the input signal period, and 3dBs respectively.

The random scaling due to Y_1, as well as the random delay due to $T_0 + X_1$ are only applied to the sub-harmonics. The only reason for doing such oversimplification is to reduce the computational cost of the algorithm since with this only half of the peaks to which the random variables should be computed and applied are actually processed. The delay is applied in frequency domain by adding the corresponding constant slope phase offset to the phase of the sub-harmonics spectrum as represented in the spectrum of Figure 2.44c. Moreover, only sub-harmonics inside the $[f_0, 8000]$ Hz band are added to the spectrum. Upper sub-harmonics are not significantly relevant in terms of acoustic perception to

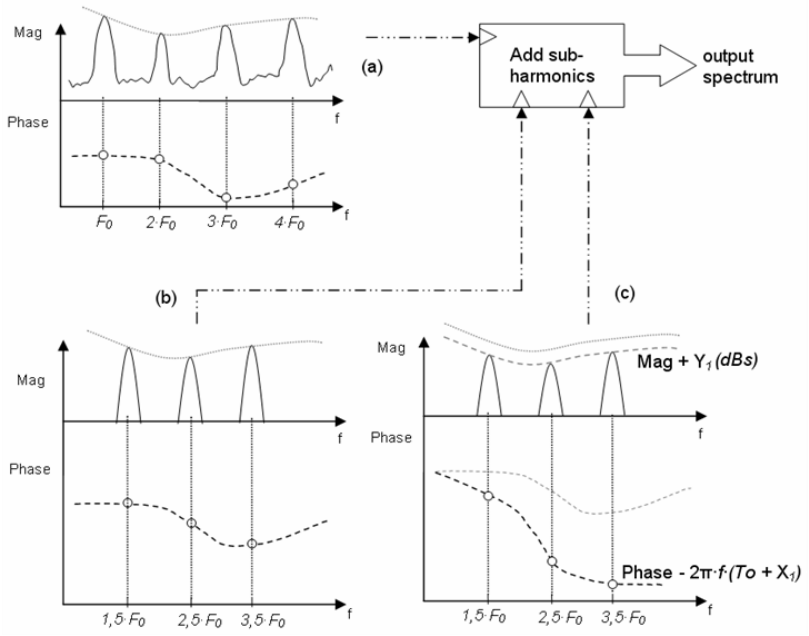


Figure 2.44 Simplified rough transformation algorithm

reproduce the rough effect, and the first sub-harmonic (placed at $0.5f_0$) is assumed to be, based on the observations, almost always masked by the fundamental peak.

❖ GROWL TRANSFORMATION

Singers in jazz, blues, pop and other music styles often use the growl phonation as an expressive accent. Perceptually, growl voices are close to other dysphonic voices such as hoarse or creaky, however, unlike these others, growl is always a vocal effect and not a permanent vocal disorder. According to (Sakakibara, et al. 2004) growl comes from simultaneous vibrations of the vocal folds and supra glottal structures of the larynx. The vocals folds vibrate half periodically to the aryepiglottic fold vibration generating sub-harmonics.

The growl algorithm presented here adds these sub-harmonics in frequency domain to the original input voice spectrum to try to emulate the growl phonation. These sub-harmonics follow certain magnitude and phase patterns that have been extracted from the spectral analysis and observation of real growl voice recordings. The behavior of the growl sub-harmonics in terms of magnitude and phase vary quite a lot from one voice to another, from one pitch to another, from one phrase to another, etcetera. However, certain patterns appear quite frequently. These patterns, which are explained next, are the ones that the proposed growl effect applies.

If a growl utterance is observed in time domain, it is most of the time easy to recognize which is the real period of the signal and which is the macro period due to growling as it is in Figure 2.45. In the observations made growl phonation appeared to have from two to five sub-harmonics. In Figure 2.45 example, the spectrum presents three sub-harmonics between harmonics, placed at frequencies $f_0 \cdot \left(h+\left(k+1\right)/4\right)$ for $h \in \left[0, H-1\right]$ and $k \in \left\{0,1,2\right\}$. Consequently, four inner periods can be distinguished in between a growl macro period.

Regarding magnitudes, in the frequency band that goes from the fundamental up to approximately 1500 Hz, the sub-harmonic peaks are commonly located below the spectral envelope defined by the harmonic peaks. In this band, the closer the sub-harmonic is to the nearest harmonic, the higher its magnitude is. In the upper band, from approximately 1500Hz to half the sampling rate, sub-harmonics go along with the harmonic spectral shape.

Regarding phases, for a growl utterance with N sub-harmonics, the typical behavior of the phases of the sub-harmonics is to get approximately aligned with the phase of the left harmonic peak every $N+1$ periods as illustrated in Figure 2.46A. Concerning harmonic

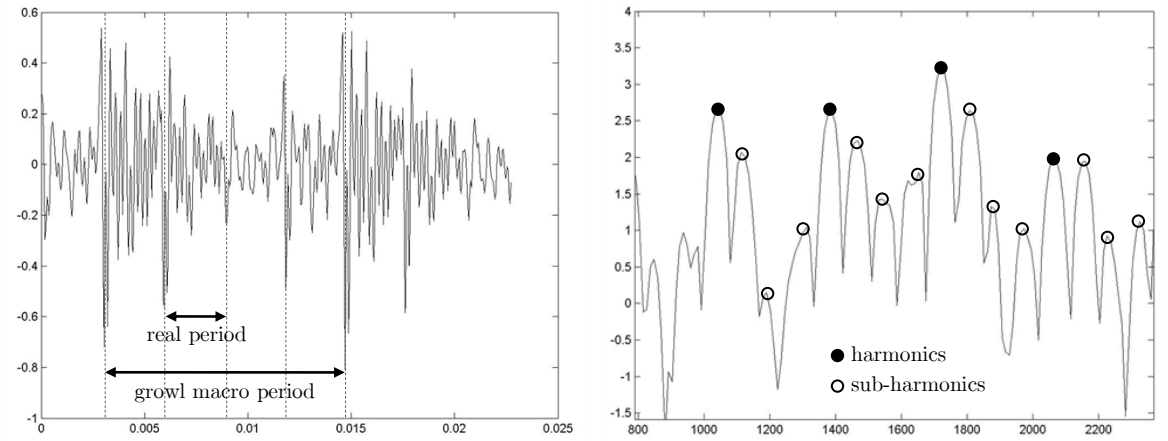


Figure 2.45 Waveform (left) and spectrum representation of a growl utterance

peaks, during a fundamental period harmonic h advances one cycle less than the next harmonic $h+1$. Thus, in between them, sub-harmonic phases can be generally expressed as

$$\phi_h^k = \phi_h + \frac{2\pi}{N+1}(k+1)p \quad , \quad \text{for } k=0,1,2 \text{ and } p=0,1,2,3 \tag{2.45}$$

being p the inner period index ($p=0$ for Figure 2.46Aa and $p=3$ for Figure 2.46Ad), k the sub-harmonic peak index in between consecutive harmonic peaks, and N the number of sub-harmonics.

Based on the most frequently observed growl spectral symptoms, the proposed method fills the original spectrum with sub-harmonics. However, since growl is not a permanent disorder, the effect cannot be applied all along the performance. For this reason, the algorithm includes an automatic growl deep control (as shown in Figure 2.46C) by which we determine how much of the effect has to be applied at each time depending on the input singing voice. This control is mainly based on the first derivatives of the fundamental frequency and energy. It sets how many sub-harmonics have to be added, plus their phase and magnitude patterns.

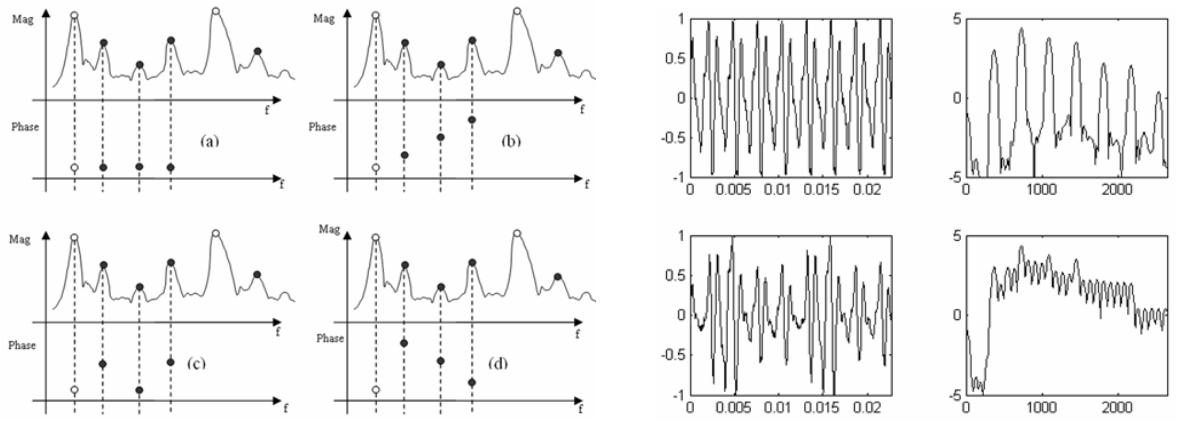
With such method, the transformation is able to reproduce growl sub-period amplitude patterns as the one shown in Figure 2.46B. In the waveform view of the transformed voice we can observe how each one of the four periods of the growl macro period is set to have a different amplitude. The amplitude modification observed is achieved by applying phase alignment patterns extracted from real growl analysis to the sub-harmonics.

DISCUSSION

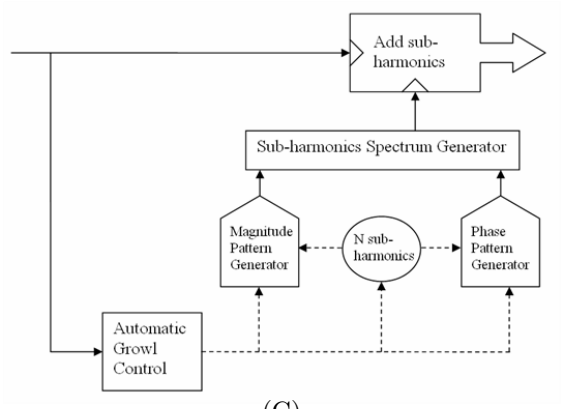
The rough and growl algorithms presented here have proven to be suitable in changing the voice character. However, the naturalness of the effect is highly dependent on the input voice characteristics. For different types of voice, different tessitura, different expressions, etcetera, different values of the transformation parameters are required. In that sense, a dynamic automatic control over these parameters should be found. In the growl effect, this control would have to combine and work together with the automatic growl deep control.

In the growl effect patterns extracted from real growl recordings are roughly reproduced in synthesis. This means that the period-to-period amplitude envelope inside a growl macro-period does not only depend on the phase alignment of the sub-harmonics but also on their amplitudes. However, it is a tedious task to find the sub-harmonic amplitudes and phase alignment required for a certain made-up amplitude envelope. It is also remarkable that no control over the jitter is available with the current growl algorithm.

Concerning the rough effect, two interesting directions come up from the current method. First, perform a study of the system without any of the simplifications made. Second, take into account the hop-size to input period relationship and the analyzed frame history so that the system could follow



(A) (B)



(C)

Figure 2.46 Growl transformation. (A) shows the representation of the phase behavior of a growl voice in the beginning of four consecutive periods (a,b,c,d) for $N=3$ sub-harmonics. (B) shows the waveform *(1)* in seconds and magnitude spectra *(2)* in Hz from both original *(a)* and transformed *(b)* voices. (C) shows the block diagram of the growl implementation.

the period. In that situation, increasing N would really improve the resolution of the algorithm. This could be considered as going towards a fusion of both techniques in which we would have control over the period-to-period jitter and shimmer.

2.2.5 Synthesis of harmonic trajectories

Equation (2.23) indicates how the original signal $s(n)$ can be reproduced out of the sinusoidal components estimated at equidistant time instants. However, what is really interesting and useful is to modify the parameters of the sinusoidal model and synthesize a new sound with different characteristics. In that case, adding a prime to distinguish the synthesis variables, we obtain

$$s'(n) = \sum_{m=0}^{M'-1} s'_m\left(n - m\frac{\Delta'_t}{T'_s}\right) \cdot w'_{ov}\left(n - m\frac{\Delta'_t}{T'_s}\right) \tag{2.46}$$

where $s'(n)$ is the synthesized signal, Δ'_t is the synthesis time increment (or *hop size*), T'_s the synthesis sampling period, M' the number of frames, and s'_m the synthesis frame signal, computed from the synthesis sinusoidal parameters as

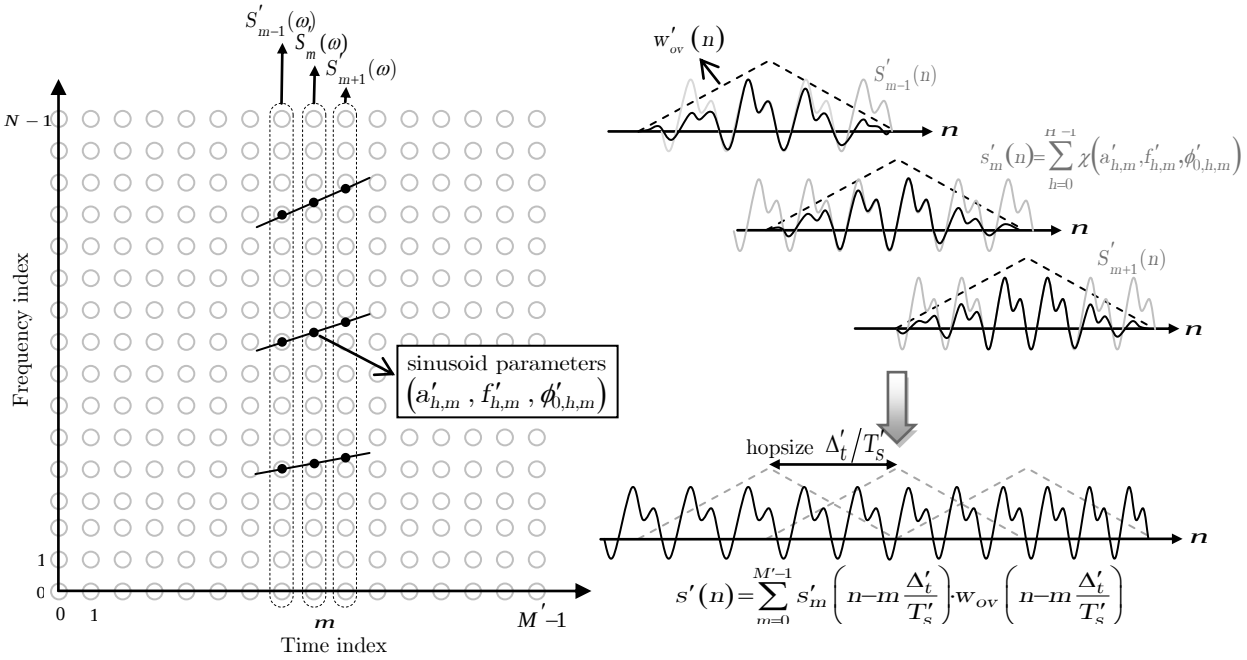


Figure 2.47 Harmonic trajectory synthesis. On the left side, there is the discrete time-frequency plane with a set of estimated sinusoid components. On the right we see three synthesized frames ($s'_{m-1}(n)$, $s'_m(n)$ and $s'_{m+1}(n)$), and how they are overlapped so to obtain the synthesis signal $s'(n)$. Note that both frame and overlapping window lengths can be different and longer than the synthesis hop size.

$$s'_m(n) = \sum_{h=0}^{H'-1} \chi\left(a'_{h,m}, f'_{h,m}, \phi'_{0,h,m}\right) \tag{2.47}$$

being $\chi\left(a'_{h,m}, f'_{h,m}, \phi'_{0,h,m}\right)$ a function or method that computes the discrete time domain signal of a sinusoid with parameters $a'_{h,m}$, $f'_{h,m}$ and $\phi'_{0,h,m}$. Figure 2.47 illustrates the different steps involved in the synthesis. The most typical methods for rendering harmonics are discussed in the following.

◊ *HARMONICS AS SINUSOIDS*

With this method, harmonics are modeled as sinusoids stationary along the frame window. The simplest synthesis method is to use a bank of time-domain oscillators as

$$s'_m(n) = \sum_{h=0}^{H'-1} \chi\left(a'_{h,m}, f'_{h,m}, \phi'_{0,h,m}\right) = \sum_{h=0}^{H'-1} a'_{h,m} \cos\left(2\pi f'_{h,m} n T'_s + \phi'_{0,h,m}\right). \tag{2.48}$$

It is straightforward with the oscillators to synthesize non-stationary sinusoids by using time-varying parameters $a'_{h,m}(n)$, $f'_{h,m}(n)$, $\phi'_{0,h,m}(n)$. For example, parameter estimations of consecutive frames can be linearly interpolated; in that case the amplitude and frequency would be computed as[11]

[11] In this equation the synthesis frame length is assumed to be shorter or equal to twice the distance between consecutive frames.

$$a'_{h,m}(n)=\begin{cases}\left(1+\dfrac{nT'_s}{\Delta'_t}\right)a'_{h,m}-\dfrac{nT'_s}{\Delta'_t}a'_{h,m-1} & if \quad n<0\\ \left(1+\dfrac{nT'_s}{\Delta'_t}\right)a'_{h,m+1}-\dfrac{nT'_s}{\Delta'_t}a'_{h,m} & if \quad n\geq 0\end{cases}$$
$$f'_{h,m}(n)=\begin{cases}\left(1+\dfrac{nT'_s}{\Delta'_t}\right)f'_{h,m}-\dfrac{nT'_s}{\Delta'_t}f'_{h,m-1} & if \quad n<0\\ \left(1+\dfrac{nT'_s}{\Delta'_t}\right)f'_{h,m+1}-\dfrac{nT'_s}{\Delta'_t}f'_{h,m} & if \quad n\geq 0.\end{cases} \tag{2.49}$$

Regarding the phase, assuming frequency changing linearly between consecutive frames, it would be computed out of the frequency values as

$$\phi'_{0,h,m}=\phi'_{0,h,m-1}+2\pi\frac{f'_{h,m-1}+f'_{h,m}}{2}\Delta'_t. \tag{2.50}$$

For simplicity, it makes sense to raise the frame rate to match the sampling rate and to avoid using overlapping windows. In that case, harmonic parameters would be computed for each time index n obtaining

$$s'(n)=\sum_{h=0}^{H'-1}a'_h(nT'_s)\cos\left(2\pi\sum_{k=0}^{n-1}f'_h(kT'_s)T'_s+\phi'_{0,h}(n)\right). \tag{2.51}$$

However, time-domain oscillators are computationally expensive, especially when many sinusoids have to be synthesized in the case of low-pitch utterances[12]. Therefore, it is preferable to use more efficient methods. Probably the most common one is the additive synthesis by IFFT, proposed in (Depalle and Rodet 1990) and (Dutoit 1993), where sinusoids are rendered in frequency domain by computing a few complex bin values around each sinusoid's center frequency, and adding those to the output spectrum. This method considers sinusoids to be stationary along the frame window, which is reasonable for high frame rates. In that case,

$$\begin{aligned}S'_w(k)&=\sum_{n=0}^{N-1}w(n)s'(n)e^{-j2\pi nk/N}=\sum_{n=0}^{N-1}w(n)\left[\sum_{h=0}^{H-1}a'_h\cos\left(2\pi f'_h nT_s+\theta'_{0,h}\right)\right]e^{-j2\pi nk/N}=\\&=\sum_{h=0}^{H-1}\frac{a'_h}{2}\left(W\left(k-f'_hT'_sN\right)e^{j\theta'_{0,h}}+W\left(k+f'_hT'_sN\right)e^{-j\theta'_{0,h}}\right).\end{aligned} \tag{2.52}$$

$W(k)$ can be approximated by the bins with more energy. A Blackman-Harris 92dB window (see Figure 2.48) is a good choice because almost all its energy is located in the main lobe, being the side lobes below 92dB from the maximum. The main lobe is 8 bins wide when no zero-padding is applied. Using the real IFFT algorithm it is enough to fill the positive frequency bins, so then in total only $8H$ bins need to be computed. This does not mean we can forget about the negative frequency components. Indeed, their contribution must be considered as well at boundaries, around zero and half sampling rate frequencies. Therefore,

$$\begin{gathered}S'_w(k)\approx\sum_{h=0}^{H-1}\frac{a'_h}{2}\left(W\left(k-f'_hT'_sN\right)e^{j\theta'_{0,h}}+W\left(k+f'_hT'_sN\right)e^{-j\theta'_{0,h}}\right)\ \forall k\in\left[0,1,...,\frac{N}{2}\right]\ and\ \left|k-f'_hT'_sN\right|\leq 4\\ s'_w(n)\approx rIFFT\left(S'_w(k)\right).\end{gathered} \tag{2.53}$$

[12] Each time domain sample is computed as $s'(n)=\sum_{h=0}^{H'-1}a'_h(nT'_s)\cos\left(2\pi\sum_{k=0}^{n-1}f'_h(kT'_s)T'_s+\phi'_{0,h}(nT'_s)\right)=$ $=\sum_{h=0}^{H'-1}a'_h(nT'_s)\cos\left(2\pi f'_h((n-1)T'_s)T'_s+\phi'_h((n-1)T'_s)\right)$, therefore H cosinus need to be computed plus 3H multiplications $\left(a\cdot\cos(),(2\pi T'_s)\cdot f\cdot n\right)$ and 2H additions. The computational cost is proportional to H and sampling rate, in the order of $F_s\cdot\left(H\cdot Cost_{\text{cosinus}}+3H\cdot Cost_{\text{multiplication}}+2H\cdot Cost_{addition}\right)$ cost per second.

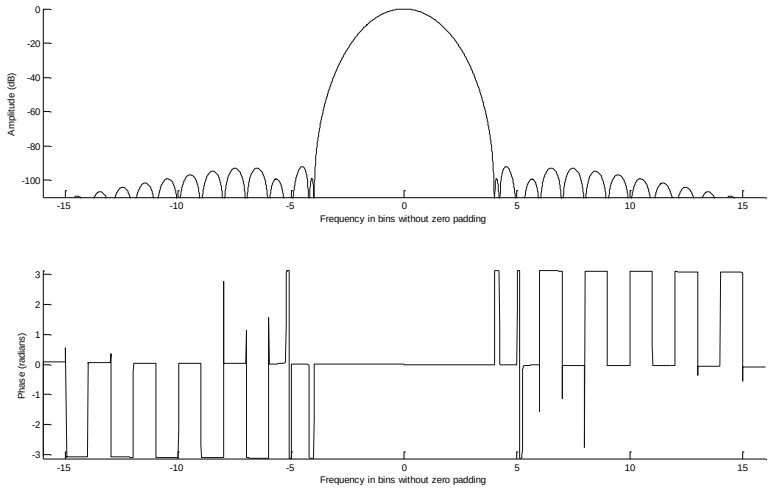


Figure 2.48 Blackman-Harris 92dB window. The main lobe width is about 8 bins when no zero-padding is applied. The side lobes amplitude is below 92dB from the maximum of the main lobe.

Frames are overlapped in order to avoid discontinuities. First, $s'_w(n)$ is divided by the synthesis window $w(n)$ so to approximate a rectangular window. Afterwards, it is multiplied by the overlapping window $w'_{ov}(n)$. Summing up, the synthetic signal is computed as

$$s'(n) = \sum_{m=0}^{M'-1} s'_m\left(n - m\frac{\Delta'_t}{T'_s}\right) \cdot \frac{w'_{ov}\left(n - m\frac{\Delta'_t}{T'_s}\right)}{w\left(n - m\frac{\Delta'_t}{T'_s}\right)}. \tag{2.54}$$

A more refined method was proposed in (LTD 2001), where there is no overlap between consecutive frames, and sinusoids are not restricted to be stationary. In fact, linear time-varying sinusoids are rendered in the spectrum by means of window transform templates[13] to ensure continuity in adjacent blocks.

◊ *HARMONICS AS SPECTRAL REGIONS*

Strictly speaking, this method should be better considered a transformation technique more than a synthesis method. It is based on the phase-locked vocoder (Puckette 1995) and was first proposed in (Laroche and Dolson 1997) for polyphonic signals, and afterwards adapted to harmonic signals in (Laroche 2003). As we have seen before, the STFT of a stationary periodic signal can be modeled as a sum of analysis window transforms located at the harmonic frequencies. In other words, as the convolution of the analysis window transform with a train of frequency domain deltas:

$$S_w(k) = \sum_{h=0}^{H-1} \frac{a_h}{2}\left(W\left(k - f_h T_s N\right)e^{j\theta_{0,h}} + W\left(k + f_h T_s N\right)e^{-j\theta_{0,h}}\right). \tag{2.55}$$

Modifying the parameters of an arbitrary harmonic would result into the window transform shifted to the new frequency position f'_h, scaled by the modified amplitude a'_h, and rotated by the resulting phase $\phi'_{0,h}$.

[13] Those window transform templates are obtained from the start-end amplitude, frequency and phase values of each sinusoidal component in each frame.

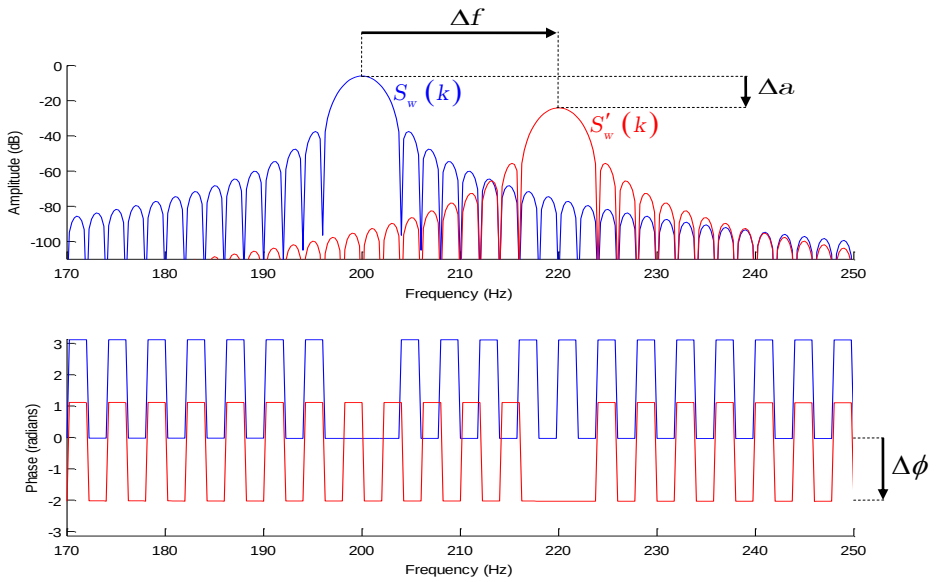


Figure 2.49 Local spectra of two sinusoids. In blue, a sinusoid of 200Hz, amplitude 1 and phase 0 radians at the center time. In red, a sinusoid of 220Hz, amplitude 0.125 and phase -2 radians. The red spectrum $S'_w(k)$ can be approximated as a modification of the blue spectrum $S_w(k)$: a frequency shifting of $\Delta f = 20\text{Hz}$, an amplitude scaling of $\Delta a = -18.0618\text{dB}$, and a phase rotation of $\Delta\phi = -2\,\text{radians}$

$$\begin{aligned}\text{input} &\rightarrow \frac{a_h}{2}\left(W\left(k - f_h T_s N\right)e^{j\theta_{0,h}} + W\left(k + f_h T_s N\right)e^{-j\theta_{0,h}}\right)\\ \text{output} &\rightarrow \frac{a'_h}{2}\left(W\left(k - f'_h T'_s N\right)e^{j\theta'_{0,h}} + W\left(k + f'_h T'_s N\right)e^{-j\theta'_{0,h}}\right)\end{aligned} \tag{2.56}$$

Let us now consider, for simplicity, only positive frequencies, the same sampling rate T_s and a signal $s(n)$ containing a single sinusoidal component with parameters a, f and ϕ_0. In that case, we can approximate[14] the modified spectrum $S'_w(k)$ as a transformation of the input one $S_w(k)$ as

$$\begin{aligned}S_w(k) &= \frac{a}{2}W\left(k - fT_s N\right)e^{j\theta_0}\\ S'_w(k) &\approx \frac{a'}{a}S_w\left(k - (f' - f)T_s N\right)e^{j(\theta'_0 - \theta_0)}.\end{aligned} \tag{2.57}$$

The resulting spectrum $S'_w(k)$ is a frequency shifted version of $S_w(k)$ by $(f' - f)T_s N$ bins, scaled in amplitude by a'/a, and rotated by $\theta' - \theta$ radians. This is illustrated in Figure 2.49. The phase rotation can be performed for each bin of the complex spectrum as a complex multiplication, or as an addition in the case of a spectrum in polar coordinates. The frequency shift is more difficult; the ideal interpolation method would be to apply an enormous zero padding factor when computing the STFT, what would increase the computational cost and the memory requirements to unfeasible levels. Instead, a small zero padding factor plus a first or second order interpolation method would be good enough for most cases. Interpolation results better in polar coordinates than in complex, but thinking of computational cost it might make sense to do it directly in the complex spectrum (there would be no need to do the complex-polar conversion).

Let us now consider the case of a non-stationary sinusoid. We saw in section 2.1 several examples of how the spectrum changes for amplitude and frequency variations in the form $(x_0 + x_1 t)^p$. In all

[14] This is an approximation because boundary conditions are not considered and the discrete spectrum $S_w(k)$ is supposed to be ideally interpolated.

cases, the amplitude spectrum stills shows a clear peak. Applying the above method would mean first to estimate the sinusoid parameters, and then to modify the spectrum as in equation (2.57). Interestingly, the variations of the input sinusoidal parameters are mostly preserved in the output signal because the local spectral shape *(result of the linear process of convolution between the analysis window and the frequency components of the signal)* is approximately the same one. This is a great advantage because if we want to transform audio preserving the temporal behavior of harmonic parameters there is no need to estimate those variations, but just to estimate the spectral peak values. However, the drawback is that since there is not an underlying model for the parameter functions then there is no way to transform them either.

In the case of a voice signal the spectrum contains several harmonics. However, the spectral regions corresponding to each frequency component are not separated but do actually overlap. Even in the case of a perfectly periodic utterance without aspirated noise, the spectrum would be the result of overlapping analysis window transforms at the multiples of the fundamental frequency. Still, it makes sense to process harmonic spectral regions independently as soon as the contribution of other harmonics is negligible or low enough. One way to attain this is to adapt the size of the analysis window to cover several periods, thus increasing the frequency resolution, but unfortunately loosing temporal resolution at the same time. So, the first thing to do is to segment the spectrum into frequency regions associated to each harmonic. As discussed in (Laroche 2003), several strategies are possible. The simplest one is to set the boundaries at the middle frequency between consecutive harmonics. Another approach is to set the boundary at the frequency bin with minimum amplitude.

The sound synthesis process is performed in a similar way to the technique detailed in the previous section (*harmonics as sinusoids, p80*). It consists on computing the synthesis spectrum by iteratively adding to it the spectral regions corresponding to each harmonic to synthesize. This correspondence is determined by a mapping function $\gamma(h)$ from output harmonic indices to input harmonic indices, where the indexing is zero-based. This can be expressed as

$$S'_w(k) = \sum_{h=0}^{H'-1}\left[\frac{a'_h}{a_{\gamma(h)}} e^{j\left(\theta'_{0,h}-\theta_{0,\gamma(h)}\right)} S_w(k-x)\,\Upsilon_{\gamma(h)}(k-x)\right]$$
$$\Upsilon_h(k) = \begin{cases} 1 & if \quad k \in [b_h, e_h] \\ 0 & if \quad k \notin [b_h, e_h] \end{cases} \qquad x \triangleq \left(f'_h - f_{\gamma(h)}\right) T_s N, \tag{2.58}$$

where $\Upsilon_h(k)$ is a square window function that is equal to 1 for all the bins within the region corresponding to the h^{th} harmonic, and each region is determined by the begin and end indices b_h and e_h.

HARMONIC MAPPING FUNCTION

Regarding the mapping function $\gamma(h)$, the simplest mapping is the identity ($\gamma(h)=h$). However this is often not a good choice because, as we will see in section 0, the spectral region contains both harmonic and noisy components, and therefore the aspirated noise in voiced utterances is shifted in frequency. For instance, in an octave up pitch transposition, the noise component of the h^{th} harmonic, which is located around hf_0' Hz, will be generated from the $\gamma(h)=h$ harmonic of the input signal located around hf_0 Hz. The effect on the timbre of the aspirated noise is similar to the typical *Mickey Mouse* effect and sounds unnatural. Moreover, in a downwards transposition the higher harmonics would have no source spectral region or, if the assignment is limited to the available indices, would probably use noisy and unstable spectral bands. Instead, (Laroche 2003) proposes to select the closest input harmonic in frequency[15]. This approach effectively reduces the frequency shifting of the aspirated noise, which is limited to half the synthesis fundamental frequency $\Delta f \le |f'_0/2|$ for each harmonic. This is illustrated in Figure 2.50 and by the input reference audio [2] and its transformations: [3] and [4] with identity mapping, [5] and [6] with closest frequency mapping. Listening to them it becomes obvious that the latter examples sound better and much more natural.

[15] $\gamma(h) = g$ where g minimizes $\left|f'_h - f_g\right|$, which leads to $\gamma(h) = round\left(h\frac{f'_0}{f_0}\right) = round\left(h \cdot T_{pitch}\right)$, being T_{pitch} the pitch transposition factor.

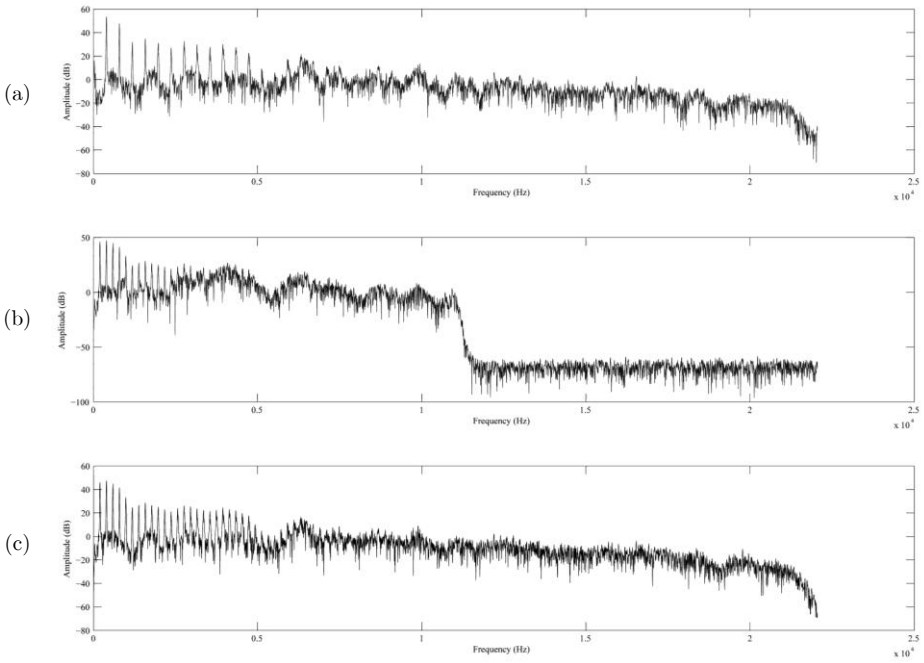


Figure 2.50 Comparison of different harmonic mapping strategies in a one octave down transposition ($T_{pitch} = 0.5$). (a) original audio [2]. (b) identity mapping, audio [3]. (c) closest frequency mapping, audio [5]. In (a) clear spectral peaks appear below 5 Khz, whereas the noise is comparable to the harmonic amplitude at higher frequencies. In (b) the right part of the spectrum has no source region so the amplitude decays strongly. Leaving this aside, the spectral envelope defined by the harmonics follows the one of the input signal. However, the frequency band below approximately 2.5 Khz shows clear peaks, while the rest do not, thus the aspirated noise timbre has been compressed. In (c) clear peaks appear up to approximately 5 Khz, so both harmonic and noise timbres are preserved.

However, another problem occurs when peaks in valleys are raised. Those peaks are typically noisy and unstable, often due to the contributions of surrounding peaks with higher amplitude. Therefore, artifacts that were not previously heard will become audible and sound unnatural. This is illustrated in Figure 2.51, in the spectrum (f), resulting from a timbre expansion $T_{timbre} = 1.18$ transformation, where the fifth harmonic (index 4) is raised. One way to improve these artifacts is to use the joint formant-pitch-modification mapping method proposed in (Laroche 2003)[16], which uses both timbre scaling $\overline{T}_{timbre}(f)$ function and pitch transposition T_{pitch} factor in the mapping function

$$\gamma(h) = round\left(\frac{\overline{T}_{timbre}^{-1}\left(hT_{pitch}f_0\right)}{f_0}\right). \tag{2.59}$$

Indeed, in the same figure, spectrum (g) shows the same transformation but using the latter mapping function. The fifth harmonic uses the source index 3 spectral region that is not noisy and has a higher amplitude. Nevertheless, this harmonic assignment does not solve all artifacts. Certainly, (h) view

[16] In fact, the author does not propose this mapping as a strategy conceived for minimizing artifacts produced by raising noisy peaks, but conceived for roughly achieving the target timbre envelope without modifying the amplitude of each harmonic's spectral amplitude. In that case equation (2.58) would become

$$S'_w(k) = \sum_{h=0}^{H'-1}\left[e^{j\left(\theta'_{0,h}-\theta_{0,\gamma(h)}\right)}S_w(k-x)\,\Upsilon_{\gamma(h)}(k-x)\right]$$

$$\Upsilon_h(k) = \begin{cases} 1 & if \quad k \in [b_h, e_h] \\ 0 & if \quad k \notin [b_h, e_h] \end{cases} \qquad x \triangleq \left(f'_h - f_{\gamma(h)}\right)T_S N$$

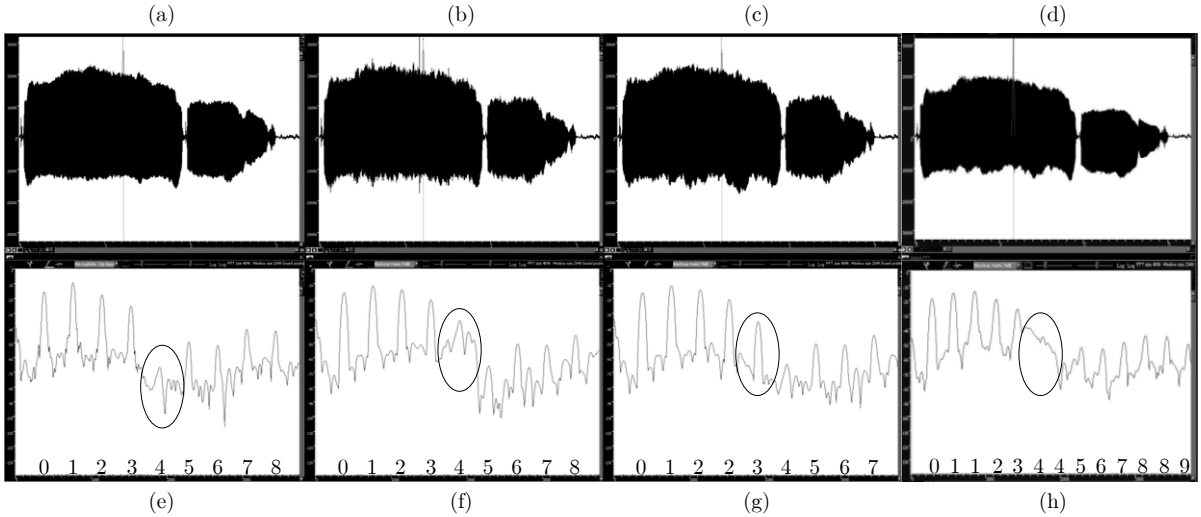


Figure 2.51 Problems found when rising peaks in valleys. On top we see (a) the waveform of a female singing the word "gardens" (audio [7]), and three modifications of it using (b) the closest frequency mapping (audio [8]) and (c) and (d) the joint formant-pitch-modification mapping (audios [9] and [10]). The first two modifications correspond to a timbre expansion of $T_{timbre} = 1.18$ without pitch transposition, whereas (d) is a pitch transposition of $T_{pitch} = 0.75$. The bottom figures show their respective amplitude spectra and the harmonic mapping indices. The harmonic around 1.9Khz is surrounded by an ellipse in all four views. That harmonic is located in an amplitude valley in the original signal spectrum (e) and the harmonic-to-surrounding-noise amplitude ratio is very low. In (f) it is raised and adds unnatural artifacts to the voice. In (g) index 3 harmonic is used instead and sounds much better. However, in (h) index 4 harmonic is used for the two transposed harmonics around 1.9Khz, producing audible artifacts.

corresponds to the spectrum of a $T_{pitch} = 0.75$ transformation. Due to the rounding operation the source index fourth harmonic is used for rendering the sixth synthesis harmonic[17], being raised in amplitude and introducing artifacts (listen to audio [10]).

In order to remove those annoying artifacts we propose to choose the best region among the selected source harmonic and its two neighbors, considering the following constraints:

1. Select the most-stable/less-noisy peak.
2. Minimize the number of region index changes.

The first constraint aims to avoid the artifacts just exposed. On the other hand, the second constraint is important because whenever the source index changes, two spectral regions coming from different harmonics are concatenated and, even though the harmonic components concatenate smoothly[18], the frequency components around them probably do not and could occasionally introduce artifacts.

[17] The sixth harmonic corresponds to $h = 5$ since the indexing is zero-based. We get then

$$\gamma(h) = round\left(\frac{\overline{T}_{timbre}^{-1}\left(hT_{pitch} f_0\right)}{f_0} \right)\Bigg|_{\substack{T_{timbre}=1 \\ T_{pitch}=0.75 \\ h=5}} = round(3.75) = 4$$

Besides, given that in the rounding operation the value 0.5 is mapped to the left, $\gamma(6) = round(4.5) = 4$, as shown in Figure 2.51

[18] Assuming synthesis harmonic trajectories are smooth, then harmonic components concatenate smoothly because, as stated in equation (2.58), each source spectral region is transformed so that its amplitude, frequency and phase match those ones of the harmonic trajectory.

We could think of several ways of estimating the noisiness factor of the h^{th} spectral region, noted as N_h. For instance, as the ratio between the estimated harmonic amplitude and the mean amplitude of the spectral region, or also as the ratio between its arithmetic and geometric means. In both cases, the greater the noisiness factor the less noisy the spectral region. However, the above computations require considering all the bins within the spectral region, and therefore might be computationally intensive. As an alternative, we propose to consider the slope of the harmonic envelope as follows. Since noisy peaks are often located in amplitude valleys, they will have less amplitude than at least one of the surrounding harmonics. Therefore, a significant amplitude difference is expected. We propose to set a threshold α for the amplitude difference function $d_a(h)=a_h-a_{h-1}$ as the decision criterion. Thus, if $d_a(\gamma(h))<-\alpha$ we choose $\gamma(h)-1$, else if $d_a(h+1)>\alpha$ then we choose $\gamma(h)+1$, and otherwise we select $\gamma(h)$. In our experiments $\alpha=15$dB has shown to be a reasonable threshold value. Moreover, adding to this threshold a hysteresis of width β helps to fulfill the second constraint we have proposed. With this approach the artifacts present in the previously referred audio [10] almost disappear (listen to audio [11]).

SPECTRAL REGION RENDERING STRATEGIES

The method previously detailed modifies the harmonic spectral regions in three aspects: frequency shifting, amplitude scaling and phase rotation. However it does not modify their width. Indeed, all the frequency components within the spectral region are shifted in frequency by the same amount, so that the frequency distances between each inner component are preserved. Therefore, in the case of upwards pitch transposition transformations, this synthesis method produces output spectra with amplitude gaps between consecutive harmonics of width proportional to the transformation ratio T_{pitch}. Those gaps might somehow affect the perception of the aspirated noise, maybe reducing its perceived energy, or might introduce audible artifacts. On the contrary, in the case of a downwards pitch transposition there will be no amplitude gaps, but spectral regions will overlap and the perception of the noise level might rise. In order to evaluate this and other aspects we propose the following different strategies for rendering spectral regions.

❖ Strategy 1: adaptive region segmentation

We propose to set the source spectral region width to match the synthesis pitch f_0' independently of the transposition factor. This way, the source region boundaries are computed as $b_h=f_h-f_0'/2$ and $e_h=f_h+f_0'/2$. In the case of downwards transposition synthesis regions will not overlap, whereas in the case of upwards transposition there will be no amplitude gaps but harmonics present in the input signal might appear at non-harmonic positions in the synthesized signal. Therefore, source spectral bins around input harmonics different to the one set by the mapping function should not be used. This can be expressed by modifying the Υ_h function as

$$\Upsilon_h(k)=\begin{cases}1 & \text{if} \quad \begin{cases} k\in[b_h,e_h] \\ |k-f_j|>\alpha \quad \forall j\neq h \end{cases} \\ 0 & \text{otherwise}\end{cases} \tag{2.60}$$

where α specifies the distance in bins to a harmonic.

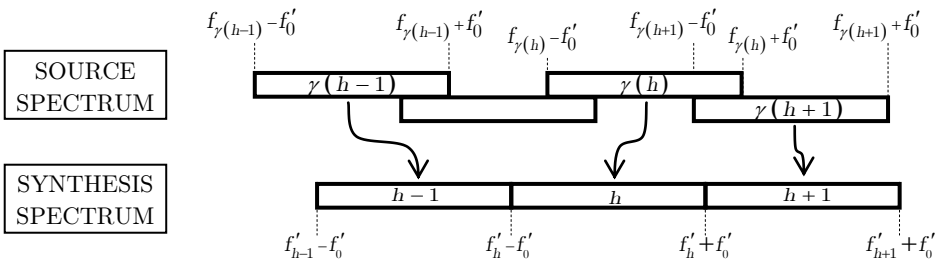


Figure 2.52 Spectral region rendering strategy 1: adaptive region segmentation.

❖ Strategy 2: fill gaps with source spectrum

The idea is to set the source spectral region width to match the input pitch f_0, i.e. $b_h = f_h - f_0/2$ and $e_h = f_h + f_0/2$. Besides, with the aim of filling the amplitude gaps, in the case of non-rendered synthesis bins between harmonics use the same source bin only if it is not around a harmonic. The timbre envelope is approximated as the spectral envelope defined by the harmonics, noted as $H_{harm}(k)$. We can rewrite equation (2.58) as

$$S'_w(k) = \sum_{h=0}^{H'-1}\left[\frac{a'_h}{a_{\gamma(h)}} e^{j\left(\theta'_{0,h}-\theta_{0,\gamma(h)}\right)} S_w(k-x)\,\Upsilon_{\gamma(h)}(k-x)\right] + S_w(k)\frac{H_{harm}\left(T_{timbre}(k)\right)}{H_{harm}(k)}\Upsilon'(k) \tag{2.61}$$

$$\Upsilon_h(k) = \begin{cases} 1 & if \quad k \in [b_h, e_h] \\ 0 & if \quad k \notin [b_h, e_h] \end{cases} \qquad \Upsilon'(k) = \begin{cases} 1 & if \\ 0 & otherwise \end{cases} \left|k - f_j\right| > \alpha \ \forall j \qquad x \triangleq \left(f'_h - f_{\gamma(h)}\right) T_s N .$$

Notice that the source bins used to fill the gaps are multiplied by the ratio between the estimated target and source timbre amplitudes, so as to apply the desired timbre transformation $T_{timbre}(k)$.

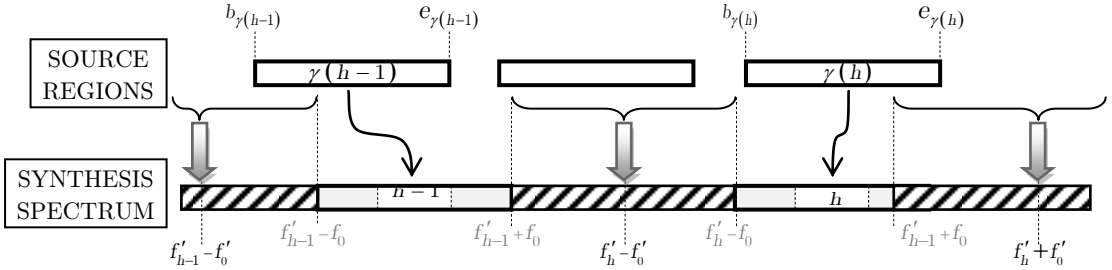


Figure 2.53 Spectral region rendering strategy 2: filling gaps with source spectrum.

❖ Strategy 3: fill gaps with frequency shifted copies of synthesis spectral regions

We propose to set the source spectral region width to match the input pitch f_0, i.e. $b_h = f_h - f_0/2$ and $e_h = f_h + f_0/2$. Use frequency shifted segments of the synthesis spectral regions to fill the gaps. For the h^{th} synthesis region $S'_w([b'_h, e'_h]) = S'_w([f'_h - f_0/2, f'_h - f_0/2])$, the left and right gaps are filled using shifted copies of the segments $S'_w([b'_h, f'_h - \alpha f_0])$ and $S'_w([f'_h + \alpha f_0, e'_h])$ respectively. The frequency shifting amount is $\Delta f_l^{left} = l(\alpha - 1) f_0$ and $\Delta f_l^{right} = l(1-\alpha) f_0$ Hz for the l^{th} copy. This shifting requires a phase correction to avoid discontinuities between consecutive frames, which can be computed as the difference between the target and source ideal phase rotations per hop size[19], i.e. $\theta_l^{left} = 2\pi l(\alpha - 1) f_0 \Delta'_t$ and $\theta_l^{right} = 2\pi l(1-\alpha) f_0 \Delta'_t$ and has to be accumulated frame by frame.

[19] The difference between the ideal target and source phase rotations per hop size depends on the frequency difference, but not on the absolute frequency values. Indeed, the phase rotation of an ideal sinusoid at the source frequency f_{source} is $\theta_{\text{source}} = 2\pi f_{\text{source}} \Delta'_t$. For the target frequency we have $\theta_{\text{target}} = 2\pi f_{\text{target}} \Delta'_t$. Then $\Delta\theta = \theta_{\text{target}} - \theta_{\text{source}} = 2\pi\left(f_{\text{target}} - f_{\text{source}}\right)\Delta'_t$.

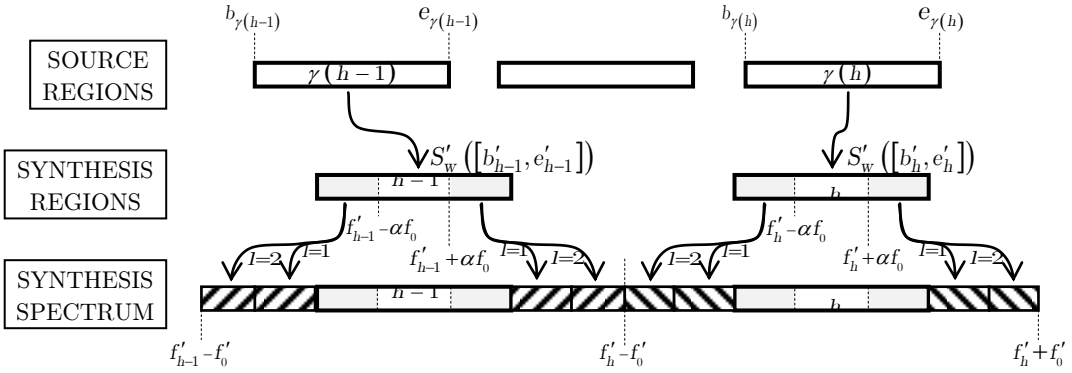


Figure 2.54 Spectral region rendering strategy 3: filling gaps with frequency shifted copies of synthesis spectral regions.

❖ Strategy 4: frequency warping

Here we propose add one more transformation to the spectral regions: non-linear frequency scaling of the spectral region, i.e. frequency warping (Bonada, et al. 2006). The idea is that around each harmonic the frequency bins are not scaled but mainly shifted in frequency, thus almost preserving the linear convolution of the analysis window with the harmonic frequency component. On the other hand, scaling is applied to the areas between harmonics. Using a warping function $\psi_h(k)$ defined to behave in such a way, equation (2.58) becomes

$$S'_w(k) = \sum_{h=0}^{H'-1}\left[\frac{a'_h}{a_{\gamma(h)}} e^{j\left(\theta'_{0,h}-\theta_{0,\gamma(h)}\right)} S_w\left(\psi_h(k) - \left(f'_h - f_{\gamma(h)}\right)T_s N\right)\Upsilon_{\gamma(h)}\left(\psi_h(k)-x\right)\right]$$

$$\Upsilon_h(k) = \begin{cases} 1 & if \quad k \in [b_h, e_h] \\ 0 & if \quad k \notin [b_h, e_h] \end{cases} \qquad x \triangleq \left(f'_h - f_{\gamma(h)}\right)T_s N \tag{2.62}$$

$$\psi_h(k) \approx k \text{ around the harmonic.}$$

If we assume a perfect harmonic distribution (as expected for a voice signal) and set the spectral boundaries to the middle frequency between consecutive harmonics, a single warping function $\psi(k)$ might be applied to all the harmonics and the previous equation can be rewritten as

$$S'_w(k) = \sum_{h=0}^{H'-1}\left[\frac{a'_h}{a_{\gamma(h)}} e^{j\left(\theta'_{0,h}-\theta_{0,\gamma(h)}\right)} S_w(k+x)\Upsilon_{\gamma(h)}(k+x)\right]$$

$$\Upsilon_h(k) = \begin{cases} 1 & if \quad k \in \left[(h+0.5)f_0, (h+1.5)f_0\right] \\ 0 & if \quad k \notin \left[(h+0.5)f_0, (h+1.5)f_0\right] \end{cases}$$

$$x \triangleq \psi(k-k_h) - (k'_h - k_h) \tag{2.63}$$

$$k'_h \triangleq (h+1)T_{pitch} f_0 T_s N$$

$$k_h \triangleq (\gamma(h)+1) f_0 T_s N$$

where the warping function ψ has been computed as a linear interpolation between the identity function $\psi_I(k)=k$ and the linear scaling function $\psi_L(k)=k/T_{pitch}$.

$$\psi(k) = \psi_I(k) + \frac{2|k|}{f_0 T_{pitch}}\left(\psi_L(k) - \psi_I(k)\right) = \frac{2\left(1-T_{pitch}\right)}{f_0 T_{pitch}^2}|k|k + k \quad , \quad k \in \left[\frac{-f_0 T_{pitch}}{2}, \frac{f_0 T_{pitch}}{2}\right] \tag{2.64}$$

Figure 2.56 shows a comparison of an upwards pitch transposition with and without frequency warping.

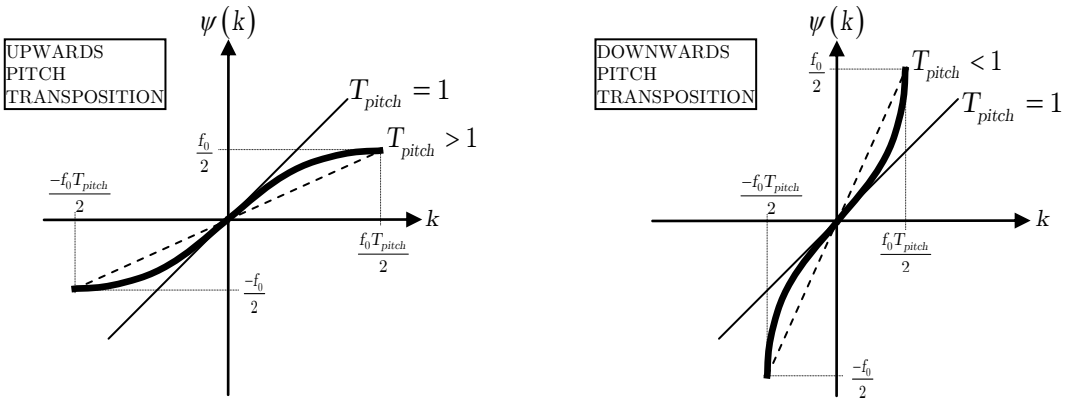


Figure 2.55 Warping function used for the spectral region rendering strategy 4.

RESULTS

We have processed several voice utterances using the different proposed strategies with the aim of covering a wide range of voice types. Table 2.3 contains the list of processed audio files together with a brief description of each one. We have performed an informal subjective evaluation of the sound quality obtained for each strategy. The preferred audio files are marked in bold. If there is no a clear preference all samples are in bold. The results are next presented.

- For upwards transposition, the preferred strategies are A, C and E
  - The basic technique (A) sounds mostly good, sometimes a little more closer or sharp than the others
  - Strategy 1 (B) sounds too noisy
  - Strategy 2 (C) sounds mostly good, especially for breathy utterances
  - Strategy 3 (D) sounds distorted and too noisy
  - Strategy 4 (E) sounds mostly good but smoothes the utterances a bit
- For downwards transpositions, there are no significant differences between the different strategies. Only in one example, the growl in gospel singing, strategy 4 sounds too smooth and worse than the others

Therefore, it seems that none of the different strategies proposed improve the overall quality of the basic technique. Only the second strategy (C) yields a comparable quality, being in some cases better, especially for breathy utterances.

REGARDING LOW FREQUENCIES BELOW THE FUNDAMENTAL

We have seen that both harmonic and surrounding frequency components are synthesized by transforming spectral regions. However, the low frequency components below the frequency boundary of the first region (b_0) do not contribute to the synthesis output. In other words, regarding those frequencies the transformation works as a high-pass filter. This might be good in some cases to remove unwanted low frequency noises such as guttural utterances or vibrations caused by the singer accidentally touching the microphone. Nevertheless, in some other cases environment sounds that enhance the sensation of reality of the voice might be removed. Therefore, we propose to add to this synthesis method an adaptive low-pass filter with amplitude control.

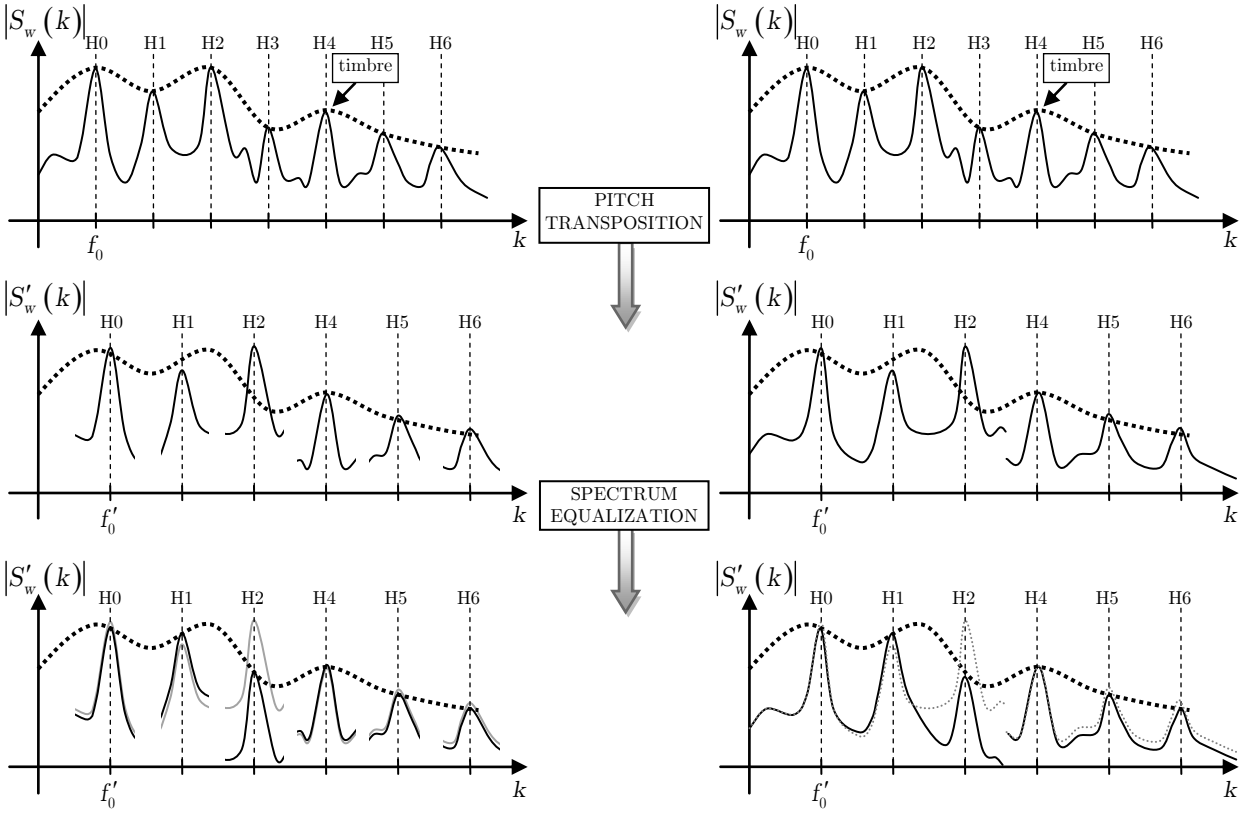


Figure 2.56 Example of upwards pitch transposition by spectral region transformations. On the left side, spectral regions are segmented, shifted in frequency and scaled in amplitude. Clear amplitude gaps are visible between consecutive regions. On the right side, spectral regions are non-linearly scaled in frequency and amplitude gaps have disappeared.

Audio	Description	Transformation *A=basic technique, B=strategy 1, C=strategy 2, D=strategy 3, E=strategy 4*									
		$T_{pitch} = 1.75$					$T_{pitch} = 0.65$				
		A	B	C	D	E	A	B	C	D	E
[12]	female singing $f_0^{range} \approx 350 - 386\,\text{Hz}$	[13]	[14]	[15]	[16]	[17]	[60]	[61]	[62]	[63]	[64]
[18]	female *scat* fast singing $f_0^{range} \approx 127 - 318\,\text{Hz}$	[19]	[20]	[21]	[22]	[23]	[65]	[66]	[67]	[68]	[69]
[24]	male *slow* singing $f_0^{range} \approx 100 - 291\,\text{Hz}$	[25]	[26]	[27]	[28]	[29]	[70]	[71]	[72]	[73]	[74]
[30]	male *soul* singing $f_0^{range} \approx 98 - 211\,\text{Hz}$	[31]	[32]	[33]	[34]	[35]	[75]	[76]	[77]	[78]	[79]
[36]	male speech $f_0^{range} \approx 59 - 100\,\text{Hz}$	[37]	[38]	[39]	[40]	[41]	[80]	[81]	[82]	[83]	[84]
[42]	female speech $f_0^{range} \approx 190 - 410\,\text{Hz}$	[43]	[44]	[45]	[46]	[47]	[85]	[86]	[87]	[88]	[89]
[48]	male *breathy* singing $f_0^{range} \approx 270 - 396\,\text{Hz}$	[49]	[50]	[51]	[52]	[53]	[90]	[91]	[92]	[93]	[94]
[54]	male *gospel* singing $f_0^{range} \approx 145 - 357\,\text{Hz}$	[55]	[56]	[57]	[58]	[59]	[95]	[96]	[97]	[98]	[99]

Table 2.3 This table contains the results of the subjective evaluation of the sound quality obtained with each spectral rendering strategy.

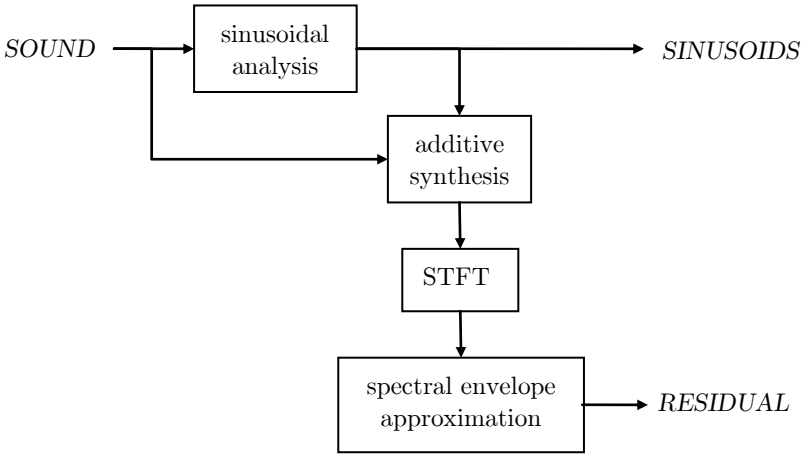


Figure 2.57 SMS analysis block diagram

2.2.6 Modeling Residual

Voice signals are not exclusively built of harmonic components. Depending on the phonation mode, different levels of noise are present in the sound and perceived as a breathy characteristic. In addition, the voice source airflow is not an ideal train of pulses at the pitch rate, but rather an irregular train of pulses, often characterized by the amount of amplitude and frequency modulation (i.e. shimmer and jitter). From the point of view of a strictly harmonic sinusoidal model, both aspects are a source of noise. Simplifying, while the former can be considered as filtered white noise, the latter adds other sinusoidal components often referred to as subharmonics.

Current section §2.2 focuses on interpreting voiced signals as a collection of harmonic trajectories. We have studied two different approaches for modeling those harmonics: as spectral regions and as sinusoids. In the former model, spectral regions contain not only the quasi-stationary sinusoidal component spectral data but also the surrounding frequency band of noise. In consequence, rendering harmonic trajectories with this method implies that both harmonic and noise components are transformed and synthesized together at once. In addition, without transformations the original voice signal is perfectly reconstructed. However, when modeling harmonics as sinusoids only harmonic components are transformed and synthesized, what results in artificial voice sounds. Hence, it is necessary to add some processing for modeling and transforming the noise component. What follows is a brief overview of most common approaches that pursue this goal.

Serra (1989) incorporated the noise component of the sound into an extended sinusoidal model: Spectral Modeling Synthesis (SMS). In this approach, sinusoids model only the stable partials of a sound, and residual models what is left, which is supposed to be a stochastic component (see Figure 2.57). The input sound $s(t)$ is decomposed as

$$s(t)=\sum_{p=1}^{P} a_p(t)\cos\left(\varphi_p(t)\right)+e(t) \tag{2.65}$$

where $a_p(t)$ and $\varphi_p(t)$ are respectively the instantaneous amplitude and phase of the p^{th} partial, and $e(t)$ is the noise component. Since the sinusoids are used to model solely the stable partials of the sound, they are referred to as the deterministic component. The residual $e(t)$ is assumed to be a stochastic signal and it can be described as filtered white noise

$$e(t)=\int_0^t h(t,\tau)u(\tau)d\tau \tag{2.66}$$

where $u(t)$ is white noise and $h(t,\tau)$ is the response of a time varying filter to an impulse at time t.

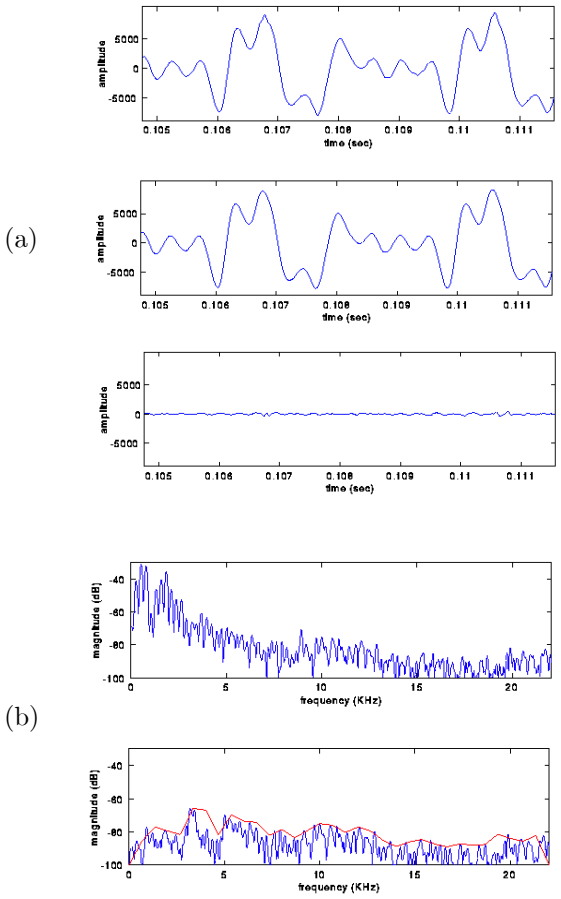


Figure 2.58 (a) Waveform representation of the original, deterministic and residual signals. (b) Discrete-time STFTs of the original and residual signals, showing the approximated residual amplitude envelope.

The residual is obtained by subtracting the deterministic component from the original sound, as illustrated in Figure 2.58a. The deterministic component can be obtained in time domain by using additive synthesis, where each sine wave oscillator is controlled by the estimated parameters (amplitude, frequency and phase) out from the analysis, smoothly interpolated frame by frame.

Since SMS assumes that the residual is a stochastic signal, it should be fully described by its amplitude and its general frequency characteristics. Discarding its phase information, it could be reproduced as white noise filtered through a time-varying filter defined by the approximated envelope of the residual amplitude spectra (see Figure 2.58b). This envelope approximation can be performed by computing, for each frame, a curve fitting of the magnitude spectrum (Sedgewick 1998, Strawn 1995). Some useful standard techniques are spline interpolation (Cox 1971), the method of least squares (Sedgewick 1998), or a straight-line approximation. Another way to approximate the envelope is to step trough the magnitude spectrum, find local maxima in each of several defined sections, and interpolate linearly between them. Another alternative, is to use linear predictive coding, LPC (Makhoul 1975, Markel and Gray 1975), a popular technique used in speech research for fitting a magnitude spectrum with an n^{th} order polynomial. A comprehensive collection of different approximation techniques for the residual component is found in (Goodwin 1997).

An interesting alternative is to link the stochastic components to each harmonic by using stochastic modulation to spread spectral energy away from the harmonic's center frequency (Fitz 1999). With this technique each partial has an extra parameter, the bandwidth coefficient η, which sets the balance between noise and sinusoidal energy. Each harmonic is modeled by

$$s_h(n) = \tilde{a}_h\left(\sqrt{1-\eta} + \sqrt{2\eta}\left[\xi_n * h_n\right]\right)e^{j2\pi f_h n T_s} \tag{2.67}$$

where ξ_n is a noise sequence that excites a low-pass filter with impulse response h_n. With this approach, both sinusoidal and noise components can be manipulated in an intuitive way, and transformations such as frequency or time-scaling do not modify the noise character neither introduce audible artifacts. Besides, this method has shown to be useful for controlling voice quality.

One drawback of the previous approaches is that often voice residual contains non-stochastic components, such as transients, which do not fit well with the proposed model. Hence, it might make sense to avoid the stochastic model and process instead the residual signal obtained by subtraction. By doing this we lose control flexibility, but we gain in sound quality. This residual signal can be transformed with high quality using for instance the TD-PSOLA algorithm (e.g. SINOLA algorithm (Peeters 2001)). This allows improved transient processing and a better preservation of the residual time-structure, which is perceptually relevant, and correlated with the glottal voice source phase (Kob 2002). Nevertheless, transient modeling can be added to the residual model and improve the quality of representation without losing control flexibility (Verma, et al. 1997).

On the other hand, the second source of noise comes mainly from the amplitude and frequency modulations often found in the voice source (i.e. shimmer and jitter). These modulations produce extra spectral components in the form of subharmonics, which behave according to complex patterns, and can be modeled with sinusoids (Loscos and Bonada 2004), although a real independent control of individual voice pulses with such approach remains yet an open question.

2.2.7 Discussion

We have explored the interpretation of voiced utterances as a collection of harmonic trajectories, in the context of spectral models. We have shown the relevance in terms of sound quality of working with shape invariant processing techniques, and proposed a method for estimating voice pulse onsets that fits very well within both constant and variable hop-size spectral model frameworks. We have also shown the complexity of modeling pulse sequence irregularities with sinusoidal components (sub-harmonics), and compared two methods for rendering harmonic trajectories: as sinusoids and as spectral regions. Finally, we described the most common approaches to model the voice residual.

If we compare the two methods for rendering harmonics, we would find that for processing stationary sinusoids both methods produce very similar results, almost indistinguishable. However, voiced utterances tend to contain non-stationary components, especially in note and phonetic boundaries, even more in expressive singing. For such contexts, modeling harmonics as sinusoids requires estimating the inner evolution of sinusoid parameters along the analysis window, which is a complex task that demands special care to choose underlying non-stationary models that match the observed spectral peaks. Otherwise, those estimators might produce extreme values, for instance in the presence of transients or unresolved sinusoids due to reverberation. In addition, for preserving the relevant perceptual characteristics of the input voice it is necessary to estimate and model a residual obtained by subtracting the harmonics from the input. One advantage of the residual models is that they are able to generate synthesis signals that sound quite similar independently of the transformation ratio, although not exactly as the original one.

On the other hand, modeling harmonics as transformed spectral regions from the original signal allows us to preserve much of the parameters evolution (amplitude, frequency, phase) within the frame window. In addition, the noise components surrounding harmonic peaks are included in the spectral regions. Therefore, the input sound is perfectly reconstructed when no transformations are applied, and noise components are transformed together with the sinusoids. However, the control in the residual is less flexible than with a residual model, and the sound quality of the synthesized noisy components degrades significantly for high modification ratios. One possible future research direction is to use a combination of spectral peak shape descriptors and harmonic spectral envelope to compute the mapping function between harmonic trajectories and spectral regions. Another interesting idea to explore is to use different mapping functions for harmonic trajectories and spectral regions.

2.3 Voice Pulse Modeling

As mentioned at the beginning of this chapter, voiced utterances can be interpreted as a sequence of time domain voice pulses occurring at the rate defined by the inverse of the fundamental frequency. In addition, each voice pulse can be modeled to be the result of an air glottal pulse filtered by the vocal tract and radiated trough the mouth. Several algorithms and techniques have been conceived out of this characterization of the voice signal, most of them based in time domain. Probably the most well known one is TD-PSOLA (Moulines, et al. 1989). It estimates the sequence of voice pulse onsets and segments the audio into overlapping windows of two periods length. The synthesis is generated by overlapping time and amplitude scaled versions of such pulses at the synthesis rate, allowing repeating or dropping pulses if necessary. This technique is able to achieve pitch transposition, time scaling and timbre scaling transformations. However, timbre scaling is limited to be linear. Its basic procedure is illustrated in Figure 2.59.

Our intention here is to propose novel techniques based on modeling radiated voice pulses in frequency domain, with the goal of providing the transformation flexibility of typical frequency domain techniques together with the ability of independently controlling each voice pulse found in time domain techniques. In order to achieve that goal, the first thing to do is to study the characteristics of the spectrum resulting of overlapping several voice pulses in time domain. The discrete-time STFT of a windowed pulse $x(n)$ can be expressed as

$$\begin{aligned} x(n) &= s(n)\cdot w(n) \\ X(e^{j\Omega}) &= \sum_{n=0}^{N-1} x(n)e^{-j\Omega n} \end{aligned} \tag{2.68}$$

where $s(n)$ is a single voice pulse, $w(n)$ is the window function and $\Omega = 2\pi f T_s$. Let's assume a rectangular window of N samples and a finite pulse shorter than N and covered by the window. If the pulse is delayed by Δn samples, then its STFT will be

$$\begin{aligned} S_{delayed\,\Delta n}\left(e^{j\Omega}\right) &= \sum_{n=0}^{N-1} s(n-\Delta n)e^{-j\Omega n} = \sum_{n=0}^{N-1} x(n-\Delta n)e^{-j\Omega n} \\ &\overset{m=n-\Delta n}{=} \sum_{m=-\Delta n}^{N-1-\Delta n} x(m)e^{-j\Omega(m+\Delta n)} \\ &\approx X(e^{j\Omega})e^{-j\Omega\Delta n} \end{aligned} \tag{2.69}$$

where the last approximation becomes an identity if the delayed signal is completely covered by the window. Let's now consider $y(n)$ as the sum of R identical pulses $s(n)$ delayed by multiples of Δn samples, where overlap is possible. We can calculate its STFT as follows

$$y(n) = s(n) + s(n-\Delta n) + s(n-2\Delta n) + \ldots + s(n-(R-1)\Delta n) \tag{2.70}$$

$$\begin{aligned} Y(e^{j\Omega}) &= \sum_{n=0}^{N-1} y(n)e^{-j\Omega n} = \\ &\approx X(e^{j\Omega})\left[1 + e^{-j\Omega\Delta n} + e^{-2j\Omega\Delta n} + \ldots + e^{-(R-1)j\Omega\Delta n}\right] = \\ &= X(e^{j\Omega})\sum_{r=0}^{r=R-1} e^{-j\Omega\Delta n r} = X(e^{j\Omega})\frac{1-e^{-j\Omega\Delta n R}}{1-e^{-j\Omega\Delta n}} = \\ &= X(e^{j\Omega})e^{-j\Omega\Delta n\frac{R-1}{2}}\frac{\sin\left(0.5\Omega\Delta n R\right)}{\sin\left(0.5\Omega\Delta n\right)} \triangleq \\ &\triangleq X(e^{j\Omega})\,\mathrm{sinc}_R\left(\Omega\Delta n\right). \end{aligned} \tag{2.71}$$

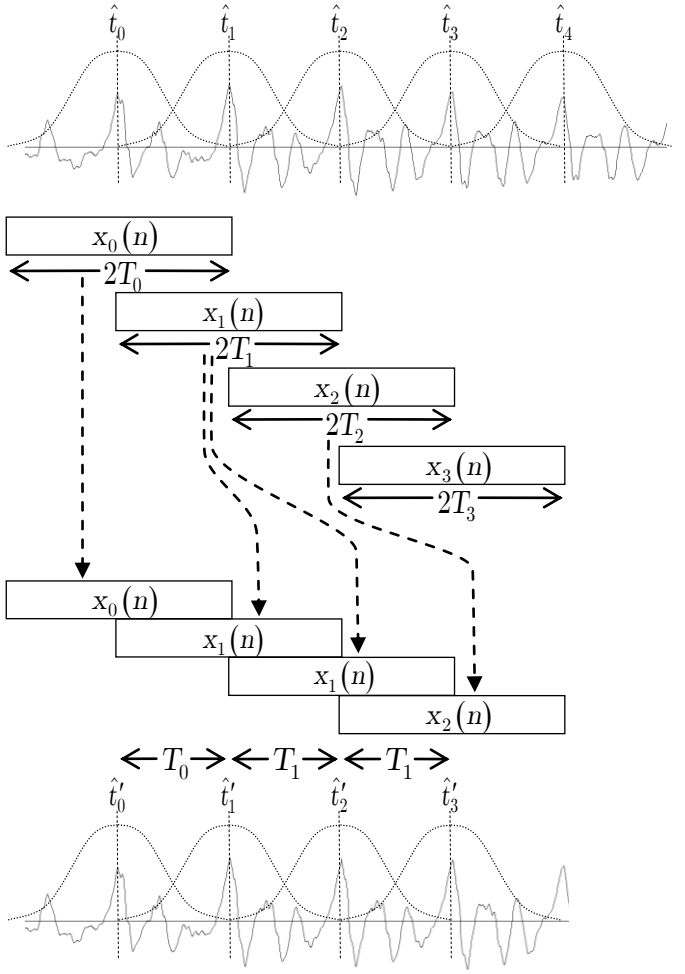


Figure 2.59 Example of a time-scaling transformation using the TD-PSOLA algorithm. The second pulse $x_1(n)$ is repeated at synthesis in order to slow down the voiced utterance without altering the fundamental frequency.

The effect of the term $\mathrm{sinc}_R(\Omega\Delta n)$ is somehow sampling the spectrum of $X(e^{j\Omega})$, since this term is actually a train of equidistant pulses located each $2\pi/\Delta n$ radians (see Figure 2.60), with a constant amplitude of R and phase zero. In effect, computing the limits of the $\mathrm{sinc}_R(\Omega\Delta n)$ function around Ω values of $r2\pi/\Delta n$ for $r\in[0,1,\ldots,\Delta n-1]$, we get

$$\begin{aligned}\lim_{\Omega\to r\frac{2\pi}{\Delta n}}\left|\mathrm{sinc}_R(\Omega\Delta n)\right| &= \lim_{\Omega\to r\frac{2\pi}{\Delta n}}\left|e^{-j\pi r(R-1)}\frac{\sin(r\pi R)}{\sin(r\pi)}\right| = \lim_{x\to 0}\left|\frac{\sin((r+x)\pi R)}{\sin((r+x)\pi)}\right| \\ &= \lim_{x\to 0}\left|\frac{\sin(a\pi+x\pi R)}{\sin(b\pi+x\pi)}\right|\end{aligned} \tag{2.72}$$

where a and b are natural numbers. Using the trigonometric equivalence $\sin(k\pi+x)=\sin(x)$ for even values of k and $\sin(k\pi+x)=-\sin(x)$ for odd values, we get

$$\lim_{x\to 0}\left|\frac{\sin(a\pi+x\pi R)}{\sin(b\pi+x\pi)}\right| = \lim_{x\to 0}\left|\frac{\sin(x\pi R)}{\sin(x\pi)}\right| \approx \left|\frac{x\pi R}{x\pi}\right| = R. \tag{2.73}$$

Regarding the phase, we obtain

$$\begin{aligned}\lim_{\Omega\to r\frac{2\pi}{\Delta n}} \measuredangle \operatorname{sinc}_R(\Omega\Delta n) &= \lim_{\Omega\to r\frac{2\pi}{\Delta n}} \measuredangle e^{-j\pi r(R-1)} \frac{\sin(r\pi R)}{\sin(r\pi)} = \lim_{x\to 0} \measuredangle e^{-j\pi(r+x)(R-1)} \frac{\sin((r+x)\pi R)}{\sin((r+x)\pi)} \\ &= \lim_{x\to 0} \measuredangle e^{-j(c\pi+\pi x(R-1))} \frac{\sin(a\pi+x\pi R)}{\sin(b\pi+x\pi)}\end{aligned} \tag{2.74}$$

where a, b and c are natural numbers, which can be odd or even depending on the values of r and R. For all the possible combinations, in the end we get as the result of the limit a positive real number[20], and therefore the resulting angle is zero:

$$\lim_{\Omega\to r\frac{2\pi}{\Delta n}} \left|\operatorname{sinc}_R(\Omega\Delta n)\right| = 0. \tag{2.75}$$

The resolution of this sampling process does not depend on the number of pulses covered by the window, but actually on the delay Δn, a bigger value meaning better resolution. In fact, we can consider such delay as the period $T=\Delta n$ of the resulting periodical signal, and therefore the *spectrum samples* as its harmonics, located at multiples of its fundamental frequency $f_0 = 1/T = 1/\Delta n$. Although these concepts might seem obvious when thinking of a pure periodical signal and its spectrum composed of harmonics, the interesting point here is that if we model the periodical signal as the result of overlapping identical pulses, then the harmonics are actually sampling the spectrum of a single pulse. This is a well-known effect and actually the basis of the TD-PSOLA algorithm. Indeed, as pointed out in (Hamon, et al. 1989) regarding the spectral interpretation of the TD-PSOLA synthesis, if a short-time signal $x_0(n)$ is repeated at the synthesis period $1/f_0'$ then it can be shown that the discrete-time STFT of the resulting signal using a window function $h(n)$ is given by the convolution of the response of the window function $H(f)$ by the spectrum of $X_0(f)$ sampled at the harmonic frequencies kf_0', i.e.

$$Y(f) = \sum_k H(f - f_k')X_0(f_k'). \tag{2.76}$$

It is now time to introduce the concept of narrow and wide-band analysis of a periodic signal, which relates to the ratio between frequency resolution and fundamental frequency, and therefore to the number of periods covered by the analysis window. Narrow-band analysis takes several periods so that in quasi-stationary conditions harmonics appear as clear and separated peaks in the spectrum. By contrast, wide-band analysis uses one or two periods so that the frequency distance between harmonics is similar to the spectral resolution, and therefore the spectra produced by each harmonic affects significantly its neighbor harmonics, what complicates the estimation of individual frequency components. Moreover, narrow-band analyses perform with lower temporal resolution than wide-band analyses. In general, algorithms based on modeling and tracking spectral peaks use a narrow-band approach to facilitate the detection of individual frequency components. This is the case of phase-locked vocoder (Laroche 2003) and sinusoidal models (McAulay and Quatieri 1986). On the other hand, typical time-domain algorithms such as TD-PSOLA (Moulines, et al. 1989) or LP-PSOLA (Moulines, et al. 1989) use two-period long frames, so they work in wide-band conditions. In the following sections we approach the task of modeling voice pulses in frequency domain from the points

---

[20] Being $a=rR$, $b=r$ and $c=r(R-1)$, and $A=\sin(a\pi+x\pi R)$, $B=\sin(b\pi+x\pi)$ and $C=e^{-j(c\pi+\pi x(R-1))}$, we obtain $\lim_{\Omega\to 2\pi r/\Delta n} \measuredangle \operatorname{sinc}_R(\Omega\Delta n) = \lim_{x\to 0} \measuredangle C\frac{A}{B}$ which values 0 if $\lim_{x\to 0} C\frac{A}{B} > 0$ and π otherwise. The possible combinations of odd/even values lead to

$$r \text{ odd / } R \text{ even} \rightarrow a \text{ even / } b \text{ odd / } c \text{ odd} \rightarrow \lim_{x\to 0} \measuredangle C\frac{A}{B} = \lim_{x\to 0} \measuredangle(-)\frac{\sin(x\pi R)}{-\sin(x\pi)} \approx \measuredangle(-)\frac{x\pi R}{-x\pi} = 0$$

$$r \text{ odd / } R \text{ odd} \rightarrow a \text{ odd / } b \text{ odd / } c \text{ even} \rightarrow \lim_{x\to 0} \measuredangle C\frac{A}{B} = \lim_{x\to 0} \measuredangle(+)\frac{-\sin(x\pi R)}{-\sin(x\pi)} \approx \measuredangle(+)\frac{-x\pi R}{-x\pi} = 0$$

$$r \text{ even / } R \text{ even} \rightarrow a \text{ even / } b \text{ even / } c \text{ even} \rightarrow \lim_{x\to 0} \measuredangle C\frac{A}{B} = \lim_{x\to 0} \measuredangle(+)\frac{\sin(x\pi R)}{\sin(x\pi)} \approx \measuredangle(+)\frac{x\pi R}{x\pi} = 0$$

$$r \text{ even / } R \text{ odd} \rightarrow a \text{ even / } b \text{ even / } c \text{ even} \rightarrow \lim_{x\to 0} \measuredangle C\frac{A}{B} = \lim_{x\to 0} \measuredangle(+)\frac{\sin(x\pi R)}{\sin(x\pi)} \approx \measuredangle(+)\frac{x\pi R}{x\pi} = 0$$

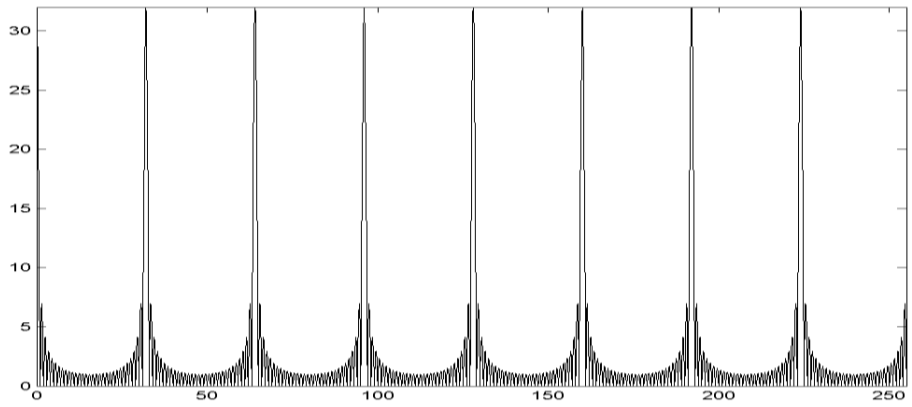


Figure 2.60 $\left|\mathrm{sinc}_R\left(\Omega\Delta n\right)\right|$, R=32, Δ=8, N=256.

of view of narrow (Bonada 2004) and wide-band (Bonada 2008) conditions and discuss the benefits and drawbacks of each strategy.

2.3.1 Narrow-Band Voice Pulse Modeling (NBVPM)

Equations (2.70) and (2.71) tell us the effect in frequency domain of periodically repeating a short-time signal $x(n)$. If $Y\left(e^{j\Omega}\right)$ is the discrete-time STFT transform of the resulting signal $y(n)$, then $Y\left(e^{j\Omega}\right)$ is a sampled version of $X\left(e^{j\Omega}\right)$, the discrete-time STFT transform of the periodically repeated signal $x(n)$. In the case of voiced utterances in a speech or singing voice recording, $x(n)$ corresponds to a radiated voice pulse, i.e. the resulting signal of radiating a single glottal pulse filtered by the vocal tract. According to the characteristics of the human voice, we expect the spectral envelope of the voice pulse to be band-limited, slow varying along frequency. Indeed, the voice source spectral envelope can be almost characterized as an exponentially decaying envelope, and the vocal tract adds roughly one resonance per KHz. Consequently, we expect to approximate reasonably well the transform of the voice pulse $X\left(e^{j\Omega}\right)$ if we sample it with a frequency resolution higher than the vocal tract filter envelope bandwidth. In other words, the lower the fundamental frequency of the analyzed signal, the better the approximation will be. In practical terms, this means that if we consider $\left|X(e^{j\Omega})\right|$ and $\measuredangle X(e^{j\Omega})$ to vary slowly along frequency, we can estimate $X(e^{j\Omega})$ from $Y(e^{j\Omega})$ by interpolating the values at Ω_I frequencies. The interpolation algorithm could be just a linear interpolation but higher order or spline methods are preferred, since we know that the spectral envelope is smooth. We can compute the reconstructed pulse signal $x'(n)$ by means of the inverse STFT of the estimated signal $X'\left(e^{j\Omega}\right)$, i.e.

$$x'(n)=\frac{1}{N}\sum_{n=0}^{N-1}X'\left(e^{j\Omega}\right)e^{j\Omega n}. \tag{2.77}$$

Since we considered that $y(n)$ consists of both identical and equidistant repetitions of the signal $s(n)$, the accuracy of this approximation depends a lot on whether the input signal is stationary or not. What we usually deal with is the result of several consecutive voice pulses, as illustrated in Figure 2.61, which are overlapped along time but not fully covered by the window. In that case Δn is directly related to the voice fundamental frequency by $\Delta n=f_s/f_0$, where f_S is the sampling rate and f_0 the fundamental frequency in Hertz. In addition, the radiated voice pulses are never identical because the voice production system is changing its characteristics continuously along time. However, if the window is short enough so that the signal becomes quasi-stationary, then the proposed method would allow estimating a single radiated voice pulse.

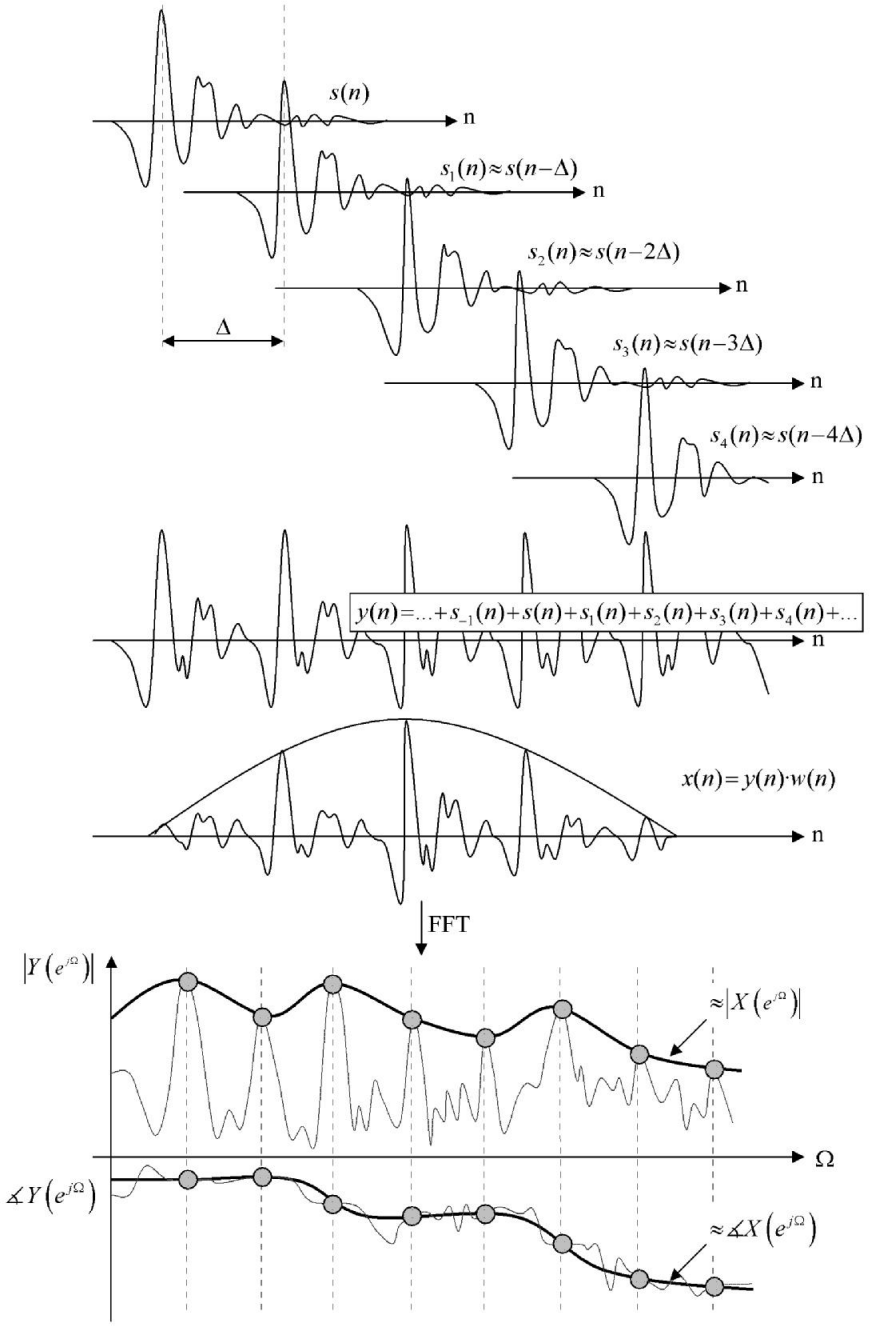


Figure 2.61 Spectrum estimation of a single radiated voice pulse

Once we have estimated the voice pulses, we can handle them similarly to time domain PSOLA methods. Thus, for instance, we can overlap voice pulses at the original pitch rate to reconstruct the input signal, or we can modify the pitch rate to perform a pitch transposition. In addition, we can compute a residual signal by subtracting the reconstructed signal to the original one, and transform it independently of the harmonic content. This framework theoretically combines the transformation flexibility of frequency domain techniques with the ability of time domain methods to handle each voice pulse independently. Figure 2.62 shows the block diagram of the different steps involved in the NBVPM analysis, transformation and synthesis processes. In the following sections we describe each of these aspects, and discuss different methods for modeling the residual signal.

ANALYSIS

The analysis consists of a constant frame-rate process where, for each step, we estimate the STFT of a centered radiated pulse (see Figure 2.61). In our experiments we use a constant hop size of 256 samples and a sampling rate of 44.1Khz, what gives us about 172 frames per second. The analysis starts windowing the input voice signal and computing its discrete-time STFT. Then the spectral peaks are detected out of the amplitude spectrum using quadratic interpolation and inputted to the pitch detection algorithm, actually an extension of the TWM algorithm (Cano 1998). The pitch is then used by the peak selection module to choose the harmonic peaks, considering as a guide both the ideal harmonic distribution and the amplitude and frequency of spectral peaks. Next, the phase of the harmonic peaks is modified by the *maximally flat phase alignment* (MFPA) algorithm (see §2.2.2), which centers the voice pulse in the window. The final step is to interpolate the harmonic peaks and to estimate the spectrum of the radiated voice pulse $X'(e^{j\Omega})$. We use a 3rd order spline method to interpolate the amplitude spectrum, and afterwards we scale it by $1/R$, where R is the number of pulses contained in the analysis window. R is computed from the pitch by

$$R = \frac{N \cdot pitch}{f_s}. \tag{2.78}$$

Regarding the window size N, a good frequency resolution is required in order to precisely detect the spectral peaks, therefore a long window should be used. However, in such case, the transitions, attacks and releases become considerably smoothed due to spectral leakage, since radiated pulses with different characteristics (energy, timbre) are processed together and pitch is not approximately constant along the window. The adopted compromise has been to adapt the window size to cover around three periods, which mostly assures enough resolution to discern the peaks while at the same time increases the temporal resolution and improves the handling of non-stationary parts. Besides, zero padding is applied in order to further increase the spectral resolution.

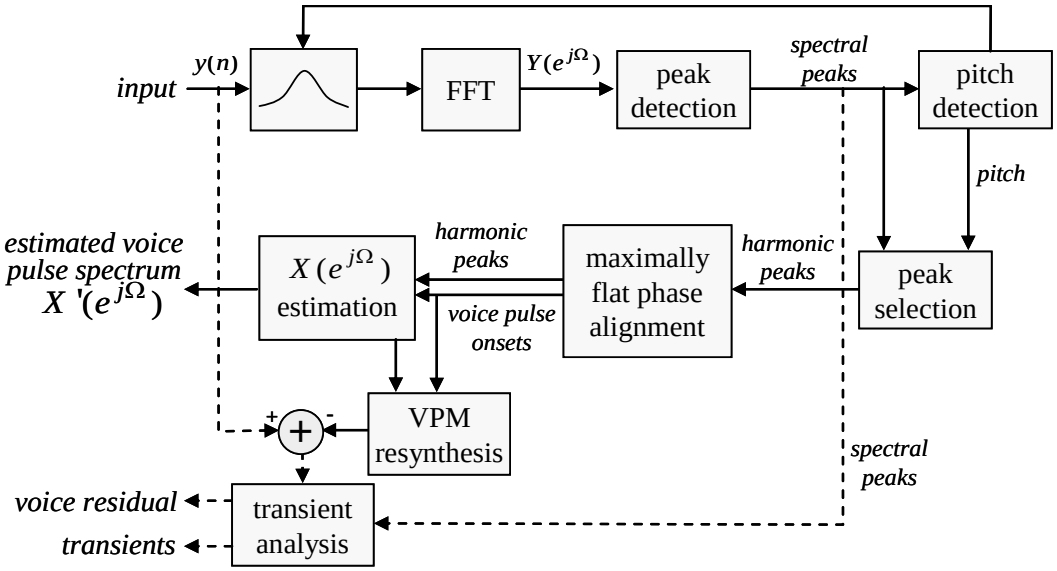
SYNTHESIS

The analysis outputs the estimated pitch and spectrum of radiated voice pulses centered in the window. In order to synthesize a voiced utterance, we first generate a sequence of pulse locations. Such sequence can be derived from the pitch envelope by separating consecutive pulses by one period duration, $T = 1/pitch$. If we want to better match the input signal pulse sequence, we can also use the unwrapped fundamental phase envelope and the outputs of the MFPA to determine the position of each pulse in the original sequence. We do not need to care about each pulse's amplitude since the original amplitude is preserved through the estimated spectral amplitude itself.

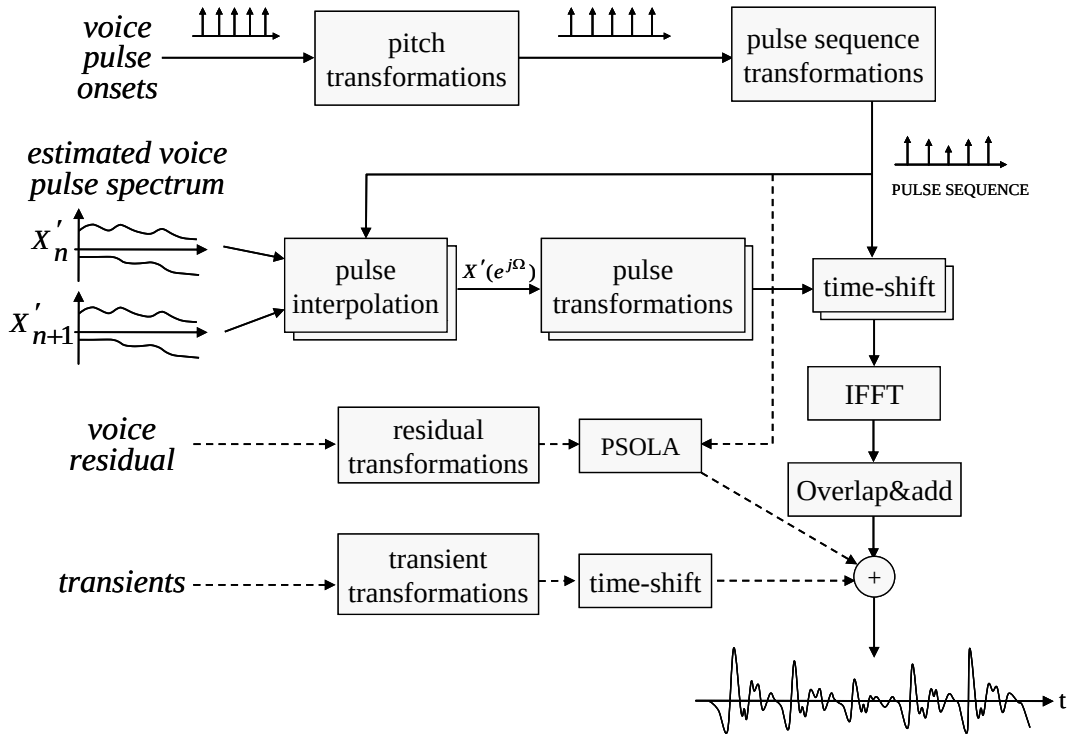
Once we have filled the sequence, we map each pulse to one analysis frame and compute the radiated voice pulse spectrum. We can choose the nearest frame or interpolate the spectrum of the two closest frames to get smoother timbre transitions, which can be specially useful when applying a time-stretch transformation. At this point we can synthesize each pulse independently by means of an IFFT and then position it in the desired location. This requires an interpolation of the samples in time domain because pulse time onsets are not quantized to the sampling rate. However, the positioning can also be directly made in frequency domain by adding a linear phase slope to the phase spectrum proportional to the required time-shift, which can written as

$$\Delta\phi_r(\Omega) = (t_r - t_{frame})\Omega \tag{2.79}$$

where $\Delta\phi_r(\Omega)$ is the phase increment for the r^{th} pulse and t_{frame} is the center time of the current frame. We have to be careful here about the hop and window size values because there is a limit of how much the signal can be shifted by this method, since the window is finite and short. If the window size is N we cannot time-shift the pulse more than $N/2$ samples, otherwise the pulse will appear in the opposite part of the window as an aliasing effect. This time-shift operation can be efficiently computed on a complex spectrum since the phase shift increases by a constant amount for consecutive bins, so it is enough to compute only two cosines for the whole spectrum and then two complex multiplications per bin.



(a)



(b)

Figure 2.62 Block diagrams of (a) analysis and (b) synthesis phases of the narrow-band voice pulse modeling

Depending on the synthesis configuration, several pulses can be combined into a single IFFT so to speed up the process. This way we generate a complex spectrum for each pulse and add it to the IFFT buffer. The computational cost for each synthesis frame includes then a polar to complex conversion for each used analysis frame plus a time shift of the whole spectrum (eq. (2.79)) for each synthesized pulse. The computational cost would then be given by

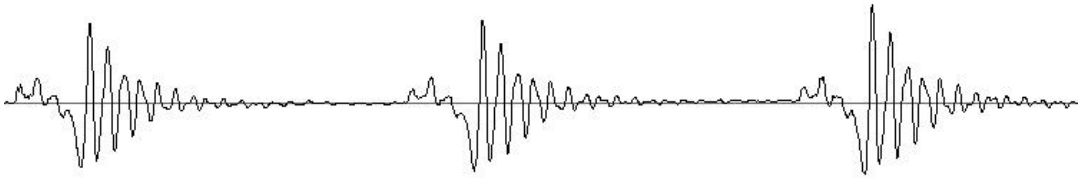


Figure 2.63 Recording of a vocal fry phonation where the whole radiated glottal pulse is visible

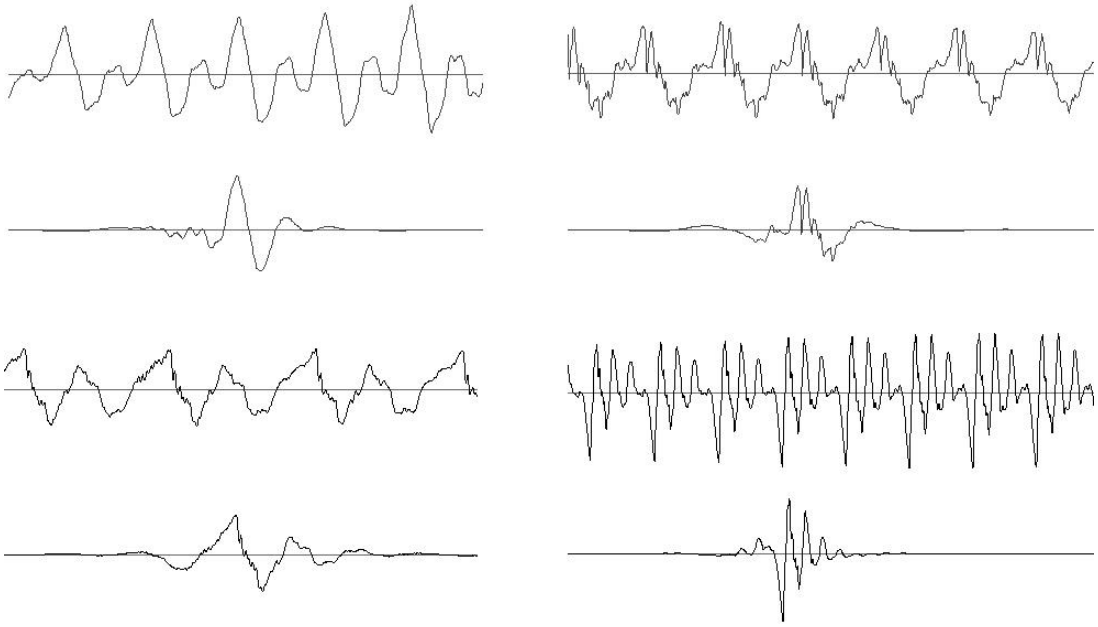


Figure 2.64 Synthesis of a single pulse from the analysis of a recorded voiced utterance. On three of the examples the synthesized pulse location has been synchronized to the original one

$$
\begin{aligned}
C &= n_A \cdot p2c + 2 \cdot n_S (cmul \cdot N + cos) + IFFT_N \\
p2c &= Polar\ to\ complex\ conversion \\
cmul &= complex\ multiplication \\
cos &= cosinus\ calculation \\
N &= synthesis\ window\ size \\
n_A &= analysis\ frames\ used \\
n_S &= synthesis\ frames\ used = \frac{pitch \cdot hopsize}{f_S}
\end{aligned}
\tag{2.80}
$$

However, in certain contexts the polar to complex conversion can be done in the analysis stage, this way significantly reducing the synthesis computational cost. This would be the case, for example, of a sample based singing voice synthesizer that reads a pre-analyzed database.

The resulting time domain signal after the IFFT must be multiplied by an overlapping window and added to the output sound buffer. A triangular window would be fine but it must have a size shorter than N to avoid artifacts due to time aliasing (the edges of the synthesized buffer will not be used). It is interesting to point out here that the estimated spectrum of the radiated voice pulse $X'\left(e^{j\Omega}\right)$ is not convolved anymore with the analysis window, thus the reconstructed signal does not need to be divided by it before overlapping.

In Figure 2.64 we observe several examples where a single pulse has been synthesized out of the analysis of the voiced utterance. Besides, Figure 2.63 shows several almost isolated radiated voice pulses obtained from a recording of a vocal fry phonation with an extremely low pitch of about 30 Hz.

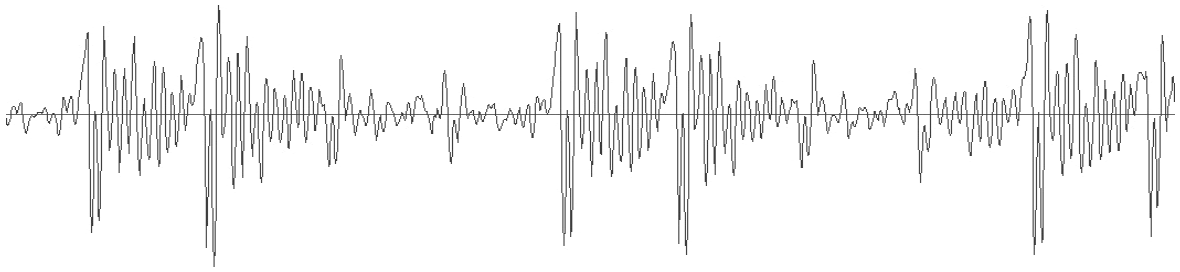


Figure 2.65 recorded waveform of a growl utterance

TRANSFORMATIONS

Several voice transformations can be achieved using this technique, from high level (pitch-shifting, time-scaling, timbre modifications) to low level ones (independent pulse control). In this section we will briefly present how some of them can be applied.

Pitch and time-scaling transformations can be implemented in a similar way to TD-PSOLA, controlling the speed at which the sequence of analysis frames is read and the distance between pulses. The timbre will be preserved as long as the estimated amplitude spectrum is not modified.

Timbre can be modified by scaling, warping or equalizing the estimated spectral amplitude of each pulse. In the real case, if a formant is shifted in frequency the related phase shift is shifted as well. This means it would be desirable to scale and warp the spectral phase envelope as well. This could be done by applying the same transformation to both amplitude and phase envelopes of the polar spectrum or just applying it once to the complex spectrum.

Several voice disorders, intentional or not, can be characterized by irregularities in the excitation glottal pulse sequence, both in time (jitter) and amplitude (shimmer). These irregularities can be described mostly by the appearance of subharmonics in the spectrum, which are hard to manipulate in frequency domain. However, in our case, we have an independent control of the location and amplitude of each pulse, thus easily different patterns of irregularities can be synthesized and even vary along time. We have observed that sometimes in a growl pattern (as the one shown in Figure 2.65) voice pulses have different timbres that differ by some equalization. In this example the two pulses with more amplitude experience a significant boost of energy around 4 Khz. This behavior can also be reproduced with the presented technique by filtering differently each pulse's amplitude spectrum.

RESIDUAL

In the estimation of $X'\left(e^{j\Omega}\right)$ we have obtained a spectrum from the interpolation of the harmonic peaks. It is clear that we have disregarded all the data contained in the spectrum bins between harmonic frequencies. This data often explain irregularities of the pulse sequence (time and amplitude) plus noisy or breathy characteristics of the voice. These pulse sequence irregularities could be approximated by properly analyzing the unwrapped fundamental phase envelope and the outputs of the MFPA. On the other hand, the breathy part could be synthesized using part of the original spectrum data.

Initially we tried to use the residual obtained by subtracting in frequency domain perfect sinusoids with the amplitude, frequency and phase values of the detected harmonics to the original spectrum. However, sometimes we found that some harmonic information was kept, especially in transitions, probably due to the fact that we were actually subtracting stationary sinusoids to each frame. Results were improved by applying comb filters to this residual that attenuate the frequency bands around harmonic frequencies, therefore minimizing the presence of the original pitch in the residual. Adding this residual to the synthesized signal produced a breathy characteristic very close to the original one. Nevertheless, better results are obtained by subtracting the NBVPM resynthesis to the input signal, therefore ensuring a perfect reconstruction when no transformations are applied. It is

well known that the aspirated noise produced during voiced utterances has a time structure, which is perceptually important, and is said to be correlated with the glottal voice source phase (Kob 2002). Hence, with the aim of preserving this time structure, the residual is synthesized using a PSOLA method synchronized to the synthesis voice pulse onsets. Finally, for processing transient-like sounds we adopted the method in (Röbel 2003), which by integrating the spectral phase is able to detect transients and discriminate which spectral peaks contribute to them, therefore allowing translating transient components to new time instants.

UNVOICED SIGNALS

The NBVPM algorithm was conceived to be used in voiced sections, since it assumes a periodic signal as input. Actually, in our experiments we have combined it with a phase-vocoder based technique for processing unvoiced segments, in which a white noise source is filtered with the spectral amplitude envelope of the input signal. Although this method allows in certain cases applying transformations with high quality results (e.g. time-scaling of fricative consonants), it fails to properly handle transients, so plosive consonants are smeared and intelligibility is degraded. In order to improve the results we have adapted a processing algorithm that is able to detect transients and discriminate which spectral peaks contribute to them (Röbel 2003).

DISCUSSION

NBVPM is similar to old techniques such as FOF (Rodet, et al., The CHANT Project: from the Synthesis of the Singing Voice to Synthesis in General 1984) and VOSIM (Kaegi, et al. 1978), where voice is modeled as a sequence of pulses whose timbre is roughly represented by a set of ideal resonances. However, in NBVPM the timbre is represented by all the harmonics, allowing capturing subtle details and nuances of both amplitude and phase spectra. In terms of timbre representation we could obtain similar results with spectral smoothing by applying restrictions to the poles and zeros estimation in AR, ARMA or PRONY models. However, with NBVPM we have the advantage of being able to smoothly interpolate different voice pulses avoiding problems due to phase unwrapping, and decomposing the voice into three components (harmonics, noise and transients), which can be independently modified.

NBVPM works in narrow-band conditions and successfully allows having an independent control of each voice pulse while at the same time providing flexible transformations. Maybe its main drawback is precisely the fact that it works in narrow-band conditions. The reason for that relies in the intrinsic non-stationary characteristics of the human voice, in the fact that the glottal pulse sequence is inherently irregular and that the articulatory system is continuously moving during voiced utterances. Working in narrow-band conditions implies that the analysis window contains several consecutive voice periods, which might be radically different in note and phonetic boundaries. Thus, the initial hypothesis of NBVPM analysis that assumes a periodical repetition of a short-time voice pulse becomes false and estimated voice pulses get smoothed and smeared.

Modeled voice pulses represent only the harmonic content of the voice signal. Thus, NBVPM adds a residual signal obtained by subtracting overlapped voice pulses to the original signal. This residual signal is processed with TD-PSOLA method so to effectively preserve its time-structure synchronized to the glottal pulse sequence. However, a transient model is still required to preserve the naturalness of plosive consonants and other impulsive signals present in voice utterances. To achieve a natural sounding synthesis by transformation and combination of all these models and signals requires doing a very good and precise analysis job. Small analysis errors can sometimes produce annoying artifacts perceptually relevant. Nevertheless, NVBPM is able to produce voice transformations of high quality. However, it makes sense to explore the possibility of working in wide-band conditions, this way increasing the temporal resolution and the likelihood to analyze a quasi-stationary signal, and at the same time aim to improve the robustness to analysis imprecisions. The following sections describe our attempts in this direction.

2.3.2 Wide-Band Voice Pulse Modeling (WBVPM)

One of the main interests of working in wide-band conditions is that of increasing the temporal resolution. Hence, our intention is to estimate the harmonics parameters of a periodic signal (in this case the singing voice) in the widest possible band conditions by means of a STFT. The proposed voice transformation method relies on the hypothesis that voice signals consist of a train of pulses at the pitch rate, where here pulse is defined as the audio between consecutive pulse onsets, not as the result of filtering a glottal pulse through the vocal tract, which might be shorter or longer than the actual pitch period. As it will be shown, this method models each of these harmonic pulses in the frequency domain with a vector of pure sinusoids. It can be considered a hybrid method in the context of this chapter because it uses harmonic guides while at the same time it models voice pulses individually, but also in the sense that it models both harmonic and noise components with sinusoids. Moreover, the proposed method overcomes the typical time-frequency constraints of frequency-based techniques were several periods of signal are required in order to achieve enough frequency resolution for estimating harmonic components, thus avoiding introducing smearing in the synthesized signal and decreasing the temporal resolution.

Let $s(n)$ be a periodic signal. We assume that the period of the signal has been already estimated by any appropriate technique (e.g. (Cheveigné and Kawahara 2001)). Let's define $s(n)$ as a stationary periodic signal sampled at a rate of f_s, composed of $T/2$ sinusoids with constant amplitude, frequency and initial phase values, and a known fundamental period of T samples,

$$s(n)=\sum_{k=1}^{T/2} a_k \cos\left(2\pi\frac{f_k}{f_s}n+\theta_k\right) \quad , \quad f_k=\frac{kf_s}{T}. \tag{2.81}$$

The discrete-time STFT of $s(n)$ using a rectangular window $w_R(n)$ is given by

$$x(n)=s(n)w_R(n) \tag{2.82}$$

$$X(f)=\sum_k W_R(f_k-f)S(f_k) \tag{2.83}$$

where f_k denotes the harmonic frequencies, and $S(f)$ and $W_R(f)$ are respectively the DTFT of $s(n)$ and $w_R(n)$. Thus the value of $X(f)$ at an arbitrary frequency f is the result of the contribution of all harmonic components multiplied by the transform of the window evaluated at the frequency difference f_k-f.

The DTFT of a normalized rectangular window of N samples is given by

$$w_R(n)=\begin{cases}1/N & for \quad 0\le n\le N-1\\ 0 & for \quad n\notin[0,N-1]\end{cases} \tag{2.84}$$

$$W_R(f)=\sum_{n=0}^{T-1}\frac{1}{N}e^{-j2\pi\frac{f}{f_s}n}=e^{-j\pi\frac{f}{f_s}(N-1)}\frac{\sin\left(\pi\frac{f}{f_s}N\right)}{N\sin\left(\pi\frac{f}{f_s}\right)}. \tag{2.85}$$

Note that it has zeros at frequencies $f_g=(gf_s)/N$, $g=1,2,...,N-1$.

Since the fundamental period is known, the harmonic frequencies are also known ($f_k=kf_s/T$) and therefore we would like to arrive to $X(f_k)=S(f_k)$. Observing equation (2.83), this will be true if the energy contribution of a given harmonic to other harmonic frequencies is zero. In other words,

$$W_R\left(\frac{kf_s}{T}\right)=0 \quad \forall k\in[1,T-1]. \tag{2.86}$$

The previous condition will happen whenever the length of the rectangular window is a multiple of the signal's period ($N=gT, g\in\mathbb{N}$). Therefore, the maximum widest-band condition is achieved for $N=T$, when the rectangular window covers exactly one period of the signal.

In practical implementations it is inefficient to compute the DTFT. Instead, the DFT is used, which actually samples the DTFT at frequencies equidistant by f_s/N . Denoting the DFT of $x(n)$ as $\bar{X}(k)$ we obtain

$$\bar{X}(k)=\sum_{k=0}^{N-1}x(n)e^{-j2\pi\frac{k}{N}n}=X\left(\frac{kf_s}{N}\right). \tag{2.87}$$

For $N=T$ we obtain

$$\bar{X}(k)=X\left(\frac{kf_s}{T}\right)=X(f_k)=S(f_k). \tag{2.88}$$

This means that each bin of the DFT actually corresponds to one harmonic of $s(n)$, and that from its complex value we can simply compute harmonic parameters as

$$\begin{aligned} f_k &= \frac{kf_s}{T} \\ a_k &= \left|\bar{X}(k)\right| \qquad k=1,\ldots,\frac{T}{2}. \\ \theta_k &= \measuredangle\bar{X}(k) \end{aligned} \tag{2.89}$$

This way we can efficiently estimate the harmonic parameters from one individual signal period without spectral smoothing due to the windowing process. For computational efficiency, it is preferred to use the FFT algorithm for computing the DFT. However, if T is not a power of 2, using the FFT algorithm will require us to zero-pad the signal and this will modify the frequency of the spectral bins so that they will not correspond anymore to a harmonic. Moreover, the FFT is limited to an integer number of samples but not the period T, which is a real value.

NON-INTEGER SIZE FFT

There are several ways for computing the spectrum of a non-integer number of samples using the FFT algorithm:

- PERIODIZATION: one period of the input signal is windowed with $w_R(n)$, and repeated several times at the rate defined by T so that the FFT buffer of length M covers in the end several periods. The repetition implies interpolating both the signal samples and the window function. Then the resulting signal $s_r(n)$ is windowed by an analysis window function $w_A(n)$, and the spectrum obtained is actually the convolution of such analysis window response $W_A(f)$ by the spectrum of $S_r(f)$ sampled at harmonic frequencies, i.e.

$$X_r(f)=\sum_k W_A(f-f_k)S_r(f_k) \tag{2.90}$$

  where actually $S_r(f)$ is the STFT of length T. In general, the frequencies of the spectral bins do not correspond to the harmonic frequencies but to

$$\bar{X}_r(b)=\sum_{b=0}^{M-1}x_r(n)e^{-j2\pi\frac{b}{M}n}=X_r\left(\frac{bf_s}{M}\right). \tag{2.91}$$

  Therefore estimating harmonic parameters (i.e. frequency, amplitude and phase) requires interpolating the spectral bins around harmonic peaks. Besides, zero-padding can help to improve the estimation accuracy. This method is depicted in Figure 2.95, although a rectangular window of T samples is not used but a longer one so to overlap samples at borders and therefore avoid discontinuities. In the following section it will be shown the need for this overlapping method.

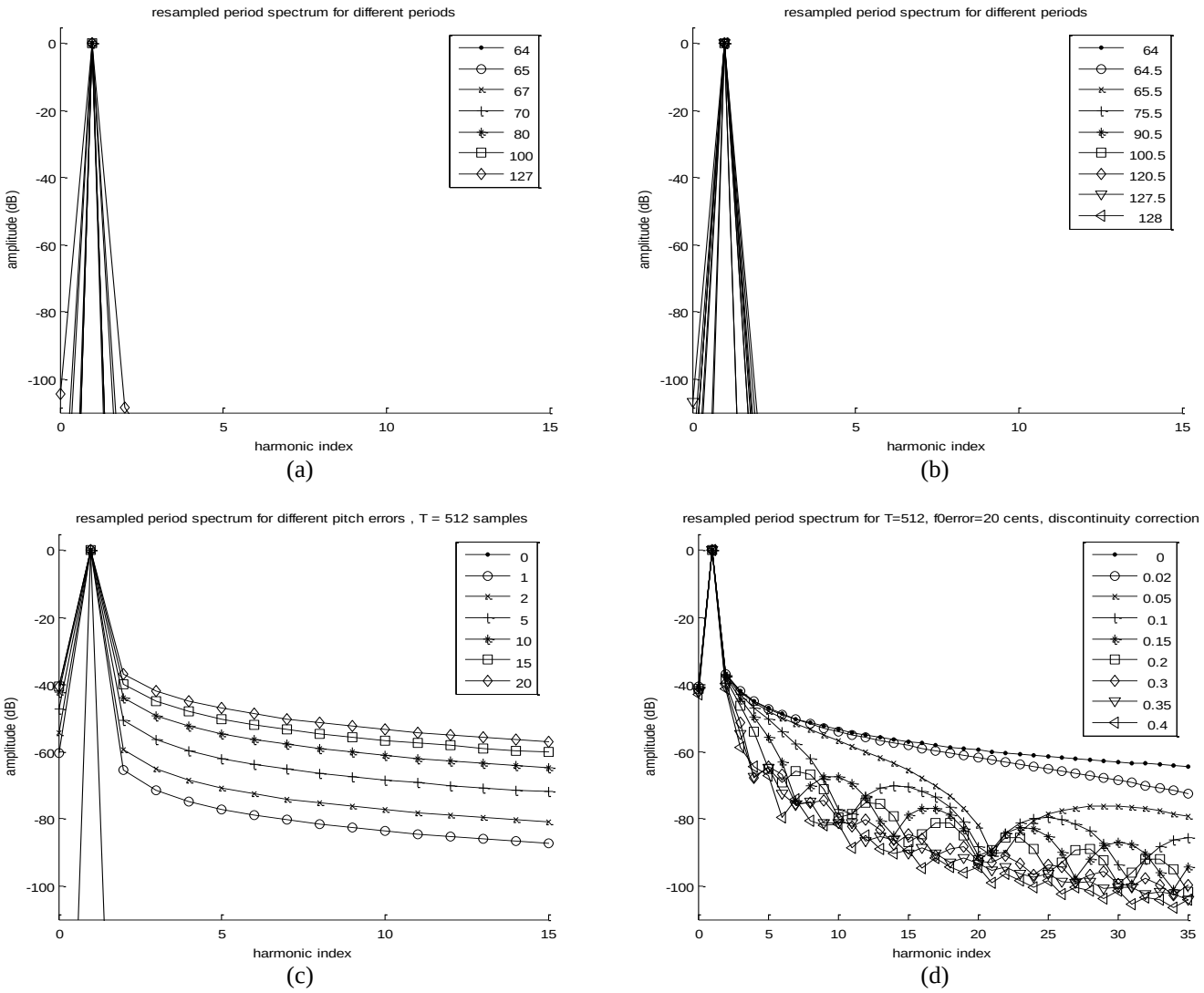


Figure 2.66 Inter-harmonic energy contribution

- UPSAMPLING: Another way of computing the STFT of a non-integer number of samples is to upsample the input signal so that one period matches the closest FFT size M, i.e. $M = 2\hat{}\left(\lfloor \log_2(T) \rfloor + 1\right)$. Downsampling is not desirable in this case because some of the higher harmonics should be removed to avoid aliasing. Computing the FFT of the upsampled signal $s_u(n)$ would result into

$$\begin{aligned} \bar{X}_u(k) &= X_u\left(\frac{kf_s}{M}\right) = \sum_g W_R\left(f_g - \frac{kf_s}{M}\right) S_u(f_g)\Big|_{f_g = \frac{gf_s}{T}} \\ &= S_u(f_k) \end{aligned} \tag{2.92}$$

where $S_u(f)$ is the STFT of length T and only the first bins up to $T/2$ would be relevant.

Ideally both methods would output exactly the same results. However, due to inaccuracies of the sample and spectral interpolation methods used some differences are expected, although insignificant.

INTER-HARMONIC ENERGY CONTRIBUTION

We saw before that in order to achieve $\bar{X}(k) = S(f_k)$ the energy contribution of each harmonic to other harmonic frequencies should be zero. Thus, in order to have an initial evaluation of the

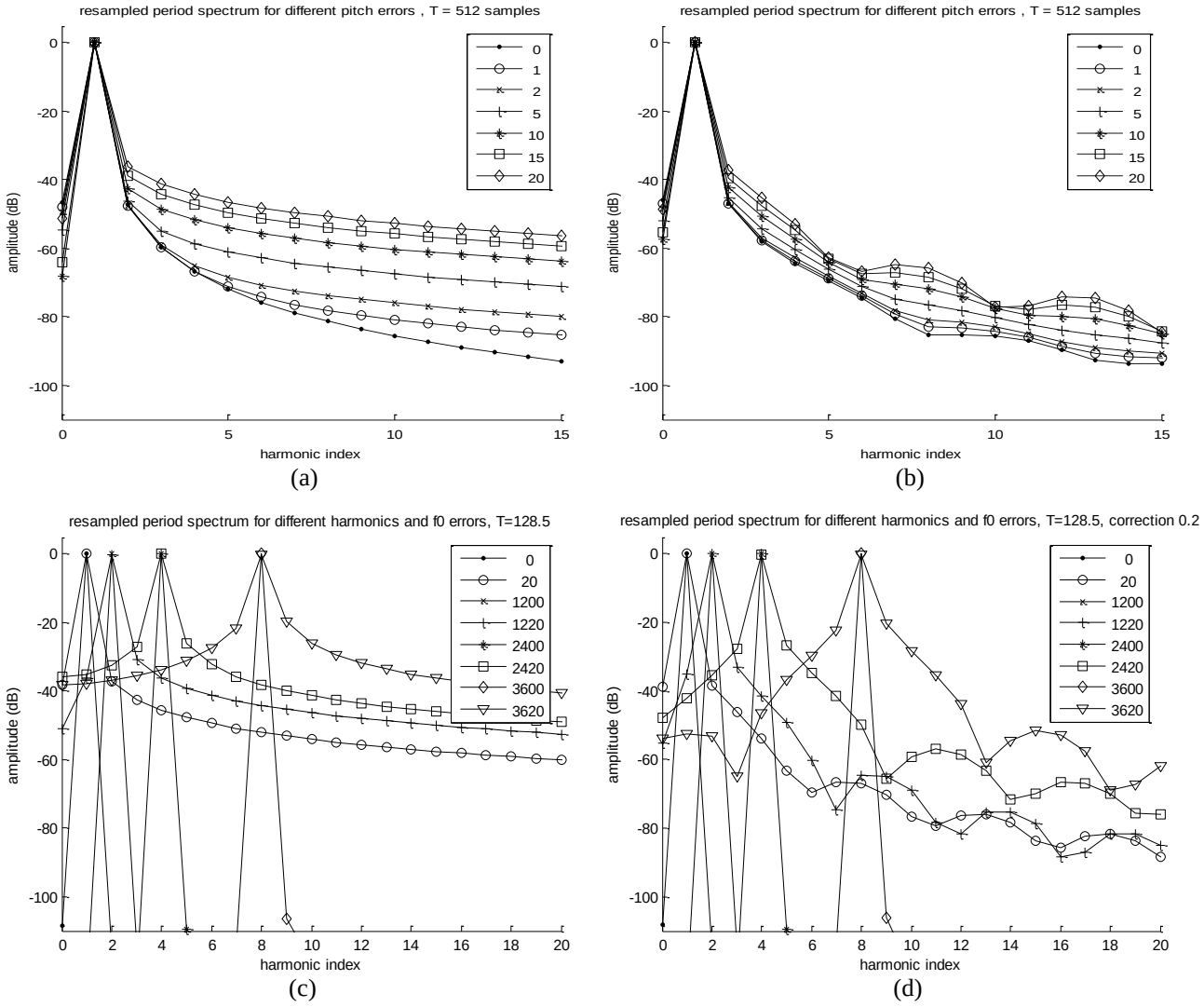


Figure 2.67 Inter-harmonic energy contribution

goodness of the proposed approach, we explore the inter-harmonic energy contribution by computing the one-period-STFT of a sinusoid with the upsampling method. This gives us a measure of the noise present at other harmonic frequencies. Figure 2.66abcd and Figure 2.67ab show the one-period-STFT of a signal containing only one sinusoid at the fundamental frequency, whereas Figure 2.67cd show the result when the sinusoid frequency corresponds to different multiples of the fundamental frequency. Inaccuracies introduced by any of the estimators or interpolation methods will degrade the analysis performance. We consider the following aspects:

- UPSAMPLING: the upsampling process is performed using a polyphase implementation. Figure 2.66a shows the case of integer period values where contributions are negligible since they fall below -100dB. Periods between 65 and 127 are upsampled to have a length of 128 samples.
- NON-INTEGER PERIODS: In Figure 2.66b we observe negligible contributions for several real-valued periods between 64 and 128.
- PITCH ESTIMATION ERRORS: Figure 2.66c shows the contributions for pitch estimation errors up to 20 cents. The contribution to the adjacent harmonic goes from -65.45dB/1cent to -36.73dB/20cents. The reason for this bias relies both in the discontinuity between borders of the STFT input signal and the fact that bin

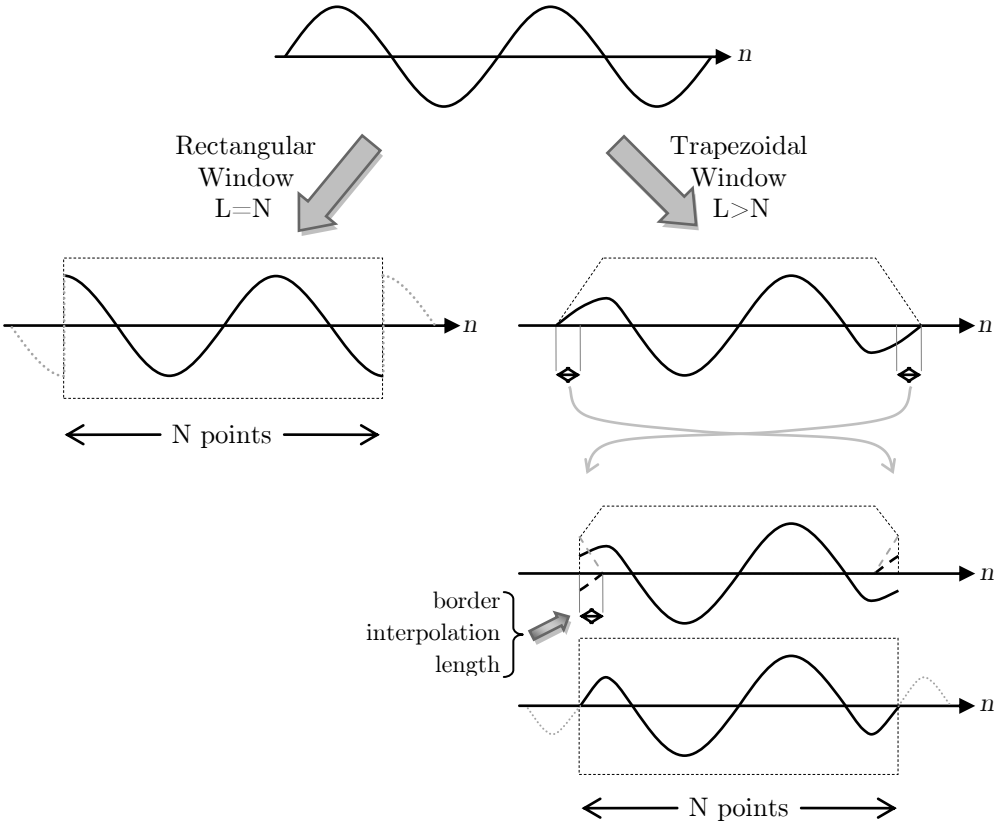


Figure 2.68 Border interpolation in the upsampling case. This figure illustrates how using a longer trapezoidal window minimizes discontinuities found with the rectangular window

frequencies depart from harmonic frequencies. In (d) we see how the numbers can be greatly improved by interpolating the values around borders in the way shown in Figure 2.68 and Figure 2.95.

- NON-STATIONARY SIGNALS: Figure 2.67a shows the results in the case of both non-stationary sinusoids and pitch estimation errors. The sinusoid frequency shifts approximately from 125 to 133Hz along the analysis window, and the estimation errors go from 0 to 20 cents. Obviously the best case is when the fundamental frequency is well detected, with contributions around -50 and -60dB for the two closest partials, and slowly decaying to -90dB for the 15^{th} harmonic. These values are good enough for real world signals. However, the worst case of 20 cents is not that good. The contributions range from -38dB for the 2^{nd} harmonic to -45dB for the 15^{th} harmonic. Applying interpolation around borders as previously exposed the results can be greatly improved, as shown in Figure 2.67b, with values falling from -39 to -80 dB, good enough for practical uses.
- OTHER HARMONICS THAN FUNDAMENTAL: Figure 2.67c shows the comparison between the contribution from fundamental and higher harmonics (2^{nd}, 4^{th} and 8^{th}), with and without pitch estimation errors of 20 cents. Clearly, the contribution increases significantly for higher harmonics. For instance, the 8^{th} harmonic estimated with a bias of 20 cents contributes to the surrounding ten harmonics with more than -40dB. Overlapping around the borders improve the results, as shown in Figure 2.67d, increasing significantly the contribution decay. It is important to mention that most common musical sounds and human voice tend to present spectra with energy decaying along frequency. Therefore, the observed increase of inter-harmonic contribution along frequency is not that relevant for achieving good results.

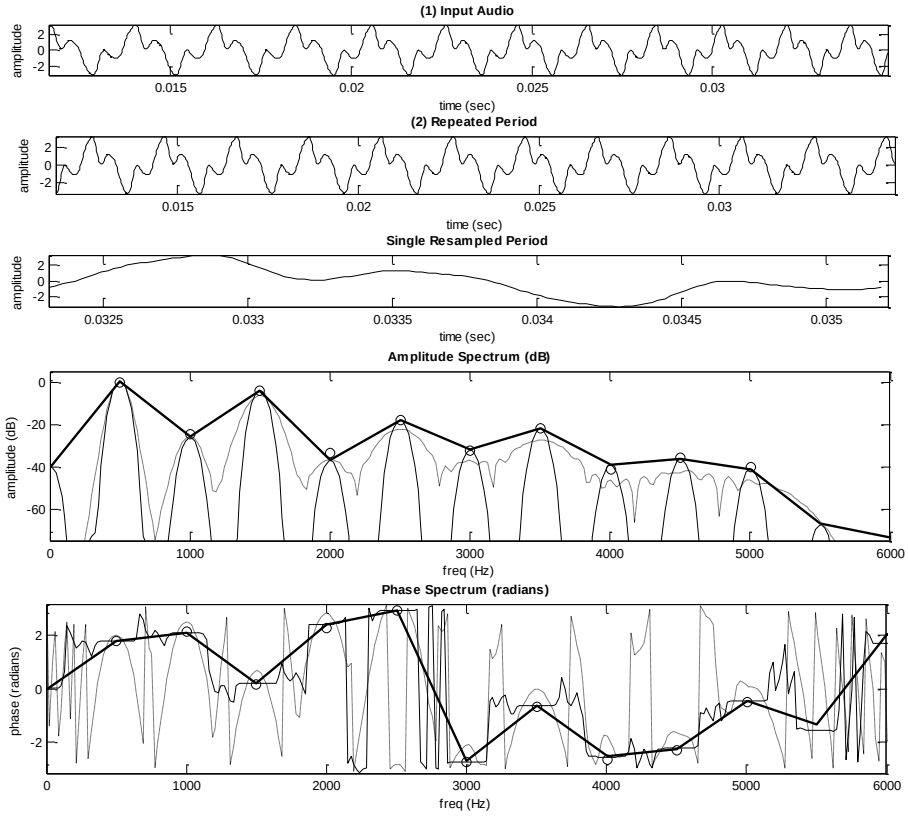


Figure 2.69 Wide-band versus narrow-band analysis of a synthetic periodic signal

SINUSOIDAL MODELING

Figure 2.69 shows the spectra obtained from a synthetic signal using those methods and a regular narrow-band STFT. The input signal consists on ten sinusoids whose frequencies are multiples of the lower one. The fundamental frequency increases along time, as can be seen in the top view (a) where periods on the left are longer than those on the right. (b) shows the waveform resulting of repeating the period in the center, whereas in (c) we see the upsampled period. Finally, (d) and (e) show respectively the amplitude and phase spectra of the previous signals, where (a) is drawn with dashed lines, (b) with solid lines, and (c) with thick solid lines. The STFT of (a) presents clear amplitude peaks only at the lower frequency harmonics, getting noisy for higher frequencies due to the non-stationary nature of the analyzed signal. Instead the STFT of (b) presents clear peaks at expected harmonic frequencies, but also above 5Khz where no harmonics are present. This is explained by the inter-harmonic energy contributions previously discussed. By contrast, the STFT of (c) has one bin per harmonic with values matching those of (b). The exact harmonic parameters values are displayed as circles. Clearly, (b) and (c) STFTs approximate much better the input signal than (a). For instance, (a) shows bias of up to -10 dB and 0.5 radians for harmonics above 3Khz.

With the two methods previously presented (i.e. periodization and upsampling), the resulting audio signals are purely periodic. Therefore, their spectra can be perfectly represented by a set of stationary sinusoids. It is then straightforward to use a sinusoidal model for the proposed wide-band analysis. Moreover, since the harmonic frequencies depend only on the estimated fundamental period, there is no need to use any complex method for building the harmonic trajectories along consecutive periods, but simply to connect the harmonics with the same index.

The proposed method can be divided in three main phases, namely analysis, transformation and synthesis, as shown in Figure 2.70. In the analysis phase, the input signal is segmented into consecutive periods that are modeled with a set of sinusoids as already explained previously. In the

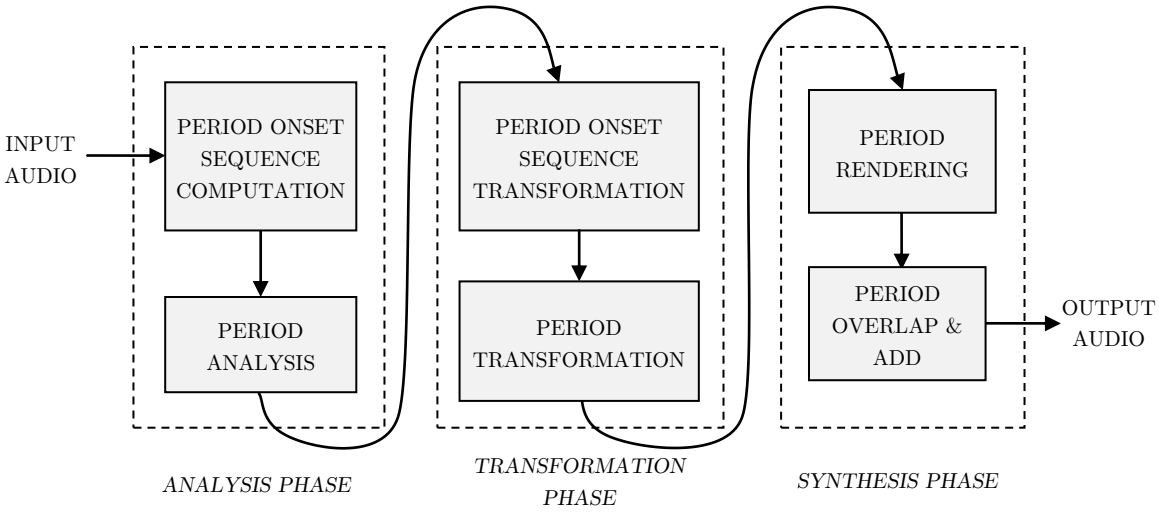


Figure 2.70 Block diagram of the different phases of the algorithm modeling voice pulses in wide-band conditions.

following sections, we study the harmonic estimation accuracy, then detail both transformations and synthesis phases, and finally discuss how the proposed method is adapted to the case of the human voice and to unvoiced signals.

HARMONIC ESTIMATION ACCURACY

We studied before the energy contribution between harmonics. Now we explore the accuracy of the harmonic estimator. In order to do that, we analyze the one-period-STFT of synthetic signals consisting of a single stationary sinusoid corresponding to one harmonic, and then we measure the resulting amplitude and phase estimation errors. Thus, each signal has only one sinusoid, either the fundamental or one of the other harmonics. Figure 2.73ab and Figure 2.74ab show the results obtained for the first 16 harmonics considering fundamental frequency estimation errors up to 20 cents and different border interpolation lengths (up to 30 percent of the period duration), denoted as ov_L and normalized by the signal fundamental period. In all cases, when the fundamental frequency is well detected, the estimation errors are insignificant. By the contrary, fundamental frequency estimation errors (let's denote them as e_{f0}) produce harmonic amplitude bias that increase along frequency. For instance, the amplitude errors of the 16^{th} harmonic are as high as 0.5dB for $e_{f0}=20$ cents. Obviously, for higher e_{f0} values we obtain higher amplitude estimation errors. The effect of interpolating analysis window borders improves the results for the first harmonics, but not for the rest. Interestingly, it smoothes the amplitude error envelopes so that for large interpolation lengths they look almost like as an exponential function as shown in Figure 2.74b. By contrast, phase estimation errors are reasonably improved by window border interpolation. While in Figure 2.73a errors seem to behave sinusoidally along frequency for $ov_L=0$ with an amplitude up to 0.006 radians, in the following figures that have higher interpolation lengths ($ov_L=0.1, 0.2, 0.3$) errors decrease along frequency and the maximum absolute values get lower.

COMPARISON TO STANDARD METHODS

For comparing the accuracy of the proposed harmonic estimator to other standard methods, we have generated a set of synthetic signals and computed both the wide and narrow-band STFT and the one-period STFT using the periodization method. Then we have estimated the harmonic parameters for each of those frequency representations, and computed the estimation errors as the difference between the estimated values and the values used when generating the signal. In the narrow-band case, the analysis window covers several periods of the analyzed signal. Harmonic parameters are computed by quadratic interpolation if an amplitude peak is found at harmonic frequencies, otherwise linear interpolation is used. The fundamental frequency does not need to be

estimated because it is already known: it is the one used when generating the synthetic signal. Figure 2.75 to Figure 2.82 show the results obtained. In all those figures we find the following representations from top to bottom:

1. Input signal time domain waveform
2. Waveform resulting of the periodization of the period centered in the input signal
3. Single period waveform with the one-period long analysis window overlapped
4. Evolution of the fundamental frequency along time. The range in semitones is specified in the title
5. Amplitude error of the estimated harmonics for both narrow (black) and the proposed wide-band (red) cases, expressed in dB. The mean absolute error is specified in the title
6. Phase estimation errors for both narrow (black) and the proposed wide-band (red) cases, expressed in radians. The mean absolute error is specified in the title
7. Amplitude spectra in narrow-band conditions (black), wide-band conditions (green) and the periodization method (red). The exact harmonic values are drawn with blue circles.
8. Phase spectra in narrow-band conditions (black), wide-band conditions (green) and the periodization method (red). The exact harmonic values are drawn with blue circles.

For simplicity, we have only considered the low frequency band between 0 and 4Khz. Besides, with the aim of resembling typical characteristics of the human voice, the synthetic signals consist of a set of harmonic sinusoids with time-varying fundamental frequency and a spectral envelope showing two significant resonances around 1 and 2 KHz. Table 2.4 presents the results obtained. The fourth column, *narrow-band number of periods*, refers to the manually counted number of signal periods covered by the analysis window, where those might have different lengths due to the fact that f_0 is time varying.

As expected, for most cases the overall error obtained with WBVPM is significantly lower than for standard narrow and wide-band approaches. In addition, amplitude errors are in general much higher for wide-band analysis than for the other two methods. The main source of error in the narrow-band case is due to the non-stationary nature of the analyzed signal, which affects the shape of the spectrum around each harmonic frequency in different ways, distorting the shape of the window transform (see §2.1). Therefore, in general for higher fundamental frequency excursions (i.e. higher non-stationary signals) we obtain higher estimation errors. In addition, the more signal periods covered by the analysis window the more likelihood to increase the non-stationary characteristic. By contrast, in the wide-band approach the estimation bias originates mainly from the poor frequency resolution, since the main lobe of the transform of the analysis window is several harmonics wide. This happens as well in the narrow-band approach for analysis windows of only a few periods long, such as in Figure 2.80 to Figure 2.82, in which spectral envelopes are very smooth and amplitude peaks at harmonic frequencies are sometimes inexistent, or shifted in frequency. WBVPM minimizes both sources of errors since (1) it forces the signal to be stationary by analyzing a single period, and (2) it matches the size of the analysis rectangular window to exactly one period of the signal, this way minimizing the windowing effect. However, the fundamental frequency estimation is a significant cause of errors in WBVPM, since f_0 determines the exact frequencies where the spectrum is evaluated, whereas in the narrow-band analysis those estimated harmonic frequencies are a guide to locate amplitude peaks, which are a better estimation if the signal is quasi-stationary.

One could say that in all these examples the analysis window length in narrow-band analyses should be adapted to cover only two or three periods of the harmonic signal to minimize the non-stationary characteristic. However, it is not rare in spectral processing to use windows that cover more than three periods of the voice signal, and therefore it makes sense to consider those cases as well. For instance, in the phase-vocoder typically the window length is constant, with a length long enough as to cover a few periods of the minimum fundamental frequency value expected (e.g. (Laroche 2003)), therefore covering more periods for higher f_0 values. Also, a long analysis window is

Figure	f_0 shape	f_0 range (semi-tones)	narrow band number of periods	harmonic amplitude mean absolute error (dB)			harmonic phase mean absolute error (radians)		
				narrow band	*WBVPM*	*wide band*	*narrow band*	*WBVPM*	*wide band*
Figure 2.75	peak	1.10	~10	0.705	0.098	7.183	0.244	0.009	0.615
Figure 2.76	raising	3.50	~9.5	3.686	0.211	7.294	0.514	0.028	0.589
Figure 2.77	decreasing	5.26	~8.5	4.673	0.413	6.429	0.559	0.047	0.687
Figure 2.78	raising	7.02	~18.5	8.976	0.209	7.294	0.747	0.028	0.588
Figure 2.79	decreasing	10.28	~17	7.691	0.414	6.428	0.875	0.047	0.686
Figure 2.80	raising	1.15	~3	0.401	0.305	6.215	0.156	0.056	0.564
Figure 2.81	decreasing	2.15	~3	1.049	0.830	6.501	0.182	0.110	0.445
Figure 2.82	peak	0.14	~3.3	0.036	0.011	3.796	0.0034	0.0018	1.001

Table 2.4 Analysis results of synthetic signals using wide-band, narrow-band and WBVPM analysis techniques

interesting to use for transforming harmonics and noisy components independently directly in the frequency domain representation, since then the width of the harmonics (i.e. the width of the window transform) is significantly narrower than the frequency distance between harmonics. A typical application is to modify the ratio between harmonic and noise (e.g. (Fabig and Janer 2004)).

It is also interesting to include some recordings of real voice signals in the comparison, although in those cases we do not have the reference values (i.e. we do not know the fundamental frequency exact values neither the harmonic parameters). Figure 2.83 to Figure 2.86 show the analysis results obtained for a male voice (reference audio [103]). In Figure 2.83 and Figure 2.84 the analysis window is centered at the same sample, but its length is different; whereas in the former figure approximately 10 periods are covered, in the latter only 5. The signal looks highly stationary in both cases, since the fundamental frequency only varies 0.54 semitones in the first figure and 0.39 semitones in the second one. While the wide-band spectra look exactly the same, the narrow-band and WBVPM spectra differ significantly, although the WBVPM spectral peaks are about the same. Obviously, the width of the window transform is much wider in the second figure where the window length is shorter. Comparing all those spectra it looks clear that the wide-band analysis performs poorly and harmonics cannot be discriminated. We cannot be sure that WBVPM estimates correctly all the harmonics, but we can argue that both narrow-band and WBVPM analyses give good estimations for the first harmonics up to 1500Hz. For the other harmonics, we can see how the narrow-band spectra get closer to the WBVPM one in the second figure. For instance, look at the harmonics between 1.7 and 2.5Khz. However, in the narrow-band case spectral peaks do not correspond anymore to harmonic frequencies, especially above 3.7Khz, what suggests bad estimation results. About the following figures Figure 2.85 and Figure 2.86 we could make similar comments. However, the voice signal is not locally stationary anymore: f_0 excursions are much higher than before, 6.14 and 3.62 semitones respectively, and we see significant energy and shape changes in the time-domain waveform. This for sure degrades especially the performance of the narrow-band estimator, whose amplitude spectral peaks do not seem to follow harmonic frequencies.

RESIDUAL ENERGY

With the aim of estimating how good the accuracy of the WBVPM harmonic estimators is, we measure the energy of the residual signal. The idea is that better estimations correspond to lower residual energies. In our experiments, we have obtained the residual signal by subtracting in time

domain from the input signal sinusoids synthesized with the estimated parameters, being the sinusoidal synthesis performed with a bank of time-domain oscillators.

- SYNTHETIC SIGNALS

  We have first considered non-stationary synthetic signals consisting of harmonically related sinusoids with strong amplitude and frequency modulations. In Figure 2.87 we see one example, where WBVPM and narrow-band analysis results are in green and red respectively. The (a) view shows the amplitude value of each harmonic at different time instants. (b) shows the frequency and amplitude functions of the 1^{st}, 2^{nd},3^{rd} and 5^{th} harmonics, together with the WBVPM and narrow-band estimations. We can appreciate that WBVPM estimations resemble more a sinusoid than the narrow-band ones, especially for the 3^{rd} and 5^{th} harmonic. It is interesting to point out that narrow-band estimations sometimes fail significantly for high amplitude values, what is likely to be perceptually relevant. WBVPM performance seems to degrade for harmonics found in valleys of the spectral envelope, such as the 2^{nd} harmonic, but comparably to narrow-band estimations. Thus, we can say that WBVPM performs better than narrow-band analysis. In (c) we see the time-domain waveforms corresponding to the original (audio [104]), resynthesized and residual signals (audios [105] and [106] for narrow-band and WBVPM respectively). The sampling rate is 44.1Khz. Finally, (d) presents the energy evolution along time of the original and residual signals, together with the fundamental frequency function. In the WBVPM analysis we have used the already known f_0 value instead of an estimation. In the narrow-band analysis, we have applied a Hanning window with a length adapted to cover 3 periods, and a constant analysis hop-size of 256 samples. The energy of the WBVPM residual is in average 11.14dB lower than the narrow-band one. Both techniques perform similarly only in f_0 peaks and valleys, mainly because the signal is locally more stationary at those segments.

  In the following Figure 2.88 we compare the residual energy obtained with WBVPM and standard sinusoidal analysis for different synthetic signals strongly modulated in amplitude and frequency whose mean fundamental frequencies vary between 50 and 850 Hz. For those signals, the fundamental frequency is modulated with a depth of a 30% of the f_0 value, whereas the amplitude is modulated with a depth of 15dB. Modulation rates are set to a quarter of the fundamental frequency. In the standard sinusoidal analyses, harmonics are estimated using quadratic interpolation of the spectral envelope, and the window length L is adapted to cover different number of signal periods, from 1 to 4. Note that strictly speaking, one and two period window lengths should be considered wide-band conditions, while three and four period window lengths would correspond to narrow-band conditions. In the WBVPM analyses, the fundamental frequency is known a priori. The results show that WBVPM residual energy fluctuates around 12dB below the original signal energy, while the standard method gives best results between -6 and -2dB for $L=2T_0$. Other window lengths $L=\{T_0,3T_0,4T_0\}$ give worse results , ranging between -2.5 and 1.5 dB. Summarizing, in this experiment WBVPM clearly outperforms the standard method.

- VOICE SIGNALS

  It is necessary to consider real world voice signals to determine the value of the proposed WBVPM approach and its possible applications. For those signals the fundamental frequency is unknown contrary to what happened with the synthetic signals. Therefore, there will be errors in the estimated f_0 values that, as we have seen before, will degrade significantly the WBVPM performance. However, the question is whether the constantly varying characteristics of the human voice that relate to non-stationary sinusoids, will degrade even more the performance of the standard narrow-band analysis. We have considered three male and two female utterances, displayed in figures from Figure 2.89 to Figure 2.93. Those figures include the waveform of the original and resynthesized signals, the energy envelopes of the original and residual signals, and the estimated fundamental frequency function. All audio files have a sampling rate of 44.1Khz and are quantized at 16 bits. We have considered two different narrow-band configurations, the former using a Blackman-Harris 92dB window of 2049 samples, and the latter using the same window type with a length adapted to cover a fixed number of fundamental periods. In narrow-band analysis, sinusoidal parameters have

been estimated at a frame rate of approximately 200 frame per second (hop size of 221 samples). In the case of wide-band analysis, the frame rate is the same as the fundamental frequency. When synthesizing sinusoids with time-domain oscillators, the actual sinusoid parameters of each sample have been obtained by linear interpolation of analysis estimations.

Figure 2.89 corresponds to a female singing expressively, with scoop and vibrato. WBVPM clearly outperforms narrow-band analyses, obtaining significantly lower residual energies. Figure 2.91 also corresponds to an expressive female voice, and shows the good performance of the WBVPM analysis. We observe that WBVPM residual energy gets more than 15 dB below the narrow-band one (fixed window) in attacks, note transitions and vibratos, actually where we expect the non-stationary characteristic of the harmonics to degrade significantly the performance of the narrow-band analysis. In turn, WBPVM achieves more than 6 dB residual energy decrease compared to the narrow-band analysis using a five-period window length. Moreover, in segments with stable energy and fundamental frequency such around second one, we observe a similar performance improvement. The second view starting from the bottom shows the sum of absolute MFCC difference values between consecutive frames, giving a measure of timbre instability (high values correspond to timbre changes). From this figure, we can state that WBVPM and narrow-band performances seem to be somehow correlated to the timbre instability descriptor, showing often residual energy peaks around timbre instability peaks. Figure 2.90 corresponds to an especially low pitch male speech utterance, with fundamental frequency ranging between 63 and 75 Hz approximately. In general, WBVPM performs better than narrow-band analyses, obtaining lower residual energies. The top view shows the waveforms of the original and reconstructed signals, together with the analysis frame times. Comparing those waveforms, we notice that the resynthesis obtained from narrow-band analysis gets smeared (e.g. around 0.35 seconds), although estimated harmonic parameters are not transformed and it uses a high frame rate (200 frames/sec). By contrast, the WBVPM reconstructed waveform looks sharper and more similar to the original one. Figure 2.92 corresponds to a male singing three long notes with fast vowel transitions, covering approximately one octave. We observe that WBVPM performs better than narrow-band analyses in glissandos and stable segments, giving about 10 and 5 dB lower residual energies with respect to fixed and adapted window lengths. However, all three methods perform similarly during fast phonetic transitions with stable fundamental frequency, especially at lower pitches.

Overall results are shown in Table 2.5. For female voices, WBVPM residual energy is about 12 dB lower than for the fixed window narrow-band analysis and 6 dB lower than the adapted window case. However, for male voices, in the first example (very low pitch speech) WBVPM is about 2 dB below both narrow-band cases, in the second example about 0.6 dB below, and in the third one (tenor voice with f_0 above 200Hz) about 6.5 and 15 dB below adapted and fixed window cases respectively. In addition to possible fundamental frequency detection errors, one possible explanation for this behavior is that for high fundamental frequencies there are less harmonics, and most of the energy is located at the first harmonics. Since, as we saw before, for higher harmonic indices the inter-harmonic contribution gets higher and the harmonic estimation accuracy gets worse, in general we should expect lower relative residual energies for higher pitches than for low pitches, thus better performance for female voices than for male voices. Another plausible explanation is that for low pitch utterances fundamental periods are long, and we cannot assume that the articulatory system is locally static along each period. This causes harmonic spectral shape to change significantly during a period, and therefore harmonics cannot be considered stationary along a single period. In that sense, WBVPM performance degrades due to the non-stationary characteristics of the voice signal, similarly to what often occurs in narrow-band analyses. However, according to these explanations, we should expect a better WBVPM performance compared to narrow-band analysis in the last note sung in Figure 2.92, the one with stable mean pitch around 180 Hz. The particularity of that example is that it contains very fast phonetic transitions between \a\ and \i\ Spanish vowels, and therefore although the pitch is not that low, during a voice period the timbre characteristics might be changing considerably. That might explain why in such note WBVPM residual energy gets about 4 dB below the

Figure	signal description	rough f_0 range (Hz)	signal energy (dB)	residual energy (dB)		
				narrow-band		*WBVPM*
				fixed window	*adapted window size*	
Figure 2.89	female voice singing the word 'yacht' with scoop and vibrato	260-420	-16.21	-29.85	-36.39	-42.30
Figure 2.90	male speech, deep voice, low pitch	63-75	-17.74	-31.65	-32.39	-34.46
Figure 2.91	female voice singing a jazz tune, deep voice, expressive with vibrato	230-380	-18.86	-35.89	-44.73	-51.03
Figure 2.92	male voice, flat singing, one octave arpeggio, \a\ and \i\ Spanish vowels with fast transitions	90-180	-21.73	-45.00	-45.02	-45.63
Figure 2.93	male tenor singer, strong vibrato	210-332	-22.00	-39.84	-48.67	-55.16

Table 2.5 Analysis results of voice signals using WBVPM and standard narrow-band analysis techniques

narrow-band (adapted window length) one for low values of the timbre variation descriptor (i.e. MFCC variation), and about 4 dB above for high values of such descriptor. However, there is still another possible reason for the observed performance behavior that we detail in the following section. Briefly, we will show that the voice signal becomes locally inharmonic when the vocal tract characteristics move fast and in the case of deep and fast vibratos. This affects especially WBVPM performance since such method relies on the harmonic structure of the input signal to minimize the analysis windowing effects.

We have proposed three reasons to explain the observations, concretely the similar performance of narrow-band and WBVPM analysis methods in certain contexts. However, those three factors will not affect always in the same way the WBVPM performance in terms of residual energy, but their effect will depend on the signal content, specifically on the fundamental frequency and timbre characteristics. Just to give an example, if the energy concentrates mostly in the harmonics below the first formant, rapid movements of the first and above formants will not increase that much the residual energy.

COMPARISON TO FREQUENCY DOMAIN DEMODULATION

(Röbel 2007) proposes a frequency domain demodulation method for estimating sinusoidal parameters in the case of linearly modulated sinusoids, and shows how it greatly reduces the energy of the residual signal compared to the quadratic interpolation (QIFFT) method. Concretely, it analyzes and computes the residual of the audio signal in Figure 2.93. Compared to the standard QIFFT method, the residual energy obtained with frequency domain demodulation decreases between 4.19 and 5.04 dB, depending on different algorithm refinements. A window of 800 samples was used in that experiment, which would be equivalent to our narrow-band analysis using an adapted window of four periods. Using WBVPM we obtained 6.49 dB of residual energy reduction compared to the standard QIFFT, so an improvement of about 1.5dB over the demodulation method. This seems to indicate that WBVPM performs better in this aspect, but this result cannot be generalized since only one sample was used in the comparison.

INHARMONICITY IN VOICE SIGNALS

In a simplified model of the harmonic phase envelope, when the analysis window is centered close to a glottal pulse onset, the voice source has an approximately flat phase envelope and each formant of the vocal tract adds a phase shift around its center frequency. This can be observed in Figure 2.18, which shows the spectrum computed out of a recorded performance. In addition, an ideal representation is shown in Figure 2.71. In the case a formant is shifting in frequency during a voiced utterance, the phase of close harmonics will be affected and therefore their instantaneous frequency modified as well. In order to demonstrate this, let us consider a voice signal $s(t)$ and its sinusoidal representation,

$$s(t) \approx \sum_{h=0}^{H-1} a_h(t)\cos\left(\phi_h(t)\right). \tag{2.93}$$

Let us also assume that the fundamental frequency is constant, and therefore $\phi_h(t) = 2\pi f_h t + \phi_{0,h}(t)$. Considering the phase shift produced by a given formant to be linear, and the formant center frequency to be a function of time $f_F(t)$, the resulting harmonic phase function will be

$$\phi_h(t) = \begin{cases} 2\pi f_h t + C & \text{if} \quad f_h \le f_F(t) - W/2 \\ 2\pi f_h t + C - \dfrac{A}{2}\left(1 + \dfrac{2}{W}\left(f_h - f_F(t)\right)\right) & \text{if} \quad \left|f_h - f_F(t)\right| < W/2 \\ 2\pi f_h t + C - A & \text{if} \quad f_h \ge f_F(t) + W/2 \end{cases} \tag{2.94}$$

where A corresponds to the phase shift excursion, W to the bandwidth of the formant's area of influence, and C is a constant. If a given harmonic h is found within the area of influence of that formant, its instantaneous frequency $\bar{f}_h(t)$ will be given by

$$\bar{f}_h(t) = \frac{1}{2\pi}\frac{d\phi_h(t)}{dt} = f_h + \frac{A}{2\pi W}\frac{df_F(t)}{dt}. \tag{2.95}$$

Therefore, the instantaneous frequency depends on the derivative of the formant center frequency, i.e. its moving speed. Considering a constant velocity of v Hz per second, we obtain

$$\begin{aligned} f_F(t) &= D + vt \\ \bar{f}_h(t) &= f_h + \frac{A}{2\pi W}v \end{aligned} \tag{2.96}$$

where D is a constant. This means the hypothesis that the voice is harmonic is not true anymore; it is in fact inharmonic. The deviation d_h in cents from the ideal harmonic frequency is given by

$$d_h = 1200\cdot\log_2\left(\frac{\bar{f}_h}{f_h}\right) = 1200\cdot\log_2\left(1 + \frac{A}{2\pi W f_h}v\right). \tag{2.97}$$

Let us make an estimation of what might be this deviation for the audio example in Figure 2.92. The MFCC variation descriptor is computed as the sum of absolute MFCC differences between consecutive frames. This gives a measure of how much the spectral envelope is changing. We expect to see higher values when formants are moving; therefore, peaks are likely to correspond to phonetic transitions. The utterance in this example consists of three repetitions at different pitches of a sustained Spanish \a\ vowel followed by the fast sequence \i\-\a\-\i\-\a\-\i\-\a\-\i\-\a\. Observing the MFCC variation descriptor, we could say that singer takes approximately takes from 4.125 to 4.25 seconds (i.e. 125ms) to change from vowel \i\ to vowel \a\. Assuming values for the second formant F2 around 2.2Khz for \i\ and 1.2Khz for \a\, we can estimate that the second formant F2 approximately takes 125ms to shift 1 Khz in frequency, thus at a mean speed around 8000 Hz/sec. Considering standard values of A=PI rad and W=250Hz, a certain harmonic h located at 1300 Hz (i.e. $f_h = 1300$) would have a frequency deviation of $d_h \simeq 21$ cents, which is quite significant. This means that a considerable bias is expected in the harmonic estimation during fast phonetic transitions, and gives a reasonable explanation for the results shown in Figure 2.92, where WBVPM performs similarly to standard STFT or even worse.

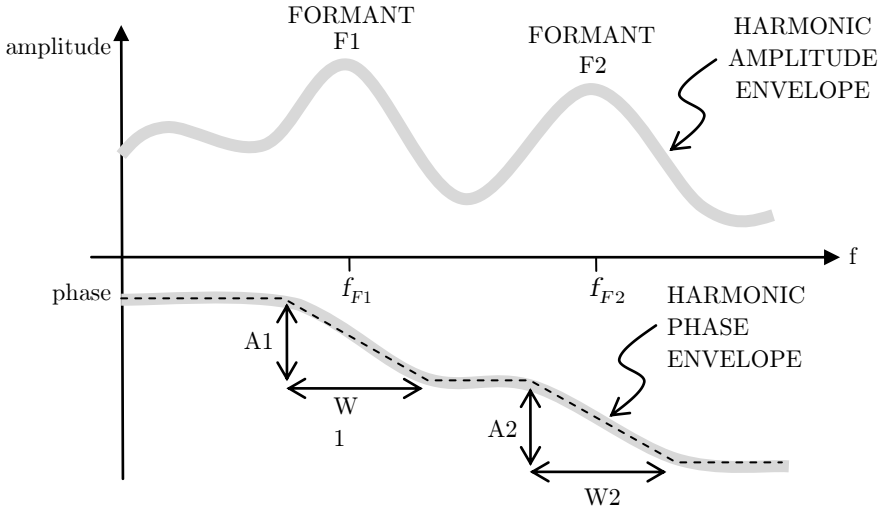


Figure 2.71 Simplified model of the harmonic phase envelope at voice pulse onsets

We could make similar arguments in the case of a sustained vowel with vibrato. The difference is that instead of the formants the harmonics are the ones that shift in frequency. Let's consider a vibrato where a fundamental frequency of C Hz is modulated in frequency with a constant rate of R Hz and a relative deep of M, i.e.

$$f_0(t) = C\left(1 + M\cos\left(2\pi R t\right)\right). \tag{2.98}$$

As we did before, let us compute the instantaneous frequency $\bar{f}_h(t)$ of an arbitrary harmonic h that moves around a formant with bandwidth W Hz and phase shift excursion A radians,

$$\bar{f}_h(t) = \frac{1}{2\pi}\frac{d\phi_h(t)}{dt} = f_h(t) - \frac{A}{2\pi W}\frac{df_h(t)}{dt} = hC\left(1 + M\cos\left(2\pi R t\right)\right) + \frac{AhCMR}{W}\sin\left(2\pi R t\right). \tag{2.99}$$

The deviation in cents from the ideal harmonic frequency is given by

$$d_h = 1200\cdot\log_2\left(\frac{\bar{f}_h}{f_h}\right) = 1200\cdot\log_2\left(1 + \frac{AMR\sin\left(2\pi R t\right)}{W\left(1 + M\cos\left(2\pi R t\right)\right)}\right). \tag{2.100}$$

The previous equation tells us that the deviation is independent of the fundamental frequency and the harmonic index. This means that the deviation will be the same for all harmonics affected by each formant. However, this conclusion would be only true for the simplified model of the formant phase function, which considers a linear phase shift. In reality, the phase shift is not linear, and therefore the different harmonics moving around the formant are affected by different deviation amounts. Figure 2.72 shows the maximum absolute deviation biases for different values of vibrato rate and normalized depths. As one might expect, the higher the vibrato deep or rate, the higher the deviation. In some cases the deviations can be quite significant, up to 50 cents, i.e. half a semitone. Let us make an estimation of what might be this deviation for the audio example in Figure 2.91. Observing the vibrato in the bottom view, we can roughly estimate its relative deep to be $M \approx 25/350 \approx 0.07$ and the rate to be $R \approx 1/0.2 = 5$ Hz. Using Eq (2.100) we get the maximum absolute deviation to be $|d_h|^{\max} = 7.8$ cents. This might be a plausible explanation for the WBVPM residual energy evolution between 1.5 and 1.9 seconds, which has peaks at points where the fundamental frequency varies rapidly, i.e. it appears to be somehow correlated with the absolute variation of the fundamental frequency.

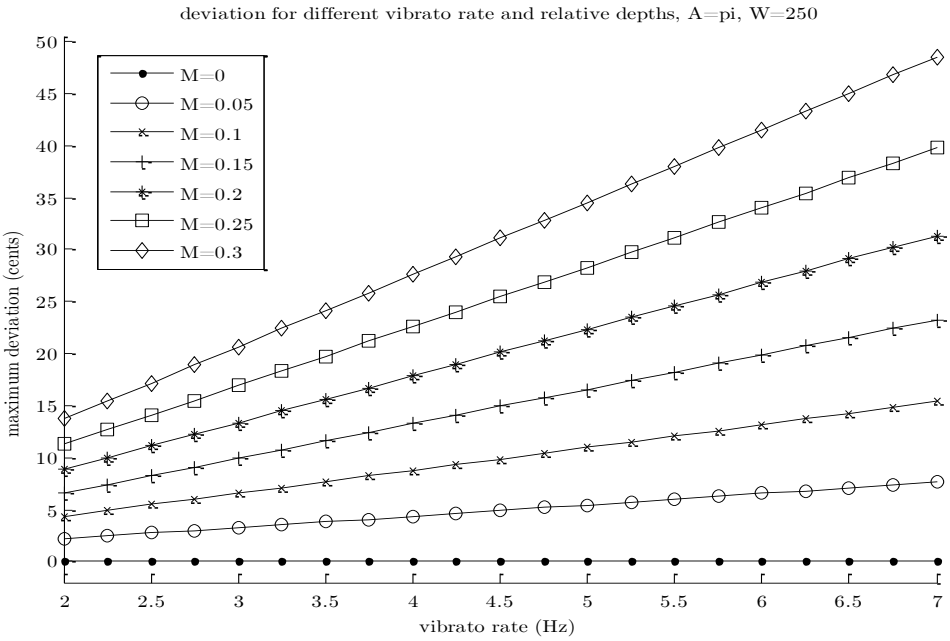
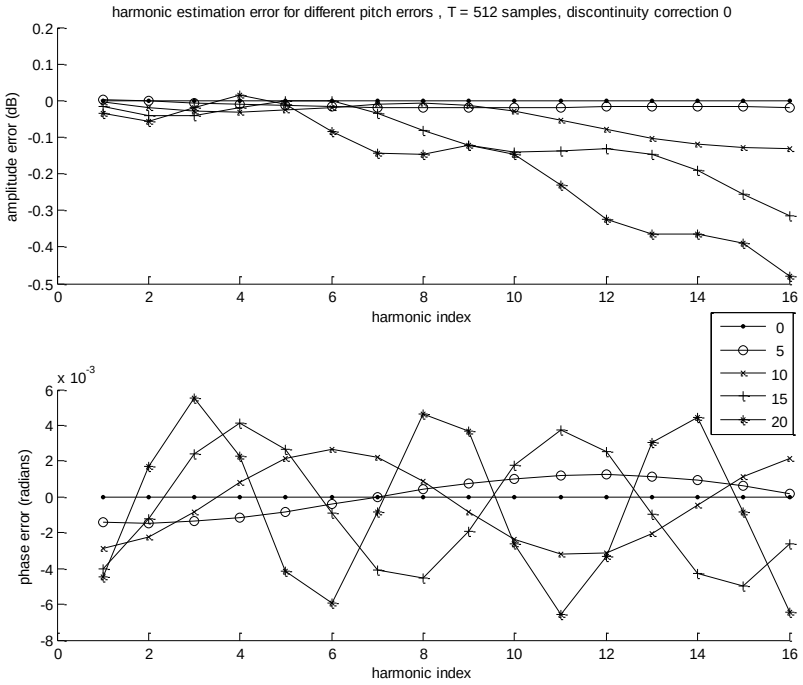
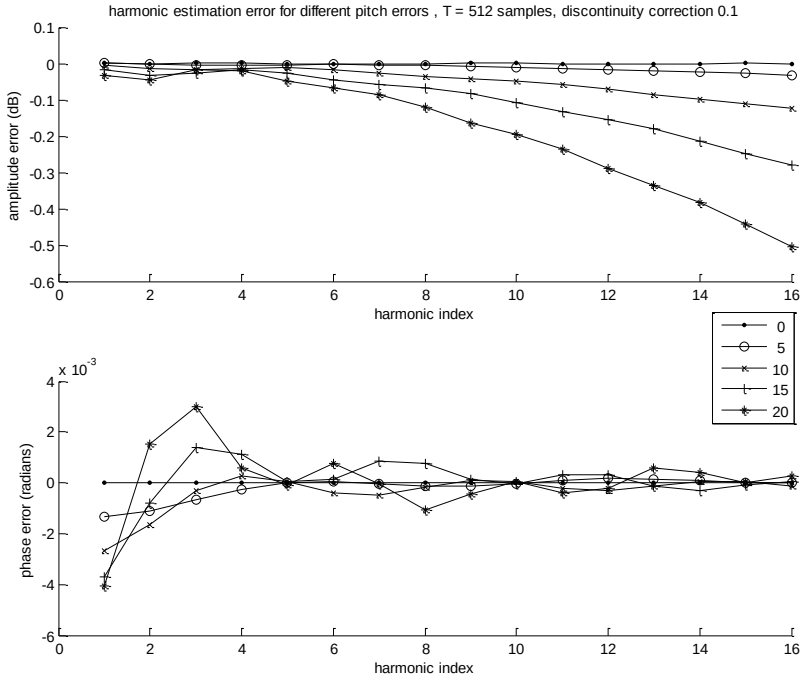


Figure 2.72 Maximum deviation in cents around a formant for different vibrato rate and relative depths

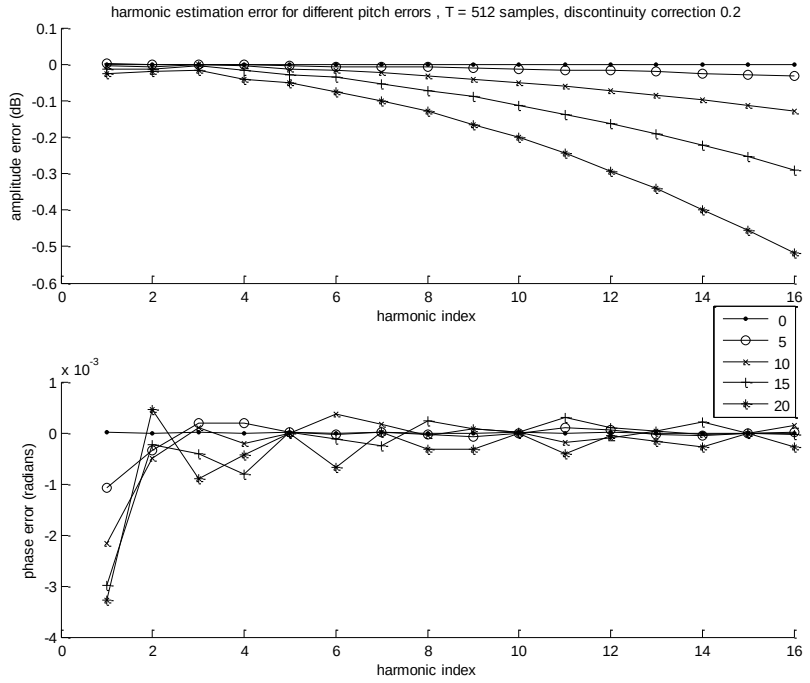


(a)

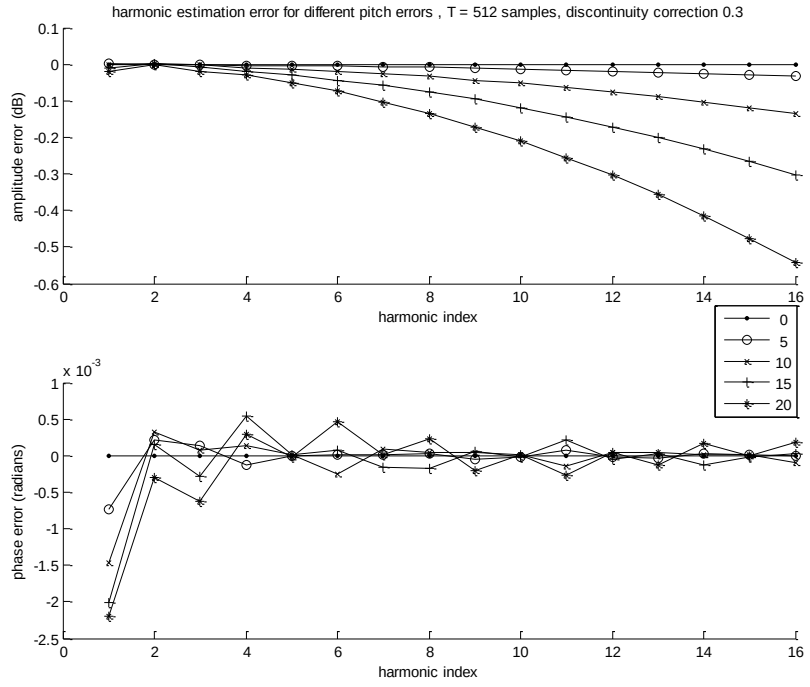


(b)

Figure 2.73 Harmonic amplitude and phase estimation errors for stationary sinusoids



(a)



(b)

Figure 2.74 Harmonic amplitude and phase estimation errors for stationary sinusoids

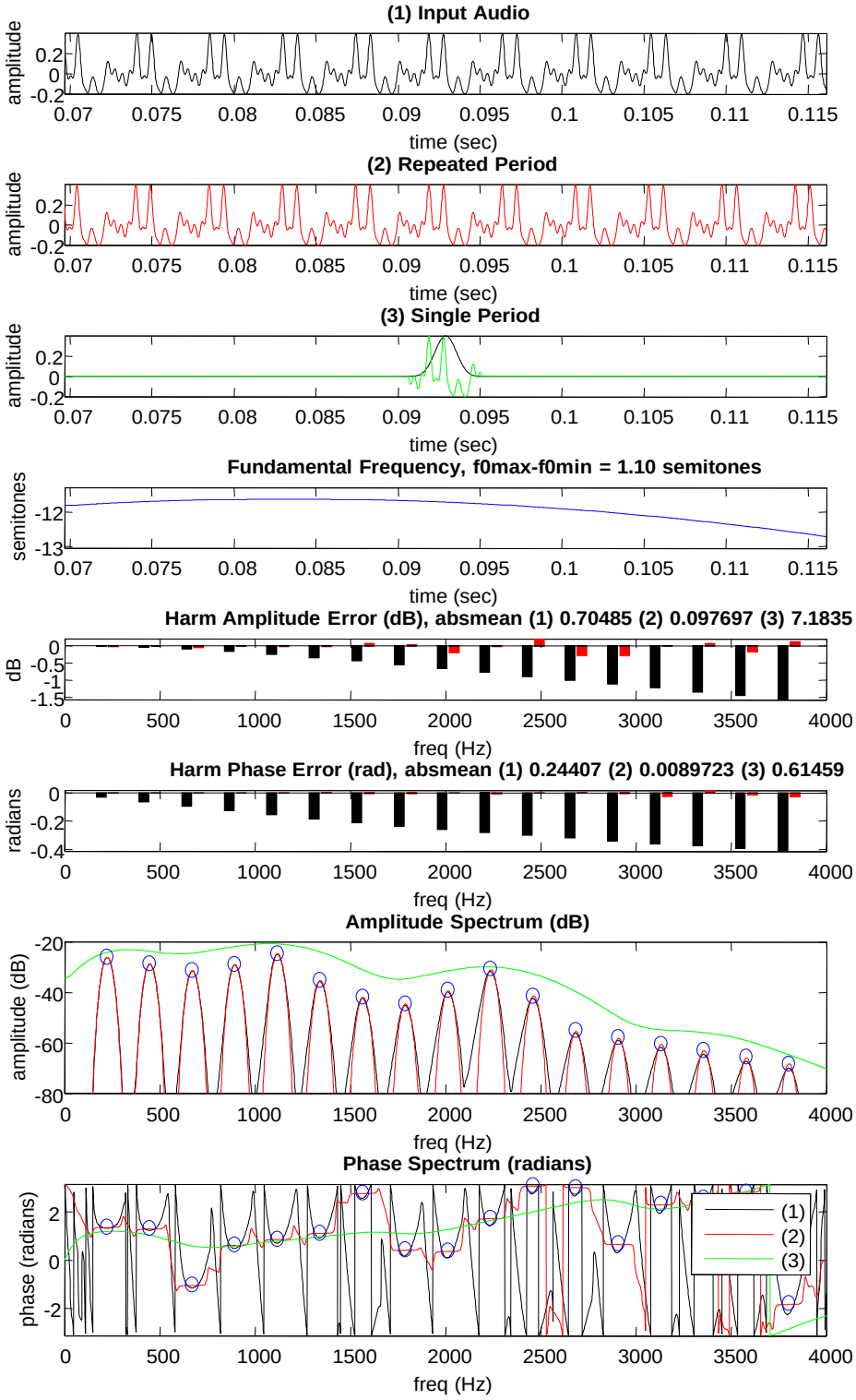


Figure 2.75 WBVPM, narrow and wide-band STFT analysis of a synthetic signal

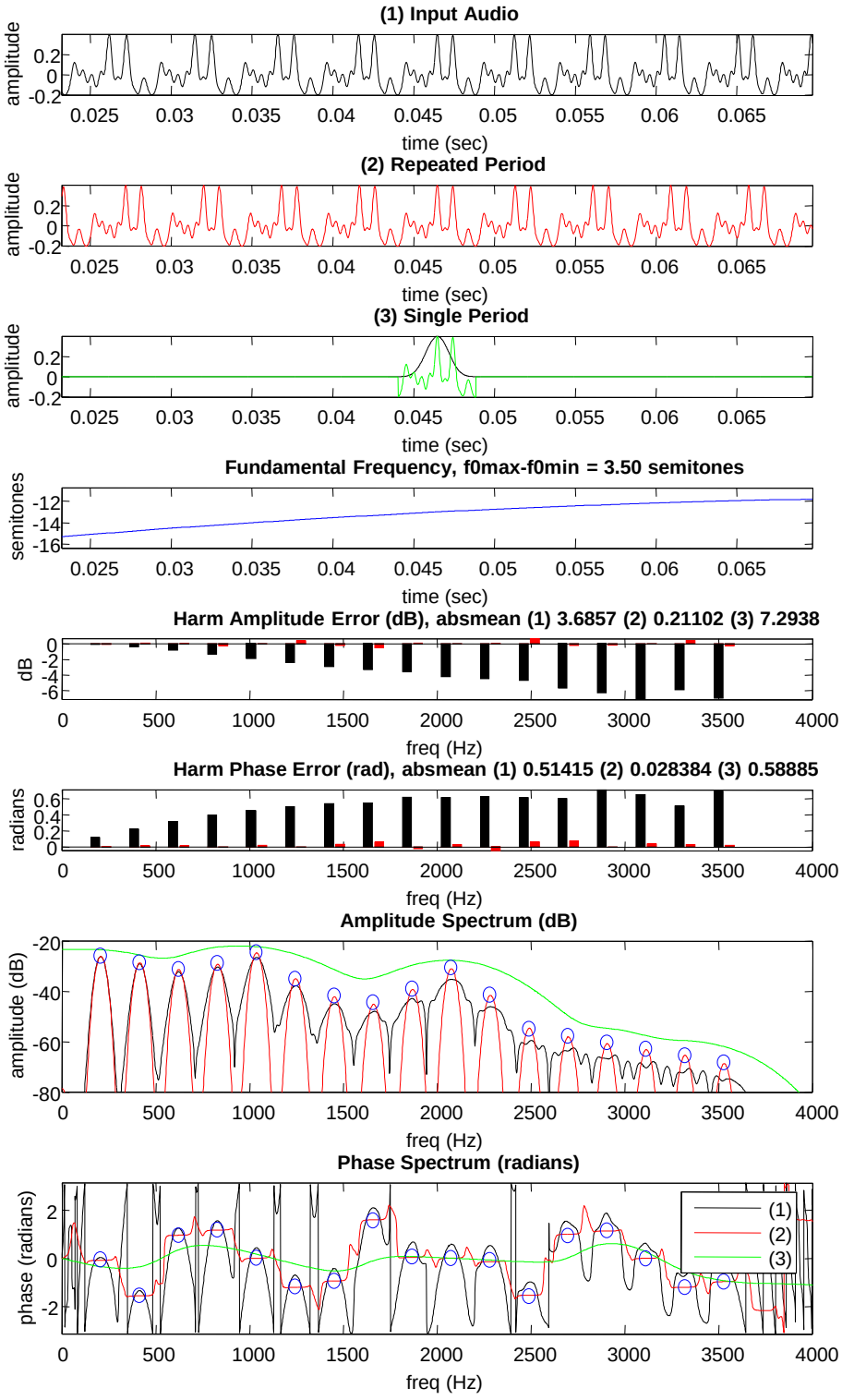


Figure 2.76 WBVPM, narrow and wide-band STFT analysis of a synthetic signal

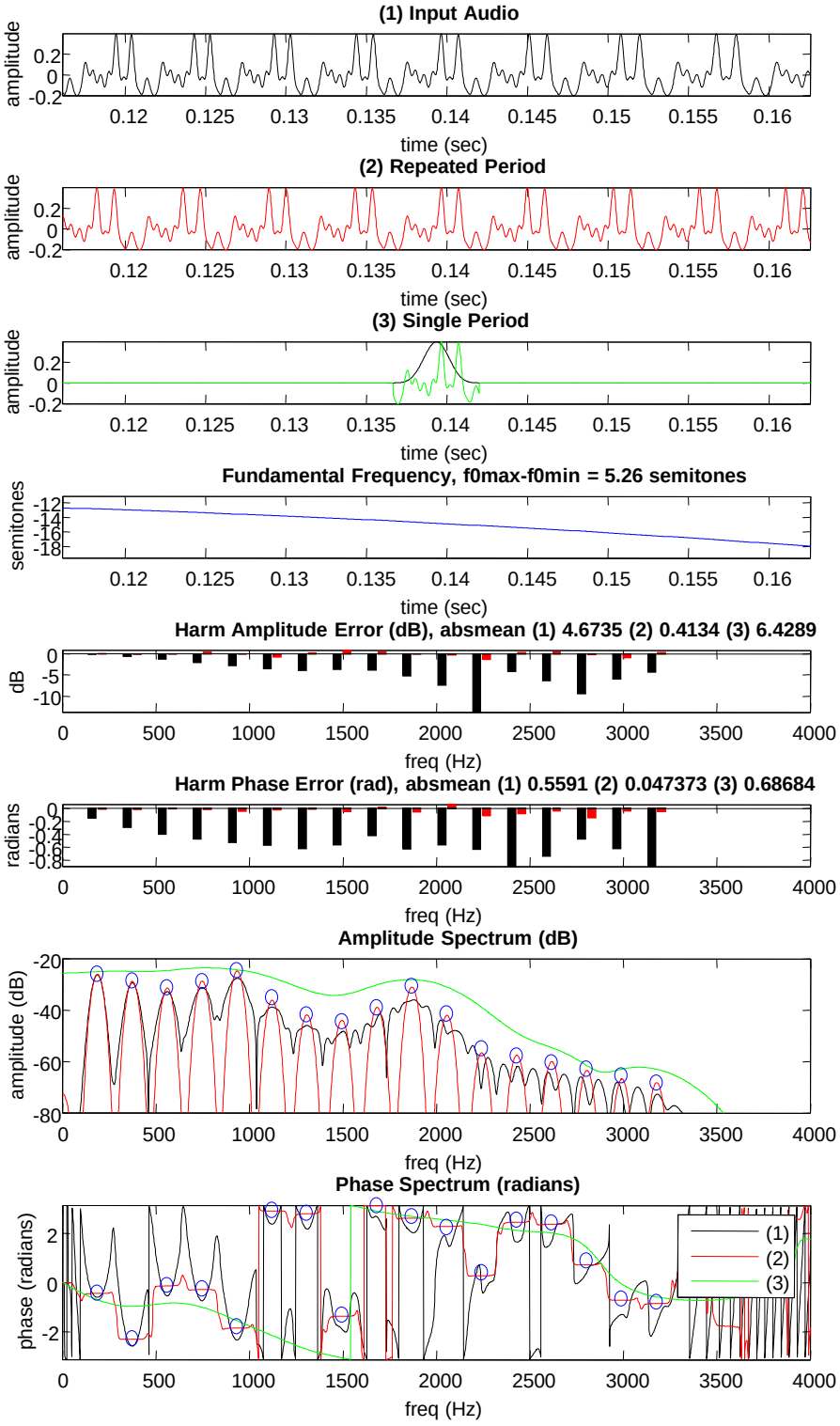


Figure 2.77 WBVPM, narrow and wide-band STFT analysis of a synthetic signal

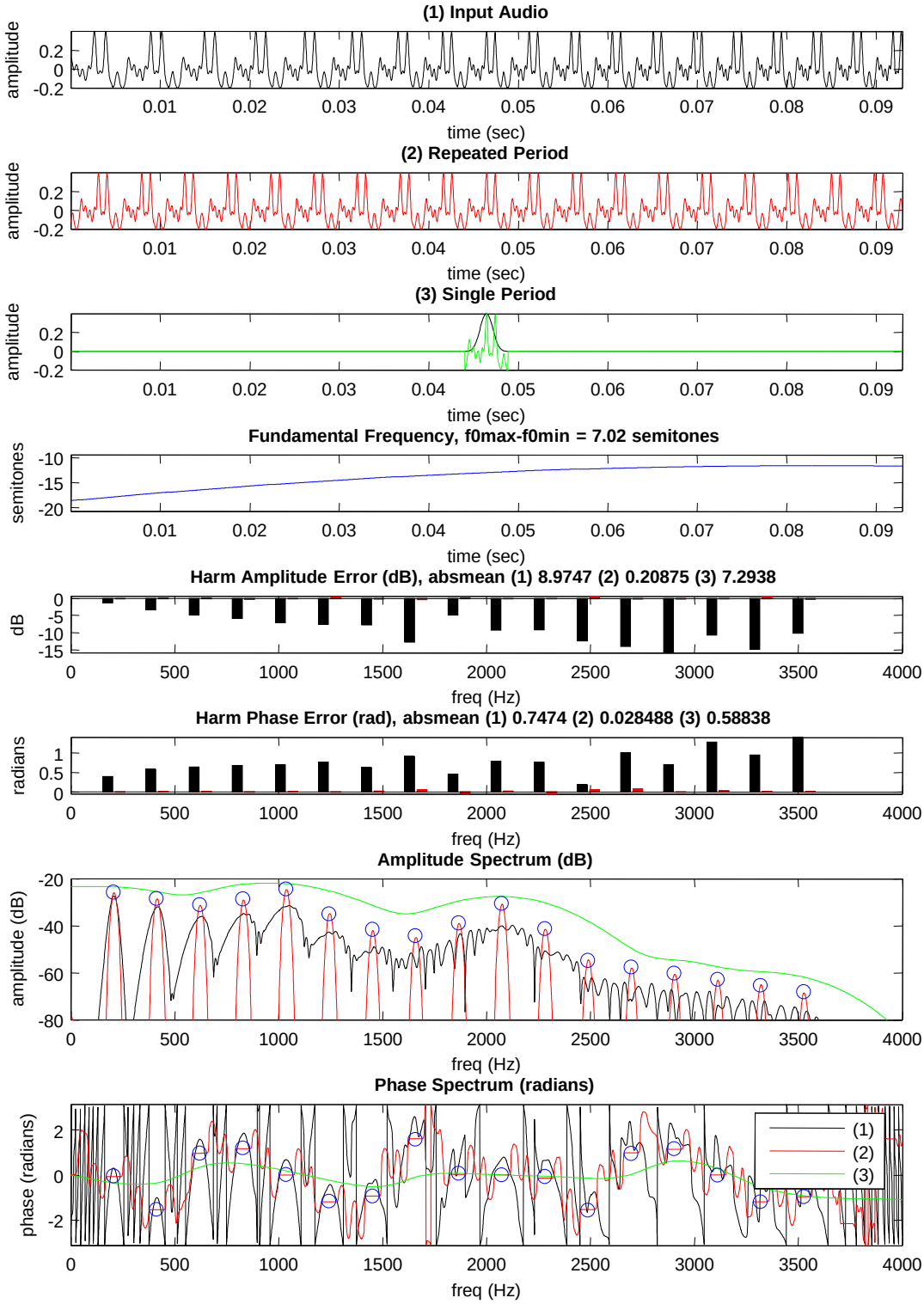


Figure 2.78 WBVPM, narrow and wide-band STFT analysis of a synthetic signal

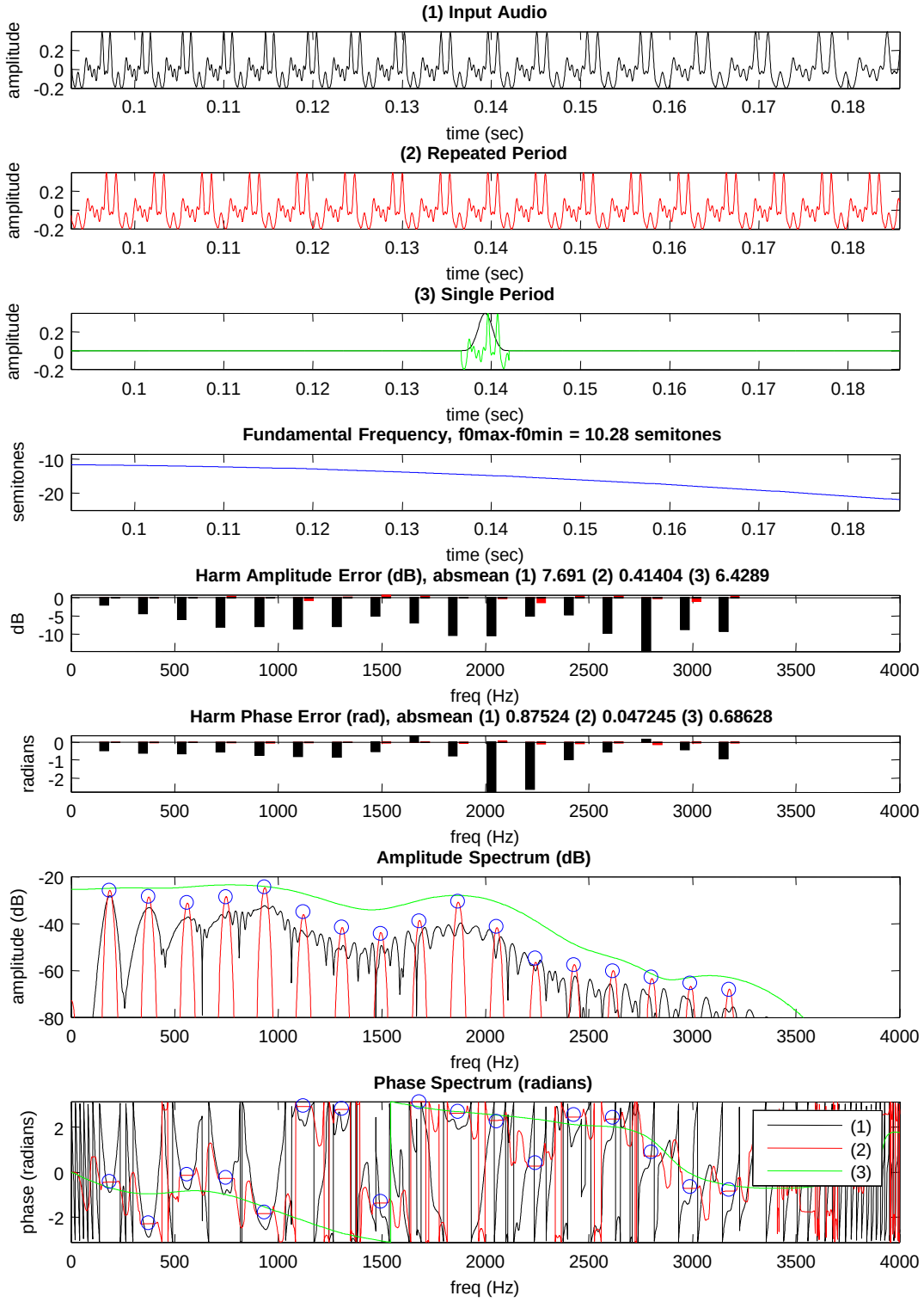


Figure 2.79 WBVPM, narrow and wide-band STFT analysis of a synthetic signal

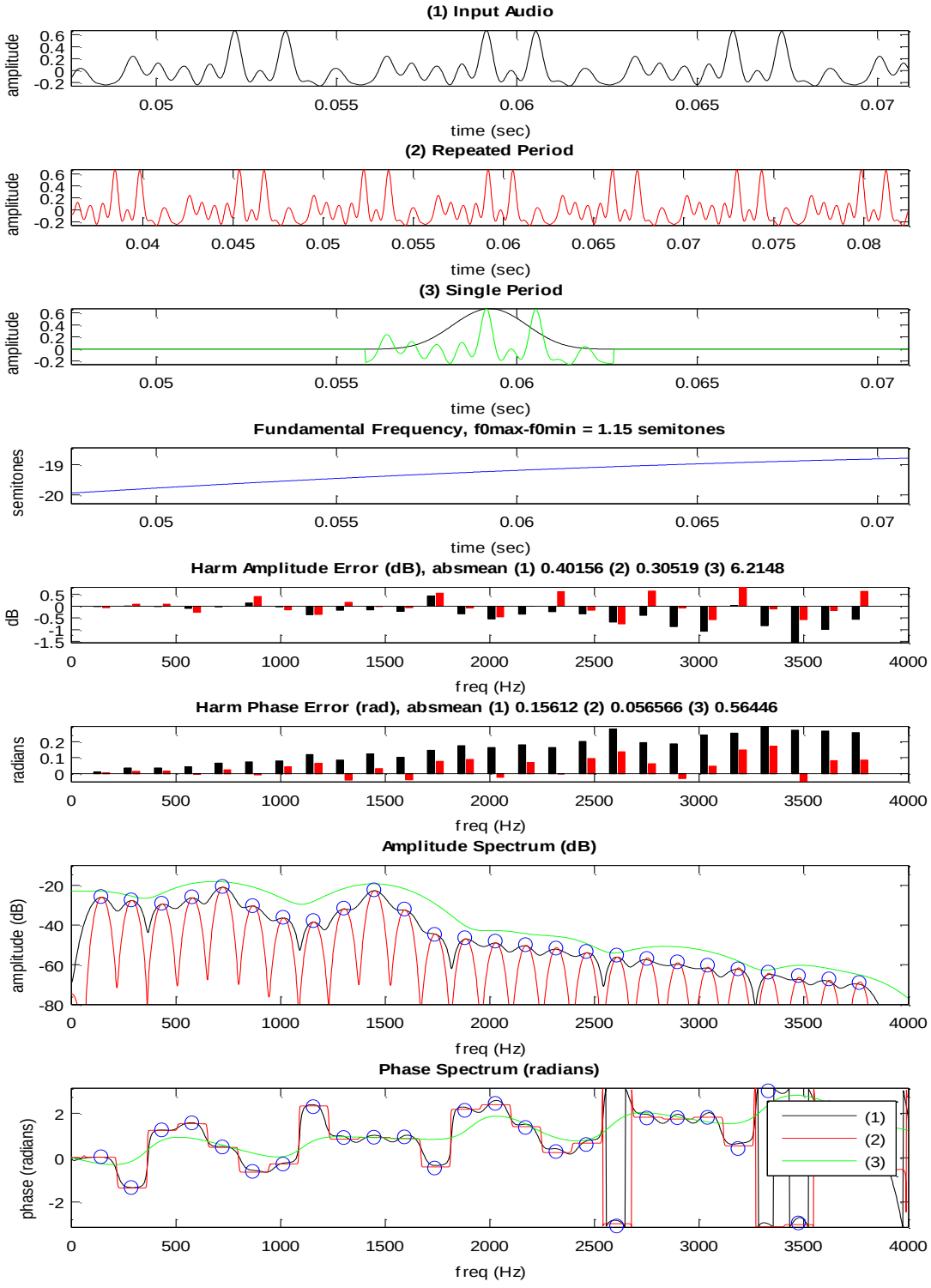


Figure 2.80 WBVPM, narrow and wide-band STFT analysis of a synthetic signal

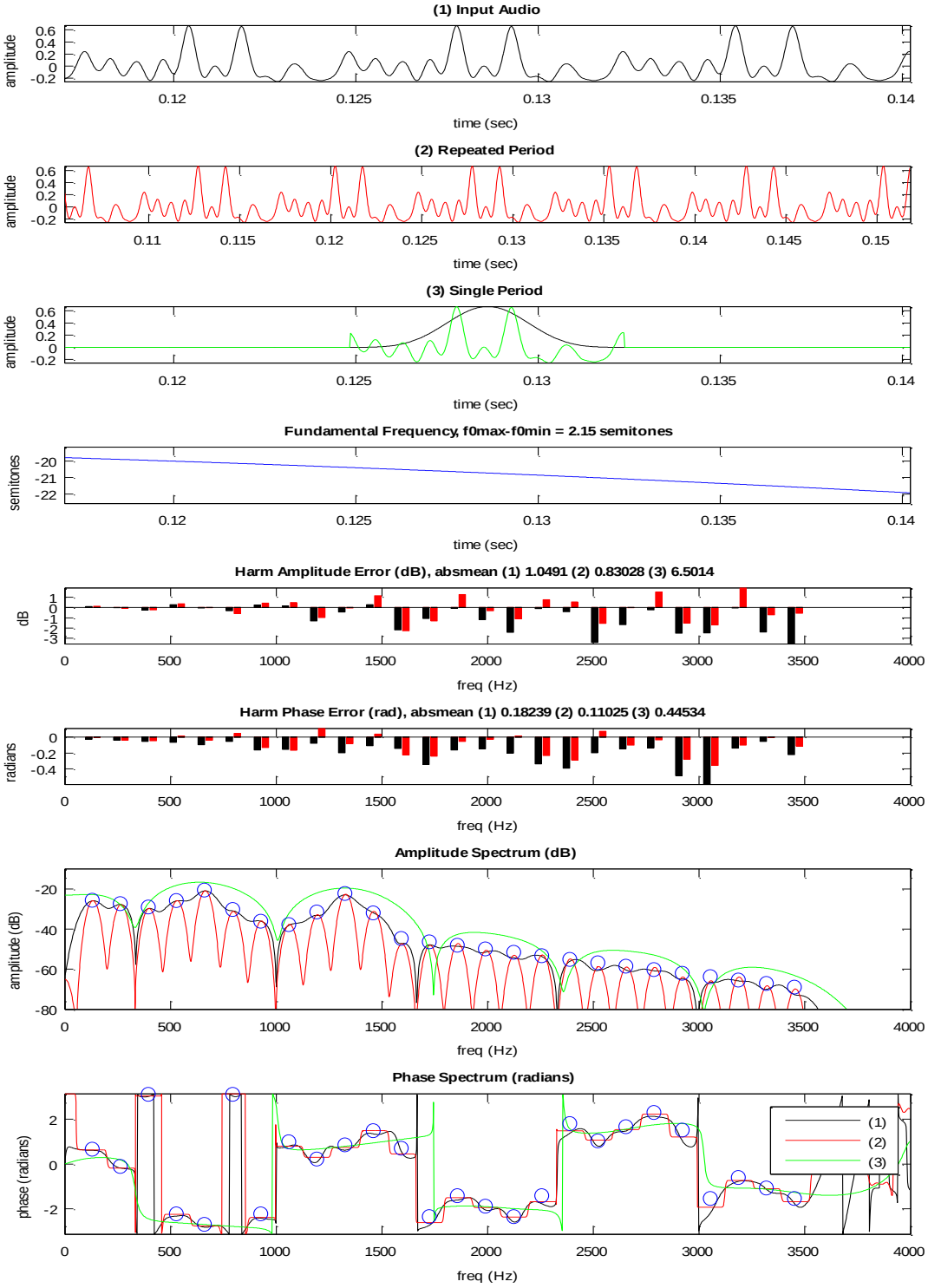


Figure 2.81 WBVPM, narrow and wide-band STFT analysis of a synthetic signal

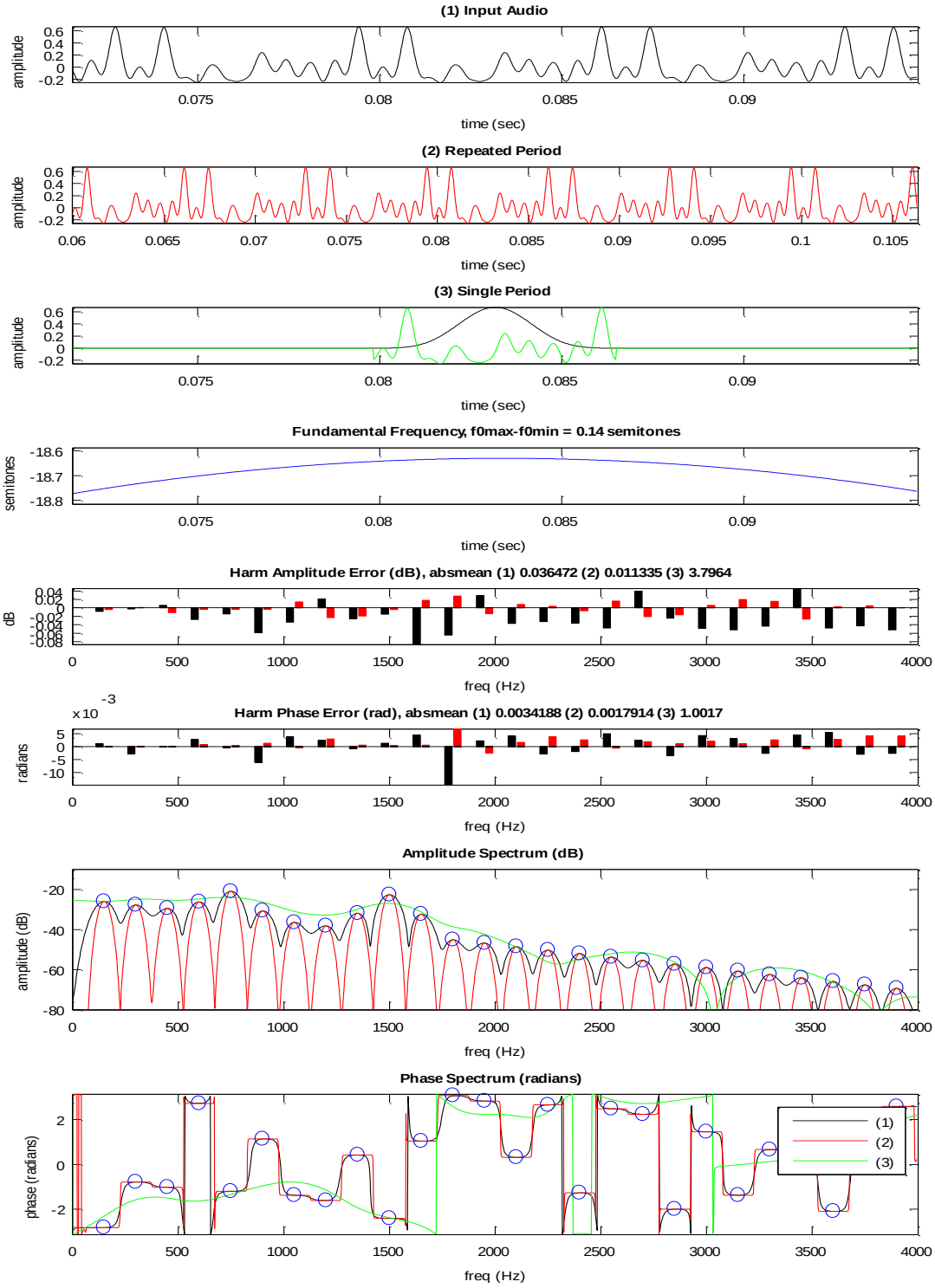


Figure 2.82 WBVPM, narrow and wide-band STFT analysis of a synthetic signal

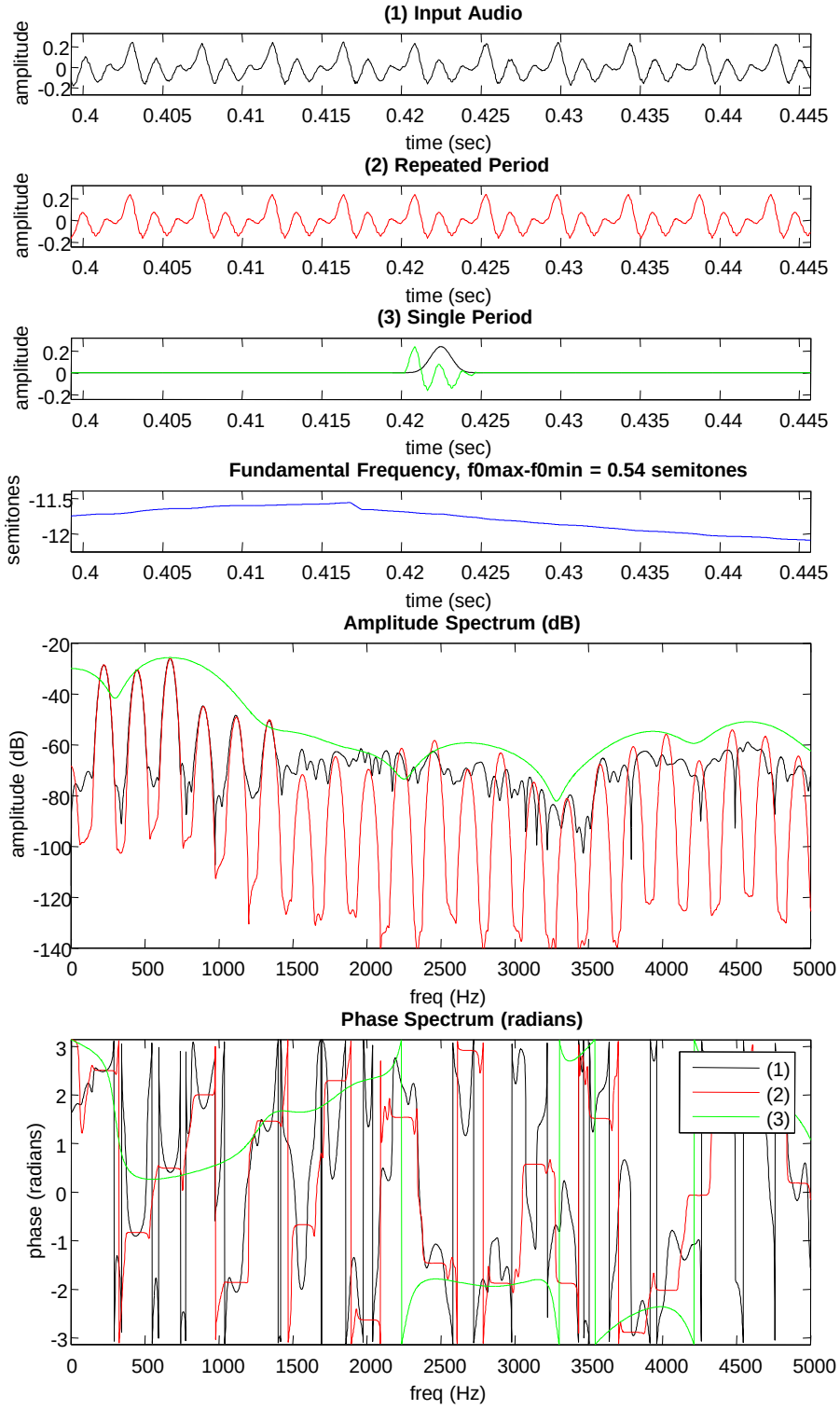


Figure 2.83 WBVPM, narrow and wide-band STFT analysis of a recorded singing male voice.

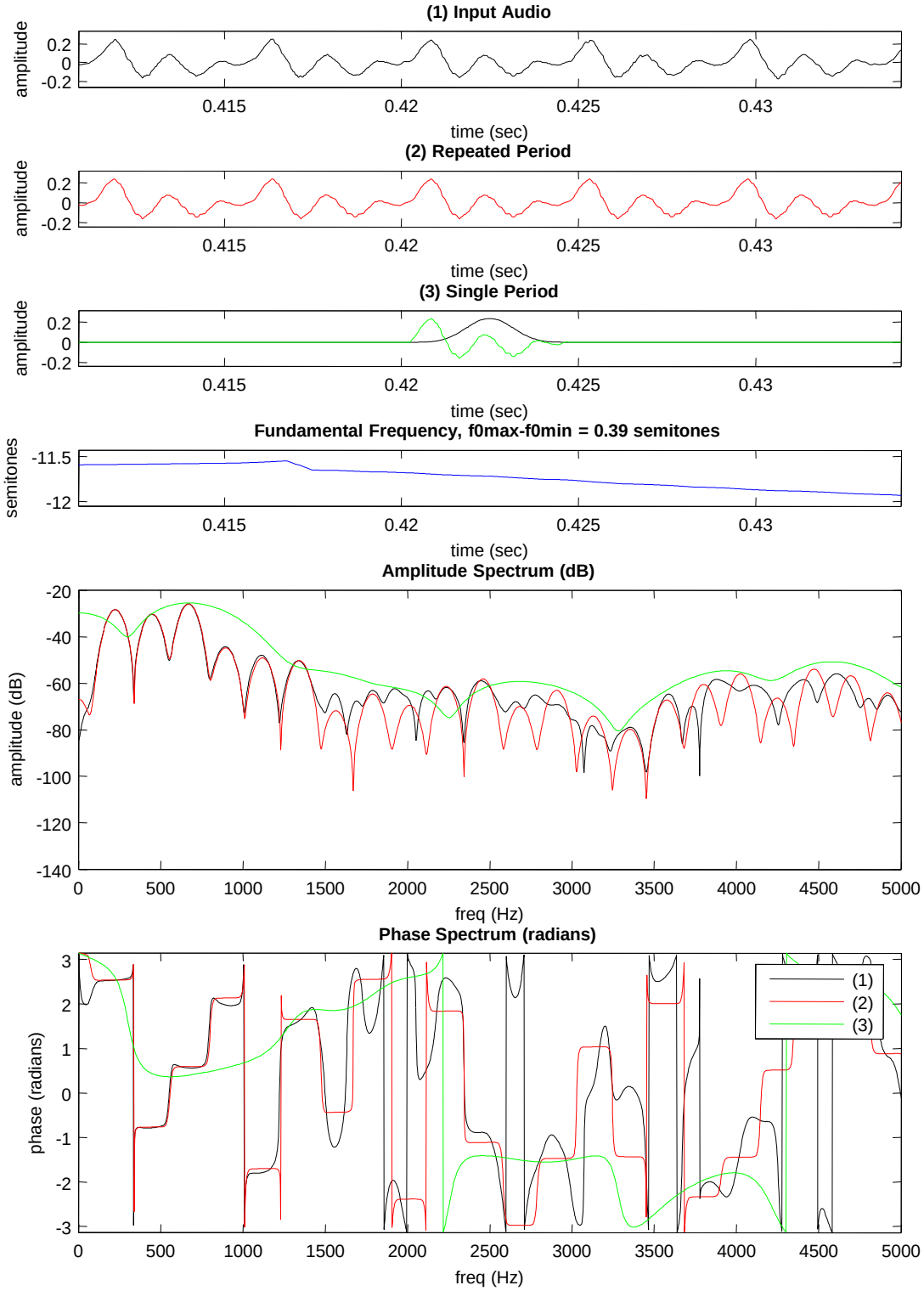


Figure 2.84 WBVPM, narrow and wide-band STFT analysis of a recorded singing male voice.

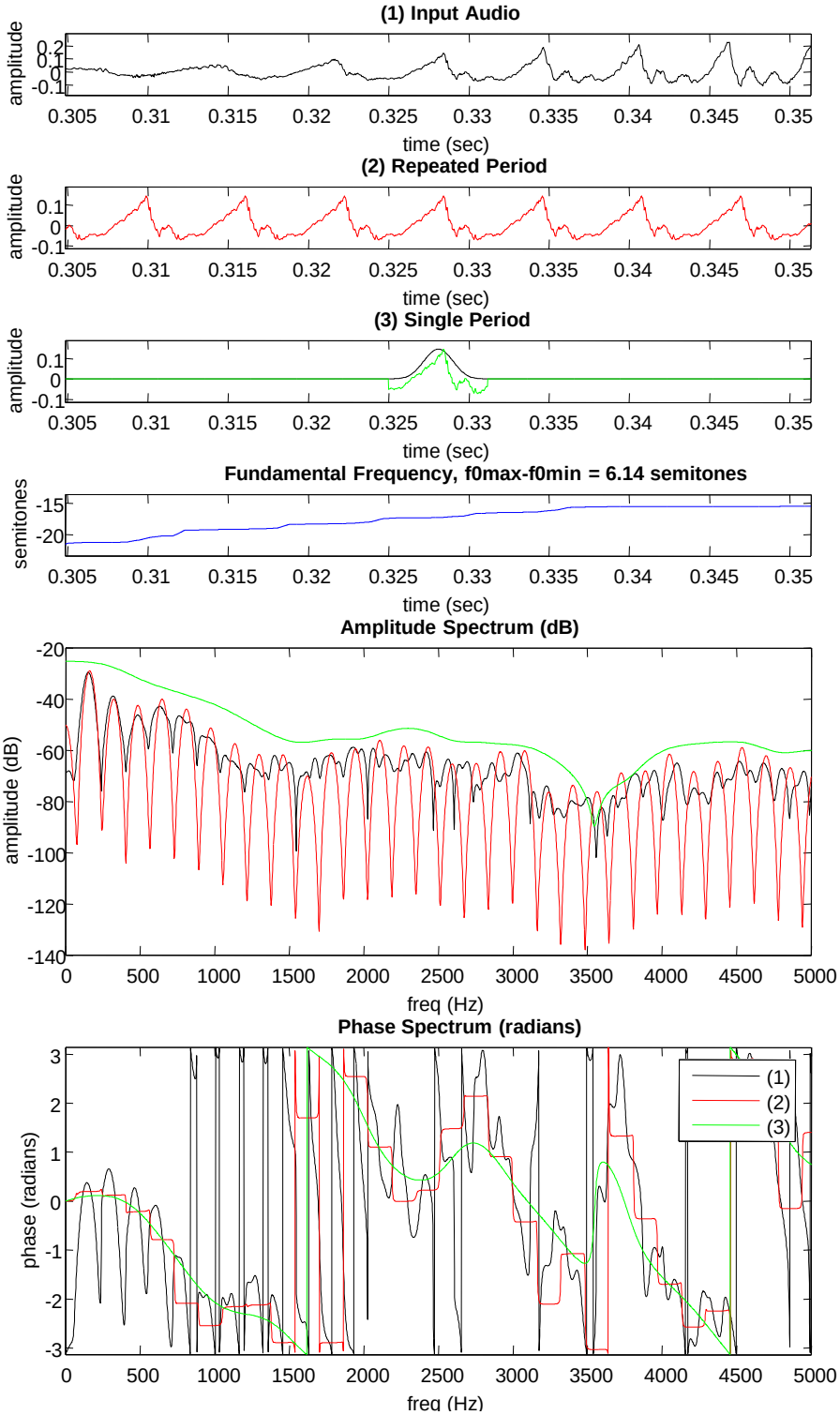


Figure 2.85 WBVPM, narrow and wide-band STFT analysis of a recorded singing male voice.

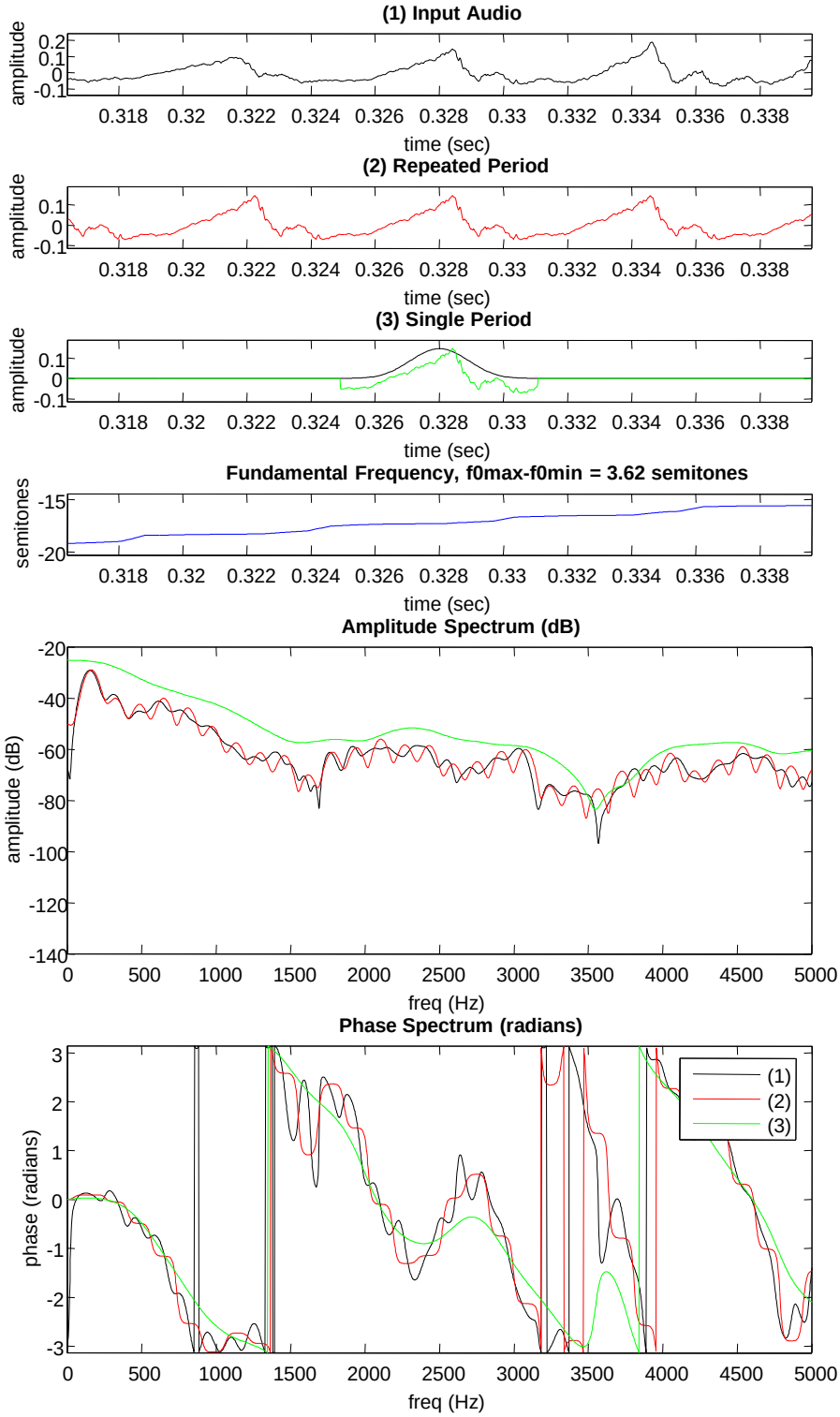


Figure 2.86 WBVPM, narrow and wide-band STFT analysis of a recorded singing male voice.

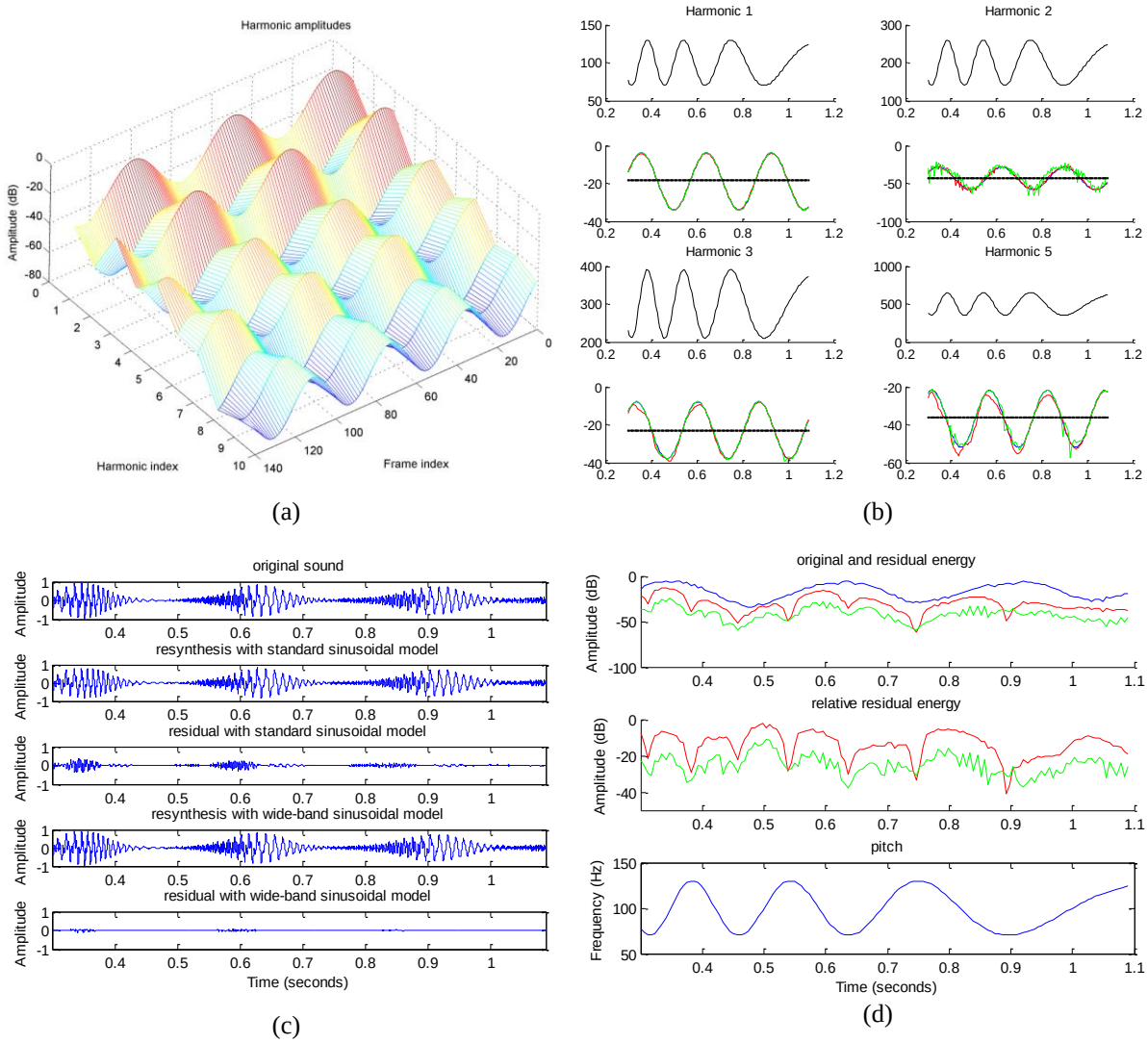


Figure 2.87 Standard narrow-band sinusoidal model versus wide-band voice pulse modeling (WBVPM). (a) shows the harmonic amplitude values of the synthetic signal (audio [104]) for all harmonics and frames. Clearly, harmonics are strongly modulated in amplitude. (b) shows in black color the frequency (in Hz) and amplitude (in dB) functions of the 1^{st}, 2^{nd}, 3^{rd} and 5^{th} harmonics. The estimated values with WBVPM and narrow-band analyses are drawn in green and red color respectively. (c) view shows the time-domain waveforms. (d) represents the energy of the input signal (in blue) and the analysis residuals, together with the fundamental frequency. In overall, WBVPM residual (audio [105]) is -11.1494dB below the standard sinusoidal model residual (audio [106]). Narrow-band analysis is performed with a Hanning window whose length is adapted to cover 3 periods. The original pitch is used in the WBVPM analysis.

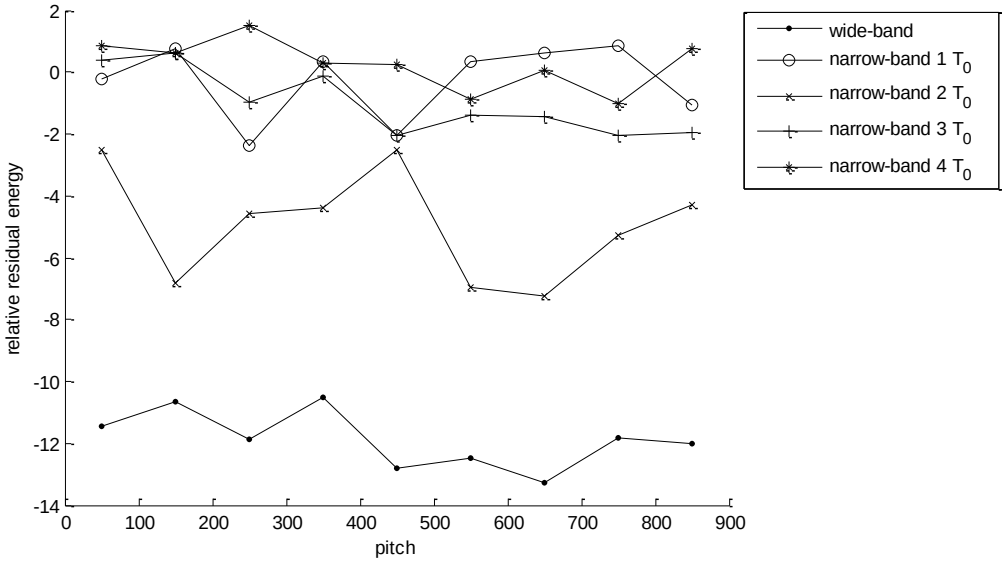


Figure 2.88 Comparison of the residual energy obtained with standard sinusoidal model and WBVPM for synthetic signals with mean pitch values between 50 and 850Hz. The analyzed signals consist of up to 10 harmonics with pitch modulated by 30% of the mean pitch value, and amplitude modulated by 15dB. The modulation frequencies are at a quarter of the mean pitch value. The sampling rate is 44.1KHz. In the standard sinusoidal analyses we have used a Hanning window with a size adapted to cover different number of periods, as specified in the legend. The pitch has not been estimated, but is the one used when generating the input signals.

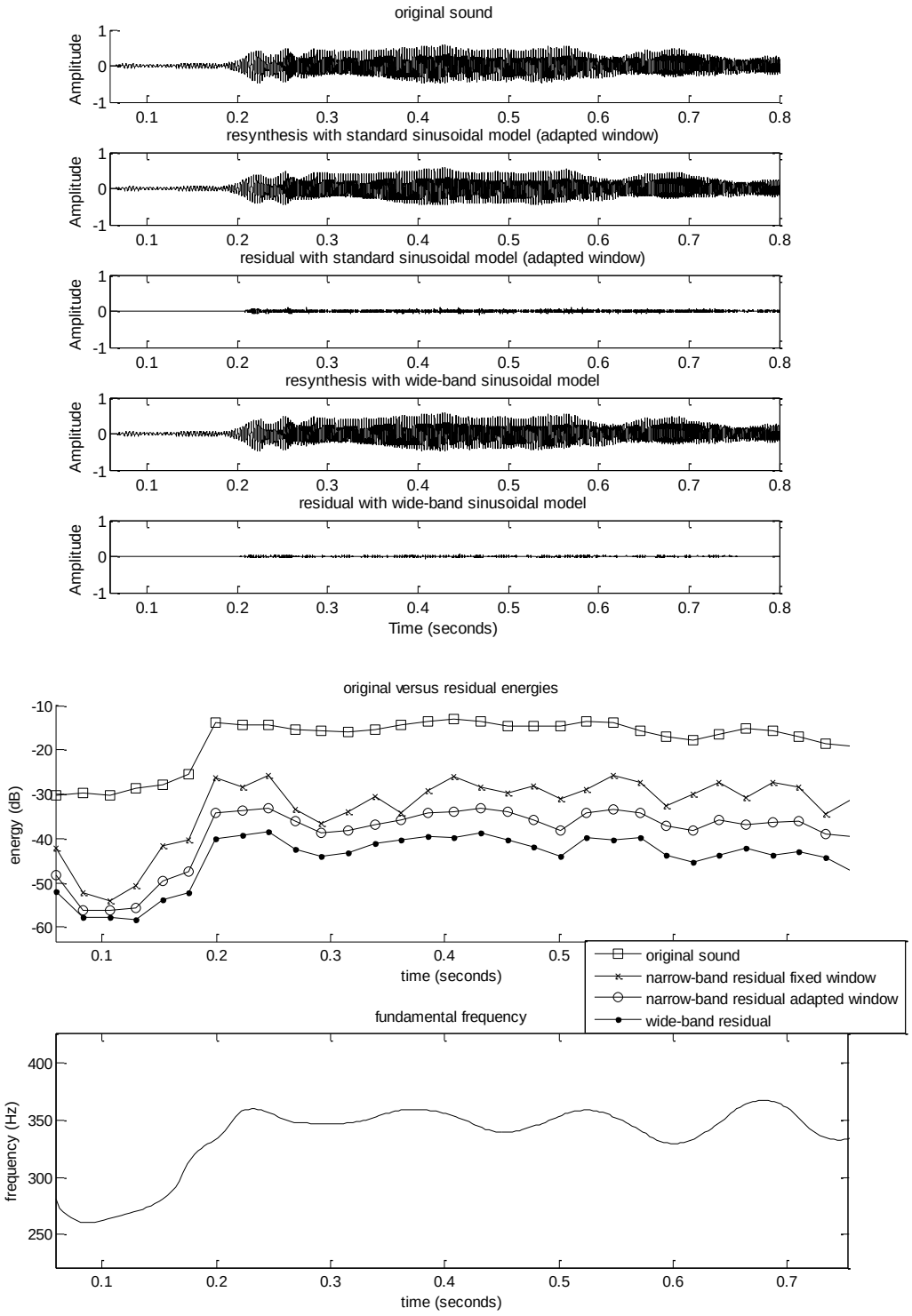


Figure 2.89 Recording (audio [107]) of a female singing the word 'yacht' expressively with scoop and vibrato. Wide-band residual energy is 5.9dB lower than the narrow-band one using a window adapted to cover five periods, and 12.44dB lower than the narrow-band case using a fixed window of 2049 samples.

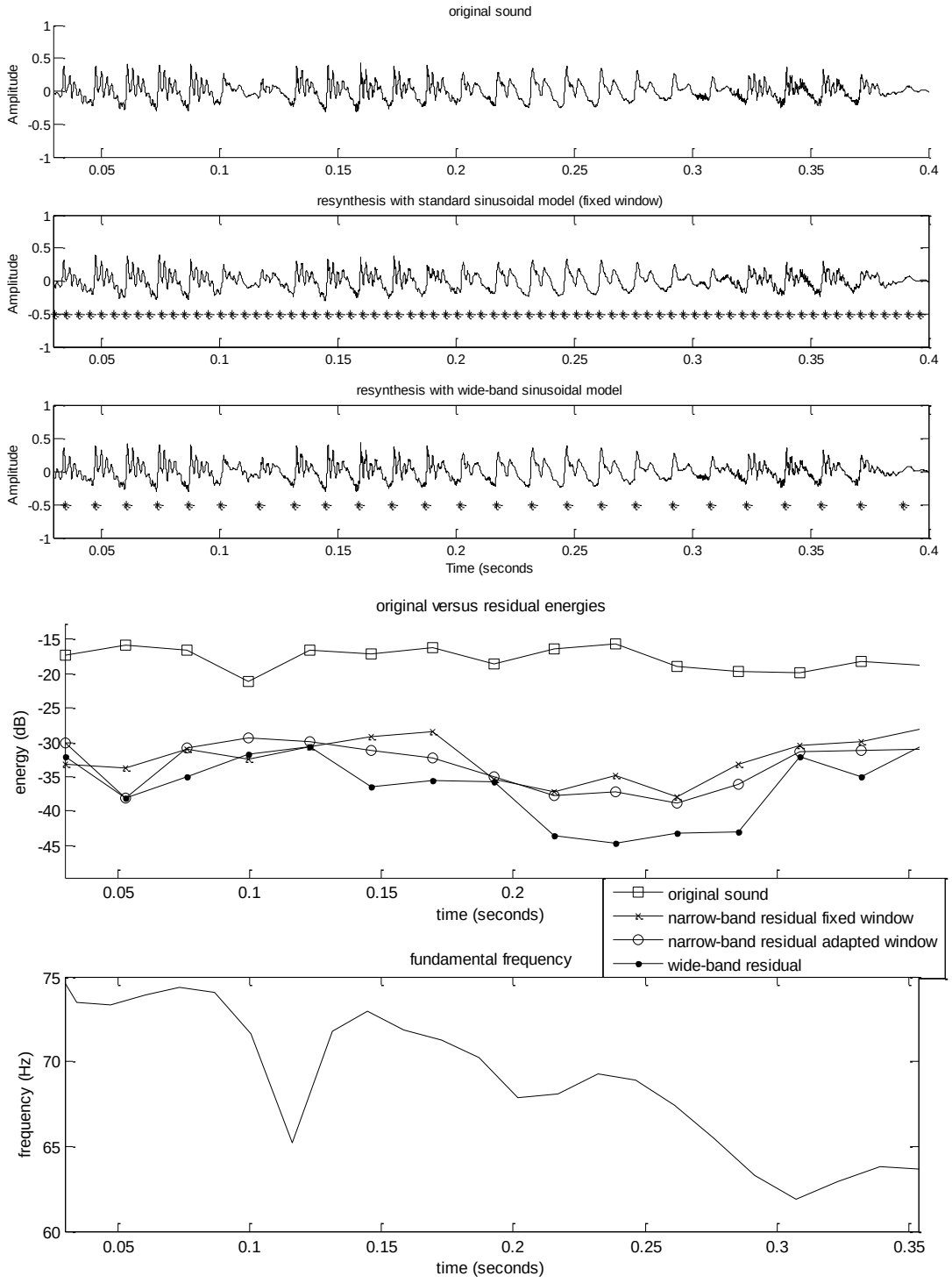


Figure 2.90 Recording (audio [108]) of a low pitch male speech utterance. Wide-band residual energy is 2.81dB lower than the narrow-band one using a window adapted to cover three periods, and 2.07dB lower than the narrow-band case using a fixed window of 2049 samples, and a constant hop size lower than the maximum period of the signal.

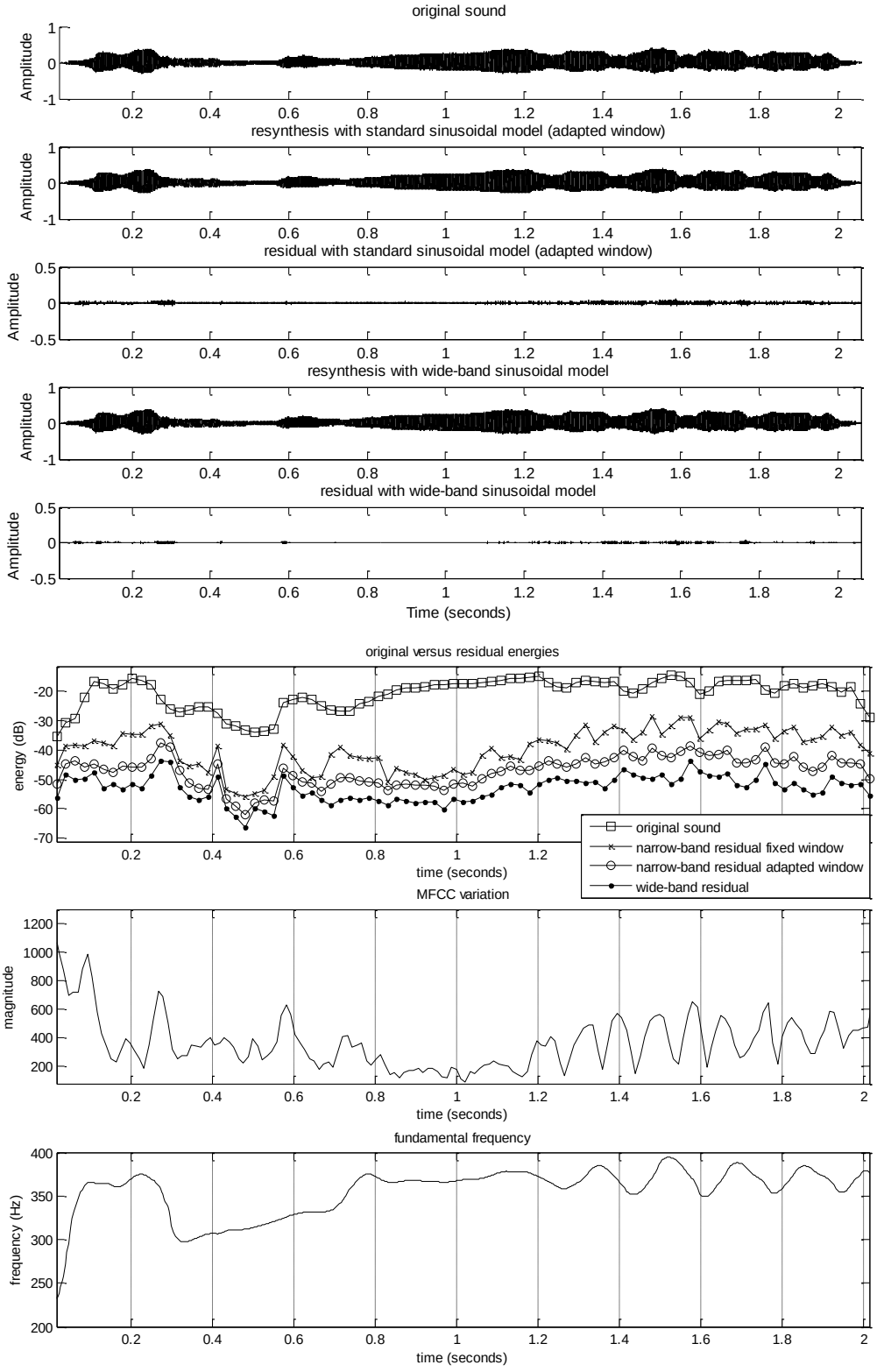


Figure 2.91 Recording (audio [109]) of a female singing expressively with a deep vibrato. Wide-band residual energy is 6.3dB lower than the narrow-band one using a window adapted to cover five periods, and 15.14dB lower than the narrow-band case using a fixed window of 2049 samples.

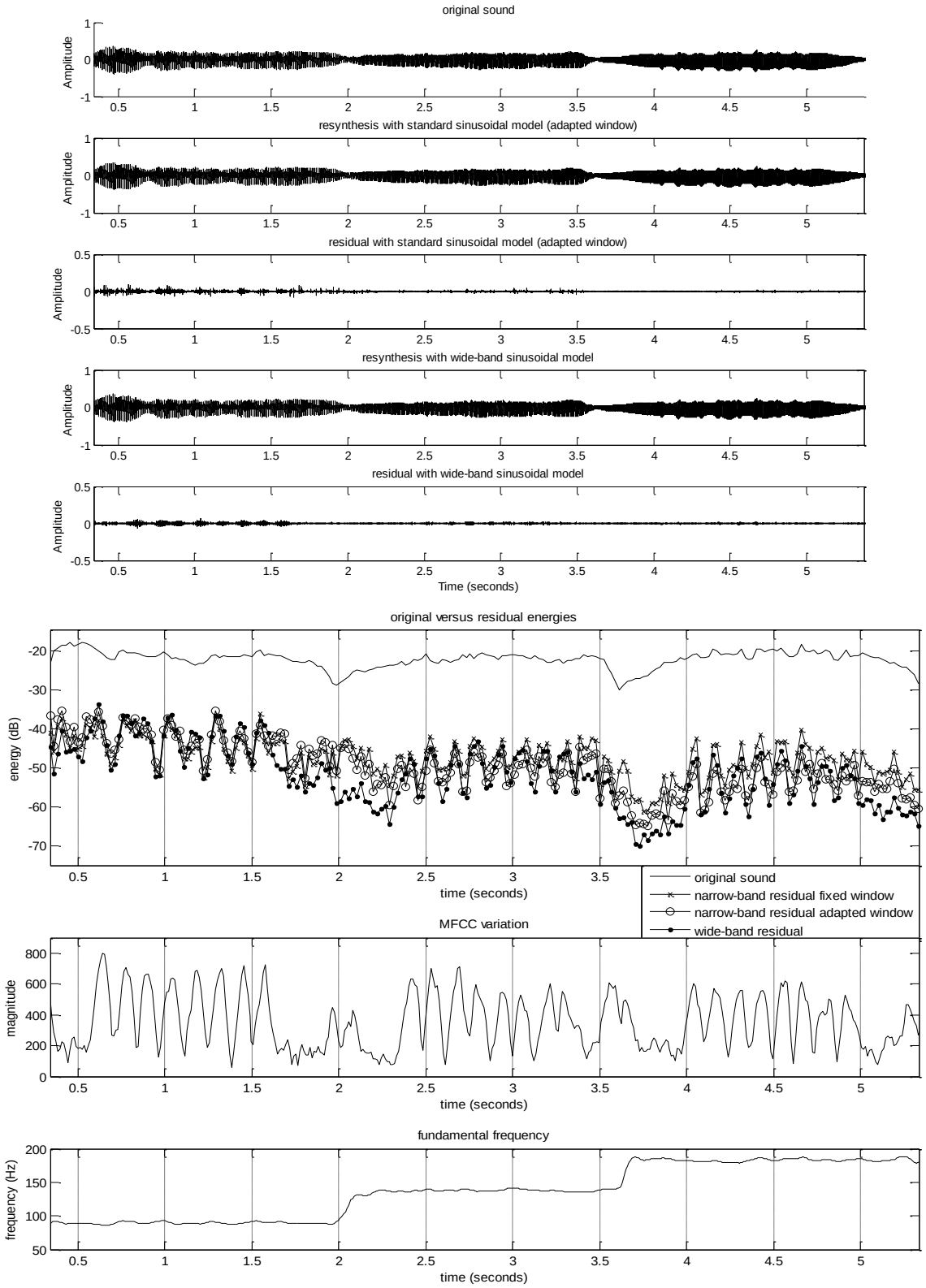


Figure 2.92 Recording (audio [110]) of a male singing three notes with flat pitch and fast vowel transitions. Wide-band residual energy is 0.6dB lower than the narrow-band one using a window adapted to cover three periods, and 0.63dB lower than the narrow-band case using a fixed window of 2049 samples.

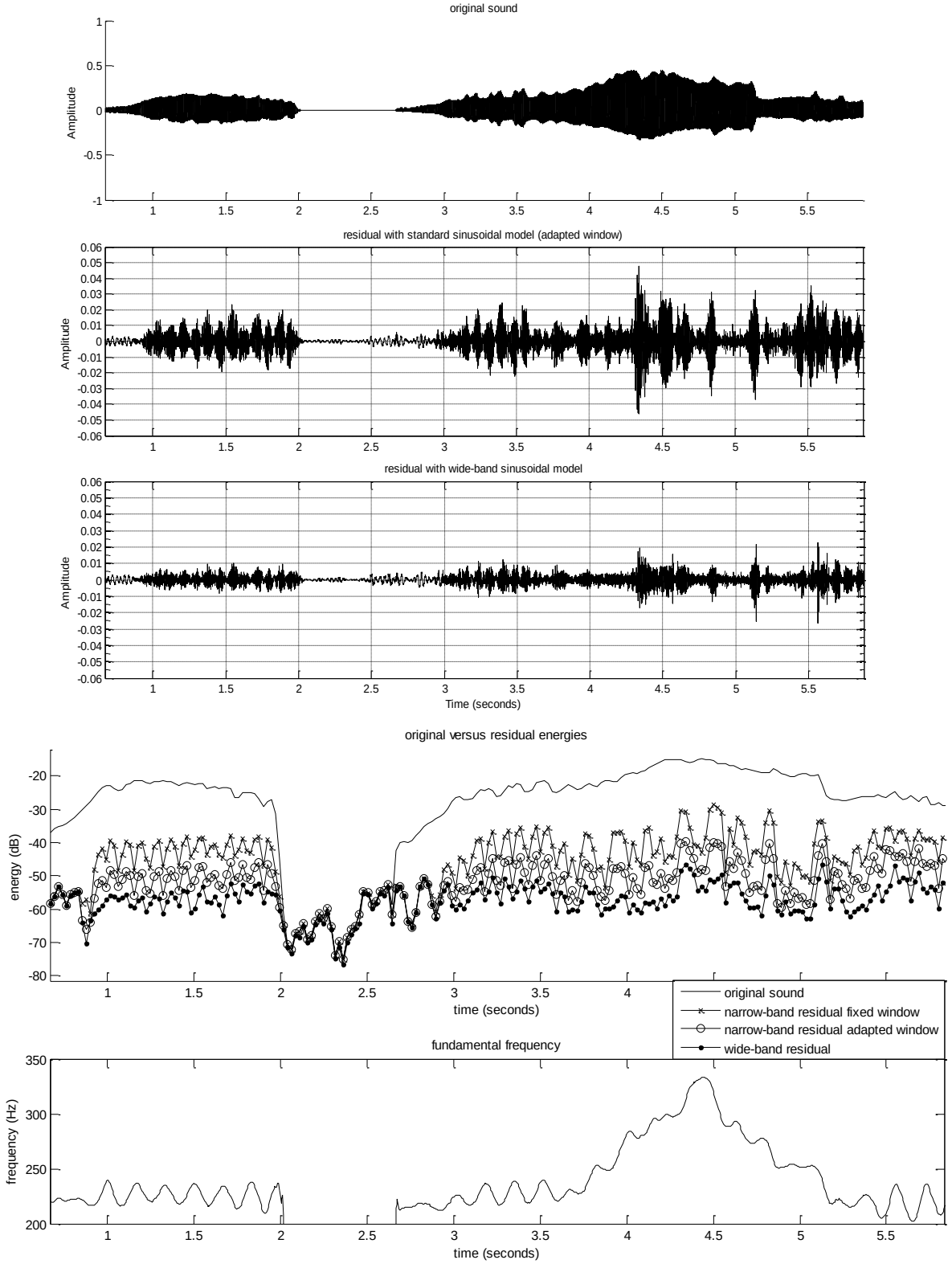


Figure 2.93 Recording of a tenor singing with strong vibrato. Wide-band residual energy is 6.49dB lower than the narrow-band one using a window adapted to cover four periods, and 15.33dB lower than the narrow-band case using a fixed window of 2049 samples.

SYNTHESIS

Figure 2.97 and Figure 2.99 show the steps involved in the synthesis phase using the periodization method. For each m^{th} period to synthesize, its spectrum $Y_r\left(e^{j\Omega}\right)$ is rendered by convolving the synthesis window transform $W'_m(f)$ by each of the harmonics. It is sufficient to use a small number of coefficients per harmonic, as proposed in (Depalle and Rodet 1990). Next, an IFFT is applied to obtain the time domain signal $y_m(n)$, consisting of a windowed sequence of identical periods at the synthesis pitch rate T'_m. Then this signal is windowed by $h_m(n)/w'_m(n)$ obtaining $p'_m(n)$, where $w'_m(n)$ is the window whose transform was used in the sinusoidal rendering process, and $h_m(n)$ is the synthesis overlapping window. All the synthesis periods are then overlapped according to the synthesis period onset sequence and the signal $y(n)$ is obtained.

It is also possible to use a synthesis method analogous to the upsampling process used in analysis. In that case each spectral bin of $Y_r\left(e^{j\Omega}\right)$ corresponds uniquely to one harmonic, and the IFFT computes the upsampled version of the synthesis period $y_m(n)$. Therefore, $y_m(n)$ has to be downsampled to the analysis sampling rate f_s, and then overlapped with the other synthesized periods according to the synthesis period onset sequence.

Both methods are equivalent and generate almost the same signal, with insignificant differences introduced by the downsampling and sinusoidal rendering steps. Although the second method is usually more efficient in terms of computation, whenever inharmonic components are being synthesized the first method is likely to be the most efficient one. On the other hand, it is important to point out that the input signal cannot be perfectly reconstructed when no transformations are applied due to the overlapping applied at the borders of the analysis window (see Figure 2.95). However, informal listening tests have shown that in most cases the synthesized signal is indistinguishable from the original one.

TRANSFORMATIONS

There are two main types of transformations, the ones related to the period onset sequence and the ones related to each individual period, as depicted in Figure 2.70. Considering the traditional source-filter voice model, we could say that the former group of transformations are related to the voice source whereas the latter to the vocal tract. Traditional transformations such as time-scaling and pitch transposition involve scaling the period onset sequence, and repeating, removing or interpolating periods, in the same way as done in typical time-domain PSOLA techniques (see Figure 2.98). However, pitch transposition also requires modifying the harmonic components of each period in order to match the target fundamental frequency, although phase continuation is not needed since consecutive period onsets are distant by one period.

Conversely, timbre transformations work as in typical frequency-domain techniques, by modifying the individual frequency components as depicted in Figure 2.96 and Figure 2.97. Initially, the spectral envelope $H_{harm}(f)$ is computed by interpolation of the estimated sinusoids and then properly modified according to the warping function $\overline{T}_{timbre}(f)$ and maybe some filtering. Preferably both spectral and phase envelopes should be modified by the same scaling function, with the aim of preserving the resonance-to-phase relationship. If the phase envelope is interpolated then the inherent phase wrapping has to be considered. Finally, synthesis sinusoidal components are computed out of the target fundamental frequency and both timbre and phase envelopes. Inharmonic components can be synthesized as well, although they require us to propagate phase so to avoid discontinuities in the synthesized signal. Figure 2.97 shows an example of transposition to a lower pitch and timbre stretching. Note that in this example the phase envelope is not interpolated but a mapping function is used to determine which input harmonic's phase is used for each output harmonic.

VOICE SIGNALS

Regarding voice signals, we already justified the necessity of using shape invariance techniques for achieving a good sound quality. In the case of the wide-band pulse modeling approach proposed here, it means that voice pulse onsets used in the analysis should be close to the actual glottal pulse onsets. Therefore, as depicted in Figure 2.95, voice pulse onsets are determined by the MFPA

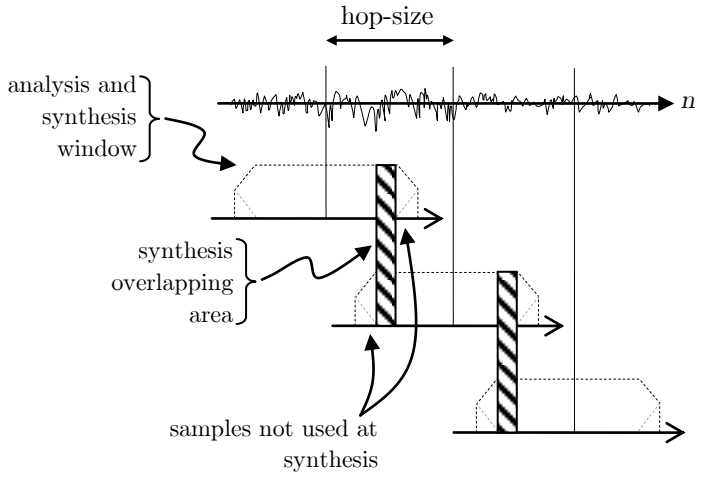


Figure 2.94 Scheme of the processing method for achieving perfect reconstruction of unvoiced signals when no transformations are applied. The samples that are actually used to generate the output sound are located out of the border interpolation area.

algorithm before performing the wide-band analysis, so that the analyzed periods are centered on them.

On the other hand it is interesting to point out that both harmonic and aspirated noise components present in voiced utterances are represented exclusively by sinusoids. Actually, since the analysis is performed pitch-synchronously, the noise produces differences between consecutive periods that result into amplitude and phase modulations of the detected harmonics.

UNVOICED SIGNALS

Unvoiced signals can be processed as if they were voiced by assigning an arbitrary fundamental frequency. However, even when no transformations are applied, the analysis fundamental frequency can be slightly perceived in the synthetic signal. This can be avoided and a perfect reconstruction achieved by using a shorter period value for the period onset sequence than for the period analysis, so that from the signal obtained by the IFFT only the section not affected by the border overlapping is used to compute the output signal. Nevertheless, the computational cost is slightly increased. This method is illustrated in Figure 2.94.

DISCUSSION

WBVPM is able to model voice pulses in frequency domain with a set of sinusoids, which represent both harmonic and noisy components. It provides an independent control of each single pulse, thus allowing pulse sequence transformations with ease. This ability is typical of time-domain methods, but complex to achieve in frequency domain, since it implies dealing with complex subharmonics patterns. At the same time, WBVPM's sinusoidal representation of the signal allows an independent control of each single harmonic component, this way overcoming typical limitations of time-domain techniques. In this sense, WBVPM combines some of the main pros of both time and frequency-domain methods while avoiding some of their main drawbacks.

WBVPM works in wide-band conditions in such a way that it overcomes the time-frequency analysis constraint where several periods of signal are required in order to achieve enough frequency resolution for estimating harmonic components. In addition, it provides a very good temporal resolution and effectively reduces the smearing characteristics typical of frequency-domain techniques. In addition, WBVPM is essentially pitch-synchronous but obtains the best results with voice signals if estimated pulse onsets match glottal pulse onsets. In this sense, the quality of transformed voices is highly dependent on a good estimation of the voice pulse onsets, especially when dealing with low pitch voices.

Compared to NBVPM, WBVPM has the benefits of increased temporal resolution, reduced smearing, and lower computational cost. By contrast, NBVPM is able to decompose the signal into harmonic, transients and noise components, and handle them independently, thus offering a higher control flexibility. However, in spite of this and according to our informal listening tests, WBVPM produces in most situations a better sound quality and naturalness, a more compact sound, and seems to be more robust against analysis imprecisions in terms of perceived artifacts.

Regarding real-world applications, as it will be shown in the following section, WBVPM can process signals in real-time and its computational cost is lowly dependant on the signal content. Moreover, whereas NBVPM was conceived solely for voiced signals and required of other techniques for handling unvoiced segments, WBVPM can process unvoiced sections within the same framework with just a few minor changes. These are desired features in practical application contexts.

One of the future directions of research should be to address the relationship between harmonic and noisy components, which WBVPM represents only with sinusoids. In quasi-stationary conditions, the noise component would be the main responsible of the subtle differences between consecutive voice pulses. We should expect this noise component to introduce amplitude and phase modulations in the harmonic trajectories, which might be modeled statistically and transformed independently of the harmonic component.

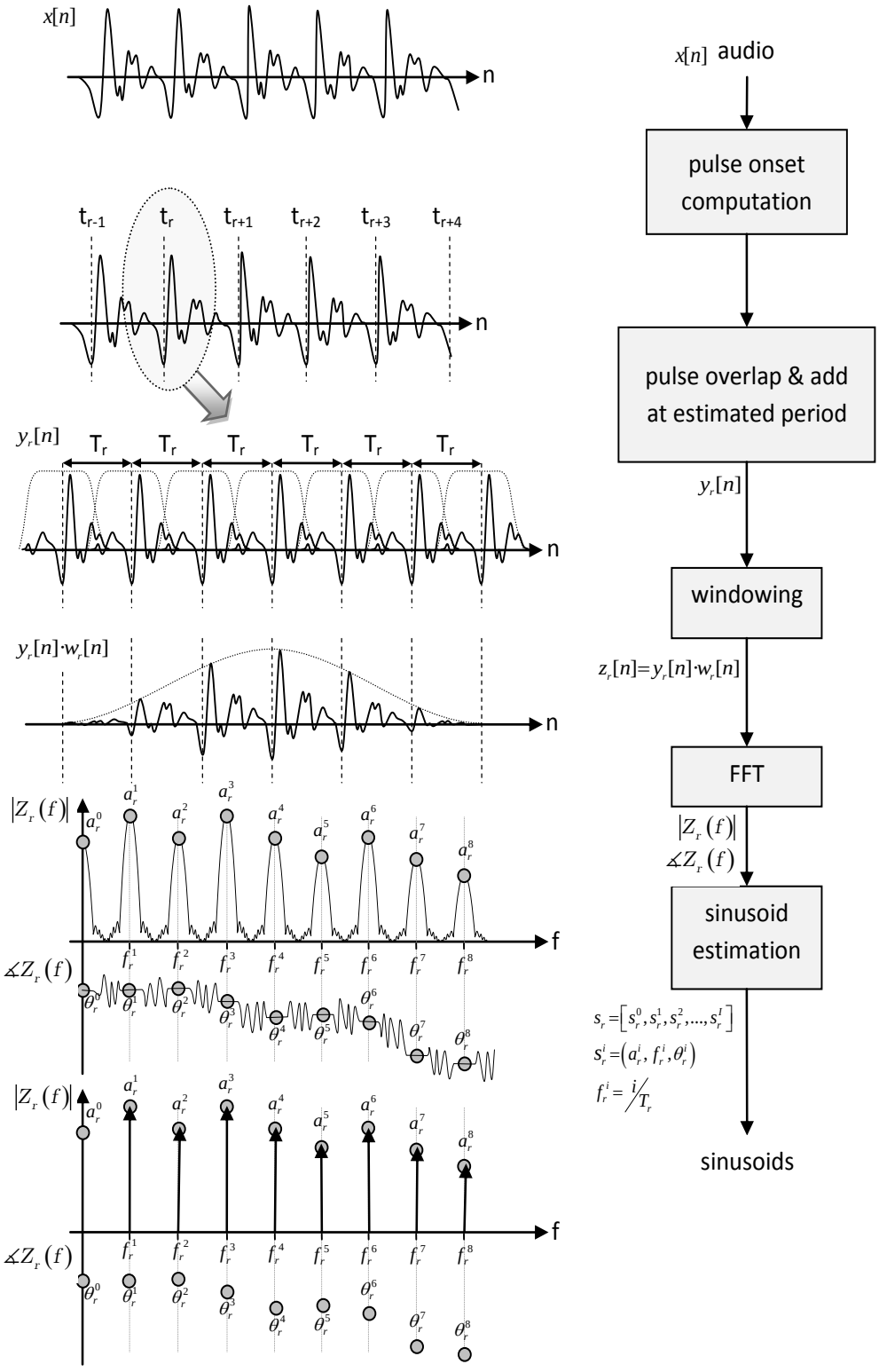


Figure 2.95 Block diagram of the analysis phase

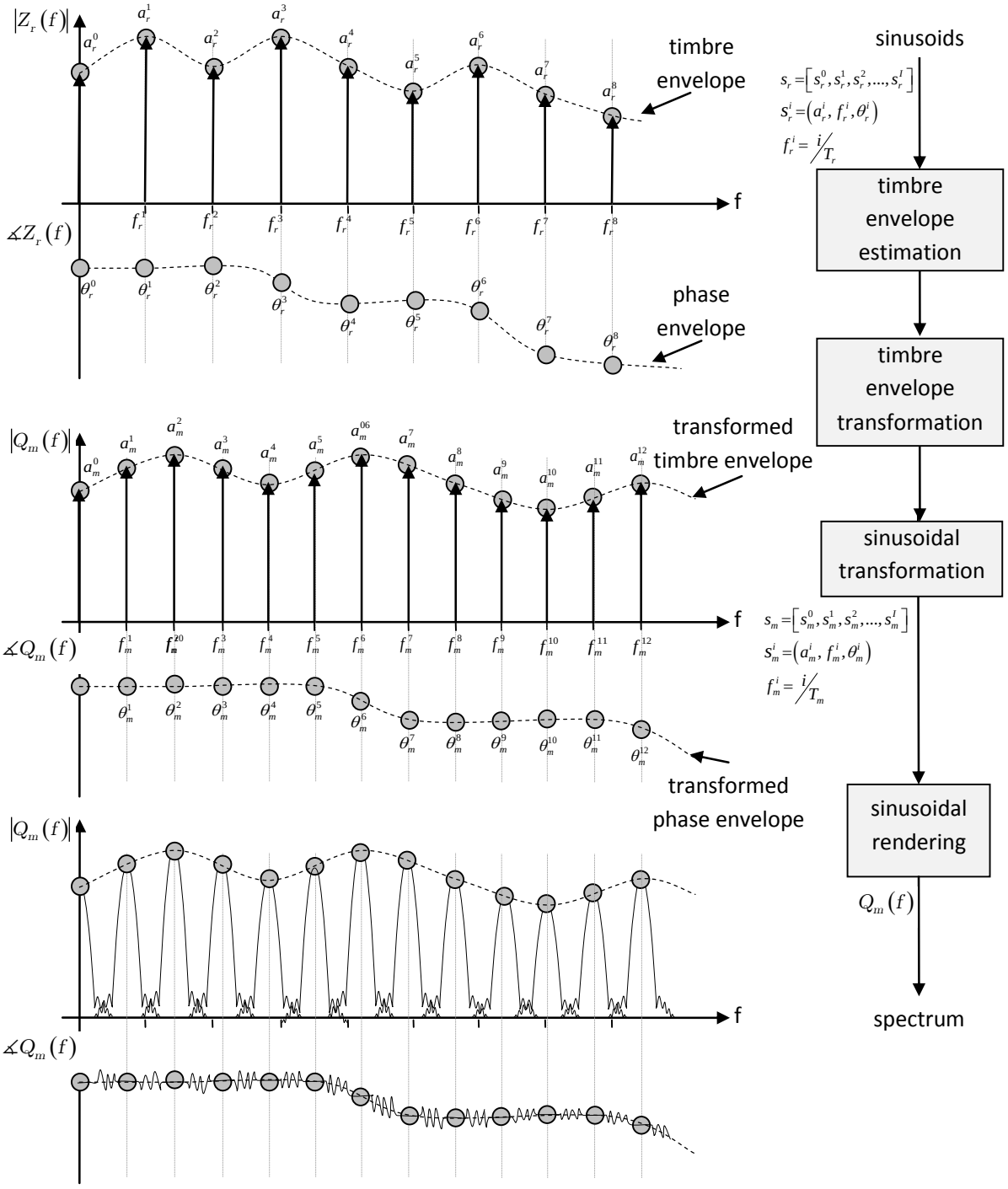


Figure 2.96 Block diagram of the pulse transformation and rendering processes. This example shows a voiced frame which is transposed to a lower pitch and whose timbre envelope is expanded.

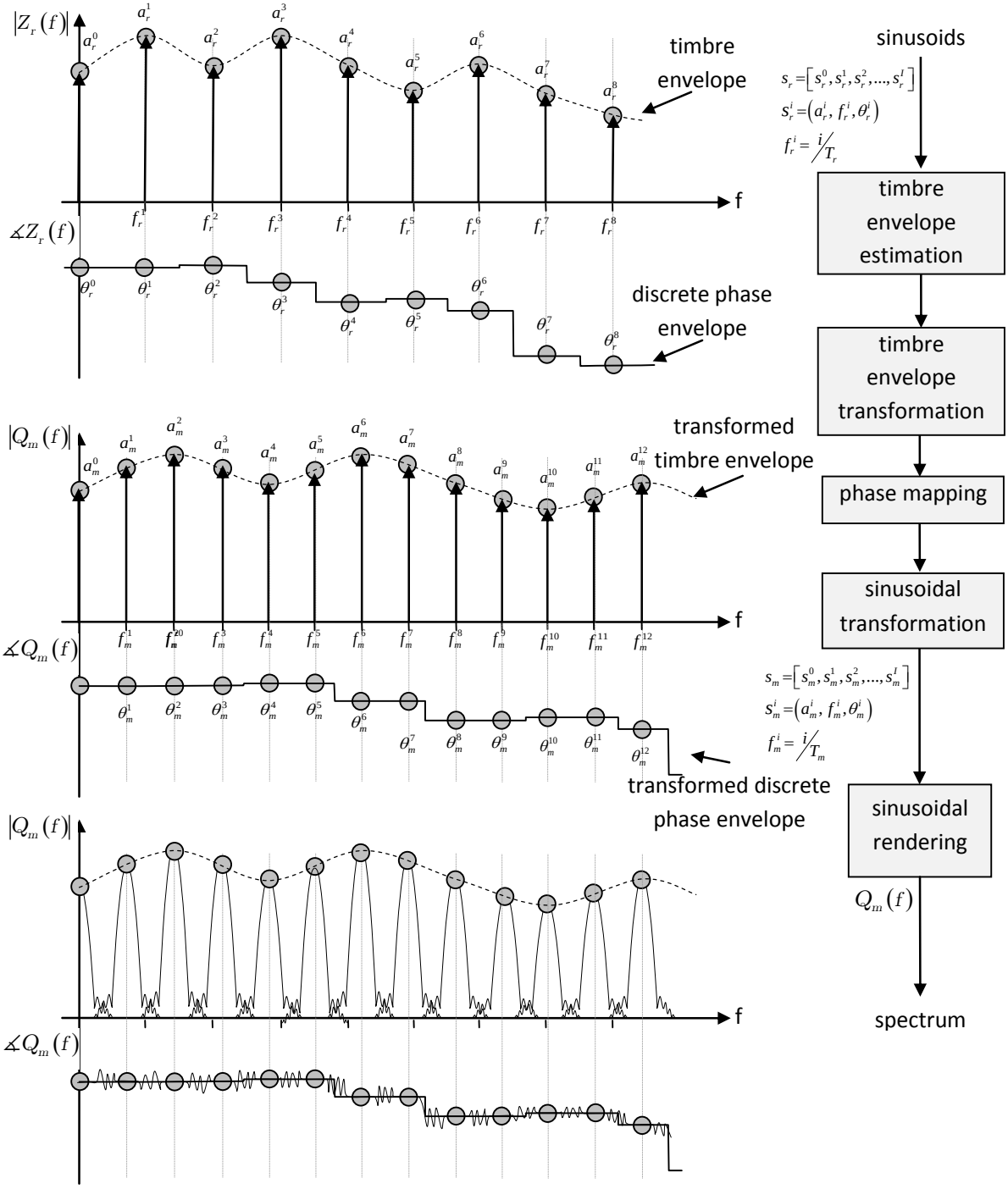


Figure 2.97 Block diagram of the pulse transformation and rendering processes. This example shows a voiced frame which is transposed to a lower pitch and whose timbre envelope is expanded. The phase envelope is not interpolated, but a mapping function is used to determine which input harmonic's phase is used for each output harmonic.

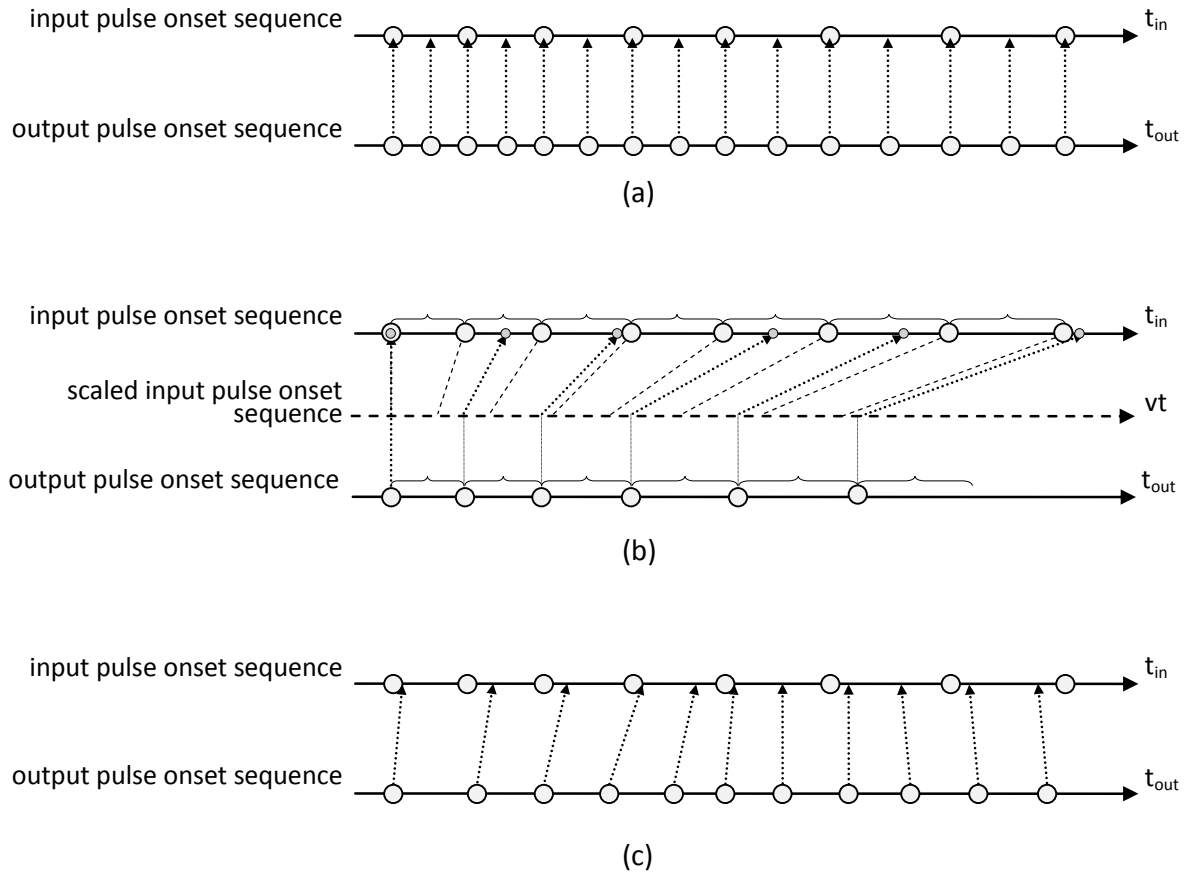


Figure 2.98 Examples of pulse onset sequence modifications. For each example, the top axis shows the estimated pulse onsets whereas the bottom axis shows the modified pulse onset sequence. The arrows point the time from which to compute the pulse data to be synthesized, which in some cases imply interpolation between input pulses. In (a) an octave up transformation is applied. In (b) a time-scaling of 1.5 times faster rate is applied. In (c) a combination of time-varying pitch transposition and time-scaling transformations are applied.

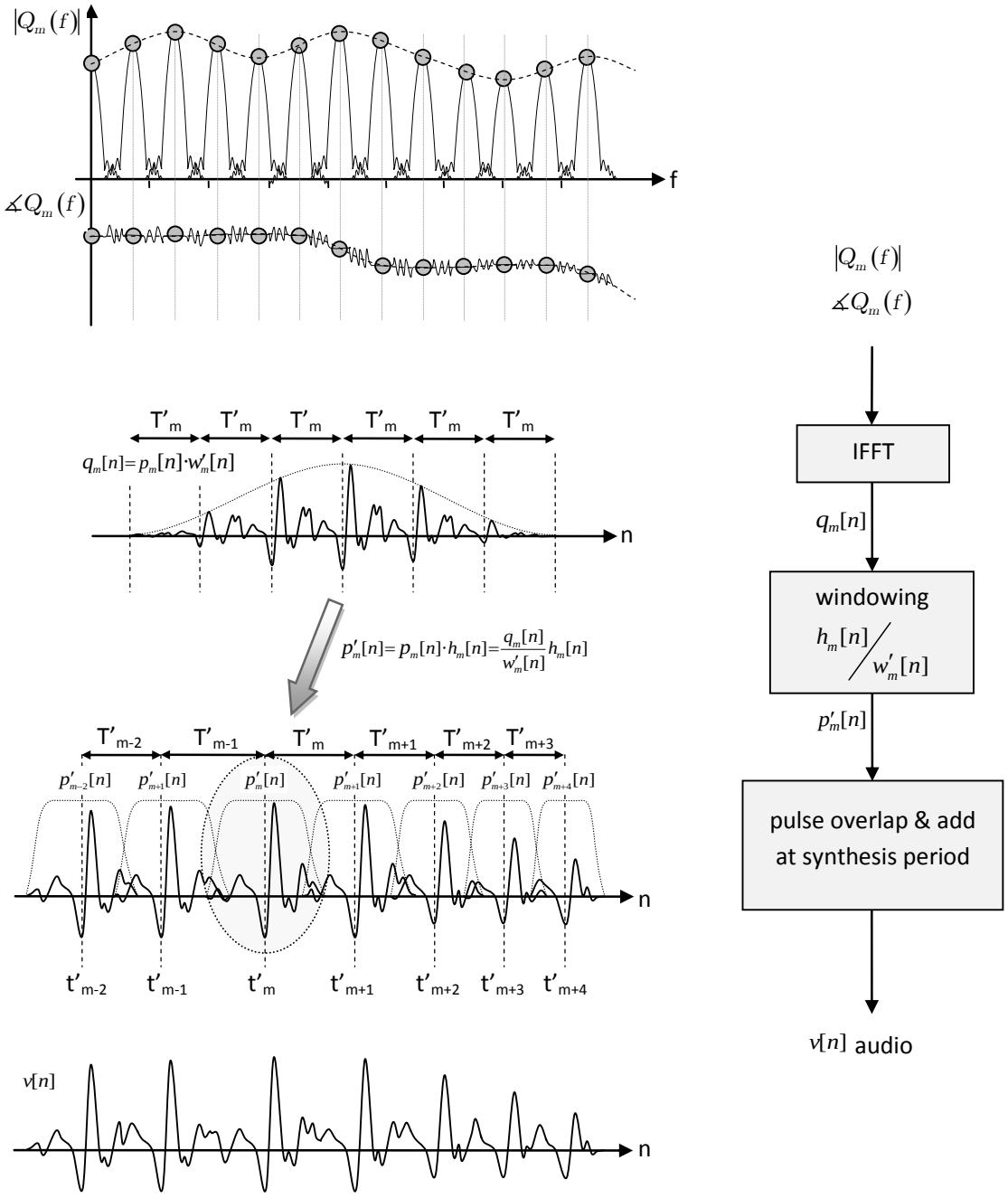


Figure 2.99 Block diagram of the several synthesis phase steps using the periodization method. For each pulse m to synthesize, with spectrum $Q_m\left(e^{j\Omega}\right)$, an IFFT is applied to obtain the time domain signal $q_m(n)$, consisting of a windowed sequence of identical pulses at the synthesis pitch rate T'_m. Then this signal is windowed by $h_m(n)/w'_m(n)$ obtaining $p'_m(n)$, where $w'_m(n)$ is the window whose transform was used in the sinusoidal rendering process, and $h_m(n)$ is the synthesis overlapping window. Finally, all the synthesis pulses are overlapped and the resulting signal $v(n)$ is obtained

(a) (b)

Figure 2.100 Voice transformation plug-ins that use NBVPM and WBVPM techniques: (a) *ComboVox* and (b) *VoiceTransform.*

2.3.3 Voice Pulse Modeling Applications

Several implementations of the proposed approaches to Voice Pulse Modeling have been incorporated in a wide range of applications including professional audio effects software for voice manipulation, museum installations and videogames (Mayor, et al. 2009). We next detail the most relevant ones.

VOICE TRANSFORMATION PLUG-INS

Voice Pulse Modeling has been integrated in audio plug-ins aimed for a wide range of users from amateurs to professionals. One example is ComboVox, a plug-in included in the Bonus DVD Pack of the Pinnacle Studio 10 software that allows the user to transform a voice in an audio track of a video using a set of predefined presets. These presets include human based modifications such as gender change or age change, but also fiction transformation like a robotizer effect, alien effect, ogre effect, and others. ComboVox uses a real-time implementation of NBVPM, with an approximately processing latency of 60ms. With this plug-in, even unskilled users can very easily transform a kid voice into an elderly woman or give a man's voice a special robot sound or monster effect. Figure 2.100a shows a screenshot of this plug-in, where each transformation is represented by a figure.

Another example is a VST plug-in developed in the context of the SALERO IST European project[21], that uses WBVPM. It allows transforming voices in real-time, based in a set of meaningful controls categorized in tuning (pitch and vibrato), sinusoidal controls (frequency shift & stretch), amplitude modulation, excitation controls (roughness, whisper, breathiness or robotizer) and timbre modification parameters. Moreover, harmonizer controls allow transforming one voice to many voices in real-time while controlling and panning each voice independently. This plug-in offers the possibility of creating presets based on the control parameters and store them for later use. In addition, it provides visual feedback of the input and output sound characteristics, including pitch (piano roll visualization), spectrum and waveform, as shown in Figure 2.100b. This prototype has been developed in C++ and runs in real-time under any compatible VST hosts. This plug-in is aimed for advanced users to design new voices in a post processing studio but also offers basic functionalities for an amateur user, that wants to transform a voice easily using predefined transformations.

[21] http://www.salero.eu

(a) (b)

Figure 2.101 Two real-time museum installations that use WBVPM: (a) *The Voice Kaleidoscope* and (b) *My Voice Produces Waves*.

REAL-TIME MUSEUM INSTALLATIONS

WBVPM real-time implementations are suitable for public installations in museums where the visitor speaks to a microphone, selects the desired voice transformation to be applied from a set of presets and is able to listen and visualize in real-time some parameters of the transformed voice. We developed two installations, which were funded by "La Caixa" foundation.

- **The Voice Kaleidoscope** (Figure 2.101a)
  This is an installation composed by an interactive kiosk with a 19'' touch display for selecting the transformations, a big 40'' screen for in/out voice parameters visualization, a microphone to capture the input voice and a set of speakers for reproducing the transformed voice. In the touch screen, allowed voice transformations are represented by image icons (male, female, elder, child, monster, robot, alien or cartoon). The user presses one of these icons to select the desired transformation. Then system analyzes the sound coming from the microphone near the kiosk, transforms it to sound like the pressed icon and reproduces it in real-time through a set of speakers situated in front of the user. The user can view in the big display a representation of some characteristics of both the input voice and the output transformed one, like the waveform, spectrum and pitch. This installation is being used by thousands of visitors daily and is gathering positive feedback, proving that the technology is reliable and robust in 24/7 situations.

- **My Voice Produces Waves** (Figure 2.101b)
  In this installation, user are children from 7 to 9 years. They talk to a microphone and by pressing some buttons they are able to transform and listen to their voice in real-time. The waveforms of input and transformed voices are drawn in real-time in a panoramic display, so that children can understand that the voice produces sound waves and that the transformation of the voice also transforms the shape and periodicity of those sound waves. Every transformation corresponds to a different light button and an image icon identifies each transformation. This installation is being used by hundreds of children visitors daily, mainly by organized school groups. The feedback gathered from monitors and instructors demonstrates the success of such installation, and the reliability and potential of the technology, allowing for instance 7 year-old kids' voice to sound like their parents.

WEB APPLICATIONS

We also implemented WBVPM in a web service application that allows transforming audio files in an offline process[22]. The client uploads a file to the server, selects a transformation preset, and the

[22] recommender-mtg.upf.es/salero

Figure 2.102 MyTinyPlanets video game application.

server returns a URL with the transformed file. This application is targeted for example to post-production studios or as web service for Text-To-Speech systems. We plan to use it to carry out a perceptual experiment with a large number of participants, with the aim of rating voice quality, naturalness and plausibility of different transformations.

VIDEO GAMES

The proposed algorithms can also be used as an offline tool to help videogame sound designers to create new fiction voices for the characters in the game, allowing real-voices to sound as if they were robots, aliens or monsters. One example is the WBVPM-based application in the context of the SALERO European project that can be found in a series of Flash based games called *MyTinyPlanets*[23]. One screenshot is shown in Figure 2.102. WBVPM is used in this application to create new virtual character voices from voices synthesized with a Text-to-Speech system as well as from some other voices recorded by speakers. For instance, from an original female voice, several new voices have been created for a male, a child, an alien, a robot and some other characters that appear in the videogame.

2.4 Computational Cost

Obviously, each algorithm implements different operations that require different computational costs (i.e. number of operations). However, if we consider the input signal characteristics, there is one issue that significantly affects the computational cost of all the algorithms previously exposed: their dependency on the estimated fundamental frequency. This dependency relies in two main factors:

- *processing frame rate*
  either constant (e.g. phase-vocoder, SMS) or pitch-synchronous (e.g. PSSM, TD-PSOLA)
- *window size*
  either constant (e.g. phase-vocoder) or adapted to the pitch (e.g. SMS, PSSM, TD-PSOLA)

Given an audio segment of length L and an arbitrary algorithm *alg* with an order of L time complexity, the computational cost of processing one segment can be expressed as $C_{segment}^{alg} = \mathrm{O}(L)$.

Let us now consider that the length of the segment is adapted to contain n periods with a fundamental frequency f_0, thus ensuring enough frequency resolution as to detect the harmonic peaks. Then we get $L = n/f_0$, and the computational cost for each segment becomes

[23] http://www.mytinyplanets.com

$C_{segment}^{alg} = \mathrm{O}(n/f_0) = \mathrm{O}(f_0^{-1})$. This means that the lower the fundamental frequency, the higher the computational cost, and vice versa.

On the other hand, the number of segments to process per second will be $m = 1/hopsize$ in the case of a constant processing frame rate. Otherwise, in the case of a pitch-synchronous algorithm and a constant fundamental frequency f_0, $m = 1/T = f_0$. This means that the lower the fundamental frequency, the less the number of segments to process, thus the lower the computational cost.

Considering the case when the input signal has a flat pitch (i.e. constant), the computational cost per second is given by the number of pulses to compute times the computational cost of each pulse

$$C_{alg} = mC_{segment}^{alg} \tag{2.101}$$

The following table shows the order of time complexity for each possible frame rate and window size combination, and enumerates some algorithms as example.

	constant frame rate $(hopsize)$	pitch synchronous (f_0)
constant window size (L)	$C_{alg} = \frac{C_{segment}^{alg}}{hopsize} = \mathrm{O}(L)$ *phase vocoder*	$C_{alg} = f_0 C_{segment}^{alg} = \mathrm{O}(f_0 L)$
adapted window size (n/f_0)	$C_{alg} = \frac{C_{segment}^{alg}}{hopsize} = \mathrm{O}(f_0^{-1})$ *sinusoidal model, SMS, NBVPM*	$C_{alg} = f_0 C_{segment}^{alg} = f_0 \mathrm{O}(n/f_0) \approx constant$ *TD-PSOLA, WBVPM*

Table 2.6 Order of time complexity for different frame rate and window size combinations

In the case of an algorithm applying a fixed window size, for low pitch values it might happen that the window size was shorter than the minimum number of periods (i.e. $L < nT$), then not being able to detect precisely the harmonic peaks. The window size should be increased to deal with such low pitched signals, but then the temporal resolution would be lowered and the processed sound probably smeared. It is desirable then to set the length of the window as n times the lowest fundamental frequency of the signal to process.

The best situation is, however, the combination of a pitch-synchronous algorithm with a window adapted to cover a certain number of pitch periods. That's a good compromise for the time-frequency resolution trade-off. In addition, for such case the cost is almost independent of the pitch of the input signal. Figure 2.103 illustrates this concept. This is a desired feature because otherwise the computational cost would increase significantly for low or high fundamental frequency values.

A Case Study: Wide-Band Voice Pulse Modeling (WBVPM)

The following section explores the relationship between pitch rate and computational cost for each of the WBVPM algorithm steps using periodization. From now on, T is the analysis pitch period length, and T' is the synthesis pitch period length.

ANALYSIS PHASE

Pulse onset sequence computation

- In the case it uses the pitch detection algorithm in (Cano 1998) and the MPFA method, its computational will not depend on the pitch of the input signal, but on the window size applied (set to cover n periods of the lowest allowed fundamental frequency). This results in a cost proportional to $n/f_0^{\min}$.

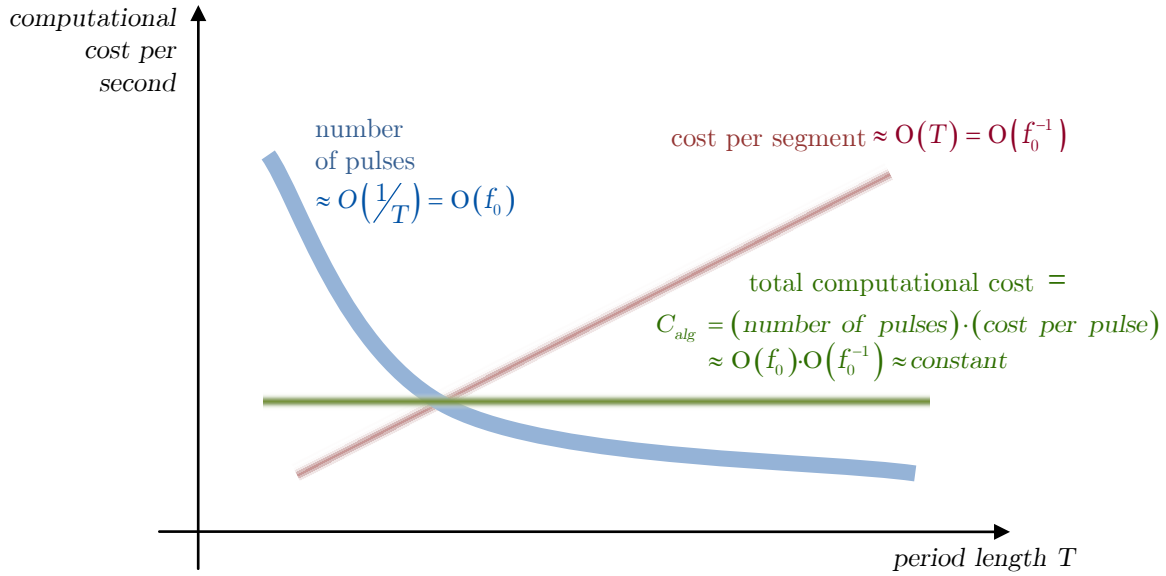


Figure 2.103 Rough estimation of the computation cost for a pitch-synchronous algorithm with a window length adapted to the fundamental frequency of the input signal. The total cost C_{alg} per second is mostly independent of the fundamental frequency of the input signal.

Pulse Analysis

- Repeating a segment using a time domain waveform interpolation method has a computational cost proportional to the length of the segment T
- Windowing has a computational cost proportional to the length of the segment T
- FFT has a computational cost in the order of $O(T \cdot \log_2(T))$
- Sinusoidal estimation has a cost proportional to the number of sinusoids, thus proportional to T.

TRANSFORMATION PHASE

Pulse Onset Sequence Transformation

- The cost is proportional to the pulse rate, so to the pitch rate (i.e. the more pulses per second in the sequence, the more pulses to compute)

Timbre Envelope Estimation

- Here the amplitude of the harmonics is computed by means of interpolation. The computational cost is proportional to the number of harmonics, thus to the length of the segment T

Timbre Envelope Transformation and *Sinusoidal Transformation*

- These two operations can be performed together in the same loop. The cost is proportional to the number of harmonics, thus to the length of the segment T

SYNTHESIS PHASE

Sinusoidal Rendering (also referred to as pulse rendering)

- The computational cost is proportional to the number of sinusoids, thus to the length of the segment T'

IFFT

- Computational cost proportional to $\mathrm{O}\left(T' \cdot \log_2\left(T'\right)\right)$

Windowing

- Computational cost proportional to the length of the synthesis segment T'

Pulse Overlap & Add At Synthesis Period

- Computational cost proportional to the length of the synthesis segment T'

Regarding the previous list of operations, we should point out that the operations performed on a single pulse or voice period (e.g. *pulse analysis*, *pulse transformation*, *pulse synthesis*) are repeated for each pulses in the sequence. Therefore, those operations will be performed as many times as the number of pulses. Consequently, for a given audio duration, the higher the pitch rate, the more pulse operations to perform, and therefore the higher the computational cost. However, on the contrary, the higher the pitch rate, the shorter the pulse length, and therefore the lower the computational cost of most pulse operations.

This is the reason why WBVPM computational cost is less dependant of the input audio characteristics, whether it features a low or a high pitch.

The WBVPM algorithm by periodization has been implemented in C++. The resulting computational cost has shown to be low enough as to afford real-time processing on a regular computer. Interestingly, the synthesis module has a much lower computational cost than the analysis module (asymmetric computational cost between analysis and synthesis phases), offering the possibility of synthesizing many voices simultaneously. Using this implementation we have processed ten times, with no transformations, a saw sweep signal with a fundamental frequency going from 13Hz up to 3kHz. The resulting performances versus fundamental frequency are drawn in Figure 2.104. One of the performances seems to be much worse than the rest (in purple), probably because the computer was stressed at that time doing other tasks. Leaving it aside, the performance appears to be quite flat in the frequency range [20Hz,1kHz], laying between 4 and 4.5 times real-time in most cases. Between 13 and 20Hz the performance increases significantly to [4,6.5] times real-time. By contrast, between 1 and 3kHz, the performance slightly decays to the range [3.5,4] times real-time.

In contraposition, Figure 2.105 shows the same experiment but using a sinusoidal model with a window size adapted to the fundamental frequency but a constant frame rate. The performance measurements are shown for both analysis (in blue) and synthesis (in red) phases. The synthesis process is around ten times faster than the analysis one, probably because its implementation is highly optimized: it is not based on a bank of time-domain oscillators, but on the rendering the sinusoids to the output spectrum using the main lobe of the synthesis window transform. Clearly, the figure shows that there is a strong correlation between the algorithm performance and the fundamental period of the processed signal. At low pitches, around 13Hz, the synthesis performance decays to around ten times real-time, but the analysis performance decays to less than 0.5 times real-time, so being impractical to perform in real-time.

Finally, we have done another experiment using a debug (therefore not optimized) implementation of the WBVPM algorithm. We have processed a 11.23 seconds long voice utterance of a male voice, with a fundamental frequency ranging from 50 to 250Hz. In addition, the speech signal has been transposed to -24, -12, +12 and +24 semitones. Then each of these processed signals has been processed again applying the same transformations. Figure 2.106 shows the results in terms of processing time versus mean fundamental frequency. It also shows the fundamental frequency histograms of the original signal and its transposed versions. The processing times last between 5.8 and 6.9 seconds for all cases.

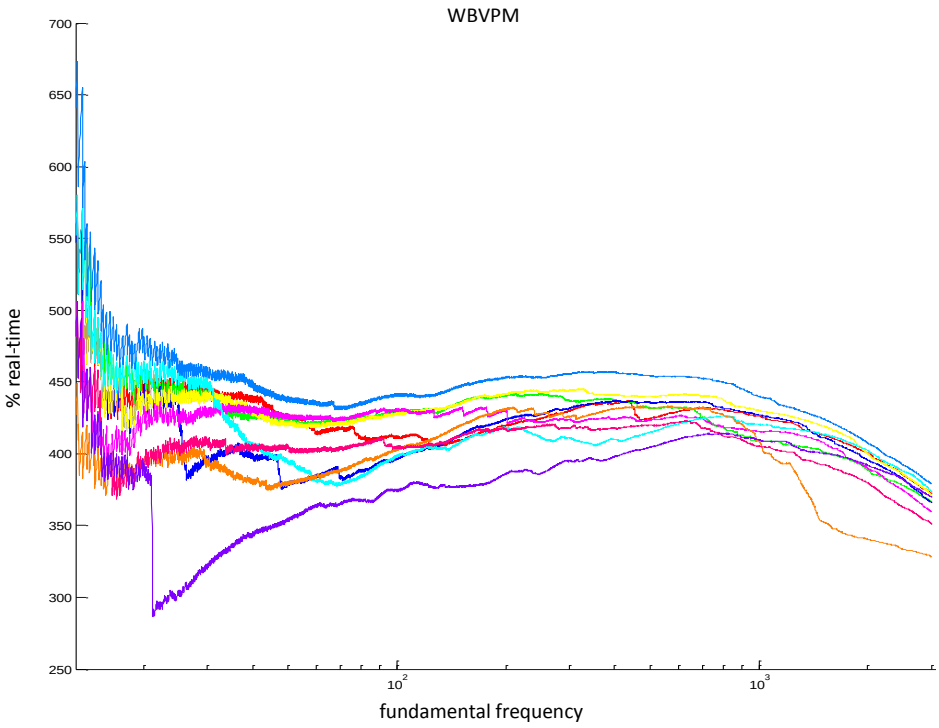


Figure 2.104 WBVPM percent of real-time versus fundamental frequency (in logarithmic scale). The input file is a sweep from 13 to 3000Hz. It has been processed ten times, each represented by a different color.

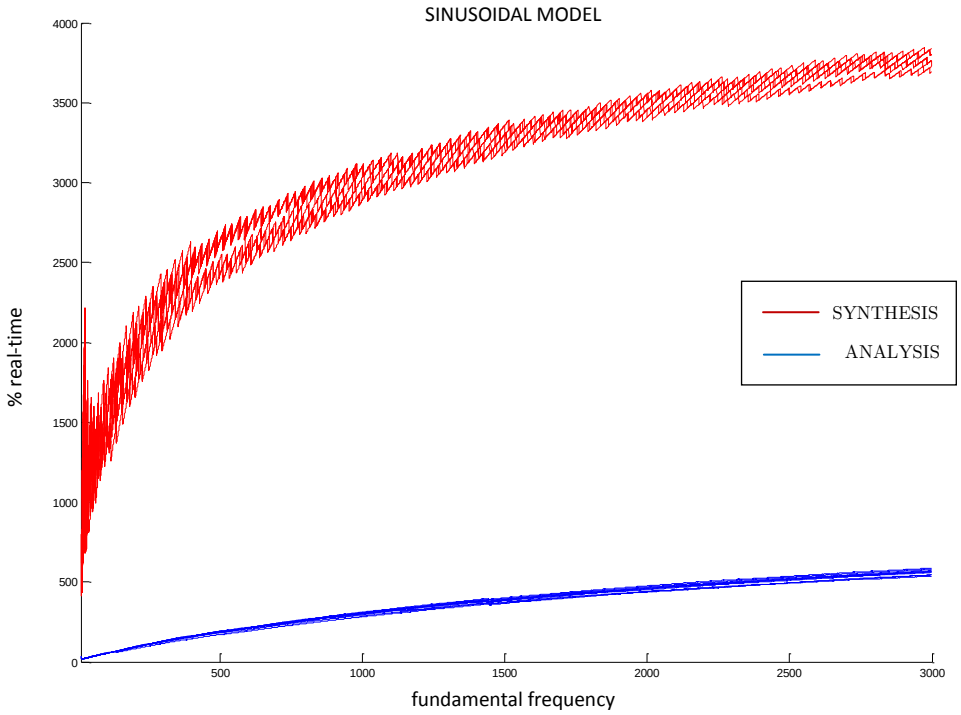


Figure 2.105 Percent of real-time versus fundamental frequency. The algorithm used is a sinusoidal model using a constant frame rate (non pitch-synchronous). The analysis window is set to cover three periods. The input signal is a sweep from 13 to 3000Hz and has been processed ten times.

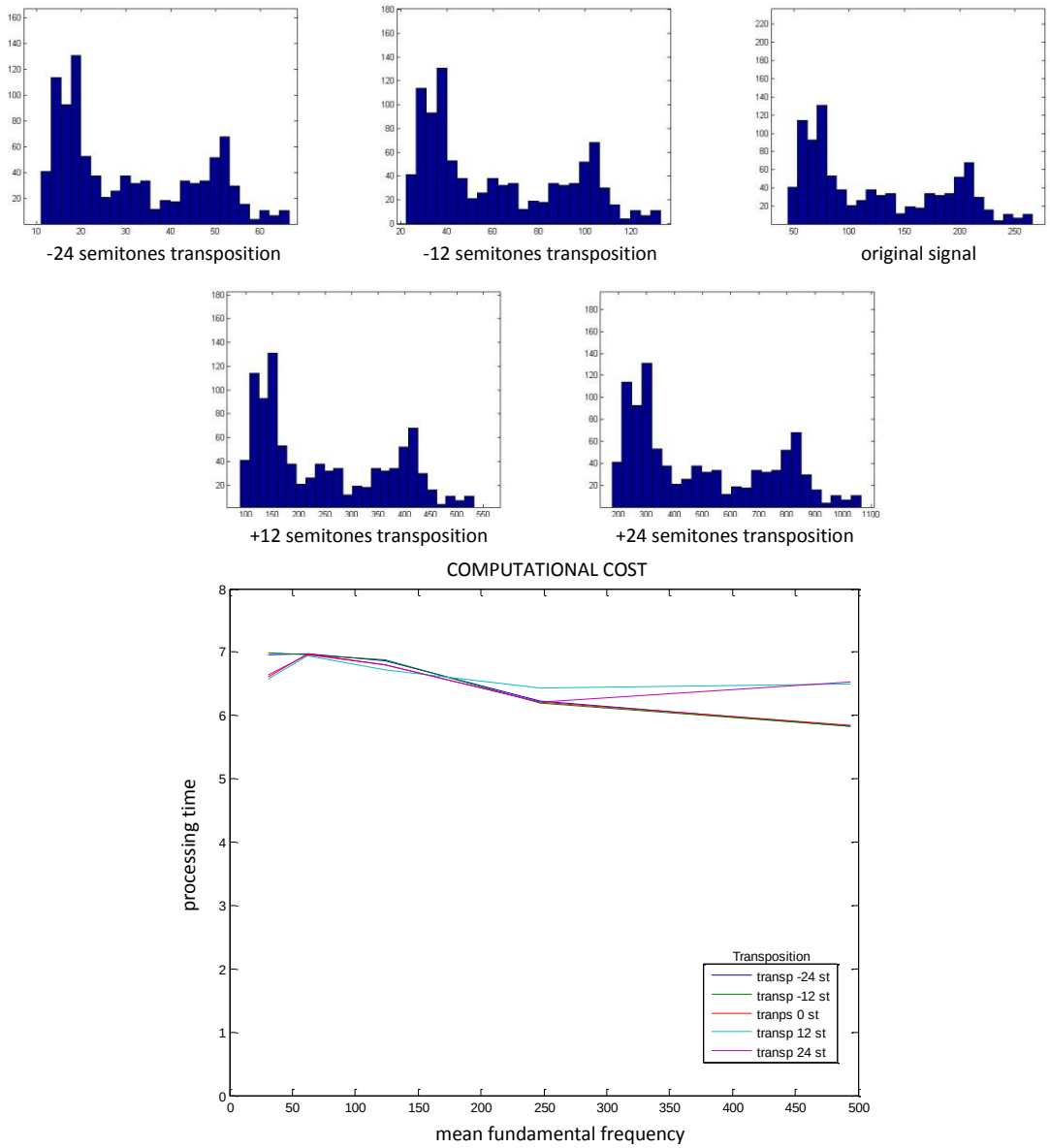


Figure 2.106 WBVPM processing time versus mean fundamental frequency. The input files are a low pitch male utterance and its -2, -1, +1 and +2 octave transpositions. Each of this input signals are processed and different pitch transpositions are applied. The resulting processing times last between 5.8 and 6.9 seconds. The top views show the fundamental frequency histograms of the original signal and its transposed versions.

2.5 Spectral Voice Model

In previous sections, we have discussed different techniques for processing voice signals, grouped into the ones that model frequency components and the ones that model voice pulses. The timbre representation was limited to be an interpolation of the harmonic amplitude values. However, there are some known intrinsic characteristics of the human voice that, if considered, might enhance the timbre representation and our knowledge about the phase relationship between harmonics and, consequently, improve also the control and quality of transformations. Traditionally, the voice has been modeled as a linear system consisting of one or more sound sources and a set of filters that shape the spectrum of those sources. The sound source can be a periodic signal, a noisy signal, or a mixture of both, and the set of filters can be regarded as the vocal tract filters. The resulting spectrum is mainly characterized by resonant peaks called formants. Thus, a vocal processor should provide means of controlling the resonant peaks of the spectrum. A timbre parameterization based on resonances is by far more natural and intuitive than a low-level interpolation of harmonics, and provides means of directly controlling subtle phonetic modifications.

In this section, we present a spectral voice model of the human voice that models the spectral envelope with a decay slope, several resonances and a residual envelope. We show that the proposed timbre parameterization is able to reconstruct perfectly all the nuances of the original singer's spectral envelope, and provides means for performing high-level voice transformations. In addition, we detail a phase model able to predict harmonic phases at voice pulse onsets.

2.5.1 Modeling the Magnitude Envelope

The EpR[24] Voice Model (Bonada, Loscos and Cano, et al. 2001) is based on an extension of the well-known source/filter approach (D. G. Childers 1994). It models the magnitude spectral envelope defined by the harmonic spectral peaks of the singer's spectrum. It consists of three filters in cascade. The first filter models the voice source frequency response with an exponential curve plus one resonance. The second one models the vocal tract with a vector of resonances that emulate the voice formants. The last filter stores the amplitude differences between the two previous filters and the original harmonic envelope. Hence, EpR can perfectly reproduce all the nuances of the harmonic envelope. Next, we describe in detail each of these filters.

◊ *EpR SOURCE FILTER*

The voice source is modeled as a frequency domain curve plus one resonance. It roughly approximates the voice timbre, here defined as the spectral envelope determined by the harmonics, i.e.

$$S_{harm}(f) = I_{\{f_h, a_h\}_{h=0...H-1}}(f) \tag{2.102}$$

where S_{harm} is the spectral envelope, h is the harmonic index, H is the number of harmonics, f_h and a_h are respectively the frequency and amplitude of the h^{th} harmonic, and I is a function that interpolates the frequency and amplitude values of the harmonics.

The EpR source curve (see Figure 2.107) is defined in a dB scale by a gain and an exponential decay as follows

$$Source_{dB}(f) = Gain_{dB} + SlopeDepth_{dB}\left(e^{Slope \cdot f} - 1\right) \tag{2.103}$$

where the *Gain*, *Slope* and *SlopeDepth* values are obtained from a linear regression of the harmonic peaks in the logarithmic frequency spectrum.

[24] EpR stands for Excitation plus Resonances.

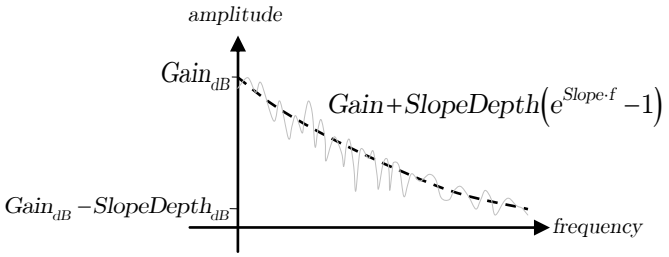


Figure 2.107 EpR source curve.

On top of the source curve, we add a resonance in order to model the low frequency content of the spectrum, below the first formant, especially in low pitch utterances. This is illustrated in Figure 2.108. This resonance does not affect the phase in the same way as the vocal tract resonances; it does not add a phase shift to the phase envelope at voice pulse onsets, as can be observed in Figure 2.111. The reason is that it is actually not a resonance in the strict sense of the word; it does not model a resonance of a tube. It just models the shape of the source spectrum at low frequencies.

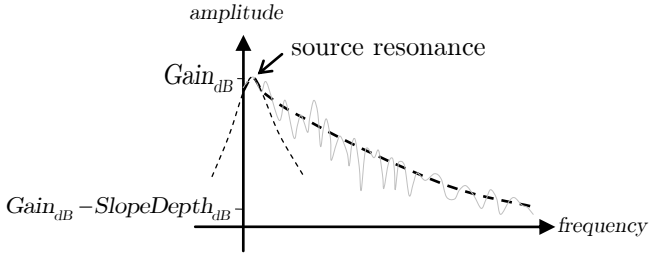


Figure 2.108 EpR source resonance.

The source resonance is modeled as a symmetric second order filter (based on the Klatt formant synthesizer (Klatt 1980)) with center frequency R_f, bandwidth R_{bw} and linear amplitude R_a. The transfer function of the resonance $R(f)$ can be expressed as follows

$$\begin{aligned} H(z) &= \frac{A}{1-Bz^{-1}-Cz^{-2}} \\ R(f) &= R_a \frac{H\left(e^{j2\pi\left(0.5+\frac{f-R_f}{f_s}\right)}\right)}{H\left(e^{j\pi}\right)} \end{aligned} \tag{2.104}$$

where

$$\begin{aligned} f_s &= \text{Sampling rate} \\ C &= -e^{\frac{-2\pi R_{bw}}{f_s}} \\ B &= 2\cos(\pi)e^{\frac{-\pi_{bw}}{f_s}} \\ A &= 1-B-C. \end{aligned} \tag{2.105}$$

The amplitude parameter R_a is relative to the source curve (a value of 1 means the resonance maximum is just over the source curve)

◊ *EpR VOCAL TRACT FILTER*

The EpR vocal tract filter is modeled by a set of M resonances $\{R_1,...R_M\}$ plus a residual envelope. Each resonance R_i is modeled as a symmetric second order filter, in the same way as the source filter (see eq. (2.104)). These resonances, represented in Figure 2.109, model the vocal tract formants.

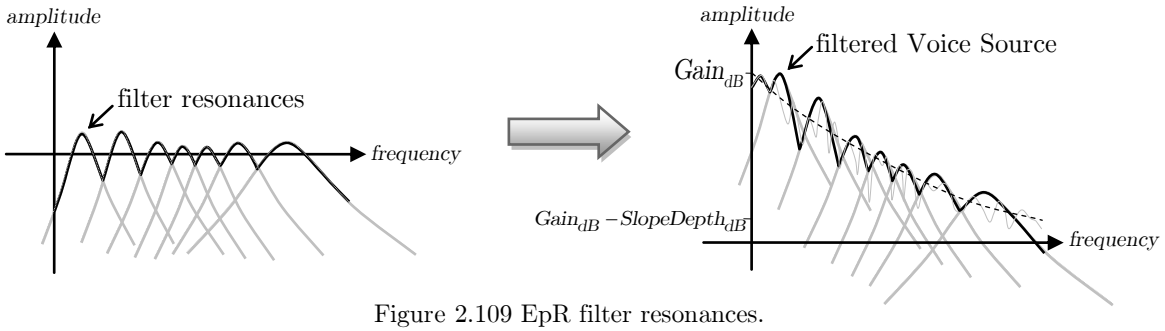


Figure 2.109 EpR filter resonances.

If the underlying processing algorithm allows independent modifications of harmonic and residual components, it makes sense to use different spectral shape models for each of them. In EpR two models are considered for harmonic and residual components that share the same representation and just differ in the gain and slope depth parameters. Those values can be approximated for instance from the harmonic and residual spectral envelopes obtained with an SMS analysis. Both filters are represented in Figure 2.110.

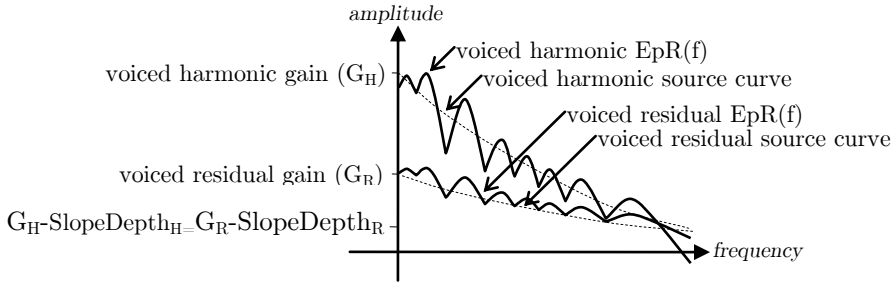


Figure 2.110 EpR residual envelopes.

The source plus resonances model produces smooth spectra that lack the richness and details of real voice spectra. Neither the voice source is a curve, neither the vocal tract an ideal tube. Moreover, not only resonances shape the voice source, but also antiresonances, i.e. frequency regions in which the amplitudes of the voice source are attenuated. These are especially present in nasal sounds because nasal cavities absorb energy from the sound wave. Hence, we propose to add a residual to the voice model to recover the nuances and details of the original singer's spectrum. The residual envelope $S_{residual_{dB}}$ actually stores the differences in dB between the source plus resonance model and the original harmonic spectral shape S_{harm} of the singer's performance. For efficiency, we only consider the differences at harmonic frequencies. If we denote the source resonance as $R_0(f)$, we can write

$$S_{residual_{dB}}(f) = I_{\left\{f_h,\, 20\log_{10}(S_{harm}(f_h)) - Source_{dB}(f) - 20\log_{10}\left(\sum_{i=0}^{M} R_i(f)\right)\right\}_{h=0\ldots H-1}}(f). \qquad (2.106)$$

Adding this residual to the EpR model, the harmonic envelope spectra is perfectly reconstructed.

Figure 2.111 shows the EpR model estimation of a sustained vowel recording. (b) displays the source curve as a dashed pink line, the source resonance in light green, the filter resonances in white, the harmonic envelope as red points, and the EpR envelope without residual in dark green, together with the harmonic predicted amplitudes as small red squares, in this case on top of the EpR envelope. In turn, (c) shows the same view but it includes the residual envelope as well. The source curve plus residual envelope is drawn in yellow color, i.e. a zero residual would fall on top of the source curve. The EpR envelope including the residual is drawn as before in dark green, with the harmonics drawn on top. Note that it perfectly overlaps the true harmonic envelope. The bottom view (d) shows the amplitude and phase spectra obtained using the window drawn in (a) on top of the waveform. The analysis window center is close to a voice pulse onset. Hence, we can appreciate a relative flat phase spectrum with shifts around each formant frequency. Note that the source resonance does not produce a shift in the phase spectrum.

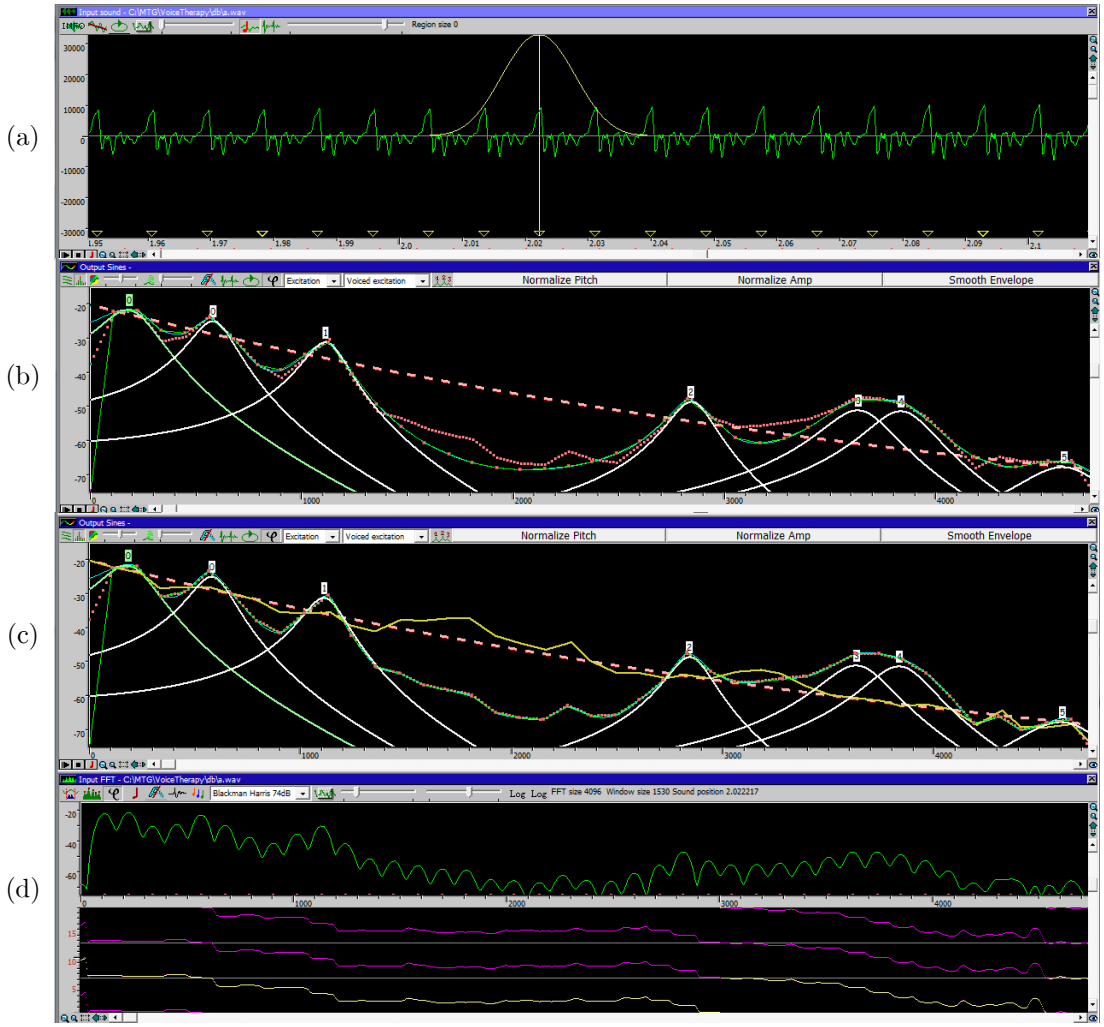


Figure 2.111 Estimated EpR voice model of a sustained Spanish /a/ vowel. (a) shows the waveform and the window used to perform the analysis. In addition, the MFPA predicted voice pulse onsets are drawn as inverted yellow triangles. In (b) we see the estimated source and filter resonances in light green and white respectively. (c) includes as well the residual envelope in yellow. (d) shows the amplitude and phase spectra.

Figure 2.112 shows all the filters together and illustrates how the harmonic spectrum is gradually filled. The resulting EpR spectral envelope is computed as the sum in dB of all three filters by

$$EpR_{dB}(f) = Gain_{dB} + SlopeDepth_{dB}\left(e^{Slope \cdot f} - 1\right) + 20\log_{10}\left(\sum_{i=0}^{M} R_i(f)\right) + S_{residual_{dB}}(f). \quad (2.107)$$

◊ *EpR TRANSFORMATIONS*

EpR aims at providing means for transforming voice sounds using perceptually relevant dimensions. Most of its parameters are related to significant voice features. The gain of the source filter is related to energy, slope and slope depth parameters to the amount of high frequencies and therefore to brightness. Resonances parameters are related to phonetics and personality. Residual envelope contains low-level data and helps to increase naturalness providing nuances and details missing in the smooth resonance model.

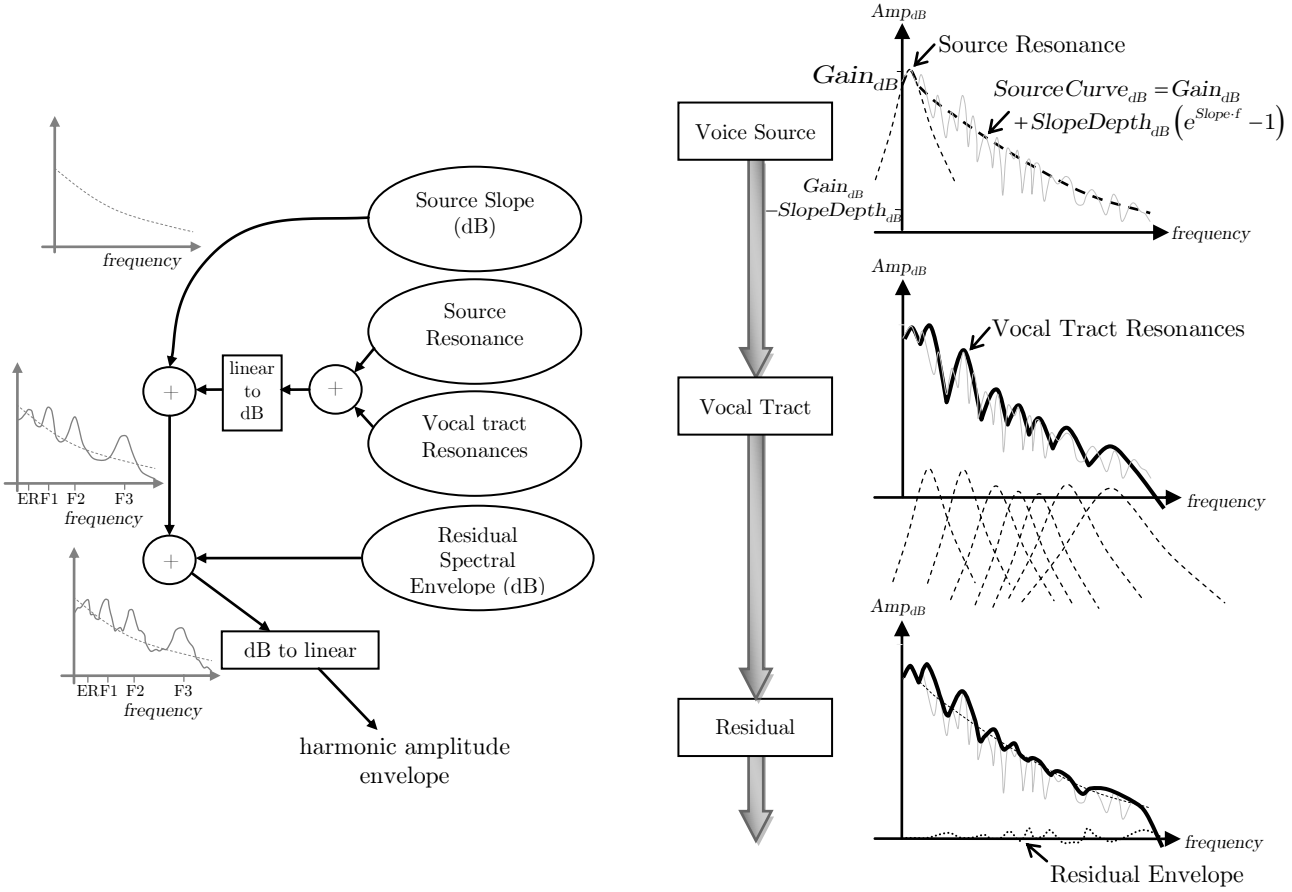


Figure 2.112 EpR Voice Model step by step.

Each of the parameters of the model can be controlled independently. However, residual envelope is intimately coupled with formants. If a formant is properly estimated, then generally we expect the formant to predict the local spectral shape quite well, and the residual envelope to have low values around the formant center frequency. This is observed in Figure 2.111. In the second view (b) the harmonic envelope (red dashed line) departs from the resonance model envelope (dark green line) between resonances. In (c) the residual envelope (draw in yellow color on top of the source curve) is close to zero around each of the resonances. In order to keep this link between resonances and residual, whenever a formant is shifted in frequency, the residual filter envelope is scaled accordingly taking as anchor points the formant frequencies. This way, the local amplitude spectral shape around each formant is preserved, and residual regions between formants might be compressed or stretched. One example is illustrated in the bottom view of Figure 2.113. Moreover, potential anti-resonances located between resonances are modified together with the residual, although we have no control on their bandwidth and center frequency, which change depending on whether the region is compressed or stretched.

A clear example of the benefits of a resonance model is found when two frames are interpolated. This is a common operation in concatenative synthesizers. Interpolating harmonic envelopes often flattens the amplitude spectrum. By contrast, using the resonance model, formants do shift in amplitude and frequency as expected. Figure 2.113 illustrates this behavior in the top view. On the left, we see the two frames that are interpolated. The main difference is that the first formant has increased its frequency. The result obtained by linear interpolation of harmonic envelopes is shown on the right at the top. The first formant seems to split into two flatter resonances, producing a flatter spectrum. By contrast, just below it is shown the result obtained with the resonance model. The first formant is clearly visible and it has actually shifted in frequency and amplitude.

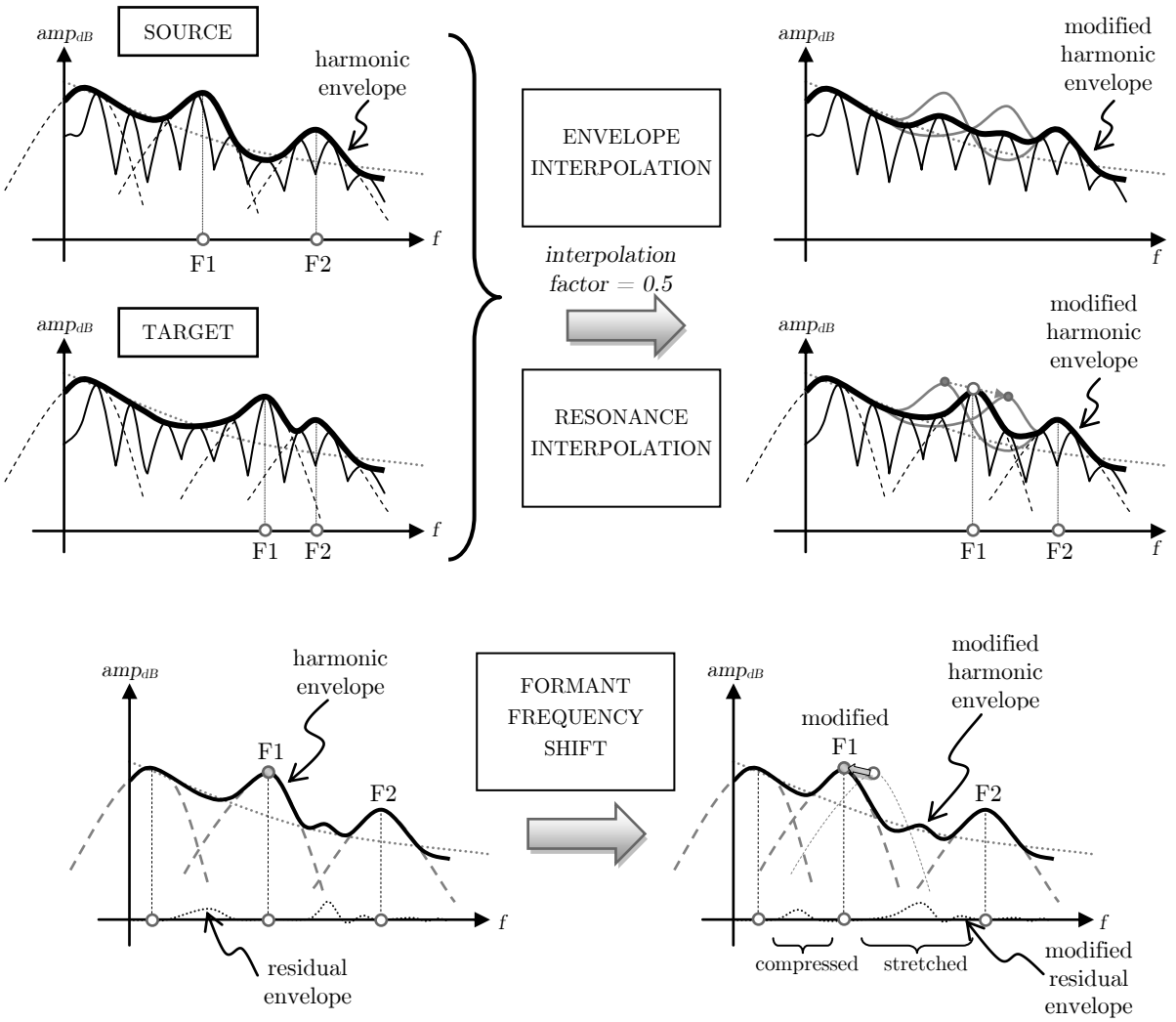


Figure 2.113 EpR transformations. The top view illustrates the interpolation between two frames. The main difference between them is that the first formant has increased its frequency. Interpolating the envelopes generates a nearly flat amplitude envelope that lacks the first formant. By contrast, the resonance model successfully shifts the formant in frequency and amplitude. The bottom view shows how when a resonance is shifted in frequency the residual envelope is scaled using as anchor points the resonance frequencies. In the end, some residual regions are compressed and others stretched.

The resonance model also improves the results in the typical case of adding vibrato to a flat utterance (see Figure 2.114). Vibrato adds frequency modulation to the fundamental frequency, and therefore harmonics shift in frequency in both directions. This implies interpolating input harmonic amplitudes to obtain the values at new harmonic frequencies. With the EpR model this interpolation is performed by first estimating the parameters of the model, and then computing the amplitude envelope values at target frequencies. In that case, we are assuming the vocal tract filter is not affected by the vibrato. The goodness of the resonance model is that the vocal tract actually has resonances, and therefore the continuous envelope obtained with the EpR model predicts quite well the local characteristics around formants, i.e. around the peaks of the envelope where energy concentrates, thus the frequency bands that are more perceptually relevant.

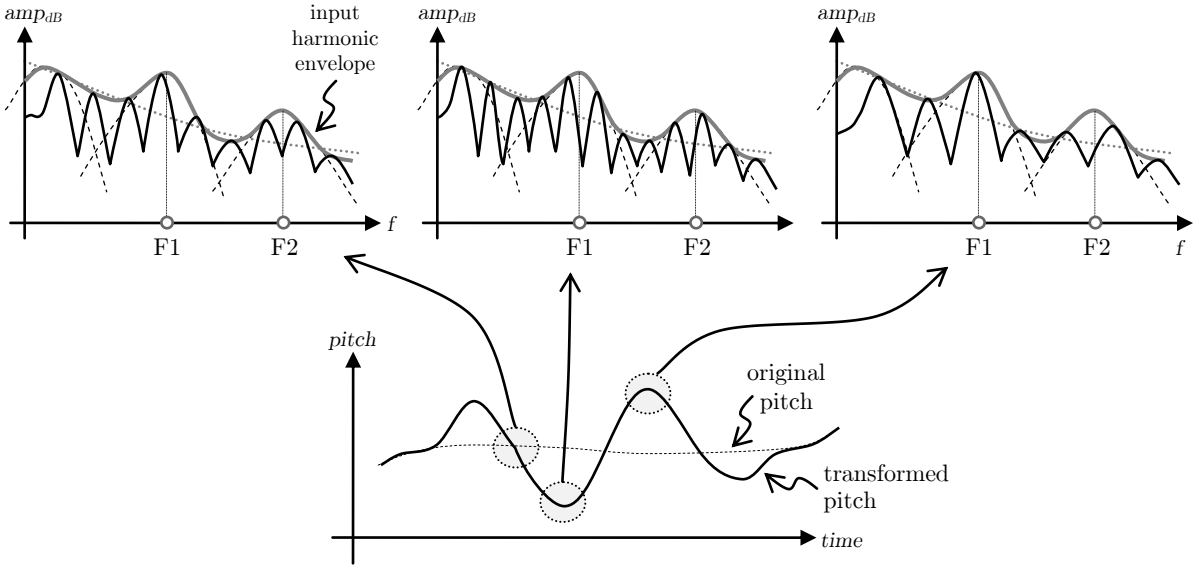


Figure 2.114 Vibrato transformation. Frequency modulation is added to the pitch of a nearly flat utterance. The synthesis harmonic amplitudes are obtained from the EpR envelope.

2.5.2 Modeling the Phase Envelope

We are interested in modeling the harmonic phase envelope at voice pulse onsets. Our aim is not to reproduce perfectly the phase envelope but to generate a phase envelope that perceptually sounds natural and similar to the original one. Our observations of many voice harmonic spectra indicate that there is a strong relation between the harmonic amplitude envelope and the phase alignment at voice pulse onsets. In those instants, there is a strong correlation between formants and phase envelope. The abrupt closure of the vocal folds often produces a prominent excitation to the vocal tract, an impulse that has minimal phase characteristics. In the scope of the source-filter model, this excitation is filtered by the vocal tract. If the vocal tract is represented with resonances, then each of those resonances affect the phase of the filter impulse response in different ways depending on its parameters (amplitude, bandwidth, frequency) and the surrounding resonances.

One might think of estimating the harmonic phase envelope out of the resonance model. However, this would require a very robust resonance estimation model, because if a given resonance appears and disappears in consecutive frames or abruptly changes its parameters, than discontinuities will emerge in the resulting phase envelope causing audible artifacts. Furthermore, resonances should match formants. Otherwise, modeling a formant with several resonances would produce too large phase shifts and distort the resulting audio quality. It is certainly difficult to build a robust voice formant estimator. Actually, one of the main difficulties of using a formant model to predict the phase envelope is that it is a discrete model that has to take *binary* decisions regarding the presence of formants, i.e. decide if a formant is present or not. Nevertheless, what about if we consider directly using the harmonic envelope instead of the formant model? There are well known methods that avoid typical problems due to the presence of noisy or masked partials when computing the harmonic amplitude envelope. For instance, the true envelope method (Röbel and Rodet 2005) is one of those. Hence, the robustness of such estimator would probably have much less impact in the results than that of a formant estimator.

Following those ideas, we first tried to find a relation between the spectral amplitude and phase envelopes. Combining scaling, shifting and offsets modifications, we could generate phase envelopes

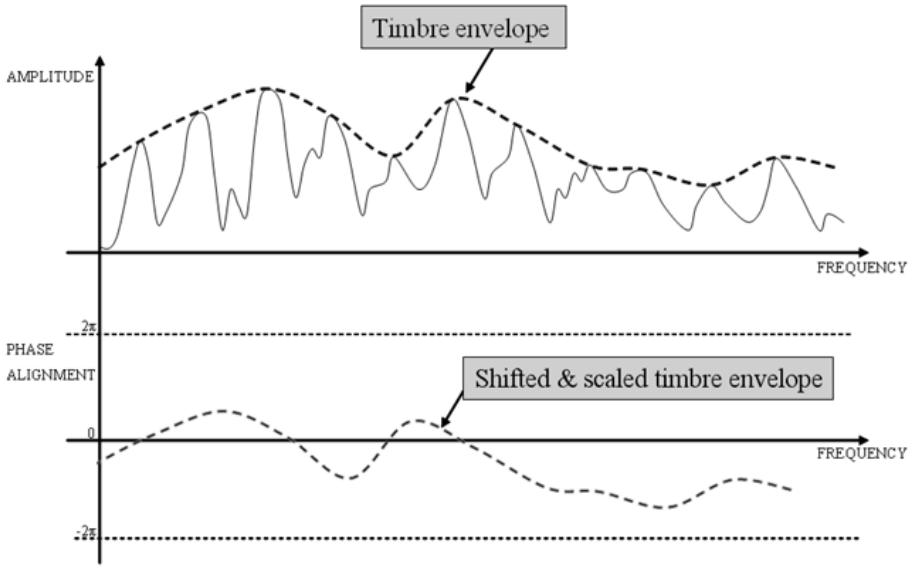


Figure 2.115 Phase alignment computation out of the spectral envelope by applying scaling, shifting and offset modifications. In the example above the envelope is shifted to the left so that the fall in the right side of formants becomes a fall in the phase envelope just around the formant center frequency.

that featured a phase shift around amplitude peaks (i.e. resonances[25]). One example is shown in Figure 2.115. However, after performing several experiments we could not find a setup with which produce convincing results for a wide variety of voices. Next, we explored the possibility of using the amplitude derivative. This approach worked much better from the very beginning. Actually, just scaling the amplitude derivative gives a good approximation of the phase envelope at low frequencies. The computation is performed as follows

$$\hat{\phi}_h = \alpha 20\log_{10}\left(\frac{a_{h+1}}{a_h}\right) \qquad for\, h = 0...H-2 \tag{2.108}$$

where $\hat{\phi}_h$ is the predicted phase for the h^{th} harmonic, α and is a scaling factor and H is the number of harmonics. $\hat{\phi}_{H-1}$ was set equal to $\hat{\phi}_{H-2}$. We found in our experiments that $\alpha = \pi/19$ is a good choice. It means that 19dB of amplitude difference between consecutive harmonics corresponds to π radians. Let us consider Figure 2.116 in detail. The top view represents several periods of waveform of the input signal. The middle view shows both the narrow and wide-band discrete-time STFT of the above signal. Obviously, visible spectral peaks belong to the narrow-band spectrum. Note that adding an offset to the wide-band spectrum we would get a good approximation of the harmonic amplitude envelope. The wide-band analysis is performed with the window function drawn in the top view. Note also that the window is centered close to the voice pulse onset. The bottom view contains the phase of the spectral bins obtained with the wide-band analysis drawn as stars. It is a good approximation of the harmonic phase envelope. The interesting story is that the derivative of the wide-band spectral amplitude envelope drawn in the bottom view (solid line) resembles quite well the spectral phase. Clearly, between 250 and 3500Hz, both envelopes have a similar behavior. Actually, the function represented is the smoothed derivative plus an offset of 4 radians. The smoothing is performed with a zero-delay running average filter of 5 coefficients. In the case of computing the phase not from the wide-band spectrum but from the harmonics, the smoothing is computed as

[25] Since we are considering here the envelope obtained by interpolating the harmonic spectral peaks, then a peak in such envelope probably corresponds to a formant or resonance.

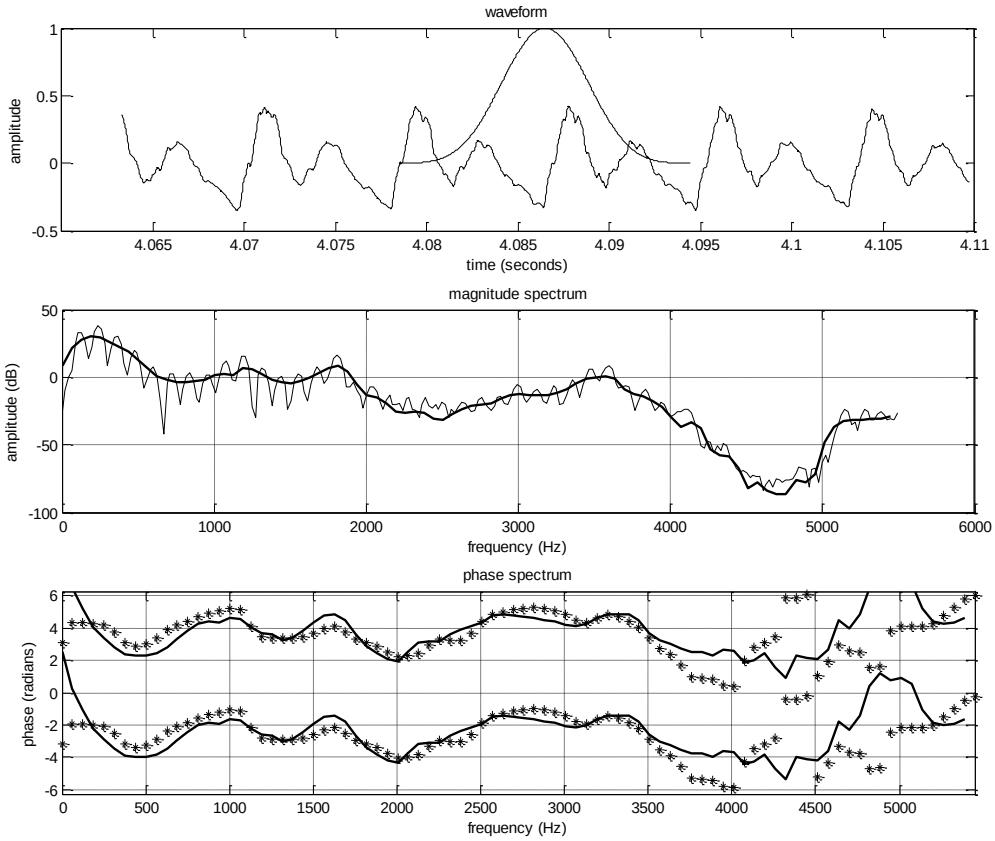


Figure 2.116 Phase envelope obtained from the wide-band spectral amplitude derivative. The top view shows the waveform of the input signal. The middle view both narrow and wide-band amplitude spectra. In the bottom view are represented the wide-band phase spectrum (stars) and the derivative of the wide-band amplitude spectrum (solid line) scaled by $\pi/10$ (rad/dB) and with an offset of 4 radians.

$$\hat{\phi}_{h,smoothed} = \phi_M + \frac{1}{o} \sum_{k=h-\frac{o-1}{2}}^{h+\frac{o-1}{2}} \hat{\phi}_{\max(0,\min(H-1,k))} \qquad for\, h = 0 \ldots H-1 \tag{2.109}$$

where o is the order of the filter (odd) and ϕ_M the phase offset. Note that applying sound transformations requires estimating the phase model out of the synthesis harmonics, not the input ones. The previous equations is then rewritten as

$$\hat{\phi}'_{h,smoothed} = \phi'_M + \frac{1}{o} \sum_{k=h-\frac{o-1}{2}}^{h+\frac{o-1}{2}} \hat{\phi}'_{\max(0,\min(H'-1,k))} \qquad for\, h = 0 \ldots H'-1. \tag{2.110}$$

In order to test the phase model, we have resynthesized a small database of voice recordings using the techniques detailed in §2.2.5, concretely rendering harmonics as spectral regions, but using the phase model to predict harmonic phases at voice pulse onsets. The database consists of

- 5 male speech
- 3 female speech
- 8 male singing
- 7 female singing

Singing examples belong to different styles (jazz, soul, dance, blues, pop, scat) and include several expressive resources (vibrato, scoop, glissando, growl). The database contains more male than female

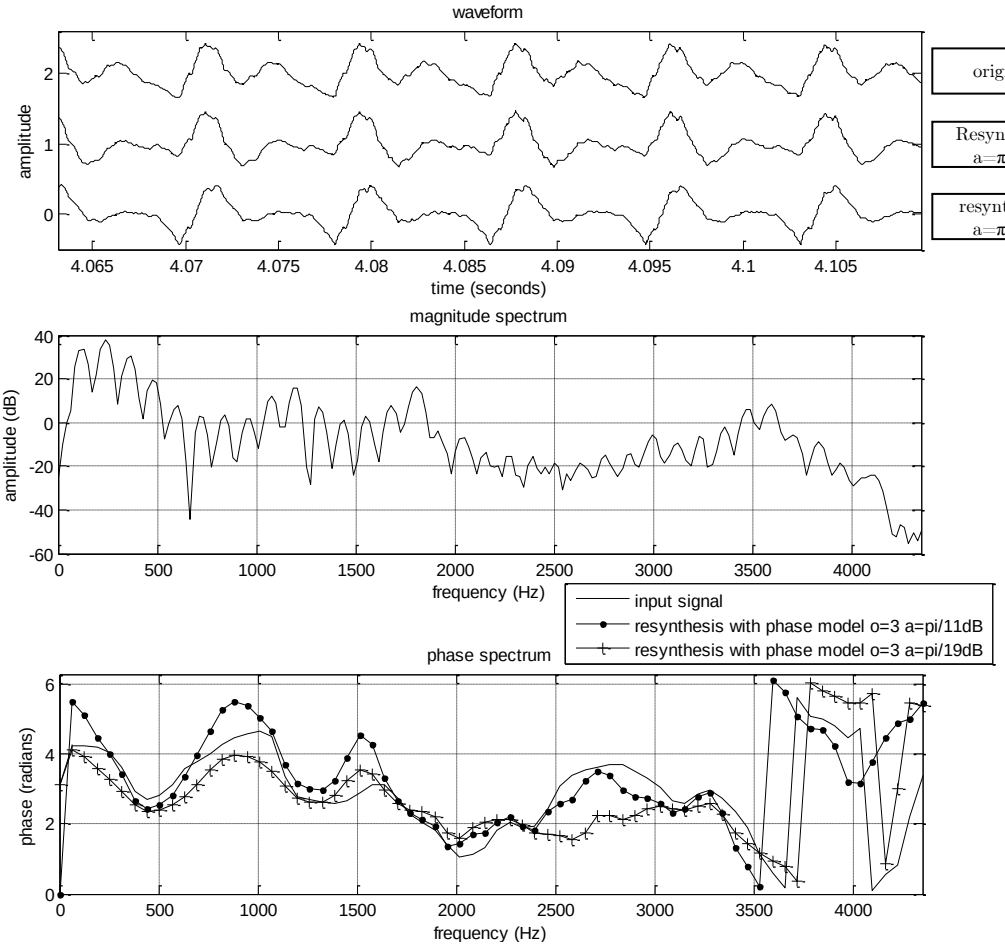


Figure 2.117 Comparison of original audio and two resynthesis using the proposed phase model. In the top view are represented the waveforms. The middle view shows the amplitude spectrum of the original signal. The bottom view displays the wide-band phase spectra of all three signals.

examples, because the perception of phase relation between harmonics (e.g. phasiness) becomes more evident at lower pitches. The audio files were sampled at 44.1Khz and quantized to 16 bits.

In order to set the parameters of the phase model, we did some preliminary tests. We processed a low pitch male speech recording (audio [111]) with a scaling value of $\alpha=\pi/19$, a phase offset of $\phi_M=0$ and several smoothing orders $o=\{1,3,5\}$. We found that phasiness was perceived for $o=1$. Note that actually no smoothing is applied in that case. According to our judgment $o=5$ and $o=3$ sounded very similar, but $o=5$ was a little bit less lively. Then, we processed again the same audio with $\alpha=\pi/19$, $o=3$ and phase offset values $\phi_M=\{-3,-2.5,...,3,3.5\}$. We found that the feeling of clarity and sharpness changed gradually, being $\phi_M=-2$ and $\phi_M=0.5$ the offsets that produced the best and worse results respectively. Next, we processed the same audio with fix values for phase offset $\phi_M=-2$ and smoothing order $o=3$, and played around with different scaling factors between $\alpha=\pi/7$ and $\alpha=\pi/21$. We found that the clarity and sharpness varied from less to more between $\alpha=\pi/7$ and $\alpha=\pi/21$. Figure 2.117 shows the results for one frame of the audio file. In the top view, from top to down, we have represented the input waveform and two resynthesis with scaling factors $\alpha=\pi/11$ and $\alpha=\pi/19$ respectively. The middle view contains the narrow-band spectrum of the input signal. The bottom view shows the wide-band phase spectra of the input and the two resynthesis signals. We can appreciate that, compared to $\alpha=\pi/19$, the predicted phase for $\alpha=\pi/11$ has a greater phase excursion around first and second formants. In other words, for $\alpha=\pi/11$ the predicted phase envelope is more sensitive to harmonic amplitude changes.

Then, we processed another male speech audio with $\phi_M=-2$, $o=3$ and $\alpha=\pi/19$, and we noticed that some voice consonants sounded strange. Carefully inspecting the problem, we found out

that when the voicing frequency is low, then middle and high frequencies contain predominantly noise components and their amplitude envelope is used to predict their phase. This might produce unnatural phase alignments of the noisy components at each pulse onset that result in time domain amplitude modulations of the pitch rate. Something similar happens in low-quality recordings with high frequency noise. One way of improving the results would be to use a voicing frequency detector and process differently the noisy frequency band. However, we tried a simpler approach that produced good enough results according to our judgment. We added a sinusoid to the predicted harmonics for frequencies higher than 8Khz as follows

$$\hat{\phi}'_{h,smoothed} = \phi'_M + \frac{1}{o}\sum_{k=h-\frac{o-1}{2}}^{h+\frac{o-1}{2}} \hat{\phi}'_{\max(0,\min(H'-1,k))} + \begin{cases} 0 & if \quad h < D \\ \pi \sin\left(2\pi \dfrac{h-D}{E}\right) & if \quad h >= D \end{cases} \qquad for\, h = 0 \ldots H'-1 \qquad (2.111)$$

where D is the index of the first harmonic above 8Khz and E is a constant that sets the rate of the sinusoid. $E=35$ was considered to be a good trade-off. This has the effect of avoiding flat phase synchronizations of partials above such frequency when the spectral envelope is nearly flat. Those values were found empirically.

The processing of the database audio files was performed with a hop-size of 256 samples and a window of 2049 samples. The phase model was generated as previously described, from the synthesis harmonic amplitude values, with parameters $o=3$, $\phi_M=-2$, $\alpha=\pi/19$ and the proposed phase correction. Table 2.7 contains the list of audio files, their corresponding descriptions and the references of processed and generated audios. In addition to resynthesis, we have included two transformations combining pitch transposition and timbre scaling. We also generated the transformed audio files without phase model for comparison.

Informal listens by the author indicate that the signals resynthesized using the phase model sound very similar to the original ones. Almost no degradation in naturalness or intelligibility has been found. In some examples we could perceive a subtle loss of clarity or presence. However, no significant or annoying phasiness has been perceived. Downwards transpositions, especially in the case of low pitches, sound better using the phase model. Transformations sound nearer, sharper, with less phasiness (e.g. audios [114], [174], [138], [132] and [216] compared to [113], [173], [137], [131] and [215]). In general, transformed sounds tend to sound softer without the phase model. We have found that in some upwards pitch transpositions some artifacts can be listened when the phase model is not used, probably due to errors in the harmonic phase continuation, for instance audio ref. [133] between 1 and 1.5 seconds compared to [134]. The same happens in audio [229] between 2 and 4 seconds ([230] with phase model). Also in [157] most of the artifacts disappear when using the phase model [158]. We explained in §3.2 the complexities of transforming harmonic phases in a shape invariant framework. Audio [201] is an interesting example. It is an excerpt of a performance of a unique and very famous singer with a very characteristic voice: Louis Armstrong. The proposed phase model is able to reproduce perfectly the typical roughness of his voice. In addition, the growl occurring in audios [189] and [195] is also perfectly reproduced.

Using a phase harmonic model has several advantages. One of the most evident is data compression, since there is no need to store harmonic phases. However, in the context of this research, probably the main advantage is found when concatenating voice segments, a typical operation of a singing voice synthesizer or a Text-to-Speech system. Amplitude and phase discontinuities often appear between segment boundaries, producing artifacts in the synthesized signal, and several techniques have been developed in order to minimize those discontinuities. Using the phase model proposed here, we can focus exclusively on the harmonic amplitude continuity and forget about phase discontinuities. This is a great simplification and, if the phase model is good enough, probably leads to a quality improvement of the synthesis results. Probably the proposed phase model has also applications in the field of voice enhancement. If harmonic phases cannot be reliably estimated, then it makes sense use predicted phases instead. In addition, some of the processing methods presented in this chapter that require to estimate voice pulse onsets could be adapted to use the phase model. Probably this would increase their robustness against onset estimation errors, and even might lead to simplifying them.

audio	signal description	re-synthesis	transformation $T_{pitch}=0.7$ $T_{timbre}=0.95$		transformation $T_{pitch}=1.25$ $T_{timbre}=1.05$	
			no phase model	*phase model*	*no phase model*	*phase model*
[111]	male speech, low pitch	[112]	[113]	[114]	[115]	[116]
[117]	male speech	[118]	[119]	[120]	[121]	[122]
[123]	male speech, highly processed, very low pitch	[124]	[125]	[126]	[127]	[128]
[129]	male speech, low pitch	[130]	[131]	[132]	[133]	[134]
[135]	male speech, low quality	[136]	[137]	[138]	[139]	[140]
[141]	female speech, storyteller	[142]	[143]	[144]	[145]	[146]
[147]	female speech, noise in the recording	[148]	[149]	[150]	[151]	[152]
[153]	female speech, very expressive	[154]	[155]	[156]	[157]	[158]
[159]	male singing, blues style, very low pitch	[160]	[161]	[162]	[163]	[164]
[165]	male singing, jazz style	[166]	[167]	[168]	[169]	[170]
[171]	male singing, scat	[172]	[173]	[174]	[175]	[176]
[177]	male singing, dance style	[178]	[179]	[180]	[181]	[182]
[183]	male singing, dance style	[184]	[185]	[186]	[187]	[188]
[189]	male singing with growl	[190]	[191]	[192]	[193]	[194]
[195]	male singing with growl	[196]	[197]	[198]	[199]	[200]
[201]	male singing, Louis Armstrong, voice over music	[202]	[203]	[204]	[205]	[206]
[207]	female singing, arpeggio with vibrato	[208]	[209]	[210]	[211]	[212]
[213]	female singing, jazz style	[214]	[215]	[216]	[217]	[218]
[219]	female singing, flat singing	[220]	[221]	[222]	[223]	[224]
[225]	female singing, pop style	[226]	[227]	[228]	[229]	[230]
[231]	female singing, dance style	[232]	[233]	[234]	[235]	[236]
[237]	female singing, soul style	[238]	[239]	[240]	[241]	[242]
[243]	female singing, pop style	[244]	[245]	[246]	[247]	[248]

Table 2.7 This table contains the audio references of a small database of voice recordings, a brief description of each file, and references to resynthesized and transformed audio files with and without the phase model.

2.6 Conclusions

In this chapter, we have discussed the most relevant problems found when processing voice signals. We have shown that voice utterances can be interpreted as a set of time-varying frequency components but also. Each interpretation leads to different processing techniques, which have been discussed in depth along the chapter. The first ones are the processing methods based on modeling harmonic trajectories. We have overviewed the main approaches to estimate and transform those trajectories. We have also justified the use of shape invariant techniques for achieving high quality voice transformations, and shown that results improve when harmonic phase envelopes are preserved at Glottal Closure Instants (CGIs). In addition, we have proposed and evaluated a novel method (MFPA) for approximating GCIs. We have then proposed some methods capable of generating irregular pulse sequences by adding subharmonics, and shown their potential to emulate expressive effects such as growls. Finally, we have detailed the issues involved in harmonic trajectories synthesis,

and proposed improvements for the case where harmonics are synthesized by transforming spectral regions.

The interpretation of voice utterances as a sequence of filtered time-domain voice pulses leads to the processing techniques based on modeling voice pulses. We have introduced two novel methods that work in narrow and wide band conditions, NBVPM and WBVPM respectively, and discussed the advantages and drawbacks of each case. WBVPM models voice pulses in frequency domain with a set of sinusoids, which represent both harmonic and noisy components. This technique provides enough temporal resolution to transform independently each of the voice pulses while at the same time it provides an independent control of each harmonic component. In this sense, WBVPM combines some of the main pros of both time and frequency-domain methods while avoids some of their main drawbacks. We have implemented the proposed techniques in C++ and integrated them into a wide range of applications, most running in real-time, that include museum installations, videogames, and professional audio effects plug-ins for voice manipulation.

In addition, we have introduced spectral models specially devised for the human voice. The first one is EpR, an amplitude spectral voice model that makes use of the source-filter decomposition and represents the vocal tract filter with resonances. The addition of a residual envelope allows reconstructing perfectly the original singer's spectrum with all its details and nuances. The second spectral model deals with harmonic phases and it is able to predict the harmonic phase relationship at voice pulse onsets without significantly altering the perceived timbre characteristics. This phase model simplifies significantly the concatenation of samples, as it will be shown in the next chapter.

Chapter 3

Singing Synthesis by Performance Sampling

Our aim is to build a singing voice synthesizer that captures the sonority of a specific singer. By sonority we mean the sound quality but also the behavioral characteristics of the singer. We assume that we have access to the individual for recording him performing predefined vocal exercises. Our task then is to decide what should be recorded and learn how to build models that are able to recreate his predominant characteristics. At the same time, we are interested into generating a virtual voice that sounds as natural as possible. We have studied in Chapter 2 different techniques for processing voice signals. Now it is time to use and adapt them to our specific target. For that purpose, we will use the approach of concatenative synthesis (Schwarz 2007). The main idea is to connect snippets of the singer recordings, and modify each of them so to recreate the target performance. Part of our efforts will focus on connecting them in a transparent way, avoiding discontinuities and unnatural transitions, intending to help hiding the fact that the system is concatenating samples and increasing the sensation of a continuous flow. In addition, transformations must include expressive resources. As we will show, most of them can be implemented in the form of parameterized templates and be obtained from singer performances.

In this chapter we will first introduce the concept of sonic space and performance sampling. We will emphasize the fact that we are actually sampling the combination of a performer and an instrument, even in the case of the singing voice. Then we will explain how to define and build a performance database. Next, we will explore the requirements of a singing performer model and explain how to create performance trajectories within the sonic space that recreate the target performance. We will then detail the aspects involved in transforming performance trajectories into sound. Finally, we will evaluate the results obtained and compare them to the state-of-the art and to real singer performances.

Part of the work presented here has been carried out in the scope of a singing voice synthesizer we have been developing in collaboration with Yamaha Corp. since 2000. This implies that some of our decisions have been determined by practical considerations related to building a synthesizer prototype in the context of a potential commercialization.

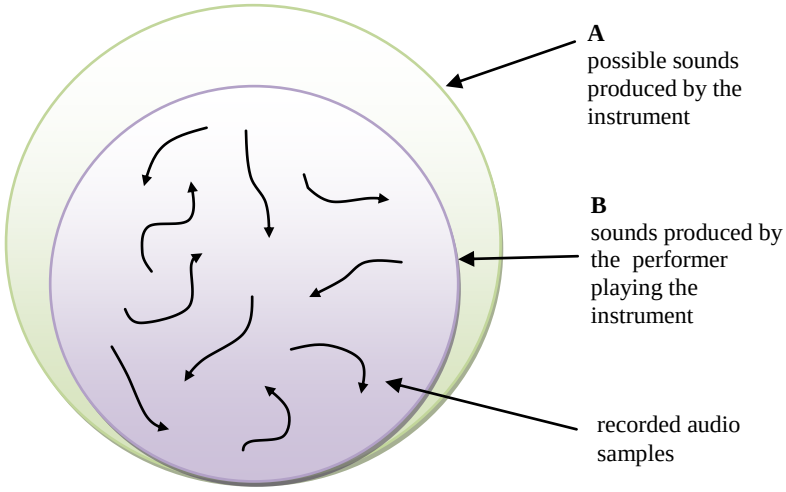


Figure 3.1 Instrument sonic space

3.1 Sampling the Sonic Space

Sampling has always been considered a way to capture and reproduce the sound of an instrument but in fact it should be better considered a way to model the sonic space produced by a performer with an instrument. This is not just a fine distinction; it is a significant conceptual shift of the goal to be achieved.

We want to model the sonic space of a performer/instrument combination. This does not mean that the synthesizer shouldn't be controlled by a performer; it just means that we want to be flexible in the choice of input controllers and be able to use high-level controls, such as a traditional music score, or to include lower-level controls if they are available, thus taking advantage from a smearing of the traditional separation between performer and instrument.

Figure 3.1 shows a visual representation of a given sonic space to be modeled. The space A represents the sounds that a given instrument can produce by any means. The space B is the subset of the space A that a given performer can produce by playing that instrument. The trajectories shown in the space B represent the actual recordings that have been sampled. The reality is that this sonic space is an infinite multidimensional one but we hope to be able to get away by approximating it with a finite space. The trajectories represent paths in this multidimensional space. The size of these spaces may vary depending on the perceptually relevant degrees of freedom in a given instrument/performer combination, thus we could say that a performed drum can be represented by a smaller space than a performed violin. This is a very different concept than the traditional timbre space; here the space is defined both by the sound itself and by the control exerted on the instrument by the performer. Thus the sound space of an accomplished performer would be bigger (and musically more interesting) than the space of a not so skilled one.

From a given sampled sonic space and the appropriate input controls, the synthesis engine should be able to generate any trajectory within the space, thus producing any sound contained in it. The brute force approach is to do an extensive sampling of the space and perform simple interpolations to move around it. In the case of the singing voice, the space is so huge and complex that this approach is far from being able to cover a large enough portion of the space. Thus, the singing voice is a clear example that the basic sampling approach is not adequate and that a parameterization of the sounds is required. We have to understand the relevant dimensions of the space and we need to find a sound parameterization with which we can move around these dimensions by interpolating or transforming existing samples.

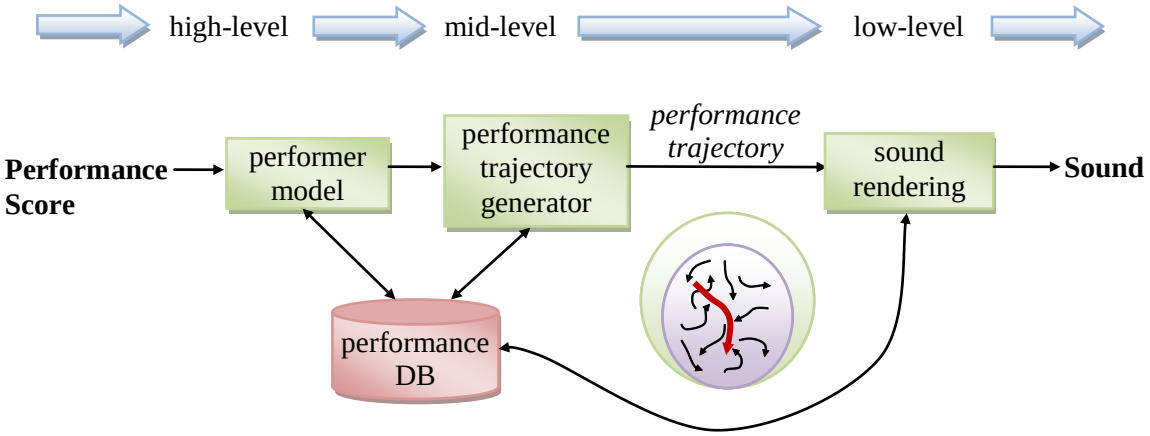


Figure 3.2 Basic modules of a performance based sampling synthesizer.

3.1.1 A Performance based Sampling Synthesizer

The basic modules of a performance based sampling synthesizer are represented in Figure 3.2. The input of the system is a generalization of the traditional score, a *Performance Score*, which can include any symbolic information that might be required for controlling the synthesizer. Generally, the performance score contains high-level controls that the *Performer Model* converts into lower level performance actions. The *Performance Trajectory Generator* creates the parameter trajectories that express the appropriate paths to move within the dimensions of the sonic space. The *Sound Rendering* module is the actual synthesis engine that produces the output sound by concatenating a sequence of transformed samples that approximate the target performance trajectory. The *Performance Database* is not restricted to performance samples but can include also models and measurements that relate to the performance space and that give relevant information to help in the process of going from the high level score representation to the output sound.

Nevertheless, the system admits other configurations. The user of the system can replace the functions of the *Performer Model* and fill the score with the appropriate expressive controls, such as for instance detailed note timings, musical articulations or dynamic envelopes. Moreover, a potential user of the system could replace as well P*erformance Trajectory Generator*. A typical case would be that of an instrument performer who, with the help of sensors, captures the parameters of his performance and inputs them to the synthesizer.

From another perspective, we consider the input of the system as a high-level type of control, and that each step in the synthesizer generates lower and lower level controls up to the lowest one, which is the synthesized sound.

In next sections we will present in more detail each of these components for the specific case of a singing voice synthesizer. Nevertheless, we consider that most of the ideas and technologies presented here can be applied to any performer-instrument combination. In fact, the author is involved in research aiming at building a violin synthesizer based on these concepts (Pérez, Bonada, et al., Combining Performance Actions with Spectral Models for Violin Sound Transformation 2007). However, before detailing each of the modules, we will first introduce a vowel synthesizer that uses the basic ideas on top of a sinusoidal signal model.

3.1.2 An additive vowel synthesizer

In this section we present an additive vowel synthesizer, which we consider a proof of concept of the capabilities of both the proposed synthesizer framework and the EpR spectral voice model. Following the overall aim of this thesis research of reproducing the voice characteristics of a specific individual, we will first record different vowels sung by a male singer. Then we will estimate the voice

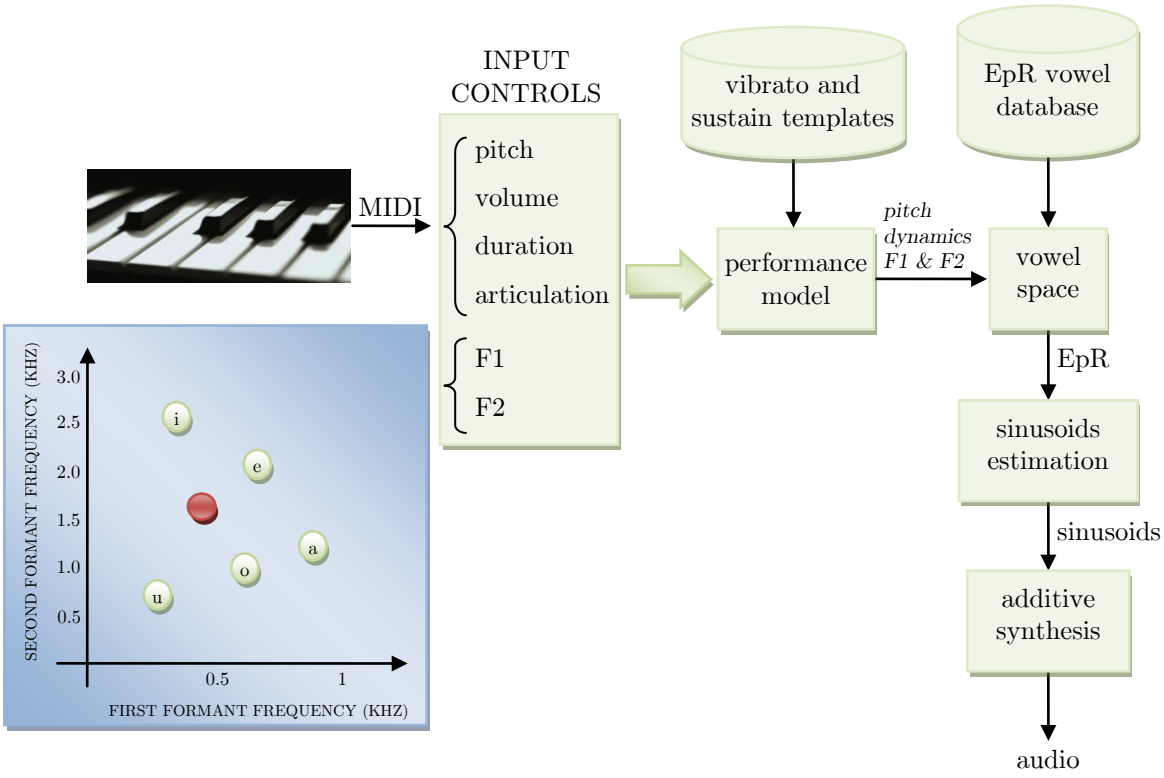


Figure 3.3 Block diagram of the additive vowel synthesizer based on the EpR voice model.

model parameters for each of them, and finally we will build an additive sinusoidal synthesizer where the amplitude and phase of each sinusoid is determined by the voice model.

◊ *OVERVIEW*

The inputs of the synthesizer are divided into melody, expression and vowel quality controls. The melody is specified as a sequence of notes with pitch and volume. Expression controls consists on articulation duration and vibrato parameters. Vowel quality controls set first and second formant frequencies. The vowel space is stored in a database containing the EpR model estimated from different vowels sung at several pitches and dynamics.

The *performance model* computes out of the input controls the instantaneous values for fundamental frequency, dynamics, and formant frequencies. The *vowel space* module receives the instantaneous controls, fetches the suitable EpR models from the database and interpolates them obtaining the EpR model to be synthesized. This is fed to the *sinusoids estimation* module that generates a vector of sinusoids with amplitude, frequency and phase parameters. Finally, the *additive synthesis* module renders the sinusoids and computes the resulting audio. Figure 3.3 shows a block diagram of the proposed synthesizer.

◊ *RECORDING VOWELS*

Our aim is to record enough audio information so to estimate afterwards the EpR parameters for different vowels, pitches and dynamics. In order to do so, we designed a simple script consisting on singing each vowel with crescendos at different pitches, in an ascendant arpeggio with pauses between each note. The crescendos go from the lowest to the highest dynamics, while the pitches cover the singer's tessitura. The five Spanish vowels (/a/, /e/, /i/, /o/, /u/) were recorded by a male amateur singer.

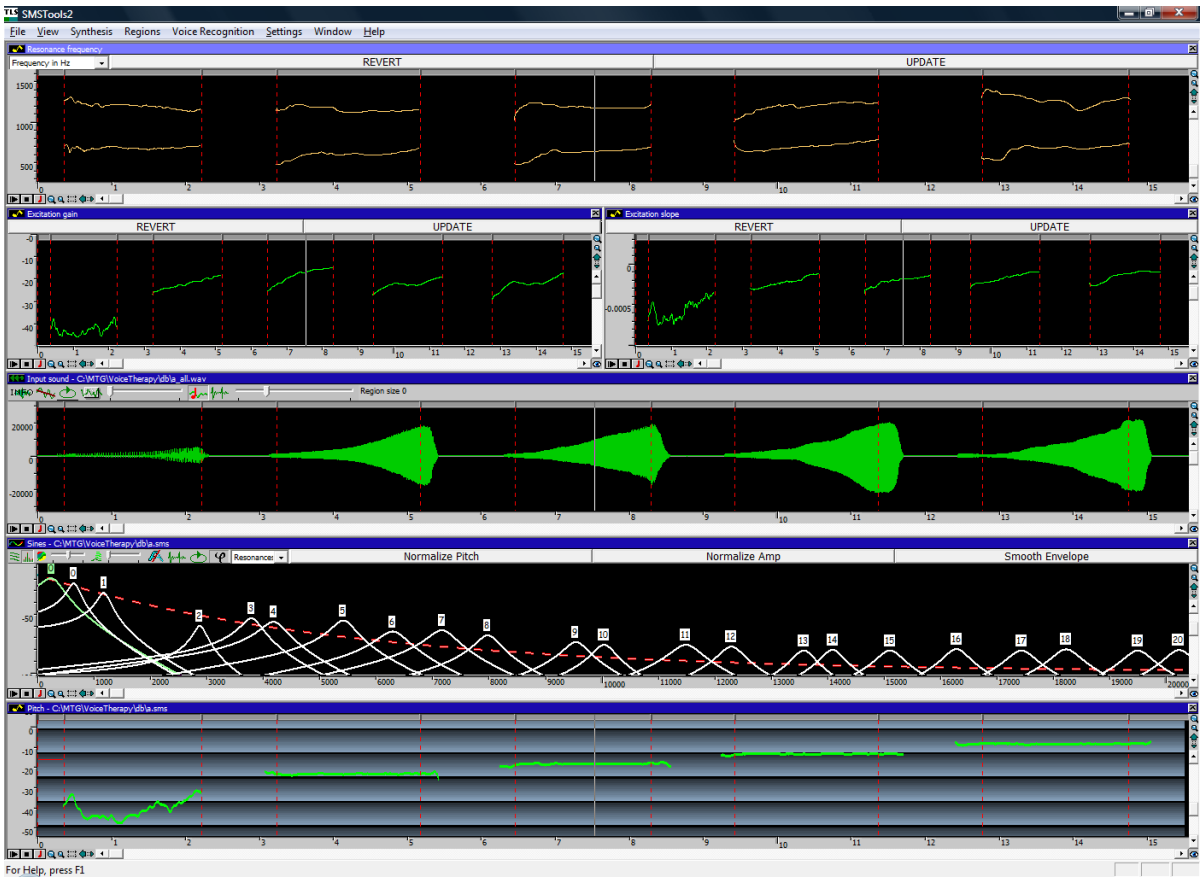


Figure 3.4 Vowel synthesizer database for Spanish /a/ vowel.

◊ *PERFORMANCE MODEL*

The performance model computes the fundamental frequency, dynamics, and formant frequency values. The pitch function is generated from the MIDI data generated by the notes pressed in the keyboard. Since the system works in real-time, some delay is added to avoid discontinuities. Whenever a new note onset is detected after a pause, the synthesis pitch value is assigned to that note reference frequency. If another note onset is received before the current note is released, then the pitch is slowly interpolated between both note frequencies in cents, producing a legato transition. If a short pause is found between two consecutive notes, then the pitch is not interpolated and a staccato transition is synthesized. Dynamics are generated in a similar way using a mapping function between MIDI note velocity and dynamics values. Moreover, note attack, sustain and release envelope functions have been manually designed to improve the naturalness of the results. Formant frequencies at each time instant are obtained from the current position in the vowel space controlled by a user. If the user is moving fast within the vowel space, than the EpR parameters between consecutive synthesis frames might change significantly and produce discontinuity artifacts. We have added a low pass filter to the EpR parameters with the aim of minimizing those artifacts. Moreover, the performance model receives input data that control vibrato deep, rate and tremolo. Vibrato is generated by adding a sinusoid to the pitch function with the user set deep and rate. Some random behavior is added to both controls in order to produce natural sounding vibratos. Finally, in sustains, a pitch template is added to the pitch function. Such template has been obtained from a recording of a sustained vowel, and it is consecutively looped using random positions in long synthesis sustains.

◊ *MODELING THE VOWEL SPACE*

Each of the recorded utterances was analyzed. The fundamental frequency was estimated using a spectral amplitude correlation method. Then the harmonics were estimated searching for local

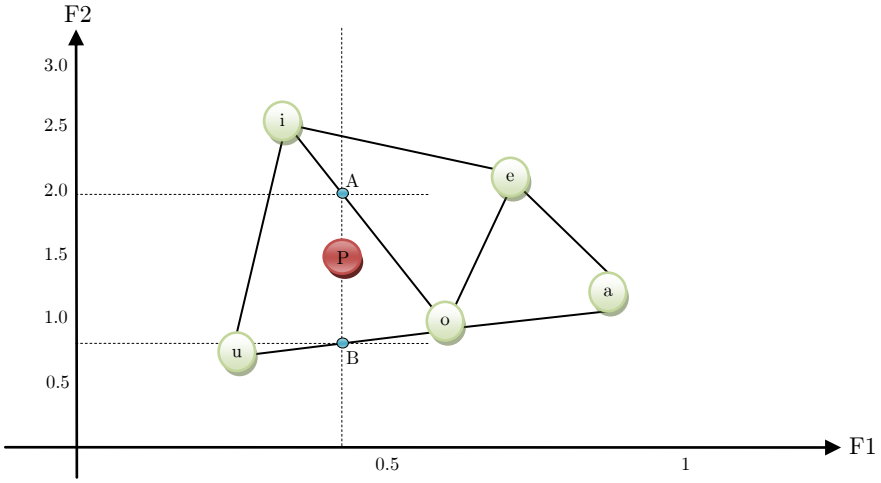


Figure 3.5 Vowel quality formant map. The five Spanish vowels /a/, /e/, /i/, /o/ and /u/ are placed at their estimated average formant frequencies. Three non overlapping triangles join all the vowels. The given point P is found to belong to the triangle formed by /u/, /o/ and /i/. Then, the voice model corresponding to points A and B is estimated by interpolating the three vowels using their first formant frequency as follows

$$A = /i/ + \frac{F1_P - F1_{/i/}}{F1_{/o/} - F1_{/i/}}(/o/ - /i/)$$

$$B = /u/ + \frac{F1_P - F1_{/u/}}{F1_{/o/} - F1_{/u/}}(/o/ - /u/)$$

Afterwards, the voice model at P is estimated by interpolating A and B voice models using their second formant frequency as follows

$$P = A + \frac{F2_P - F2_A}{F2_B - F2_A}(B - A)$$

maxima around each multiple of the estimated fundamental frequency. Finally, the EpR parameters were estimated out of the harmonic peaks.

Figure 3.4 shows a screenshot of the analysis tool we used. From bottom to top the data drawn corresponds to: pitch, EpR model at the selected time, waveform of the recorded performance, excitation gain (left) and slope (right), and finally first and second formant frequencies. The data was obtained from the analysis of the Spanish /a/ vowel recording. Each note is manually segmented, assigning begin and end positions to the lowest and maximum dynamics. The lowest note is sung with fry phonation. It is hard for most people to produce stable utterances in such phonation mode. That is why the estimated pitch looks unstable. Note that formant frequencies are not constant during crescendos, probably because the singer is opening his mouth. Nevertheless, the proposed approach should be able to reproduce such behavior.

Since EpR provides means for interpolating voice timbres in a natural way, we could situate each frame in a multidimensional space and then estimate the voice model at an arbitrary location by interpolating the closer frames. However, for simplification, we decided to represent all frames of the same vowel in a single location in a formant map, using as coordinates the average of estimated first and second formant frequency values. Then, given a vowel, we assigned all the frames of the same note to their average pitch. Finally, given a vowel and a pitch, we assigned the dynamic value of each frame by linearly interpolating the normalized time position within the note (i.e. $t_n \in [0,1]$) to the whole dynamic range (i.e. from *pppp* to *ffff*).

This way, we set the order of dimensions to interpolate as follows: (1) vowel quality, (2) pitch, (3) dynamics. Therefore, given the target parameters, we approximated the voice model as a weighted sum of three vowels, depending on their location within the formant map. Then, for each of the

vowels considered in the weighted sum, we approximated their voice model as the linear interpolation between the notes with closest pitches. Finally, for each of the used notes, their voice model was approximated as the linear interpolation between the frames with closest dynamics. Summing up, the target voice model was approximated as a weighted sum of the estimated EpR of 3*2*2=12 frames.

$$\begin{aligned} &\textit{VOWEL QUALITY} \quad EpR_{target} = \frac{\sum_v \omega_v V_v}{\sum_v \omega_v} \\ &\textit{PITCH} \qquad V_v = note_{v,n} \cdot \left(1 - intp_{v,n}\right) + intp_{v,n} \cdot note_{v,n+1} \\ &\textit{DYNAMICS} \qquad note_{v,n} = dyn_{v,n,d} \cdot \left(1 - intp_{v,n,d}\right) + intp_{v,n,d} \cdot dyn_{v,n,d+1} \end{aligned} \tag{3.1}$$

Only three vowels are used in the vowel quality interpolation for optimization reasons. In order to avoid discontinuities when one of the three chosen vowels change for consecutive frames, we designed the following method. We created a net of non-overlapping triangles having the vowels as vertices. Then, for a given target location, we computed which triangle it belonged to[26]. The weights of each of the three vertices were computed by firstly interpolating in the horizontal axis (first formant frequency) and secondly in the vertical axis (second formant frequency). One example is shown in Figure 3.5, where the EpR parameters corresponding to P are computed.

◊ *SINUSOIDS ESTIMATION*

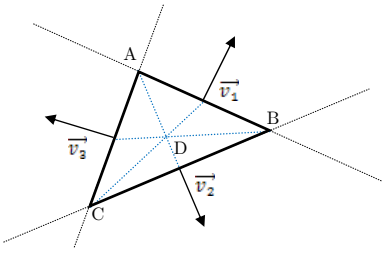
Once we have estimated the EpR voice model parameters at the target location, we have to estimate the sinusoids to be synthesized. Given the target pitch, the frequency of each sinusoid (i.e. harmonic) is set to be a multiple of the fundamental frequency

$$f_h = \left(h+1\right) f_0 \qquad \text{for } h \in 0 \ldots H-1 \tag{3.2}$$

where H is the number of harmonics. Then, the amplitude and phase of each sinusoid is estimated using the EpR voice model as detailed in previous sections.

---

[26] One way of determining if an arbitrary point is located within a triangle is by means of basic geometric operations such as vector scalar products. For a given triangle with vertices $A = \left(A_x, A_y\right)$, B and C, we first compute three vectors $\vec{v}_1$, $\vec{v}_2$ and $\vec{v}_3$, not necessary unit vectors but perpendicular to each of the sides of the triangles and pointing outside. The scalar product of two vectors $\vec{a} = \left(a_x, a_y\right)$ and $\vec{c} = \left(c_x, c_y\right)$ is defined as $\vec{a} \cdot \vec{c} = \left|\vec{a}\right|\left|\vec{c}\right| \cos\left(\alpha\right) = a_x c_x + a_y c_y$. If they are perpendicular, the cosine will be zero, and so the scalar product. For example, vectors $\vec{b} = \left(-a_y, a_x\right)$ and $\vec{d} = \left(a_y, -a_x\right)$ would be perpendicular to $\vec{a}$ but would point to opposite directions. From this, we can derive that $\vec{v}_1$ might be either $\left(-A_y, A_x\right)$ or $\left(A_y, -A_x\right)$. In order to determine the correct direction, we can use a known point inside the triangle, for example the barycenter (point D in the figure), the intersection of the medians (i.e. lines joining each vertex with the midpoint of the opposite side, drawn in blue in the figure). It is computed as $D = \left(A+B+C\right)/3$. Since vectors $\vec{v}_1$ and $\overrightarrow{AD}$ form an angle bigger than $\pi/2$, their scalar product will be negative. This way we can determine the correct direction for $\vec{v}_1$. Following the same procedure for each side of the triangle, we compute the three perpendicular vectors. Finally, given an arbitrary point P, if it is inside the triangle it must fulfill that $\overrightarrow{AP} \cdot \vec{v}_1 \leq 0$, $\overrightarrow{BP} \cdot \vec{v}_2 \leq 0$ and $\overrightarrow{CP} \cdot \vec{v}_3 \leq 0$, because all the angles will be bigger or equal than $\pi/2$.



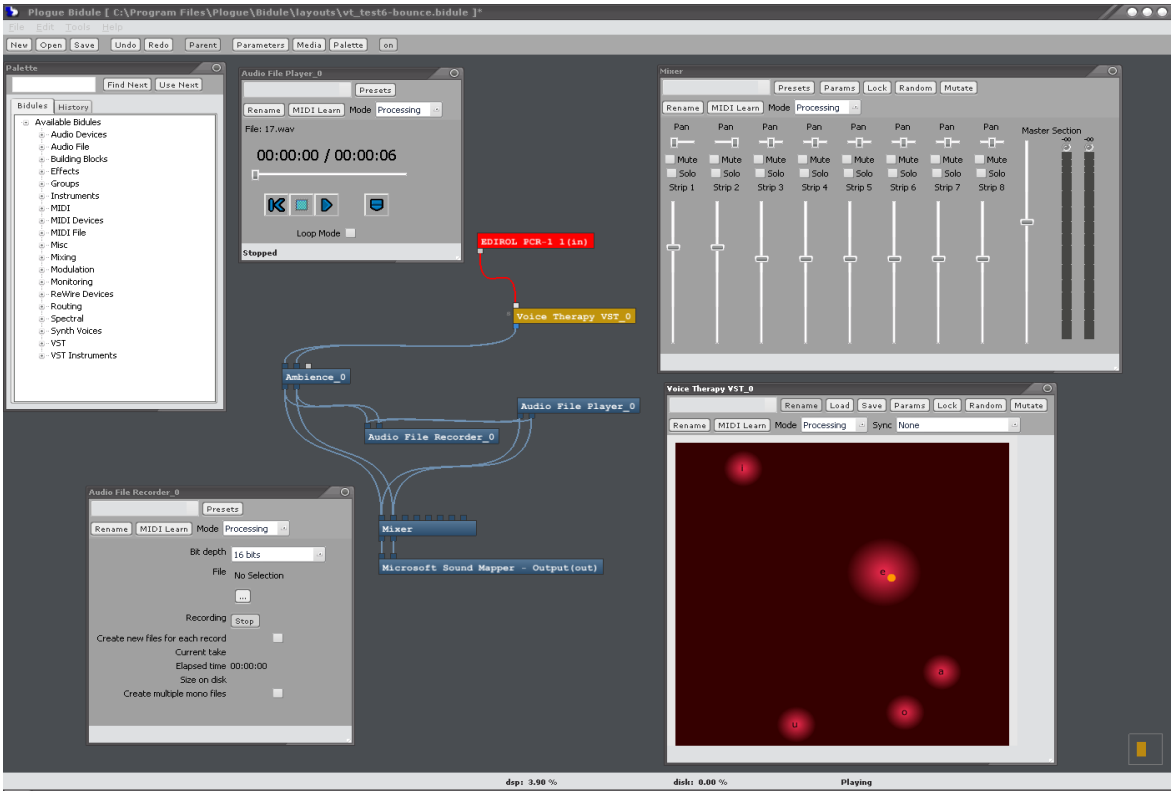


Figure 3.6 Vowel synthesizer VST Instrument plug-in interface

◊ *ADDITIVE SYNTHESIS*

The last step is the additive synthesis that converts the vector of input sinusoids into synthetic audio. For optimization reasons, instead of using a bank of time domain oscillators we decided to adopt the ideas from (Depalle and Rodet 1990), to render each sinusoid in the spectrum by convolving the main lobe of the synthesis window transform with the sinusoid transform (i.e. a delta). The Blackman-Harris 92dB window is a good choice because most of its energy concentrates in the main lobe and the rest of side lobes have amplitudes below -92 dB compared to the main lobe. Therefore, for achieving a good quality, it is enough to use the bins within the main lobe (9 bins without zero padding). Once all sinusoids have been rendered into the same spectrum, we apply an IFFT and get the windowed time domain data. Consecutive frames can be overlapped by dividing the windowed data by the synthesis window, and then multiplying it by an appropriate overlapping window that adds to a constant, in our case a triangular window.

◊ *RESULTS*

The proposed synthesizer has been implemented as a VST Instrument plug-in, and it is controlled in real-time by a MIDI keyboard and a graphical interface. In our experiments two people were involved in a performance, one playing notes with the keyboard and the other moving a mouse over a formant1 versus formant2 frequency plane on the computer screen. Figure 3.6 shows a screenshot of the plug-in interface. In the formant plane, vowels are located inside red circles, and the mouse position is drawn as a yellow point. When the mouse gets close to a vowel, its surrounding circle grows so to emphasize that such vowel contributes more to the timbre than the rest of vowels.

Audio [249] is an example of the results obtained combining two performances played simultaneously. Having in mind that this is a sinusoidal synthesizer results are really good, especially in mid and high frequencies. Also in some excerpts, the behavior sounds quite natural according to our judgment. We think the results are promising and a proof of concept of proposed synthesizer framework. On the other hand, we believe the sound quality would definitely improve if the voice

phase model from §2.5.2 was used. That is the first task for future work. Another task would be to add to the Performance Model expression templates obtained from real performances.

3.2 Performance Database

In this section, we propose a representation of the sonic space of the performer-instrument combination we are modeling, we define which are its relevant dimensions and set the appropriate sampling grid for capturing the necessary samples. In addition, we detail the steps required to build a database of those sample, always within the scope of a singing voice synthesizer.

3.2.1 Defining the sonic space

The sonic space is a multidimensional space that contains the collection of possible sounds produced by the performer-instrument combination. Ideally, its dimensions relate to significant perceptual characteristics. Our task then is to define those dimensions in the case of the singing voice. It should be mentioned, however, that in the scope of this research we must consider some of the practical aspects and limitations related to building a synthesizer prototype in the context of a potential commercialization.

In addition to the dimensions, we have to define the sampling grid, i.e. the positions or trajectories in the sonic space that we want to capture. This greatly depends on the probability of occurrence of each area in the sonic space. In other words, we want to have a better coverage of the most common voice utterances, and therefore increase the resolution of the sampling grid in those areas. However, the sampling grid also depends on which are the transformations we can apply to the samples. For each transformation, we should study the tradeoff between sound quality and transformation range, preferably in different contexts of the sonic space[27]. Besides, we should also consider which sample concatenation techniques will be used at synthesis, and find the consequent tradeoff between sample distance and natural sounding transitions.

Next, once we have set the sampling grid to be a good compromise for the previous tradeoffs, we have to come up with detailed scripts for recording each sample. In the context of our research, we want to minimize the score length and therefore maximize the density of database samples in the performance being recorded. These scripts should be as detailed as possible in order to avoid ambiguities and assure that the performance contains the target samples.

One of our goals is to capture typical performances occurring in the musical contexts in which we want to use the synthesizer. This has the advantage of actually providing the sound character and behavior we expect, so that with fewer transformations we should be able to generate appropriate synthesis results. This approach is being successfully applied in some synthesizers (e.g. (Lindemann 2007, Pérez, Bonada, et al., Combining Performance Actions with Spectral Models for Violin Sound Transformation 2007)). However, singing voice has some special characteristics that greatly affect how we face the task of designing those performances and, consequently, the sampling grid of the sonic space. Most musical instruments include acoustic resonators to enhance the sound of the instrument (e.g. sound boxes, pipes, etc). In general, those acoustic resonators exhibit approximately static resonances for the different performances of the same note[28]. One example is the violin. Its body is the acoustic resonator, and it is affordable to define a medium-size violin performance database including several bow strokes, dynamics and durations for each note out of excerpts of musical pieces. However, in the case of the singing voice, resonance frequencies are set by the vocal tract, a mobile structure constantly moving. Therefore, for different performances of the same note

[27] For example, we can think of a case where transpositions sound acceptable for the interval [-3,+4] semitones at low pitches, but the interval becomes [-2,+2] at high pitches

[28] Here we use *note* to denote the *musical note*. Therefore, we mean that the intended pitch is the same for all performances of the same note.

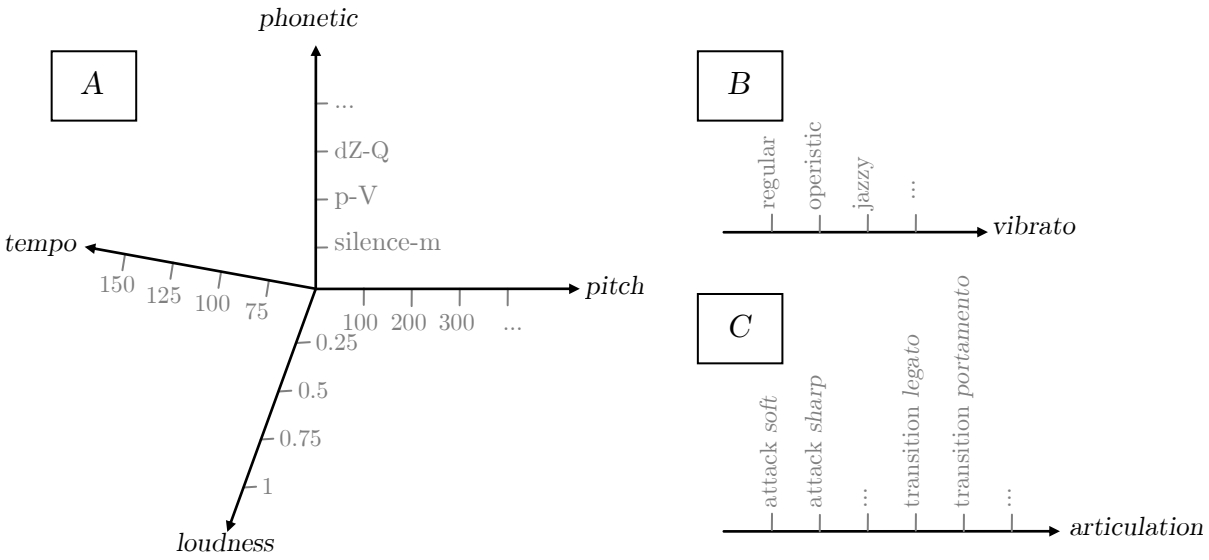


Figure 3.7 Proposed subspaces of the singing voice sonic space.

we find radically different resonance dispositions. In other words, each musical note can be sung with different phonemes, syllables or words. This implies a much larger performance database if we want to cover the whole phonetic space for each dynamic of each note. In addition, each voice has a distinct character with particular characteristics that distinguish one voice from other voices more clearly than what we find between different instruments of other types. Hence, if we want to build a singing voice synthesizer with perspectives of a posterior commercialization, we have to consider it is most likely that several voice databases will be created. Therefore, it is important to ease the process of creating a new voice and minimize the cost of database production, related to duration of the recording session in the studio, the size of the database, the number of supervised tasks to perform and so on.

Having considered all the previous issues, in the specific case of singing voice we propose to divide the sonic space into three subspaces *A*, *B* and *C*, each one with different dimensions, as illustrated in Figure 3.7. Subspace *A* contains the actual samples that are transformed and concatenated at synthesis. Subspaces *B* and *C* are one-dimensional and contain samples that once properly modeled specify how samples from *A* should be transformed to show a variety of specific expressions.

Subspace *A* has four dimensions: phonetic, tempo, loudness and pitch. The phonetic axis has a discrete scale defined as units of two allophones[29] combinations (i.e. di-allophones) plus sustained voiced allophones. We limit samples to combinations of two allophones for reducing the size of the database and the recording time, although using combinations of three or more allophones is in fact a common practice in concatenative speech synthesis (e.g. (Black 2002)). However, not all di-allophone combinations must be sampled but only a subset that statistically covers the most frequent combinations. Obviously, this study should be performed for each language. Firstly, we have to identify the phonetic units or allophones we want the synthesizer to handle, code them in a phonetic alphabet that the synthesizer understands, grouped by phonetic types, and build a phonetic dictionary. We use the SAMPA notation when available. For example, European Spanish has 33 allophones (Llisterri and Mariño 1993) and the phonetic dictionary we use is the one shown in Table 3.1. Then we analyze the phonetic transcription of a textual corpus and find the combinations required to cover most occurrences. On the other hand, we decided to include also sustained voiced

[29] An allophone is one of several speech sounds that belongs to the same phoneme; a phoneme is the smallest phonetic unit in a language that is capable of conveying a distinction in meaning.

category	allophones	examples
unvoiced plosives	p,t,k	pala, tala, cala
voiced plosives	b,d,g	vino, dar, gala
unvoiced affricates	tS,*ts*	chico, *quetzal*
voiced approximant	B,D,G	cabra, nada, luego
voiced affricates	dZ, *dl*	cónyuge, *náhuatl*
unvoiced fricatives	f,T,s,x,*h,C,S*	falso, zona, sala, jamón, *pasta, cojín, xocoyote*
voiced fricatives	jj,z,Z	ayer, desde, Yugo
nasals	m,n,J,N	mala, nada, caña, hongo
liquids	l,L,r,rr	lejos, caballo, caro, torre
semivowels	j,w	labio, agua
vowels	a,e,i,o,u	tal, tela, tila, todo, tul
other	Sil,Asp	

Table 3.1 Spanish phonetic dictionary. Latin American Spanish variants in italics.

allophones in the phonetic axis because in singing phoneme durations depend on the duration of the notes, and it is very common to find sustained vowels, liquids or nasals with durations much longer than in speech. We could instead record longer di-allophones with long sustained sections; however, this would increase the recording time excessively.

The pitch axis must be adapted to the specific range of the singer. In our case, the sound quality seems to be acceptable for transpositions up to ±6 semitones, and the sampling grid is set accordingly, being often three pitch values enough to cover singer's range (excluding falsetto) for phonetic transitions. Since there are much less sustained allophones than di-allophone combinations, we can record more sustain contexts without increasing the recording time that much. Hence, we arranged to record sustained allophones every few semitones. On the other hand, loudness axis is set to the interval [0,1], assuming 0 to be the softest singing and 1 the loudest one. In our experiments, we decided to record *very soft*, *soft*, *normal* and *loud* qualitative labels for sustained allophones and just *soft* and *normal* for articulations. Regarding tempo, we recorded *normal* and *fast* speeds, valued as 90 and 120 BPM[30] respectively. Summarizing, for each di-allophone we sampled 3 pitches, 2 loudness and 2 tempo contexts, summing 12 different locations in the sonic space. Instead, for each stationary we sampled roughly 7 pitches and 4 loudness, so 28 different contexts.

Subspaces B and C contain vibratos and musical articulations intended to capture some basic expression aspects of the singer's voice and therefore increase the naturalness of the synthesis. With the aim of reducing the size of the database, we simplify them to be independent of the dimensions in subspace A. Subspace B contains different types of vibratos, a very common expressive resource. We do not intend to be exhaustive but to achieve a coarse representation of how the singer performs vibratos. At synthesis, samples in B are used as parameterized templates for generating new vibratos with scaled depth, rate and tremolo characteristics. Each template stores voice model control envelopes obtained from analysis that are later used in synthesis to control voice model transformations of samples in A. The third subspace C represents musical articulations. We model it with three basic types: note attacks, transitions and releases. For each of them we set a variety of meaningful labels with the intention of covering different musical and intentional contexts. Note we are considering here the musical note level but not higher levels such as musical phrasing or style, aspects to be ruled by the Performer Model discussed in §3.3. Like for vibratos, samples become parameterized templates applicable to any synthesis context (Bonada, Loscos and Mayor, et al. 2003). We are aware that several other expressive resources are possible and have not been considered here. Partly this has been a decision taken given the constraints imposed by research related to the

[30] BPM means beats per minute. In this case it corresponds to the number of syllables per minute, considering a syllable is sung for each note.

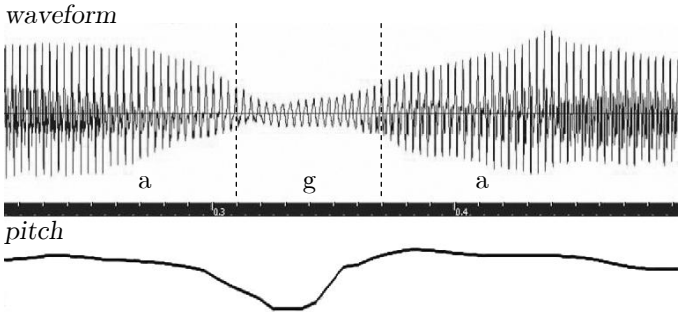


Figure 3.8 Loudness and Pitch variations inherent to phonetic articulations. In this figure we observe a valley in waveform amplitude and fundamental frequency along the Spanish /a/-/g/-/a/ transition.

development of a commercial software. However, we believe our approach provides ways of modeling and synthesizing most of the expressive characteristics of the singer.

3.2.2 Recording scripts

Recording scripts contain the description of the performances that should be recorded. In our case, they consist of a set of musical scores including lyrics and annotations explaining the specific goals of each performance. Most of them are devoted to the sampling of subspace A and contain meaningful sentences to be sung at a constant tempo, pitch and loudness values. One reason for using constant values is that we want to capture loudness and pitch variations inherent to phonetic transitions, and make them independent of the ones related to musical performance. It makes sense to refer to these phonetic related variations as phonetic intonation. Preserving this phonetic intonation is important for generating synthesis results that sound natural and intelligible. Figure 3.8 shows one example of variations occurring in a vowel/voiced-plosive/vowel transition. The idea is that in synthesis those relative variations are added to the envelopes generated out of the musical score so to increase the naturalness and intelligibility. On the other hand, another reason for using constant values of tempo, pitch and loudness is that we want to constrain the singer's voice quality and expression to ensure a maximal consistency between different samples among the database, intending to help hiding the fact that the system is concatenating samples and increasing the sensation of a continuous flow. A similar approach has been successfully used in TD-PSOLA based TTS systems in the context of emotional speech, which requires like singing synthesis more extreme pitch manipulations than normal speech (Vine and Sahandi 2000).

Next, we describe our approach to the Spanish language, although the procedure and most ideas are valid for other languages. Out of Joaquim Llisterri and José B. Mariño's paper on Spanish adaptation of SAMPA[31] (Llisterri and Mariño 1993), we consider there are 33 possible Spanish allophones, which make a list of 1089 theoretically possible di-allophone combinations. From this list, impossible and very rare combinations are removed, actually those that do not appear in the automatic phonetic transcription of our corpus, consisting of a set of classical books[32] summing up in

[31] SAMPA (Speech Assessment Methods Phonetic Alphabet) is a computer-readable phonetic alphabet based on the International Phonetic Alphabet (IPA), and originally developed in the late eighties by an international group of phoneticians.

[32] Initially we used a collection of 5000 Spanish lyrics downloaded from internet from different sources. However, they contained a lot of spelling errors and words in other languages. Thus, we decided to use the following collection of books instead: "20.000 leguas de viaje submarino" by J. Verne, "Alicia en el país de las maravillas" by L. Carroll, "El corazón de las tinieblas" by J. Conrad, "El príncipe" by N. Maquiavelo, "El público", "Bodas de sangre", "Romancero gitano", "Yerma", "Poeta en Nueva York" and "Poema del cante jondo" by F.G. Lorca, "La vida de Lazarillo de Tormes" (Anonymous), "Moby Dick" by H. Melville, "Naufragios" by A. Núñez Cabeza de Vaca, "Peter Pan" by J.M. Barrie, "Piel" by E. Barceló, "Del sentimiento trágico de la vida" by M.

MOST FREQUENT		LESS FREQUENT	
e-s	2.318%	x-tS	2.693e-005%
e-n	2.231%	rr-B	2.693e-005%
a-s	1.730%	L-x	2.693e-005%
e-r	1.655%	G-L	2.693e-005%
o-s	1.487%	rr-x	2.693e-005%
r-a	1.483%	x-B	2.693e-005%
k-e	1.428%	u-j	2.693e-005%
a-n	1.395%	L-G	2.693e-005%
t-e	1.376%	tS-B	2.693e-005%
a-r	1.375%	rr-f	2.693e-005%

Table 3.2 Most and less frequent di-allophone combinations that appear in our Spanish textual corpus.

Text	en zigzag como las oscilaciones de la temperatura
Phonetic transcription	Sil-e-n-T-i-G-T-a-G-k-o-m-o-l-a-s-o-s-T-i-l-a-T-j-o-n-e-s-d-e-l-a-t-e-m-p-e-r-a-t-u-r-a-Sil
Articulations	Sil-e T-i i-G G-T T-a G-k o-l s-o s-T i-l a-T T-j j-o o-n e-s a-t t-e m-p p-e e-r t-u

Figure 3.9 Excerpt from the Spanish recording script. The first line shows the sentence to be sung, the second one the corresponding SAMPA phonetic transcription, and the third one the list of phonetic articulations to be added to the database

total more than 300.000 words. From the previous results, we computed that we should record 521 allophone-to-allophone articulations so to cover around 94% of all possible appearing combinations, and approximately 99.9% of occurrences in the corpus. Table 3.2 shows the most and less frequent phonetic combinations we found together with their percentage.

We have not mentioned that the corpus has to be automatically transcribed to phonetics before computing any statistics. We did so by implementing in a C++ software the rules proposed in (Llisterri and Mariño 1993) for the Spanish language. For other languages, however, a rule approach might not be sufficient. For instance, for the English language we used an open source pronunciation dictionary of North American English, the Carnegie Mellon University Pronouncing Dictionary[33], which contains over 125.000 words and their transcriptions.

Once the corpus has been automatically transcribed into phonetics, we divide the text into sentences. In order to save time when recording articulations, we implemented a Perl script to find the minimum set of those sentences required to cover all the articulations to be recorded. An excerpt of one sentence of the recording script is shown in Figure 3.9. Compared to words or non-sense phonetic sequences, using sentences ease the comprehension of the required performance, reduce the recording time, and enhance the consistency of the pronunciation. On the other hand, a short recording script has several benefits; reduced recording time, less sentences to segment, and therefore less effort into creating, checking and tuning the singer database. In addition, it allows recording more pitches and loudness contexts while keeping the recording time reasonably short. ANNEX B contains the Spanish recording scripts.

Several issues have to be considered regarding the recording procedure. Singers usually get bored or tired if the scripts are repetitive and mechanical. Therefore, it is recommended to sing on top of stimulating musical backgrounds. This in addition ensures an accurate control of tempo and tuning and increases the feeling of singing. Besides, using actual meaningful sentences increases furthermore

Unamuno, “Romeo y Julieta” by W. Shakespeare, “Fábulas literarias” by T. de Iriarte, “Yo acuso” by E. Zola, “Cuesta abajo” by L. Alas Clarín.

[33] http://www.speech.cs.cmu.edu/cgi-bin/cmudict

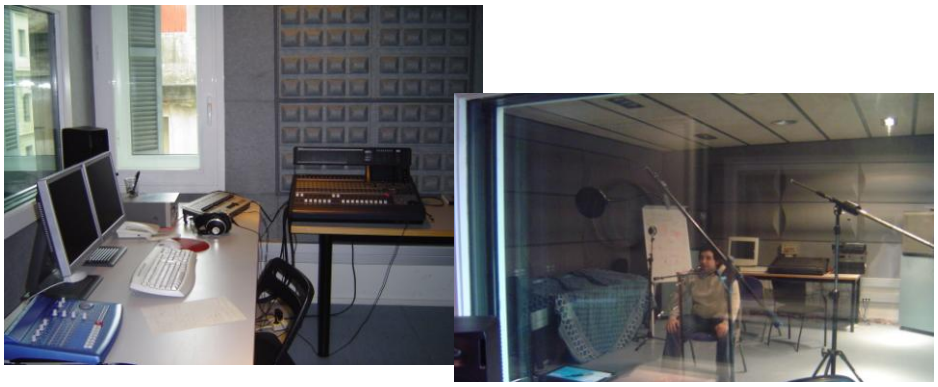


Figure 3.10 Recording studio facilities at the Institut Universitari de l'Audiovisual (IUA) of the Universitat Pompeu Fabra.

the feeling of singing. Another aspect to consider is that voice quality often degrades after singing for a long time; therefore pauses are necessary now and then. Finally, another concern is that high pitches or high loudness levels are difficult to hold continuously and exhaust singers, what suggests the necessity of combining samples of different pitches and dynamics in consecutive sequences. We considered all this issues in our recording scripts.

3.2.3 Recording session

Most of the recordings were performed at the studio lab of the Institut Universitari de l'Audiovisual (IUA), in the Universitat Pompeu Fabra[34] (Figure 3.10). Spanish recordings took between three and four hours per singer. English recordings, however, took between 5 and 6 hours, due to the larger number of phonetic contexts to cover. In both cases, the recording was split into several sessions to avoid voice quality degradation and reduce singer stress. Recordings involved several people. A sound engineer was controlling the sequencer and the mixer while two researchers were carefully listening to the singer performance and asking for repetitions whenever required in order to ensure a correct realization of the scripts.

3.2.4 Database creation

The creation of the singer database is not an easy task. Huge numbers of sound files have to be segmented, labeled and analyzed, especially regarding subspace A. That is why we put special efforts in automating the whole process and reducing manual time-consuming tasks (Bonada, et al. 2006). Figure 3.11 shows the different steps involved in the database creation and how one English sentence is gradually processed in each step. The steps are data preparation, phoneme segmentation, sample segmentation, sample analysis and database tuning.

Initially, the recorded audio files are manually cut into sentences. For each of them we add a text file including the phonetic transcription plus tempo, pitch and loudness values set in the recording scripts. Next, we perform a phonetic segmentation. In order to speed up this process we use an automatic speech recognizer (ASR) tool as an aligner between the audio and the transcription. For Spanish and English languages we used two free software toolkits, (Kawahara, et al. 2001) and (Huang, et al. 1993) respectively. Both ASR systems use HMM-based models trained with mel-frequency cepstral data. While overall this approach gives fairly good results, the main problem is that segmentation still fails in a significant number of cases. Segmentation sometimes fails completely, probably due to fundamental differences between the singing voice input and the speech corpus the ASR toolkit is trailed with, or, more commonly, the overall segmentation is correct but some boundaries are misestimated locally. This latter case is worsened by the fact that the ideal

[34] http://iua.upf.edu/recursos/laboratorios/audio/

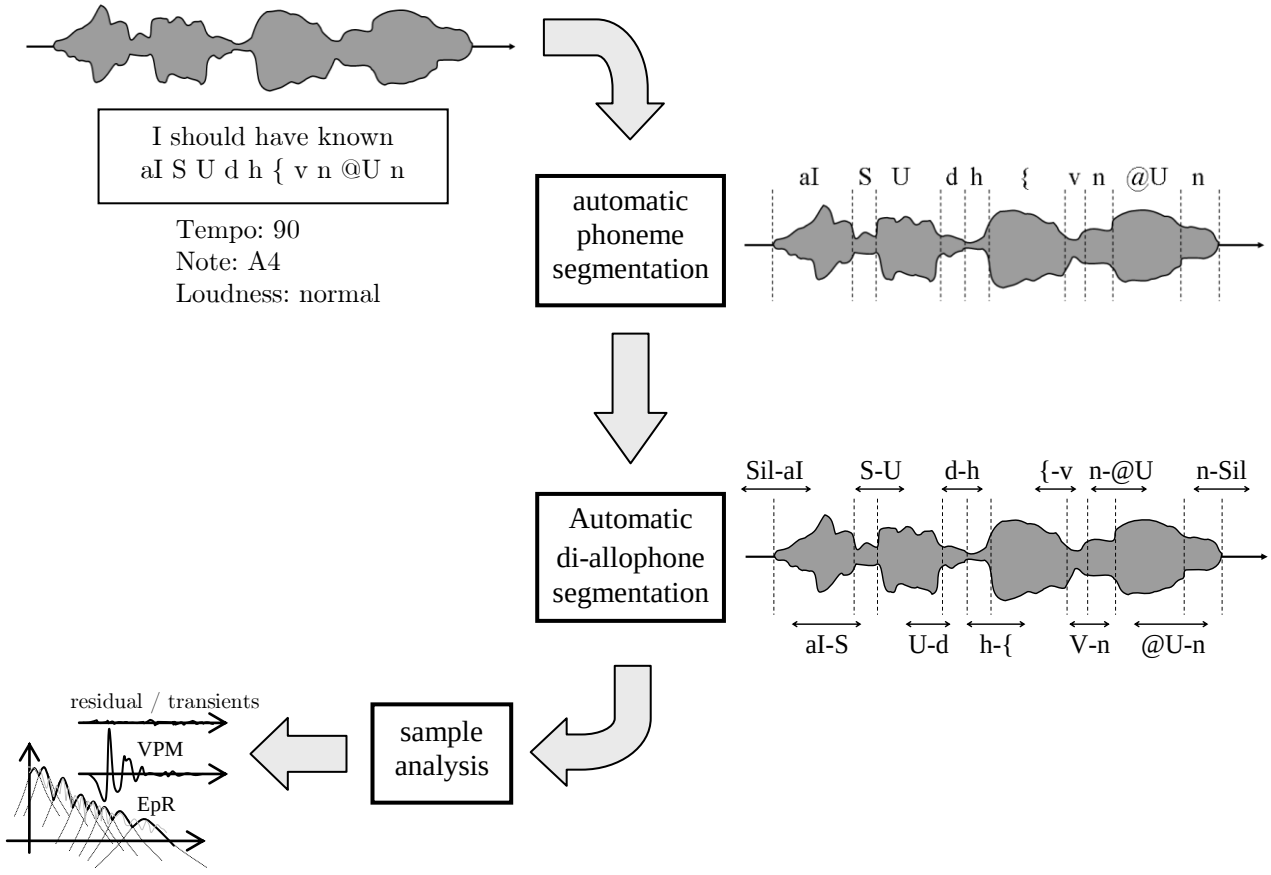


Figure 3.11 Singer database creation process

segmentation might be defined slightly differently between the ASR toolkit and the synthesizer. To improve the results of the ASR toolkit a second post-processing step was later introduced. First, a number of low-level descriptors such as amplitude envelope, derivative of mel-frequency cepstral coefficients (MFCC), zero-crossing rate, etc. are computed from the recording. Then, assuming the initial ASR segmentation is at least globally correct, the algorithm uses the low-level descriptors to improve the initial segmentation using rules based on a priori knowledge about the given phoneme types of the articulation. This post-processing step significantly improves the output of the ASR segmentation, but has the disadvantage that finding a set of rules that work well in all cases is difficult and still requires manual verification that the initial segmentation is at least globally correct. However, on the other hand, rules can be adapted to the specific characteristics of the singer.

In the next step, samples are automatically segmented fixing the precise boundaries of the di-allophone units around each phoneme onset, following several rules that depend on each allophone family[35]. The main rule is to set boundaries at stable frames whenever possible. A relevant aspect is that we detect gaps, stops and stable segments, as illustrated in Figure 3.12. The purpose is to use this information for better fitting and handling samples at synthesis. One example is to drop out those segments in the case of fast singing instead of time-compressing the whole sample.

Once samples are segmented, each one is analyzed using spectral analysis tools to obtain the specific data required for the voice processing technique used at synthesis. This data is stored in the hard disk. The last step is to manually check and tune the resulting database and correct erroneous segmentations or analysis estimations. Several tools have been implemented that automatically locate

[35] For instace, in the case of unvoiced fricatives such as the English /s/, we end the *-/s/ articulation just at the /s/ onset, and the beginning of the /s/-* articulation is set at the beginning of the /s/ utterance. We do it like this to avoid connecting different /s/ pronunciations at synthesis

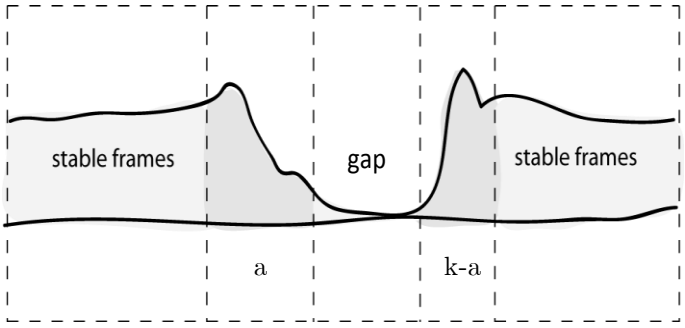


Figure 3.12 Gaps, stops and timbre stable segments detection. In this example of /a/-/k/-/a/ Spanish phonetic transition, two stable sustained parts with stationary characteristics have been found. In addition, in the middle a gap segment has been detected. Later at synthesis, frames of these segments can be drop to improve the synthesis results.

most significant errors and help the supervisor to perform this tuning without having to check every single sample.

We implemented an authoring tool to create voice databases, supervise each task involved in the creation process, and also visualize, check and modify segmentations and analysis data of each sample. It was implemented in C++ for the Windows operating system. This tool includes an implementation for most tasks to perform but calls a few external Phython and Perl scripts to carry out some specific tasks such as the phonetic alignment. One screenshot of this tool is shown is Figure 3.13. In this screenshot, the tool is showing a di-allophone sample. At the top, the user selects begin and end allophones from the available ones, in this case /t/ and /a/. Existing samples corresponding to the selected phonetic combination are drawn in the top left view, sorted horizontally by pitch and vertically by dynamics. The one chosen by the user is highlighted, in this case a sample at -2100 cents and 0.6 dynamics. The view below shows the estimated first and second formant center frequencies along time. The next one shows the waveform and the segmentation. The yellow inverted triangle is a mark that indicates the beginning of a detected quasi-stationary sustain, between 0.423 and 0.505 seconds. The second view starting from the bottom shows the EpR voice model estimated at the selected frame, including source curve, resonances, residual envelope and harmonic envelope. Finally, the bottom view shows the estimated fundamental frequency along time in cents scale.

The data obtained during the creation of the database is stored in a set of files organized in a folder tree structure. Each folder is accompanied by a binary file with the same name and "dat" extension that includes data related to the folder. The top folder is named with the singer name. Each folder contains files with data and subfolders if required. Most data files are stored in binary format using a chunk-based structure. Figure 3.14 shows a fragment of a database including sustained and di-allophone units. We have defined and implemented the structure in a way that the same source code allows storing the whole database in a single file internally organized in chunks, or in a folder tree structure otherwise. One bit of data indicates at each level whether it is stored as a folder or as a chunk. The software that calls the API to read or write the database ignores which option was selected. This way, we can take advantage of both database organizations. A folder tree structure allows using any system explorer an easy backup of specific data files, merging different databases and move or delete data. By contrast, a single file database hides the internal structure and facilitates database deployment and installation.

ON THE FLY DATABASE CREATION

On drawback of the presented database creation framework is that several aspects such as out-of-tuning singing, level recording mismatches or mispronunciations, might not be detected until the recording has finished and the database created and checked. In addition, the manual cut of the recordings into sentences might be sometimes a tedious task. Hence, we propose to implement in the form of a VST plug-in a tool to create new databases on the fly, in real-time, while recording in the studio. This tool should as well manage the recording scripts, allowing recording different takes for

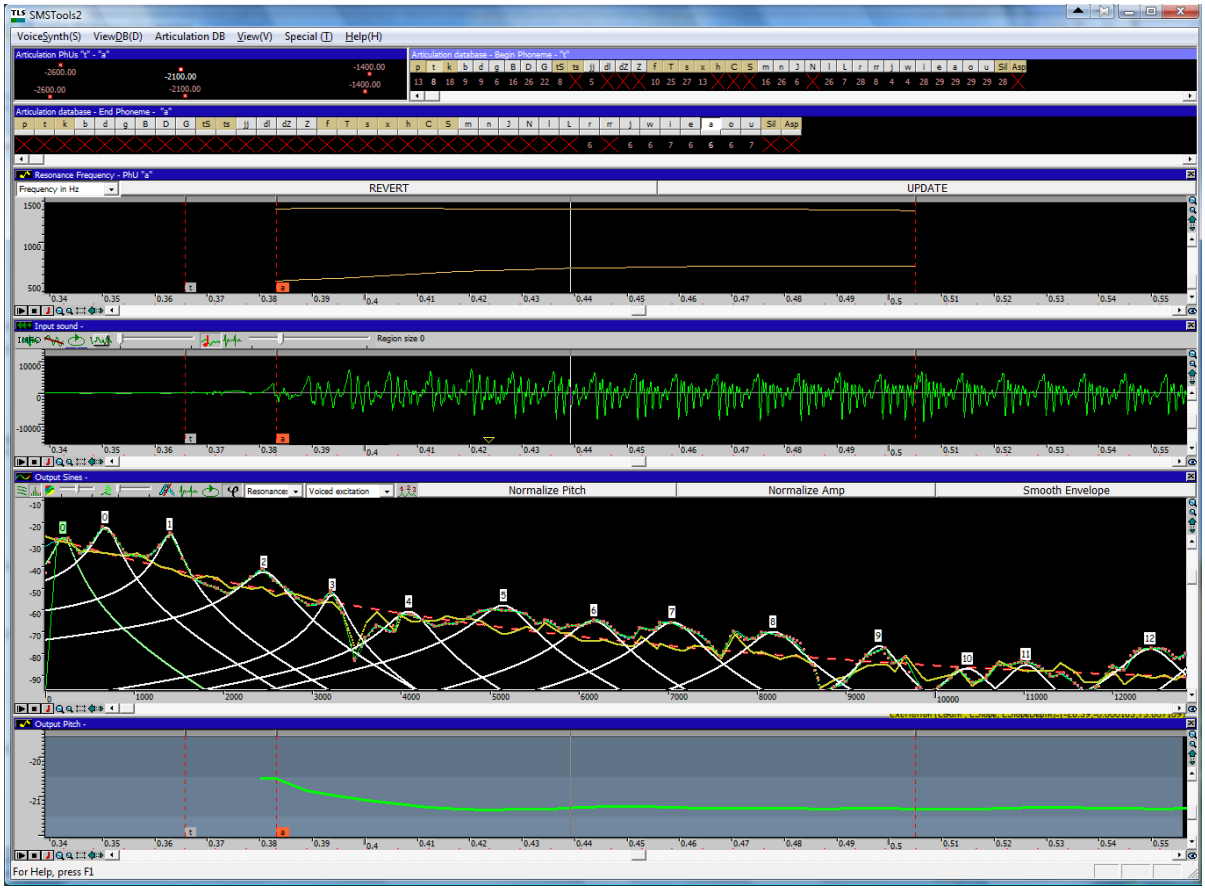


Figure 3.13 Voice database creation authoring tool

each script, and incorporate several tests to detect typical problems. At the same time, we propose a different technique for segmenting recorded performances that better fits the tool we have in mind, and might as well produce better results.

As earlier mentioned, the sample segmentation algorithm works quite well in general, but still has some problems. A possible way around these problems may be to utilize one existing correctly segmented databases to segment new databases. This would avoid the dependency on the speech-trained model from the ASR used. The problem of time-aligning two utterances of the same sentence by different speakers can be solved using the Dynamic Time Warping (DTW) algorithm (Rabiner and Juang 1993). It takes one or a combination of descriptors of the signals and then finds the optimal path through a similarity matrix of the descriptors of both utterances. Allowing using additional descriptors besides MFCCs can provide some of the benefits of the previous approach's post-processing step in a simpler manner because none of the phoneme-dependent rules is present. In particular, a combination of MFCCs and time-domain waveform envelope derivative turned out to be an effective combination. MFCCs give a good overall match and the envelope derivative helps to avoid discontinuous jumps in the optimal DTW path in cases where vowels are sustained or shortened compared to the model utterance. An important issue with the DTW technique is to accurately trim silence at the beginning and, less importantly, the end of the utterance to remove leading and trailing silence since begin and end points of the DTW path are always fixed.

Initial listening tests to evaluate the algorithm were positive. Those were realized by listening simultaneously to the target utterance and the source utterance time-scaled using the DTW predicted alignment. Quantitative evaluation proved somewhat difficult because the second reference database available was only partially corrected manually. Furthermore defining a meaningful rule to measure if segmentation was successful is problematic because this is dependent on phoneme types to a degree. The inherent disadvantage of this system is of course that it requires at least one correctly segmented database for each supported language. Consequently, the recording scripts cannot be easily changed.

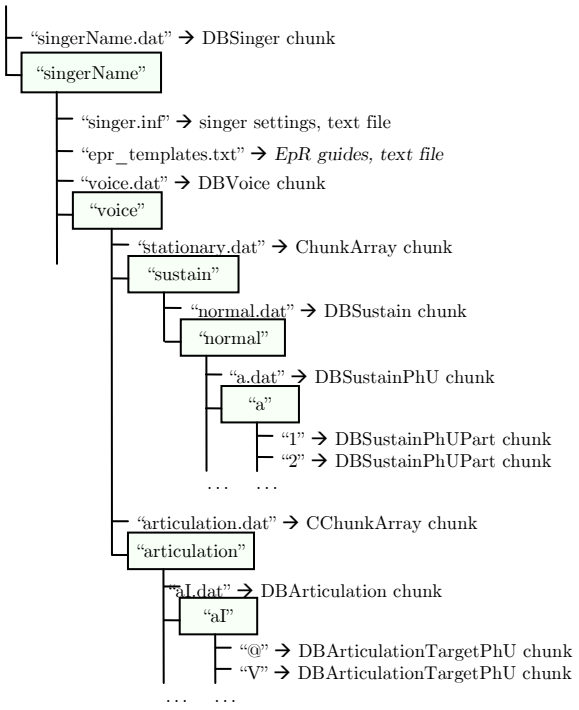


Figure 3.14 Voice database folder structure

Figure 3.15 Voice database creation VST plug-in interface.

However, creating the initial segmentation model for a specific language can still be partially automated using the previous ASR-based technique.

One problem that arose from the way databases were created was that issues with the recordings such as mispronounced phonemes, level mismatches, pitch problems, etc., usually were not found until creating or using the database, when it is usually too late to fix them. To reduce these kinds of problems, the DTW-based database creation tool was implemented as a real-time VST plug-in. This allows it to be easily integrated with many recording environments. Besides improving the database creation workflow overall, this on-the-fly system can also help flag problems as they happen and increase database consistency, which ultimately determines the synthesizer output's quality. Firstly, recordings are segmented in utterances automatically, then checked if their duration is approximately equal to those of the models and finally the DTW is applied. The total error of the DTW path-finding algorithm can indicate problems with the match such as severe mispronunciations. This

system also allows vowels levels in stationary samples and articulations to be matched more closely. Figure 3.15 shows a screenshot of the VST plug-in interface.

3.3 Performer Model

The Performer Model should emulate as precisely as possible the behavior of the real instrument player when performing a given score. However, we are not so much interested in the physical actions executed by the performer, but in the decisions taken on a higher level close to the musical interpretation of the piece, therefore to the musical gestures chosen in that context. Indeed, our intention is to come out with a clear picture of which are the intentional controls that the performer activates when playing the instrument. In other words, we want to get rid of the huge amount of unconscious movements that physically produce the effect wanted, which are inherent to the performer, to its unique way of playing, and are the result of the techniques practiced and learned over the years.

An illustrative analogy would be the case of a car driver who wants to drive past another car. The drivers thinks he wants to drive faster, so he presses downwards his foot on the accelerator. He is not actually thinking of the physical action he performs (e.g. moving the foot), but rather of the idea of going faster. We could say the same about a singer who wants to sing a certain note. He is thinking the mental representation of the pitch of the note, but not anything related to the vocal folds tension or the subglottal pressure he will apply in order to get them vibrating at the rate determined by the pitch of that note. And so on, we could apply this same concept to any performer playing any instrument. For instance, we could say that a violinist is controlling the loudness of the played notes moving his thought along a mental representation that goes from *pppp* to *ffff*, but that he is not consciously aware of the actual velocity and force he is applying to the bow.

From the input score, the Performer Model is in charge of generating lower level actions, thus it is responsible of incorporating performance specific knowledge to the system. Some scores might include a fair number of performance indications and in those cases the Performer Model would have an easier task. Nevertheless, several complex and varied issues are involved in the processes that take place when the player decides how to interpret a musical piece in a certain way. The issues involved are very diverse, going from music theory to cognition and motor control aspects. In this research work, we do not attempt to come up with a complete and automatic Performer Model in all its facets. Our aim is to provide the potential user of the synthesizer with a broad and rich set of useful tools to allow him to manipulate the performance in such a way as the real performer does. Hence, the synthesizer will ideally guide the user and propose him several interpretations of the piece, but the user will be the one who finally decides the appropriate interpretation and refines it. Nevertheless, we have to say that the complete emulation of the performer is out of the scope of this research work, yet a desirable goal.

One particular case worth to mention is that of performance driven synthesis(Meron 1999, Janer, et al. 2006), where the input score is actually a performance and contains all the controls and details required by the synthesizer, so that nothing has to be added by the Performer Model.

3.3.1 Approaches to Performance Modeling

We are far from completely understanding and being able to simulate the music performance process and therefore this is still one of the most open-ended problems in music sound synthesis. Current approaches to performance modeling only give partial, although promising, answers. A good overview and comparison of most relevant performance models is found in (Widmer and Goebl 2004). Next, we briefly describe the most successful practical approaches.

- ANALYSIS BY SYNTHESIS

  This method consists on iteratively refine performance hypotheses by judging the results produced by a computation model. First, one hypothesis is proposed about one expressive resource a musician uses when performing. Next, a computational model that implements that hypothesis is built and used to synthesize a piece. Then this piece is either evaluated by

a professional musician or compared to a real interpretation. Finally, the initial hypothesis is refined and the process starts again.

The research on this model was initiated in early eighties (Frydén, et al. 1983) and has been largely extended. A comprehensive overview is given in (Friberg 1995). This model implements a collection of performance rules that are applied to traditional music scores, modifying note timings and dynamics. Control over timbre and vibrato is also predicted. This model has been successfully implemented as a standalone application (*Director Musices*) for several platforms and is freely available online[36]. The interpretation can be controlled in real-time by specifying how much each rule affects the synthesis. Negative factors are allowed, which result into applying rules with an inverse effect. Controls are grouped in what the authors called *expressive palettes*. Much research has been devoted to define palettes that express different emotions, also for the specific case of the singing voice (Juslin and Laukka 2003). Actually, the author was supervisor of a Master Thesis that integrated those rules in a singing voice synthesizer (Alonso 2004).

- ANALYSIS BY MACHINE INDUCTION

  Machine learning techniques can be used to generate performance rules automatically. The method consists on applying to data obtained from real musician performances machine learning techniques to produce general performance rules. Those rules can be interpreted and used as predictive computational models for new unknown pieces(Widmer and Tobudic 2003). Interestingly, these techniques have shown to perform quite well in the task of identifying famous pianist players (Saunders, et al. 2004).

- INTERACTIVE SYSTEMS

  Another useful method is based on using music notation software that allows interactive playing and adjustment of the synthetic performance (Laurson, et al. 2005). In this case, the user becomes an essential part of the model. Traditionally, real-time synthesis systems and non-real-time computer assisted composition environments have been seen as separate entities. According to the authors, this is an attempt to make a bridge between them. This system features a constraint-based language focused in music related search problems.

3.3.2 Singing Voice Performance Controls

For the particular case of the singing voice system presented here we propose to use an approach based on the user interaction. The synthesizer has to provide the user with a set of controls that allow him to generate expressive performances and tune them according to the feedback of the system. Often the virtual voice will be part of a music production and therefore is going to be mixed with other audio material. Hence, the ideal feedback for the user would be precisely the real-time output of the mixer, so that he can easily perceive if the results are satisfactory.

Most approaches to performance modeling work at note and phrase levels, but do not predict intra-note behaviors. This is a handicap because singing voice is never static and its characteristics are continuously evolving in time. Generating natural sounding and expressive performances is not an easy task. Singing voice is rarely on time and, when note onsets and durations are quantized to the score metrics, the sound produced is typically perceived as mechanical and artificial. Similar statements can be made regarding tuning. A note is almost never sung exactly at the pitch associated to the note. Moreover, sung notes are connected with relatively slow time-varying pitch envelopes that never repeat exactly, and might be very diverse.

In our approach we assume that melody and lyrics are always fixed and given by the input score. We propose to use the following controls:

- VOICE MODEL

  The user can modify EpR parameters to adjust timbre characteristics, including subtle voice quality modifications. In addition, allophone transcription can be altered to refine pronunciation.

[36] http://www.speech.kth.se/music/performance/download

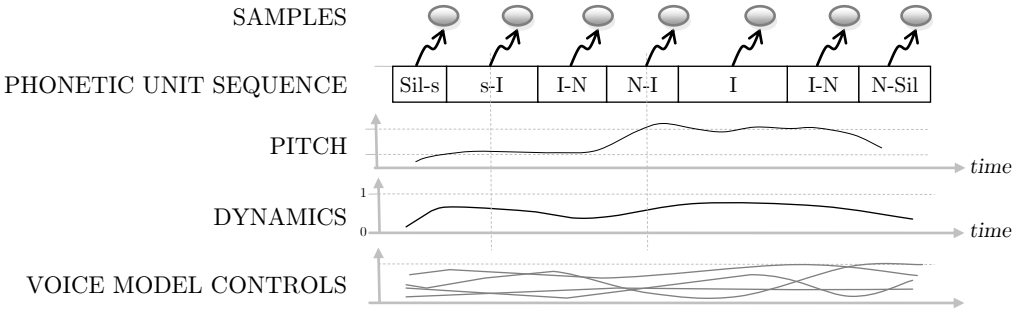


Figure 3.16 Example of a performance trajectory

- TIMING.
  The user can freely set note onsets and durations, although overlap is not allowed.
- DYNAMICS
  The user can set a dynamic value in a scale from 0 to 1 to each note. In addition, an envelope can be drawn.
- MUSICAL ARTICULATION
  The user can set for each note labels that define ways of attacking, connecting and releasing notes. The most basic expressions that we consider a synthesizer should at least cover are *soft*, *normal* and *sharp* attacks, *legato, portamento* and *scoop* transitions, and *soft*, *normal* and *sharp* releases. We consider s*taccatto* transition as notes separated by a pause. In addition, the user should be able to set the grade of application of each label and adjust its duration.
- EXPRESSIVE RESOURCES
  It is also important to provide the user control over other specific expressive resources. The most relevant one is vibrato, present in almost every musical style. The system allows the user to choose between different types of vibrato. Besides, the synthesizer must offer control over vibrato time-varying parameters: rate, deep and tremolo.

It is beyond the scope of this research to address higher levels of performance control such as different levels of phrasing.

3.4 Performance Trajectory Generation

The Performance Trajectory Generator module converts performance actions set by the Performer Model and the input score into adequate parameter trajectories within the instrument's sonic space. We characterize singing voice Performance Trajectories as phonetic unit sequences plus pitch and loudness envelopes plus voice model controls, as illustrated in Figure 3.16. Those are coordinates of the sonic subspace A_{PT}. Note that A_{PT} is not the same as subspace A, but somewhat equivalent if we assume that the tempo axis in A is already embedded in the phonetic unit track as both timing information and actual sample selection. We saw in §3.2.1 how the singing voice sonic space was split into three subspaces. The first one A includes the actual samples to be synthesized, and the rest B and C represent ways of transforming samples from A so to obtain specific musical expressions. Hence, the Performance Trajectory Generator actually applies models from subspaces B and C to the coarse Performance Score coordinates in A, in order to obtain detailed trajectory functions within subspace A_{PT}.

A representative example of the whole process is shown in Figure 3.17, starting with a high-level performance score (top left) of two notes with the lyrics *fly me*. The former note is an Ab2 played forte and attacked softly. The latter note is a G2 played piano, ending with a long release and exhibiting a *wet* vibrato. The note transition is *legato*. From this input an internal score (mid right) is built. It describes a sequence of audio samples, a set of musical articulation templates, vibrato control parameters, and loudness and pitch values. Lowering another level, musical articulation

templates are applied and the resulting Performance Trajectory (bottom right) is obtained. In this same figure, we observe another possible input performance score (bottom left), in this case a recorded performance by a real singer, obviously a low-level score. From it we can directly generate a performance trajectory with a similar expression by performing phoneme segmentation and computing pitch and loudness curves(Janer, et al. 2006).

Sample selection and transformation parameters in concatenative synthesis systems are typically obtained as the optimal solution of a constraint-based problem (Schwarz 2007, Meron 1999). In general, we can say that the more transformations applied to a signal, the more likelihood of signal degradation and presence of artifacts. Hence, in our case we compute the optimal sample sequence so that the overall transformation required at synthesis is minimized. In other words, we choose the samples that better match locally the performance trajectory. With this aim, we compute the matching distance as a cost function C obtained as a weighted sum of several transformation costs: temporal compression or expansion applied to the samples to fit the given phonetic timing ($C_{duration}$), pitch and loudness transformations needed to reach the target pitch and loudness curves (C_{pitch} and $C_{dynamics}$), and concatenation transformations required to connect consecutive samples (C_{pitch_cont} and $C_{dynamics\_cont}$). The weights have been obtained empirically. We denote $\bar{S}=\{S_0,S_1,...,S_{n-1}\}$ as a sample sequence, and S_i as the i^{th} sample in the sequence. The cost of a sequence $\bar{S}$ is given by

$$C(\bar{S})=\sum_{i=0}^{n-1}C(\bar{S}(i))$$

$$\begin{aligned}C(S_i)=&\,w_{pitch}C_{pitch}(S_i)+w_{dynamics}C_{dynamics}(S_i)+w_{duration}C_{duration}(S_i)\\&+w_{pitch\_cont}C_{pitch\_cont}(S_i)+w_{dynamics\_cont}C_{dynamics\_cont}(S_i).\end{aligned} \tag{3.3}$$

The best sequence $\bar{S}_{best}$ is the one that has a cost lower than that of all other possible sequences. If there are m possible sequences,

$$C(\bar{S}_{best})\le C(\bar{S}_k)\quad \forall k\in\{0,m-1\}. \tag{3.4}$$

Ideally, a global optimization is preferred. However, in real-life implementations, especially when the input score is a data stream or the user has a real-time control over the performance, the time-interval over which this optimization is performed must be limited. In our implementation it is fixed to the duration of two succeeding notes. While bigger optimization intervals get closer to the ideal solution, they also result in greater computational costs while only offering diminishing improvements according to our experiments. Furthermore, in real-time settings, the interval is limited by the available time between user input and synthesis output (i.e. system's latency)

3.4.1 Phonetic Sequence

One essential aspect to consider when computing phonetic timings is to ensure the precise alignment of certain phones (mainly vowels) with the notes (Sundberg 1987, Ross and Sundberg 2001). Each di-allophone sample in the database is segmented phonetically by three time-tags: t_b, t_o and t_e. t_b and t_e indicate respectively the begin and end of the sample, whereas t_o indicates the ending allophone onset. Hence, in most situations a note onset should match the frame corresponding to time t_o in the sample that contains the main vowel of the lyrics associated to that note. The selected samples have to be fitted to the duration determined by the notes. This fitting often implies to time-scale samples or remove some sections. Note that in the case of long notes usually a sustained sample will be selected in addition to di-allophone samples. Notice as well that consonants preceding a vowel are pronounced before the actual note onset and therefore when fitting have to be considered as belonging to the previous note. This can be observed in Figure 3.17. The segments to be fitted to the duration of the first note are the right part of *l-aI* sample, *aI-m* sample, and the left part of *m-I* sample, being *m* the consonant that precedes the vowel I of the following note.

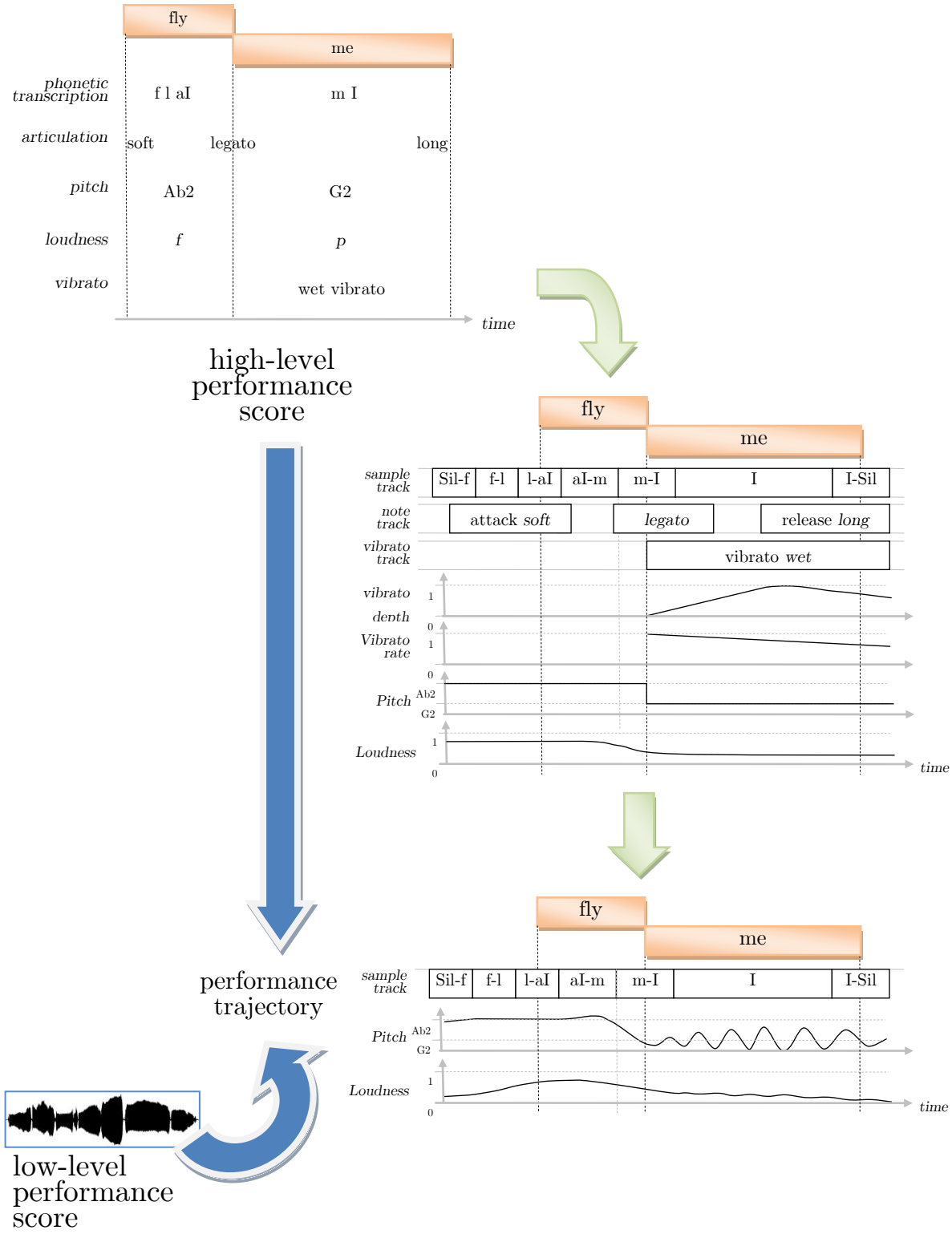


Figure 3.17 From performance score to performance trajectory

Additional markers present in samples determine which parts of the sample are stable and which are transitional. For instance, an articulation sample may include a sustained vowel section or a small silence before a plosive. Using this information the system can determine which segments of the sample are essential and which can be cut without loss of intelligibility. In many cases this allows to avoid time-compressing samples when synthesizing fast singing, especially when there are no available rapidly sung samples in the database. Moreover, time-scaling transformations often work better in stable sections than in transitions. Therefore, these marks are also used to apply different time-scaling ratios to stable segments and transitions, this way increasing furthermore the synthesis sound quality.

3.4.2 Pitch and Dynamics Envelopes

We propose to generate pitch and dynamics envelopes in two steps. First, a parametric model generates the continuous envelopes with a neutral expression and second, templates from *B* and *C* singer subspaces modify those envelopes to achieve the target expression.

EXPRESSION MODEL

We have designed an algorithm that generates both synthesis pitch and dynamics curves by smoothly interpolating a set of points obtained from normal distributions. The initial step in the design was to observe the analysis envelopes of a set of arpeggio sequences performed by several singers. Out of this observation, some predominant curve tendencies were empirically modeled by a set of points for each note, with a number depending on the absolute note duration in seconds. Figure 3.18 shows a few examples of pitch curves generated by this model. Note that obtaining values from normal distributions implies that curves generated from the same input score are always different. When several synthetic voices singing the same notes are mixed and overlapped, this randomness helps to perceive them as independent voices. However, this randomness also produces subtle changes in the perceived interpretation, what might be a drawback for certain potential users who like a particular interpretation and prefer a replicable system. Hence, to cope with both scenarios we generate the predictions using a pseudorandom process whose initialization is controlled by the user.

Regular length notes are modeled by 3 points (*A, B and C*), whereas shorter notes by just 2 points (*B* and *C* get clustered). If the duration is even shorter, then all 3 points get clustered into a single one (*A*). Each point has a pitch value computed from a normal distribution with mean equal to the note pitch and standard deviation of $\sigma = 20$ cents. Section *A*-*B* is obtained as the sum of a linear interpolation between *A* and *B* plus a valley computed subtracting a squared sinus with an amplitude obtained form a normal distribution with zero mean and standard deviation of $\sigma_B = 35$ cents. Section *B*-*C* is a linear interpolation between *B* and *C*. Durations of each section are computed using normal distributions (*onset*-*A, A*-*B, B*-*C, C*-*noteOff*). If the note duration is long, in order to avoid the perception of a pitch too static, more points are added (each 200ms) between B and C. Each of those points has a pitch computed from a normal distribution with mean equal to the note pitch and standard deviation $\sigma = 20$ cents. The different control parameters of the proposed model are listed in Table 3.3. Different parameter configurations can be used to roughly approximate different singing styles, for instance to generate performances with different grades of legato.

GENERATING PORTAMENTOS AND SCOOPS

There are two very common expressive resources that can be easily controlled and generated by this model: *portamentos* and *legatos*. In our context, we define *portamento* as reaching the note pitch before the note onset, and *scoop* as beginning the note transition after the actual note onset. Note that both expressions refer to the pitch curve but not to the phonetic track, which is not affected by either. A simple method to generate these expressive resources is to shift the actual note onsets to the expected positions before computing the points of the model, but keeping unmodified the note onsets used for computing the phonetic boundaries. Figure 3.19 shows an example of synthesis results obtained applying both expressive resources.

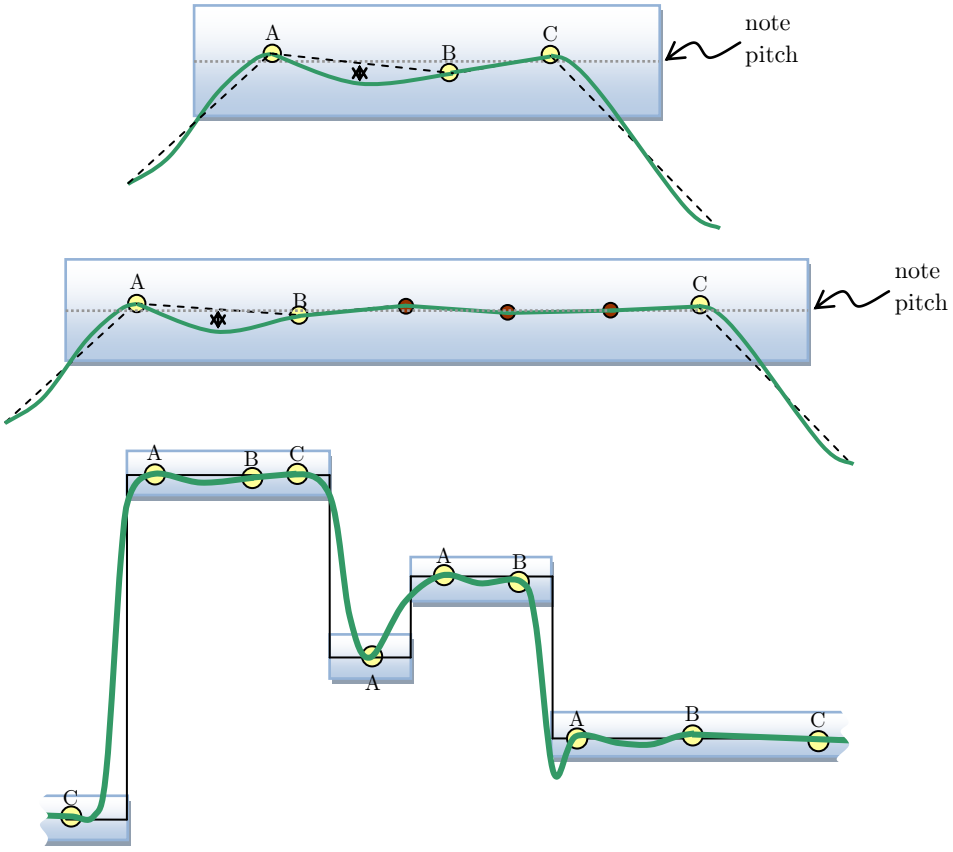


Figure 3.18 Pitch model based on smoothly interpolating a set of points. Depending on the duration of the notes, different number of points are used: 3 points (A,B,C) for regular durations, 2 points (A, B) for shorter durations, 1 point (A) for really short notes. In the top figure, we can appreciate the valley in A-B section, and the linear interpolation in the B-C section. If the note is long (middle figure), more random points are added between B and C. In the bottom figure, we see an example of the computed points and the generated pitch curve for a melody with different pitches and durations.

```
// "p" stands for pitch offset in cents
// "d" stands for duration in seconds
// "v" stands for pitch valley in cents
ctrlpitch_p1dev = 20.; // p1 std in cents
ctrlpitch_p2dev = 20.; // p2 std in cents
ctrlpitch_p3dev = 20.; // p3 std in cents
ctrlpitch_v1mean = 25.; // v1 mean in cents
ctrlpitch_v1dev = 20.; // v1 deviation in cents
ctrlpitch_d1meanFirstNote = 0.1; // d1 mean in sec when it is first note
ctrlpitch_d1devFirstNote = 0.04; // d1 std in sec when it is first note
ctrlpitch_d1mean = 0.04; // d1 mean in sec
ctrlpitch_d1dev = 0.01; // d1 std in sec
ctrlpitch_d2mean = 0.14; // d4 mean in sec
ctrlpitch_d2dev = 0.05; // d4 std in sec
ctrlpitch_d2clustering = 0.05; // d2 clustering in sec --> if d2 duration is less than 0.05 than it will be clustered
ctrlpitch_d4mean = 0.13; // d4 mean in sec
ctrlpitch_d4dev = 0.03; // d4 std in sec
ctrlpitch_pMeanBeforeOnsetFirstNote = 150.; // pitch mean (in cents) before onset of first note
ctrlpitch_pDevBeforeOnsetFirstNote = 30.; // pitch std (in cents) before onset of first note
ctrlpitch_pMeanOnsetFirstNote = 50.; // pitch mean (in cents) at onset of first note
ctrlpitch_pDevOnsetFirstNote = 30.; // pitch std (in cents) at onset of first note
ctrlpitch_dBeforeOnsetFirstNote = 0.2; // time distance before onset of first note
ctrlpitch_vMeanNoteTransition = 30.; // valley mean (in cents) added to note transition
ctrlpitch_vDevNoteTranstition = 20.; // valley std (in cents) added to note transition
ctrlpitch_NoteTransitionPow = 0.7; // contols the shape of pitch during note transition
ctrlpitch_pMeanEndingNote= 20.; // pitch mean (in cents) at offset of ending note
ctrlpitch_pDevEndingNote = 10.; // pitch std (in cents) at offset of ending note
ctrlpitch_offsetEndingNote = 0.2; // offset of ending note, where pMeanEndingNote and pDevEndingNote are applied
                                  --> onset+d1+d2+d3+this
ctrlpitch_pDecayEndingNote = 50.; // decay pitch (in cents) at the end of an ending note
ctrlpitch_dDecayEndingNote = 0.5; // decay pitch position (in sec) at the end of an ending note
                                  --> onset+d1+d2+d3+ctrlpitch_offsetEndingNote+this
```

Table 3.3 Control parameters of the proposed pitch model based on interpolating a set of points. A brief comment is added to each parameter.

EXPRESSION TEMPLATES

The aim of using templates is to generate a given synthesis performance gesture by transforming in a suitable way a recorded example of the same gesture type. Templates consist of a set of sequences of voice features extracted from real performances. Several statistics methods are suitable to be applied to each voice feature, such as the computation of global mean values, slowly varying local means, excursion ranges, standard variations, etc. Moreover, sometimes it might be appropriate to mark the time positions of important events such as synchronization points, local maxima or minima, segment boundaries, etc. We consider three different types of templates: sustain, vibrato and note articulation.

SUSTAINS

Voice parameters are never static but constantly changing along time. That is an important characteristic to emulate; otherwise, synthesized utterances are perceived as rather robotic or artificial. Sustain templates are obtained from samples in which an allophone is sustained for a few seconds. Basically, these templates store the voice model and pitch parameters values relative to a slow-varying mean. Hence, those variations are added on top of the performance trajectory values. In order to allow really long sustains, template envelopes are consecutively looped whenever required using random begin and end loop times.

VIBRATOS

Vibratos templates store pitch and voice model controls envelopes obtained from analysis, each of which is used in synthesis to control transformations of samples in subspace A. As shown in Figure 3.20, the control envelopes we use are a set of EpR parameters (gain, slope, slope depth) and pitch variations relative to their slowly varying mean, plus a set of time stamps pointing the beginning and the middle of each vibrato cycle. These time stamps are used as synchronization instants for transformations (for instance time-expansion can be achieved by repeating and interpolating vibrato cycles) and for estimations (for example vibrato rate would be the inverse of the duration of one cycle). Besides, vibrato samples are segmented into *attack*, *sustain* and *release* sections. The *sustain*

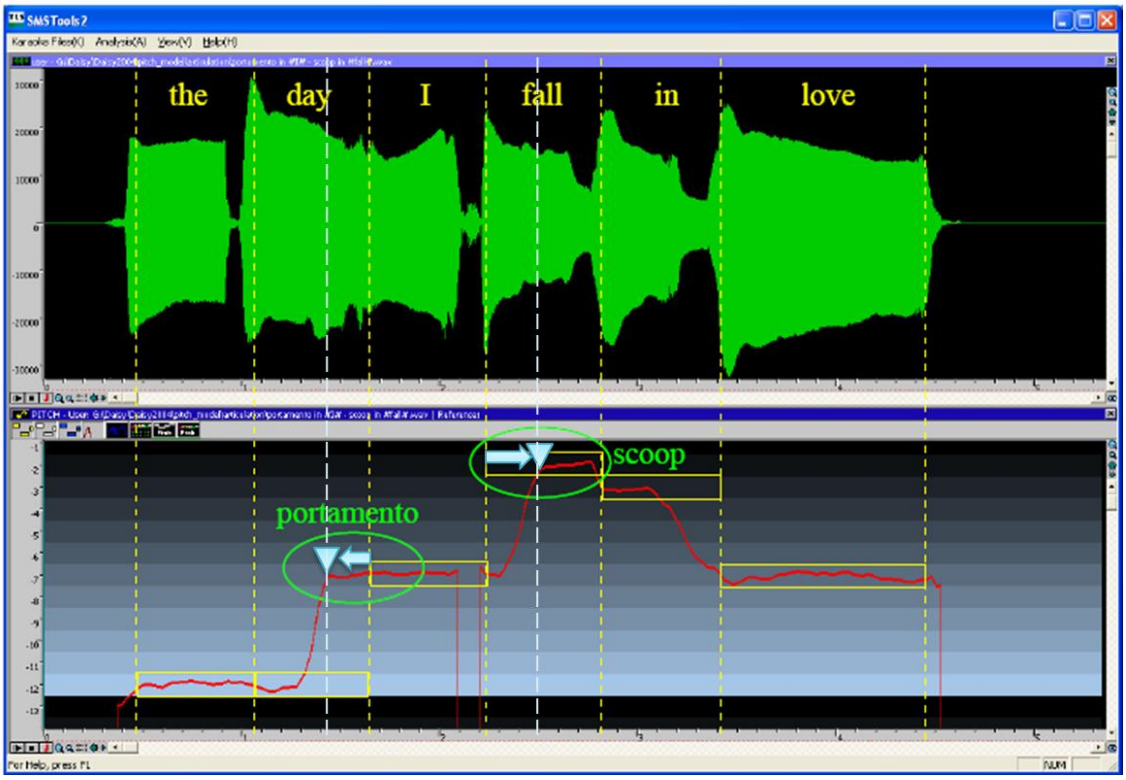


Figure 3.19 Example of a portamento and a scoop generated by the pitch model. The top figure shows the waveform of the synthesized sentence "the day I fall in love", segmented into notes (dashed yellow lines). The bottom figure displays the rendered pitch curve in red, with a *portamento* and a *scoop* highlighted in green. Changes to the note onsets are drawn as well (dashed blue lines).

section is looped in order to synthesize long vibratos. During synthesis, these templates are applied to flat samples from subspace A (i.e. without vibrato), and the EpR voice model ensures that harmonics will match the timbre envelope defined by the formants while varying their frequency (see Figure 3.21). Depending on the singing technique adopted by the singer, it is possible that formants vary in synchrony with the vibrato phase. Hence, in the specific case the template was obtained from a sample corresponding to the same phoneme being synthesized, we can use as well the control envelopes related to the formants location and shape for generating those subtle timbre variations.

NOTE ARTICULATIONS

Note Articulation templates are classified into attacks, transitions and releases. Those templates use voice model controls and pitch envelopes to transform samples in subspace A. Note attacks store the envelopes obtained by subtracting the ending values to each envelope. By contrast, note releases store the envelopes obtained by subtracting the beginning values instead. Note transitions have to be processed differently, because they are obtained from a specific interval that is often different from the synthesis one, so both begin and end values must be considered. Transition envelopes are computed as analysis envelopes normalized by the interval and afterwards, at synthesis, they are scaled by the synthesis interval. However, in order to avoid exaggerated behaviors, values above or below the interval are not normalized neither scaled. Figure 3.22 shows one example that illustrates this. The left view shows the pitch envelope of the note transition template. The right figure shows the template applied to a higher note interval using both approaches. The scaled envelope clearly exaggerates the peak above the ending note of the interval.

Obviously, the methods described work best when units are in a linear perceptual scale, so that the relative behavior is perceived as the same independently of the synthesis context. We could argue that most envelope units are roughly in such a linear perceptual scale: pitch in cents, voice model *Gain* and *SlopeDepth* in dBs. The following audio examples contain synthesis results generated using

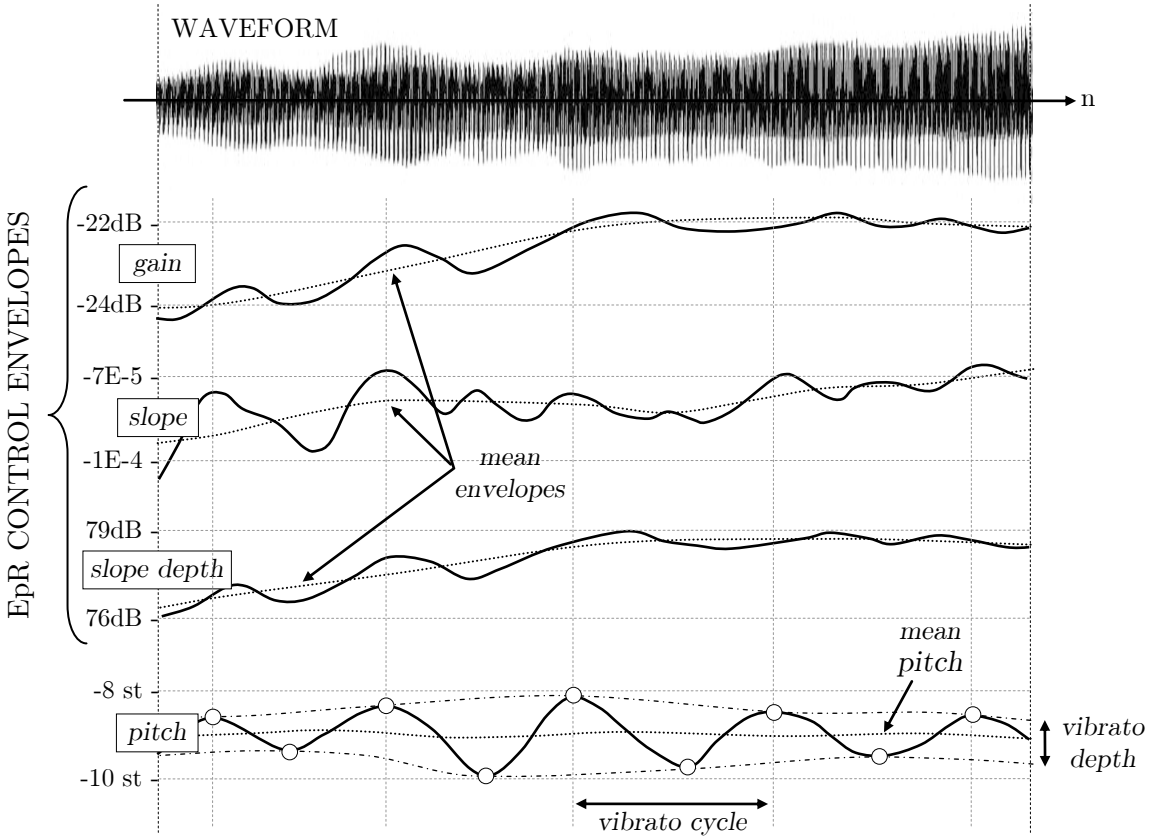


Figure 3.20 Vibrato template with several control envelopes estimated from a sample. The pitch curve is segmented into vibrato cycles. *Depth* is computed as the difference between local maximum and minimum of each cycle, while *mean pitch* is computed as their average.

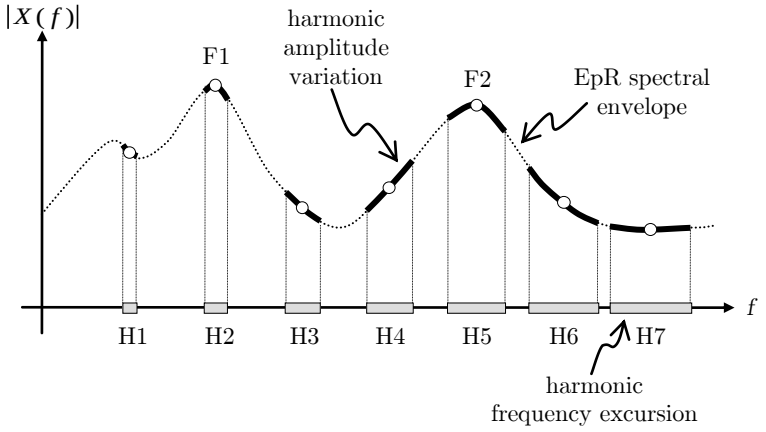


Figure 3.21 Along a vibrato, harmonics are following the EpR spectral envelope while their frequency is varying. For instance, when fifth harmonic (H5) oscillates, it follows the shape of the second formant (F2) peak.

several attack templates and show the potential of the proposed approach: *normal* ([250]), *smooth* ([251]), *smooth long* ([252]), *strong accent* ([253]), *low exaggerated* ([254]), *sexy* ([255]). One of the strong points in favour of using templates is their ability to imitate the behavior of a singer in specific contexts.

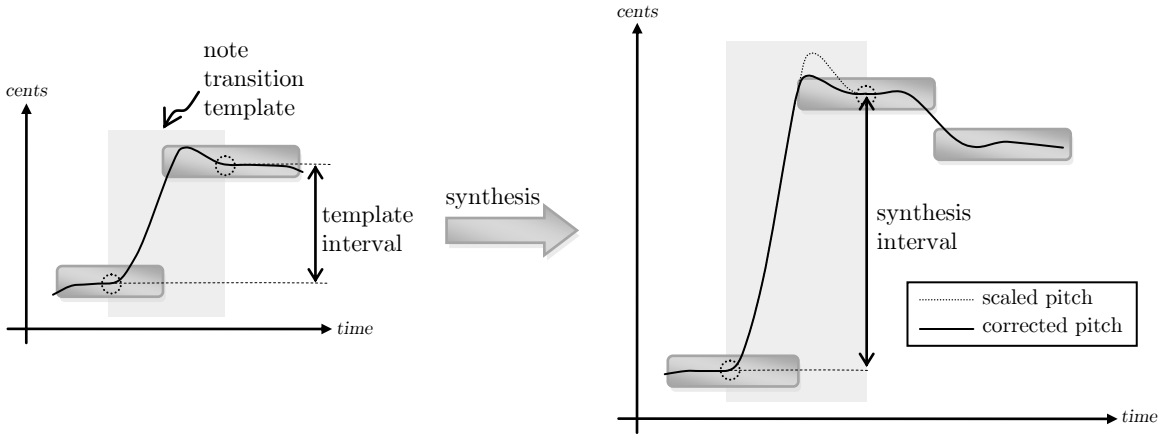


Figure 3.22 Note transition template applied to a different synthesis interval. Scaling the envelope exaggerates values outside the interval, in this case the pitch peak in the second note of the interval. Not scaling values outside the interval improves the results.

3.5 Sound Rendering

The Sound Rendering engine is the module that produces the synthesizer output sound. Its input is a Performance Trajectory within the instrument sonic space. The rendering process works by transforming and concatenating a sequence of database samples. We could think of many possible transformations to apply. However, we are mostly interested in those directly related to the sonic space axes, since they allow us to freely manipulate samples within the sonic space, and therefore match the target trajectory with ease. This concept is illustrated in Figure 3.23.

However, feasible transformations are determined by the spectral models we use and their parameterization. In our case, spectral models have been specially conceived for tackling the singing voice and allow transformations such as transposition, loudness and time-scaling, all of them clearly linked to the A_{PT} sonic subspace dimensions. Several resources related to musical articulation are already embedded in the Performance Trajectory itself, thus no specific transformations with this purpose are needed by the rendering module. Still, other transformations not linked to our specific sonic space axes and particular to the singing voice are desired, such as the ones related to voice quality and voice phonation, which might be especially important for achieving expressive and natural sounding rendered performances. For example, we could think of transformations for controlling breathiness or roughness qualities of the synthetic voice. In particular, voice excitation related transformations would be very useful for certain musical styles, such as blues, to produce growl utterances. We explored several methods for producing these kinds of alterations using our voice models in (Loscos and Bonada 2004).

3.5.1 Concatenating Samples

If we restricted our view to the sonic space we deal with, we would reach the conclusion that transformed samples do connect perfectly. However, this is not true, because the actual sonic space of the singing voice is much richer and complex than our approximation. Thus, transformed samples almost never connect perfectly. Many voice features are not described precisely by the coordinates in A_{PT} subspace, and others such as voice phonation modes are just ignored. For example, phonetic axis describes the timbre envelope with allophone labels, so rather coarsely. Hence, when connecting samples with the same phonetic description, formants will not match precisely. Another reason for imperfect connections is that the transposition factor applied to each sample is computed as the difference between the nominal note pitch specified in the recording scripts and the trajectory target

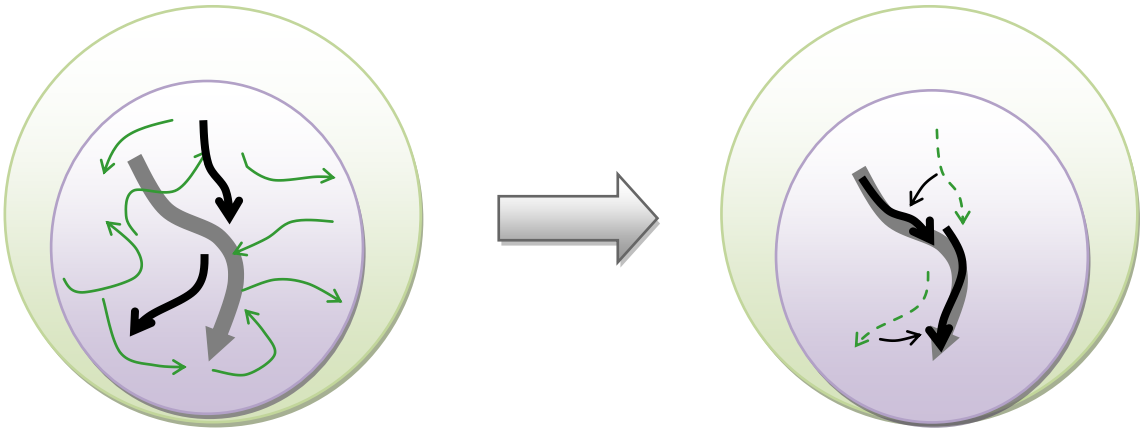


Figure 3.23 Matching a performance trajectory by transforming samples. On the left side we see the target trajectory (wide arrow) and the available samples (narrow arrows). The two selected samples are drawn in black with wider width. On the right side, we see how these samples are transformed to approximate the target trajectory.

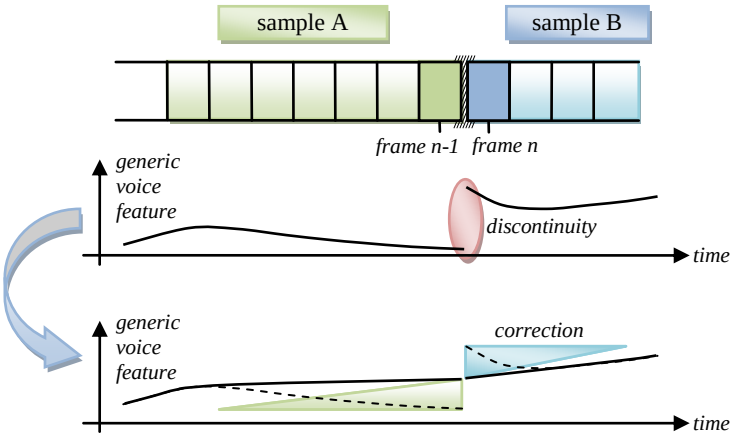


Figure 3.24 Sample concatenation smoothing

pitch, with the aim of preserving the inner fundamental frequency variations inherent to phonetic articulations, and therefore pitch rarely matches at sample joints.

Depending on the spectral processing technique chosen, different type of discontinuities might appear that require different approaches. Table 3.4 details potential discontinuities of the techniques discussed in Chapter 2.

All methods have to deal with pitch and voice model discontinuities. In order to connect samples smoothly, we compute the differences found at joint points for several voice features and transform surrounding frames by adding specific correction amounts to each voice feature. These correction values are obtained by spreading out the differences around connection points, as illustrated in Figure 3.24. We apply this method to pitch and to our voice model, including the controls of each formant (Bonada, et al. 2003).

In methods based on harmonic trajectories, harmonic frequency discontinuities exclusively depend on pitch values in the case frequencies were forced in analysis to match an ideal harmonic structure. Otherwise, discontinuities found at each harmonic frequency are handled separately with the correction spreading method.

If we use the voice phase model proposed in §2.5.2, then theoretically there will be no phase discontinuities as soon as harmonic amplitudes connect smoothly. Therefore, there is no need to apply

Harmonic Trajectories		Voice Pulse Modeling	
sinusoids plus residual *constant hop-size*	spectral regions *constant hop-size*	narrow-band (NBVPM) *constant hop-size*	wide-band (WBVPM) *pitch synchronous*
• pitch envelope • EpR controls • harmonic frequencies • harmonic phase alignment • period phase • residual energy • residual timbre	• pitch envelope • EpR controls • harmonic frequencies • harmonic phase alignment • period phase • region spectral envelopes	• pitch envelope • EpR controls • pulse phase envelope • period phase • residual energy • residual timbre	• pitch envelope • EpR controls • harmonic phase alignment • modulations present in harmonic amplitude and phase

Table 3.4 This table lists the different types of sample concatenation discontinuities associated to each synthesis technique.

any concatenation correction to phases anymore. WBVPM is a special case though, because noisy components appear as modulation factors of harmonic amplitude and phase parameters. The phase model should be adapted to cope with this fact; otherwise, noisy components would not be correctly synthesized. Nevertheless, some preliminary experiments suggest that increasing the phase model scaling factor α for middle and high frequencies partially recovers such noisy components.

In general, results are judged to be of high quality. However, when processing by spectral regions further work is needed to tackle phonation modes and avoid audible discontinuities such as those found in breathy to non-breathy connections. This is partially caused by the fact that noisy components are embedded in harmonic spectral regions and there is no residual model.

We next describe with more detail our approaches to minimize discontinuities in some cases that require more complex operations than just the correction spreading shown in Figure 3.24.

PHASE CONCATENATION USING HARMONIC TRAJECTORIES

In order to avoid harmonic phase discontinuities at segment boundaries, we have first to come out with a phase continuity condition. If we assume that harmonic frequencies vary linearly at boundary frames, then the phase continuity condition for a given harmonic h is given by

$$\phi'_{0,h,m} \approx \text{princarg}\left(\phi'_{0,h,m-1} + 2\pi \frac{f'_{h,m-1} + f'_{h,m}}{2} \Delta_t\right) \tag{3.5}$$

where $\phi'_{0,h,m}$ and $\phi'_{0,h,m-1}$ are the phases of the h^{th} harmonic at the right and left frames respectively. The phase correction $\Delta\phi^c_h$ that produces an ideal phase continuation of the h^{th} harmonic is obtained from the difference between left phase and the ideal phase by

$$\Delta\phi^c_h = \text{princarg}\left(\phi'_{0,h,m} - 2\pi \frac{f'_{h,m-1} + f'_{h,m}}{2} \Delta_t - \phi'_{0,h,m-1}\right). \tag{3.6}$$

This correction can be applied either to the left or to the right of the boundary (reversed sign), and spread along several frames in order to get a smooth transition. If it is applied to both sides, then half of the correction should be spread to each side.

Often there will appear discontinuities in the period phase at boundaries, and therefore in the voice pulse onset sequence. This produces big phase correction values that might affect significantly harmonic frequencies around the boundary, increasing or decreasing them. A way to continue the period phase is to time-shift the right sample by a certain amount Δ^c_t. This time difference can be computed from the estimated MFPA onsets. An alternative is to approximate it by continuing the fundamental phase, i.e.

$$\Delta_t^c = \frac{\text{princarg}\left(\phi'_{0,0,m-1} + 2\pi \frac{f'_{0,m-1} + f'_{0,m}}{2}\Delta_t - \phi'_{0,0,m}\right)}{2\pi f'_{0,m}} = \frac{-\Delta\phi_0^c}{2\pi f'_{0,m}}. \quad (3.7)$$

Note that it is better to time-shift the right sample than do it to the left sample, because in that case the correction would affect all the previous samples up to the beginning of the piece, and therefore it could not be implemented in a real-time system with a streaming input. Considering this correction, equation (3.6) becomes

$$\Delta\phi_h^c = \text{princarg}\left(\phi'_{0,h,m} + 2\pi f'_{h,m}\Delta_t^c - 2\pi \frac{f'_{h,m-1} + f'_{h,m}}{2}\Delta_t - \phi'_{0,h,m-1}\right). \quad (3.8)$$

VOICE MODEL RESIDUAL ENVELOPE CONCATENATION

In order to avoid spectral shape discontinuities at the segment boundaries we can make use of the EpR model. First, we estimate the EpR of the boundary frames. The left EpR is then stretched using the resonance frequencies as mapping points ($F0_{left}$ *to* $F0_{right}$, $F1_{left}$ *to* $F1_{right}$,...) and it is subtracted from the right EpR. The differential envelope obtained accounts for the spectral shape differences between the two joint frames.

For each one of the transition frames at the left of the boundary, a frequency warping function is obtained from the interpolation between the above resonance frequency mapping and the identity mapping ($y=x$) with a factor *1-SSIntp*. This factor stands for the distance from the frame to the boundary (*SSIntp* is 0 at the beginning of the interpolation zone and 1 at the boundary). The frequency warping function is applied to each frame spectral amplitude envelope and finally the differential envelope (weighted by *SSIntp*) is added to it. The process is illustrated in Figure 3.25.

Note that the spectral shape interpolation is spread along several frames in a similar way to the phase concatenation, but with the addition of the spectrum scaling.

Informal listening tests show that smoothing the voice features indeed smooth the synthesized transitions and increases the feeling of a continuous flow. For example, in Figure 3.26 we can observe two rendered waveforms of the same input score, (a) without and (b) with concatenation smoothing applied. We can clearly notice how the evident waveform discontinuities in (a) appear significantly minimized in (b).

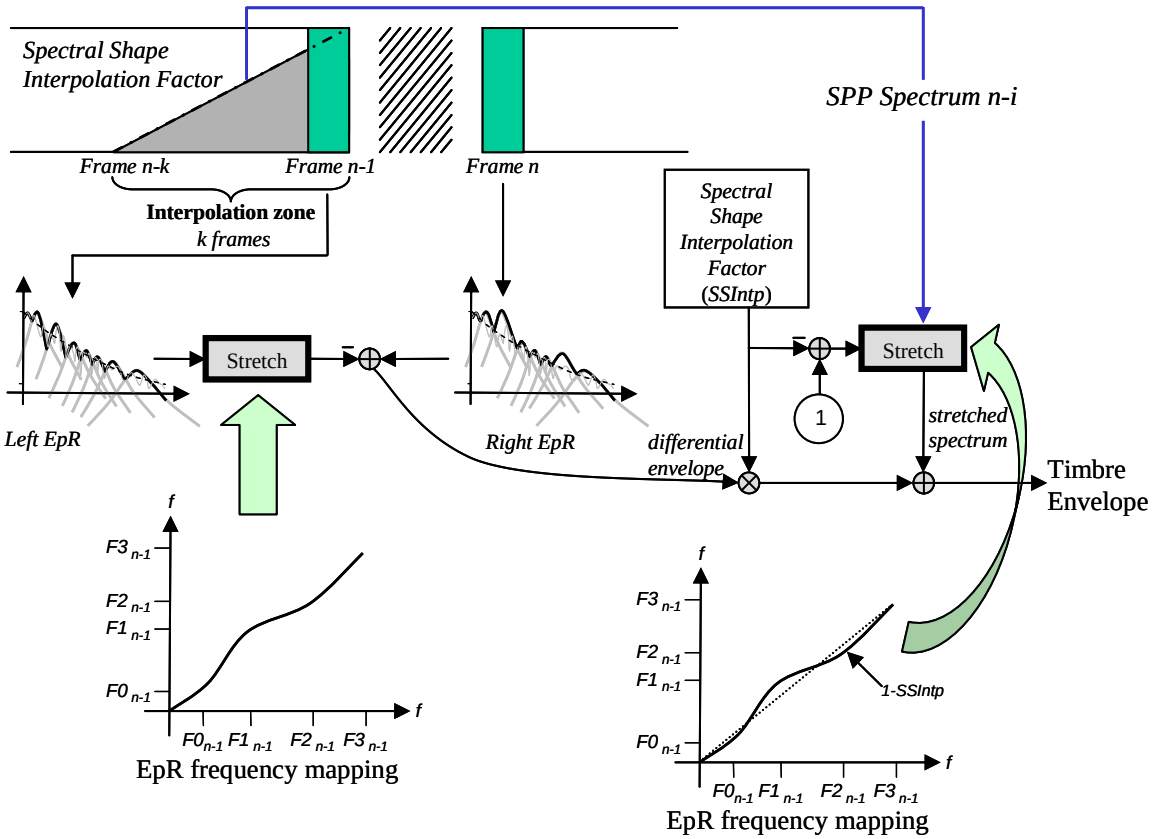


Figure 3.25 Timbre concatenation using EpR

Figure 3.26 Comparison between the waveform of two synthesis excerpts. In the top one (a), no concatenation smoothing of descriptors is applied, and we can observe several discontinuities at joint boundaries. In the bottom figure (b), smoothing of the concatenation has been applied and clearly the waveform discontinuities has been minimized.

3.6 Evaluation

It is difficult to come up with an objective evaluation of singing voice synthesis. Several approaches have been successfully applied to speech, resulting into a set of evaluation algorithms that measure the perceptual quality of speech signals (PESQ[37]). However, singing voice is quite different from speech, and its evaluation requires some specific strategies (e.g. (Garnier, et al. 2007)), also when considering synthetic singing (Rodet 2002). For instance, intelligibility is less important in singing than in speech, while expression becomes more relevant. In other words, pronunciation is partly used as an expressive asset, therefore with an aesthetic purpose. In addition, in speech prosody is determined by the text and its meaning, having the grammatical function of each word an essential role in setting the pitch curve. By contrast, in singing the pitch envelope is mostly determined by the song's melody and the expressive resources employed by the singer (e.g. vibratos, scoops, etc). Evaluating singing voice involves considering together several aspects such as expression, timbre, singing technique, naturality and tuning. It is not our intention to pursue an objective test of real and synthetic singing, or to study the link between acoustic descriptors and perception. What we attempt is to obtain useful data about different perceived attributes of both singing synthesis and real performances, and use them to rate and compare several synthesizers. Having this in mind, we have designed a listening test. Next, we detail this experiment and discuss the obtained results.

3.6.1 Test design

We have performed a listening test with the aim of evaluating and comparing real singer performances with different singing synthesizers. The test was carried out in the recording studio of the Institut Universitari de l'Audiovisual[38], in the Universitat Pompeu Fabra. We installed a computer, two speakers and twelve chairs in the recording room.

Subjects listened to 24 audio excerpts that contained both singing and background music. They were asked to rate the excerpts according to different criteria: expressiveness, naturalness, pleasantness, nasality, quality, tuning, pronunciation, understanding of lyrics and liking. For rating them, they used a scale from 1 to 5. They were also asked if they knew the song, if they recognized the singer as the same one in the previous excerpt, and if they think it was a synthesized voice. For these last questions, the options were *no*, *not sure* and *yes*. An example of the questionnaire for one of the excerpts is presented in Table 3.5. In total, there were 12 questions (Q1 to Q12) for each song.

Before each audio excerpt, they listened to a voice announcing the audio index number: "audio file 1", "audio file 2", etc. Each audio excerpt was repeated twice. Subjects had approximately one and a half minutes to answer the questions of each audio file, including the listening time. Questions were answered on paper sheets available in the listening room, and subjects took some time before the listening test to get familiarized with the questions. Finally, they also answered a set of general questions about their background and musical skills, listed in Table 3.6.

The audio excerpts were chosen to cover different techniques and systems for audio synthesis and included as well real recordings. The excerpts and their corresponding descriptions are presented in Table 3.7. The order of appearance was modified for different sessions (groups of subjects), so to reduce the effect of the order on the evaluation results. In addition, the first two audios were chosen to be two clear cases of singing synthesis and real singer performance, but were not considered in the posterior results evaluation. The subjects were not aware of this. We included them in the test so to help the subjects not only to get familiarized with the process itself but also to fix the limits of their rating scale.

[37] http://www.pesq.org/

[38] http://iua.upf.edu/recursos/laboratorios/audio/

AUDIO TRACK NUMBER *id*
Q1: "The voice sounds expressive (showing a particular feeling)"
1 2 3 4 5
inexpressive expressive
Q2: "The voice sounds natural (contrary to artificial)"
1 2 3 4 5
artificial natural
Q3: "The voice sounds pleasant"
1 2 3 4 5
unpleasant pleasant
Q4: "The voice sounds nasal"
1 2 3 4 5
not nasal nasal
Q5: "The singer is a good singer"
1 2 3 4 5
bad good
Q6: "Do you think the performance is in tune?"
1 2 3 4 5
out of tune in tune
Q7: "The words are pronounced correctly (not artificially or in a weird way)"
1 2 3 4 5
artificially correctly
Q8: Do you understand the lyrics?
1 2 3 4 5
nothing everything
Q9: "Do you like the song?"
1 2 3 4 5
not at all very much
Q10: Do you think this is the same singer as the one in the previous audio excerpt?
No / Not sure / Yes
Q11: "Did you know the song before listening to it?"
No / Not sure / Yes
Q12: "Do you think this is a synthesized voice?"
No / Not sure / Yes

Table 3.5 Questions for each audio excerpt

GENERAL QUESTIONS
Survey Starting time: 10h / 11h / 12h / 13h / 14h / 16h / 17h / 18h / 19h
Age: 0-9 / 10-19 / 20-29 / 30-39 / 40-49 / 50-59 / 60-69 / 70-79 / 80-89 / 90-99
Country:
Musical Training and Listening Habits
1. Have you had musical education in the Conservatoire or in a similar school/academy? Yes / No
2. How many years have you studied? 0 / 1-2 / 3-4 / 5-6 / 7-8 / 9-10 / more than 10 years / more than 20 years
3. What musical instrument(s) did you study?
4. Did you study Solfege as well? Yes / No
5. What musical instrument(s) do you play?
6. How much time do you spend attentively listening to music every day? 0.5 hours / 1-2 hours / more than 2 hours
7. How much time do you spend listening to music in the background every day? 0.5 hours / 1-2 hours / more than 2 hours
8. What type of music do you listen to?
Global questions for the whole listening
1. Have you ever listened to a synthesized singing voice before this survey? Yes / No / Not sure
2. What is your English language understanding level? nothing / basic / intermediate / advanced
3. What is your Japanese language understanding level? nothing / basic / intermediate / advanced
4. What is your Spanish language understanding level? nothing / basic / intermediate / advanced

Table 3.6 General questions

id	audio ref	audio name	type	language	gender	comments
A1	[256]	antonio carlos jobim–fascination rhythm	real singer	English	male	bossanova style, brasilean singing in English
A2	[257]	springsong NTT	NTT synthesizer	Japanese	female	NTT Synthesizer, Mendelssohn's "Spring Song" from https://ssli.ee.washington.edu/people/duh/projects/singing.html
A3	[258]	female synth FLINGER	FLINGER synthesizer	English	female	Flinger synthesizer (http://cslu.cse.ogi.edu/tts/flinger), song "Search" by Alex.B.Kain
A4	[259]	female synth CANTOR	CANTOR synthesizer	English	female	VirSyn Software Cantor Synthesizer (http://www.virsyn.de), song "I can do all things trough Crhist" by Gerry Rzeppa
A5	[260]	ansiedad male synth	thesis synthesizer	Spanish	male	expression includes vibratos, scoops and dynamics
A6	[261]	ansiedad male singer	real singer	Spanish	male	singer from whom the previous example was synthesized
A7	[262]	ansiedad male another singer	real singer	Spanish	male	different singer than the two previous examples
A8	[263]	kimi no u wasa male synth	thesis synthesizer	Japanese	male	deep and time-varying vibratos
A9	[264]	hoffnung opera male synth MERON	Y. Meron's synthesizer	German	male	performance driven synthesis, from (Meron 1999)
A10	[265]	days of wine and roses female synth	thesis synthesizer	English	female	flat singing, just one vibrato at the end
A11	[266]	choir synth SUNDBERG	CHANT synthesizer	la-la-la	male choir	deep vibratos, several male voices singing la-la-la
A12	[267]	choir synth	thesis synthesizer	u-u-u	mixed choir	no vibratos, /u/ vowel, legato transitions
A13	[268]	feelings female synth	thesis synthesizer	English	female	deep and constant-rate vibratos
A14	[269]	ansiedad female synth	thesis synthesizer	Spanish	female	expression includes vibratos, scoops and dynamics
A15	[270]	ansiedad female singer	real singer	Spanish	female	singer from whom the previous example was synthesized
A16	[271]	out of life male synth VocalWriter	VocalWriter synthesizer	English	male	Kaelabs software VocalWriter, song "She is out of my life" from http://kaelabs.com/vocalWriter_demos.html
A17	[272]	besame mucho female synth1	thesis synthesizer	Spanish	female	soft dynamics, few expression resources
A18	[273]	besame mucho female synth2	thesis synthesizer	Spanish	female	same as previous example but with normal dynamics
A19	[274]	besame mucho male synth	thesis synthesizer	Spanish	male	few expressive resources, deep and constant-rate vibratos
A20	[275]	bossanova child singer	real singer	English	child	out of tune, somehow irregular timings
A21	[276]	japanese female synth	Vocaloid synthesizer	Japanese	female	few expression resources
A22	[277]	mozart opera female synth CHANT	CHANT synthesizer	vowel	female	well-known example of opera singing, from (Rodet, et al. 1984)
A23	[278]	jazz female singer	real singer	English	female	Jazz recording
A24	[279]	VocaListener Dearest	Vocaloid synthesizer	Japanese	female	Vocaloid example using Vocalistener technology

Table 3.7 Audios played in the listening test

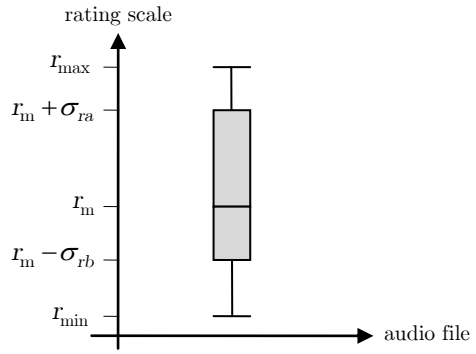


Figure 3.27 Representative example of how statistics are drawn.

3.6.2 Results

During one day, several listening sessions started each hour from 10am to 19pm and lasted for about 40 minutes each. In total 50 people participated in the test. Roughly, half of them were master students while the other half were researchers of the Music Technology Group (MTG). Moreover, some subjects were external people with no experience at all in computer music. Figure 3.28 and Figure 3.29 show some interesting user profile statistics. Most subjects were aged between 20 and 29 years old, and one third between 30 and 39. There was a varied number of countries of origin, although more than 40% of the subjects were Spanish. Around a 60% of people had received musical education in a Conservatoire or in a similar school, and almost half of them for more than 6 years. Regarding the played instrument, interviewed subjects played different instruments, such as piano, guitar, percussion, saxophone, bass, double bass, violin, accordion and jaw harp. Some of them were also amateur singers. Regarding the music listening habits, more than half of the people attentively listened to music for at least one hour every day. In addition, 55% of the subjects listened to music in the background for more than two hours every day. Subjects listened to a varied set of musical genres, including classical music, jazz, pop, rock, electronic and world music (flamenco and others). Regarding English language understanding level, most people had an intermediate or advanced level. By contrast, regarding Spanish language, a 20 % of people had a basic or negligible level.

Figure 3.30 to Figure 3.33 show the statistical analysis of the evaluated characteristics for each of the considered audio excerpts. Statistics are drawn with a black vertical line that spans from the minimum r_{min} to the maximum r_{max} rated value, and two boxes that indicate the mean rating value (r_m) and the standard deviation of ratings above and below the mean (σ_{ra} and σ_{rb} respectively). Those statistics are computed as follows:

$$r_{min} = \min(r_i) \; \forall i \in [1,2,...,I] \qquad r_{max} = \max(r_i) \; \forall i \in [1,2,...,I]$$

$$r_m = \frac{\sum_{i=1}^{I} r_i}{I} \qquad \sigma_{ra} = \sqrt{\frac{\sum_{i=1}^{I} (r_m - r_i)^2 \cdot (r_m > r_i)}{\sum_{i=1}^{I} (r_m > r_i)}} \qquad \sigma_{rb} = \sqrt{\frac{\sum_{i=1}^{I} (r_m - r_i)^2 \cdot (r_m < r_i)}{\sum_{i=1}^{I} (r_m < r_i)}} \tag{3.9}$$

where I denotes the number of subjects and r_i the rating given by the i^{th} subject. Figure 3.27 shows an example of the statistical representation. In addition, boxes are colored according to whether audios correspond to real performances (in gray), synthesis results from this dissertation (in blue), synthesis created by Vocaloid users (in orange), or synthesis performed with other synthesizers (in green). From now on, SP refers to singer performances, DS to dissertation synthesis, VS to Vocaloid synthesis and OS to alternative synthesis methods. We discuss next the results obtained for each question.

Q1: "The voice sounds expressive (showing a particular feeling)", Figure 3.30a

SP excerpts are rated as the most expressive ones with mean values around or above 4, although the song sung by a child (*bossanova child singer*) is rated with an average value of 3. DS examples with manual rich expressive controls (*ansiedad male* and *female, kimi no u wasa*) were rated with means above 3.5, while examples with less detailed control were rated around 3. VS examples obtained results slightly below 3. OS excerpts were rated around 2.5 with two exceptions: the famous CHANT synthesis of the *Queen of the Night* (around 3.6) and the example OS *Hoffnung* from Y.Meron's dissertation(Meron 1999), rated around 3.3. Both belong to the opera genre and feature deep vibratos. The former required a lot of tweaking by hand and only synthesizes one vowel, whereas the latter is controlled by a human performance. Results obtained by DS are more challenging in the sense that they use both actual words and manual controls. It is interesting to point out that the only synthetic examples that were never rated with the lowest value were two DS examples: *ansiedad male synth* and *ansiedad female synth*.

Q2: "The voice sounds natural (contrary to artificial)", Figure 3.30b

As expected, SP excerpts are rated as the most natural ones, with ratings above 4. The best-rated synthetic examples are the *ansiedad* male synthesis (DS), slightly below 4, followed by the DS choir excerpt. In the DS group, examples with richer expression generally obtain higher *naturality* ratings. *Kimi no u wasa* is an exception, maybe because the language has some influence on the ratings and it sounds a bit nasal. In the OS group, the CHANT example obtains a 3.5 rating, *Hoffnung* 2.6 and the rest around 1.5. VS examples were rated around 2.5, similar to OS *Hoffnung* and DS excerpts with basic expression. Interestingly, the only synthetic example that was never rated as artificial was DS *ansiedad male synth*. Summing up, DS examples sound in general more natural than the other considered synthesizers.

Q3: "The voice sounds pleasant", Figure 3.30c

SP examples are again rated as the most natural ones, with ratings around 4, despite of the child example, probably due to its particular expression and the presence of some out of tune notes. The best-rated synthetic excerpts are, as before, the DS *ansiedad* male synthesis (3.7) and the DS choir excerpt (3.8). Then CHANT and DS *kimi no u wasa* excerpts obtain a similar 3.3 rating. The only synthetic examples that were never rated as unpleasant are DS *ansiedad male synth*, DS *kimi no u wasa*, and DS *choir synth*. Without considering the CHANT example, DS sounds in general more pleasant than OS and VS. Nevertheless, synthetic examples are not rated in average as unpleasant (the minimum rating is 2.5).

Q4: "The voice sounds nasal", Figure 3.31a

SP excerpts are rated as the least nasal ones, with ratings below 2, with the exception of the child voice. Regarding synthetic examples, the least nasal ones are DS *ansiedad* male, DS choir and OS CHANT. In general, most synthesis excerpts are rated around 3. Note that the synthetic examples never rated as nasal are all DS excerpts: *ansiedad male synth*, *kimi no u wasa*, *days of wine and roses female synth*.

Q5: "The singer is a good singer", Figure 3.31b

Most SP examples are judged to be performed by good singers. The exception is again the child example, rated as the worst singer in the test. The best synthesis singer is CHANT, with its rich operatic vibratos, followed by DS choir, DS *ansiedad* male, and DS *kimi no u wasa*. The rest DS, OS and VS excerpts are rated with values around or below 3.

Q6: "Do you think the performance is in tune?", Figure 3.31c

OS CHANT example was judged to be the best tuned one, with a mean 4.7 score, followed by SP *jazz* and *ansiedad*, and then DS *choir* and *kimi no u wasa*. Most of the other excerpts were rated between 3.5 and 4. The SP *child* example was the worst rated with 2.3.

Q7: "The words are pronounced correctly (not artificially or in a weird way)", Figure 3.32a

Here we do not consider the synthesis of a single vowel (OS CHANT and DS *choir*). The best-rated excerpts are the adult SP examples, followed by the DS and SP *child* excerpts, then VS and finally OS examples. The worst rated ones were CANTOR, FLINGER and VocalWriter examples. Note that the only synthetic excerpts that were never identified with artificial pronunciation were DS *ansiedad male synth*, DS *besame mucho female synth 2* and VS *Japanese female synth*.

Q8: "Do you understand the lyrics?", Figure 3.32b

SP lyrics were all in English and well understood, all but the *child* rated above 4. Note that almost all test subjects had an intermediate or advanced English understanding level. OS Vocalwriter, FLINGER and DS *Feelings* obtained ratings between 3 and 4. DS *days* scored worse (2.5), although using the same singer as DS *Feelings*, maybe because the singing is on top of a flute. OS CANTOR was almost not understood. In turn, Japanese examples were not understood at all (ratings around 1.5), what was expected, as none of the subjects understood Japanese. The same probably applies to the German example (OS *Hoffnung*), although we did not ask subjects about their German knowledgd. By contrast, all DS Spanish excerpts were well understood, with ratings around 4. Figure 3.34a shows the distribution of user ratings with respect to their understanding of the language for all Spanish excerpts. There is a high correlation between lyrics and language understanding. This leads us to think that Spanish pronunciation in synthesis examples is quite correct. By contrast, Figure 3.34b shows an analogous distribution for the English language. The correlation between lyrics and language understanding is not as clear as for the Spanish case, but for almost all cases, lyrics understanding decrease with language understanding.

Q9: "Do you like the song?", Figure 3.32c

Here most songs were rated around 3, with no observed relevant differences.

Q10: "Do you think this is the same singer as the one in the previous audio excerpt?", Figure 3.33a

The motivation for this question was to evaluate if the listeners recognized real and virtual singers as the same one when comparing original performances to synthesis generated from databases obtained from the same singer. This applies only to a few excerpts in the survey. DS *Ansiedad male synth* (a) and SP *ansiedad male singer* (b) come from the same singer, whereas SP *ansiedad male another singer* (c) corresponds to a different one. In all listening sessions, (b) was played after (a), and (c) after (b). The ratings show that most subjects recognized (a) and (b) as from the same singer, with a mean between *not sure* and *yes*. By contrast, (c) was mostly identified as a different singer. Another case where this question applies is to DS *ansiedad female synth* (d) and SP *ansiedad female singer* (e). In this case, the mean answer is slightly above *not sure*. Finally, DS *besame mucho female synth 1* (f) and *2* (g) use the same database, but the former sings with *soft* dynamics whereas the latter with *normal* dynamics. Here the average answer is again slightly above *not sure*. Summing up, for Spanish DS male synthesis the singer was mostly recognized as the same one, whereas in the case of female synthesis test participants were not sure.

Q11: "Did you know the song before listening to it?", Figure 3.33b

Only two songs were known by almost all subjects: the excerpt of the Mozart's opera *The magic flute* (OS *CHANT*), and *Bésame mucho*. Conversely, three songs were unknown by everybody: OS *hoffnung*, VS *Japanese female synth*, and VS *Vocalistener Dearest*. For the rest of songs, mean values lay between *no* and *not sure*.

Q12: "Do you think this is a synthesized voice?", Figure 3.33c

This question is very interesting, as listeners have to decide whether they perceive the excerpt as synthetic or as a real performance. Any small artifact, strange timbre or slightly unnatural expression might lead listeners to rate a given example as synthetic. Therefore, a synthesis has to be convincing in most if not all facets in order to mislead a participant. DS *ansiedad male synth* (a) and OS *CHANT* (b) were perceived as real performances by many people, getting an average rating just slightly above *not sure*. Concretely, 34% of participants identified (a) as a real performance and another 26% were not sure. In turn, 30% of subjects thought (b) was a real singer and another 24%

doubted. The next more convincing synthesis were DS *ansiedad female synth* and DS *choir*, followed by DS *besame mucho female synth 2* and VS *Vocalistener Dearest*. It is interesting to note that nobody judged OS *FLINGER*, OS *CANTOR*, OS *choir* or OS *VocalWriter* excerpts to be real performances. Overall, DS and VS examples were perceived as less synthetic than OS excerpts.

DISCUSSION

In most questions, the proposed synthesizer excerpts were rated in average higher than the other synthesis examples. This is true in fundamental aspects such as expressivity, naturalness, pleasantness, and pronunciation. We then believe that these results indicate that the synthesis method proposed in this thesis makes relevant contributions to the state of the art in singing voice synthesis.

However, there is still plenty of room for improvements in most research areas. This is clearly indicated by the fact that singer performances are still better rated than most if not all synthesis excerpts. In addition, although some of the excerpts we generated were highly rated, other ones did not get such good results, especially the ones were expression was not much tweaked. This indicates that the basic expression generated by the Performer Model does not sound realistic yet, so that the synthesizer user still plays an important role for achieving convincing singing synthesis results.

The best generated example is with no doubt *ansiedad male synth*, which convinced one third of the test subjects to be a real performance, while almost another third doubted. We believe this synthesis result is much more challenging that the well-known Mozart's aria example synthesized with CHANT, probably the best singing voice synthetic example till now. The main reasons are that our synthesizer is singing sentences, not just a vowel, and did not required as much manual tweaking as the operatic one. Actually, one day was enough to generate the synthesis score including expressive resources such as vibratos, scoops and dynamics envelopes. Regarding pronunciation, Figure 3.34c shows the distribution according to the Spanish understanding level for the answers to question Q12 (*do you think this is a synthesized voice?*) about the *ansiedad male synth* example. Note that almost all subjects with basic or negligible Spanish knowledge identified the excerpt as a real performance, and only half of the subjects with intermediate or advanced level thought it was synthetic. This indicates that pronunciation still needs some further improvements.

Another relevant aspect is that many people correctly recognized synthetic singers as the same real singers from which the synthesizer database was created. This was one of our main dissertation goals: emulate the characteristics of a real singer. However, many people also doubted, so still more research has to be devoted to improve the singer modeling.

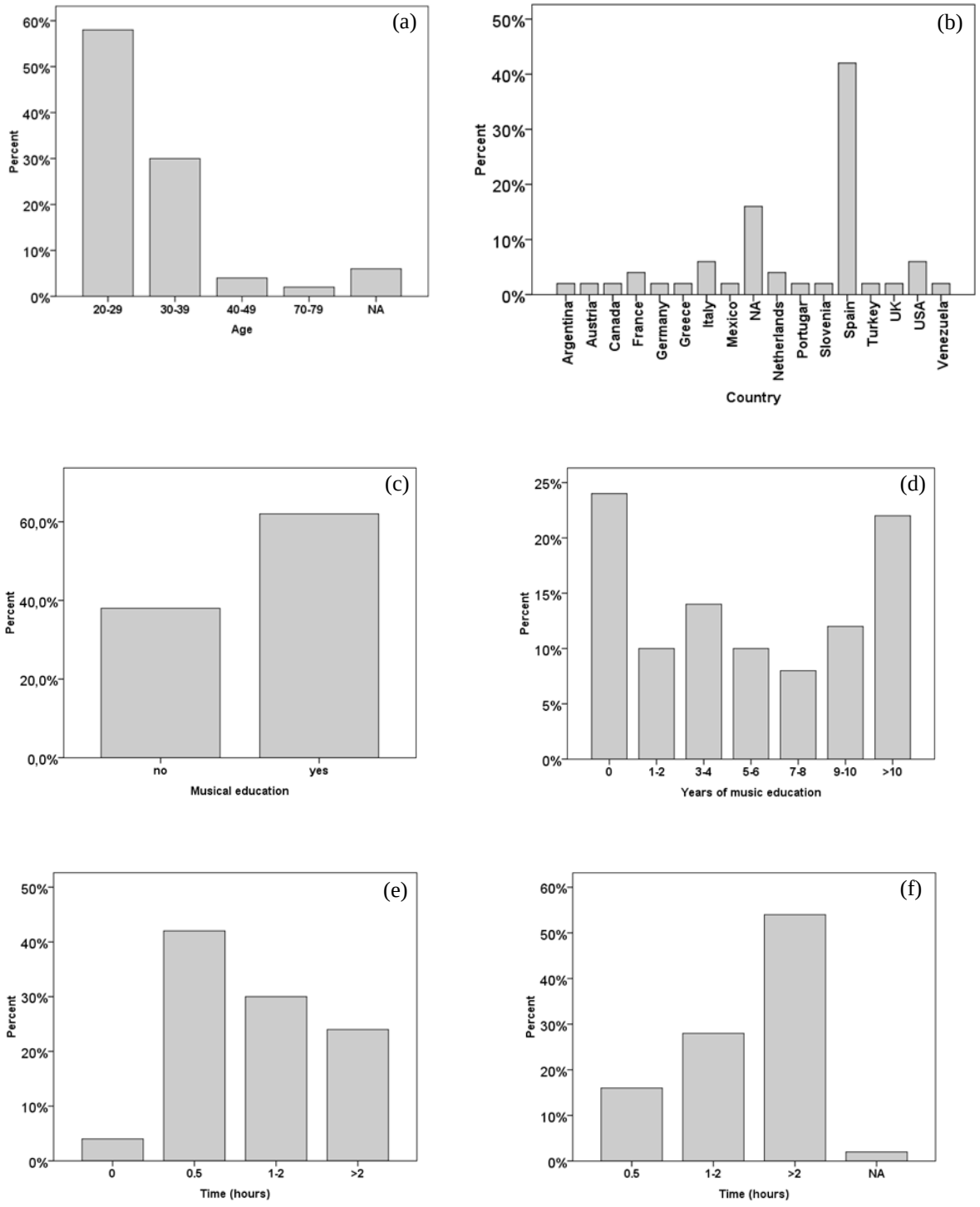
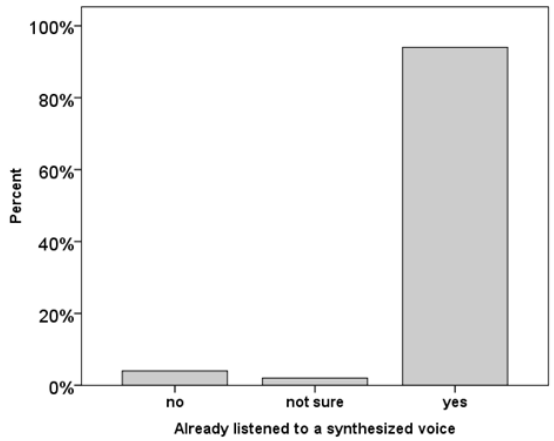
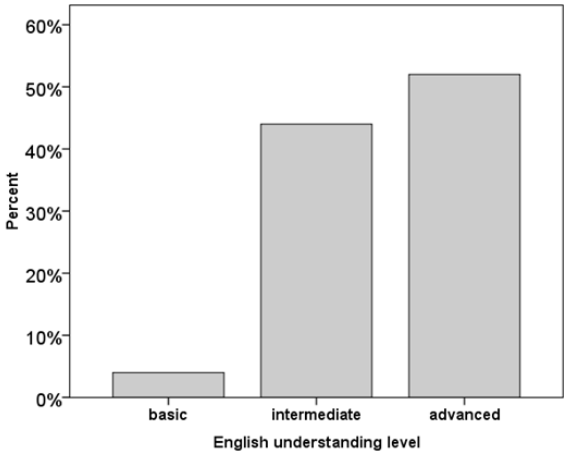


Figure 3.28 User profile statistics for different global questions of the survey: (a) age, (b) country, (c) whether the subject studied music or not, (d) how many years the subject studied music, (e) time spent during a day attentively listening to music, and (f) hours spent in a day listening to music in the background.

(a)



(b)



(c)

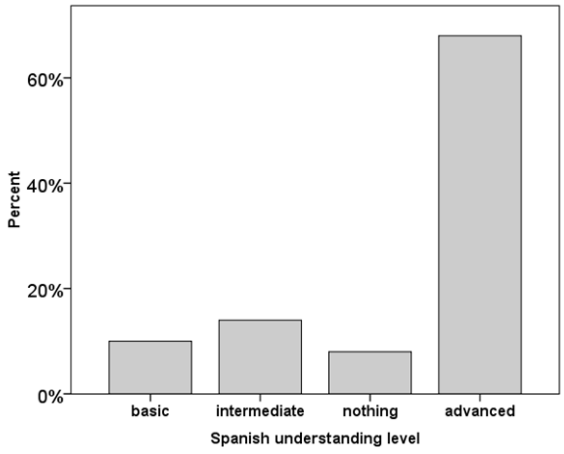


Figure 3.29 User profile statistics for different global questions of the survey: (a) whether the subject had previously listened to synthesized voices, (b) English understanding level, and (c) Spanish understanding level.

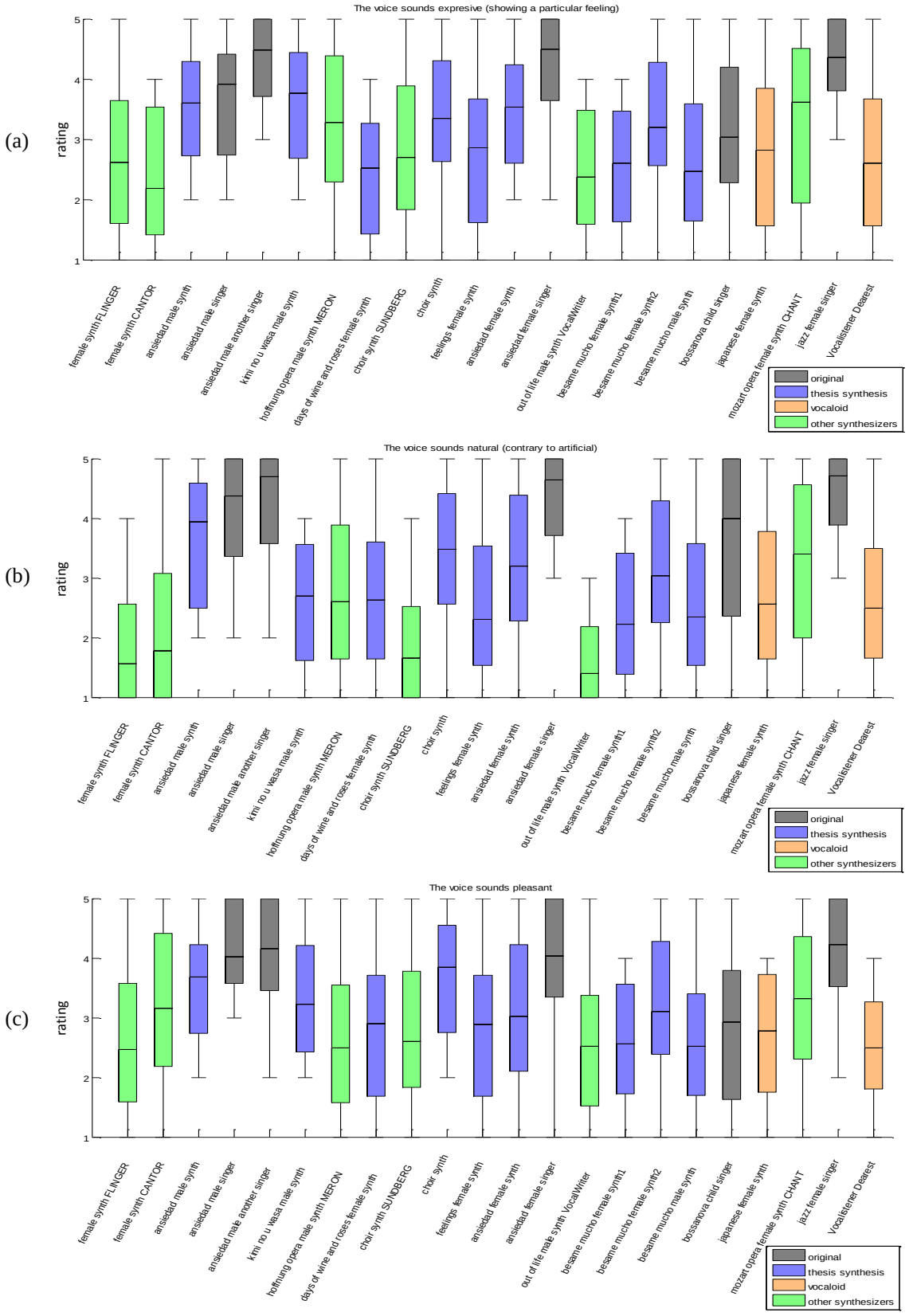


Figure 3.30 Audio ratings statistics for questions 1 to 3.

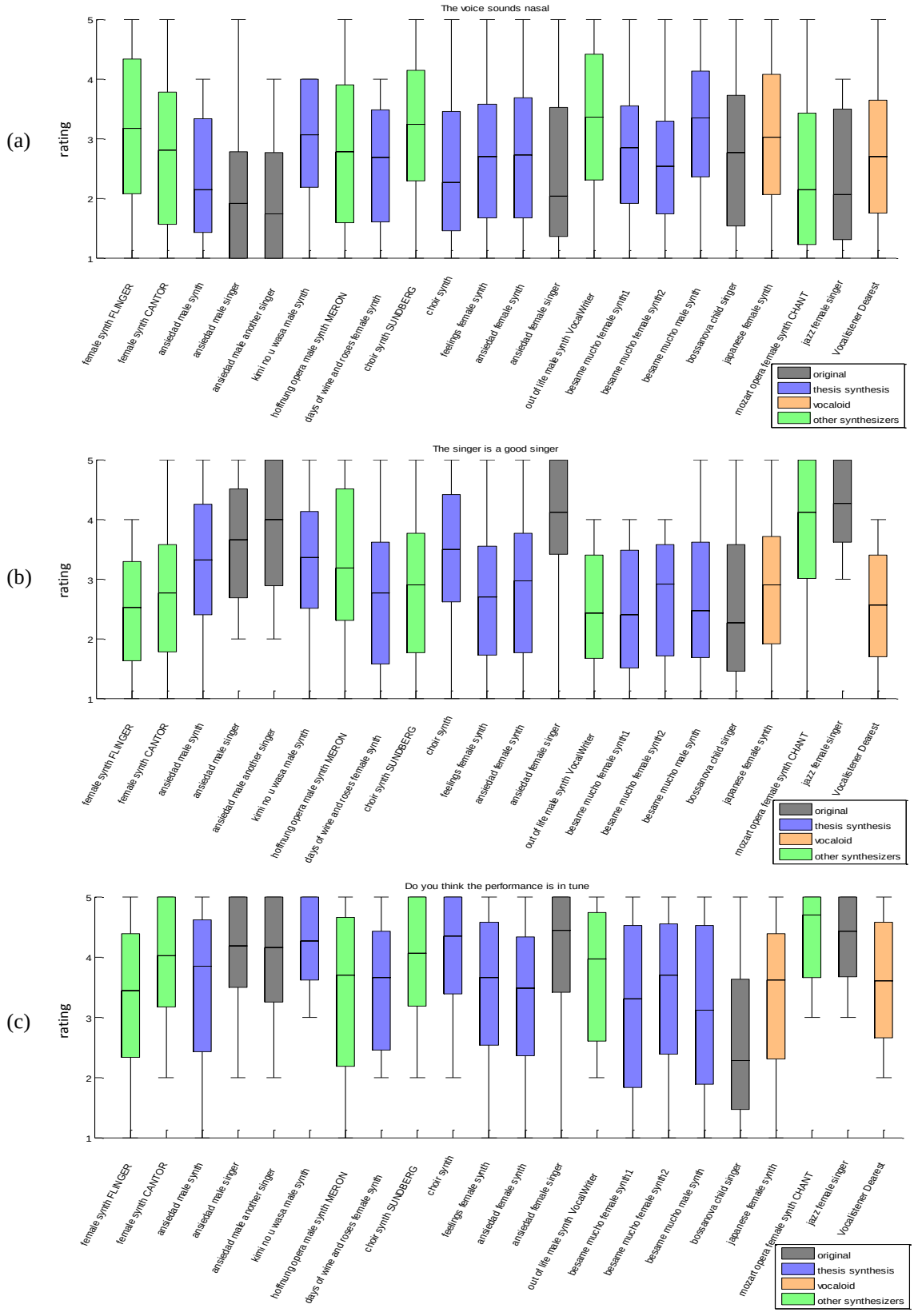


Figure 3.31 Audio ratings statistics for questions 4 to 6.

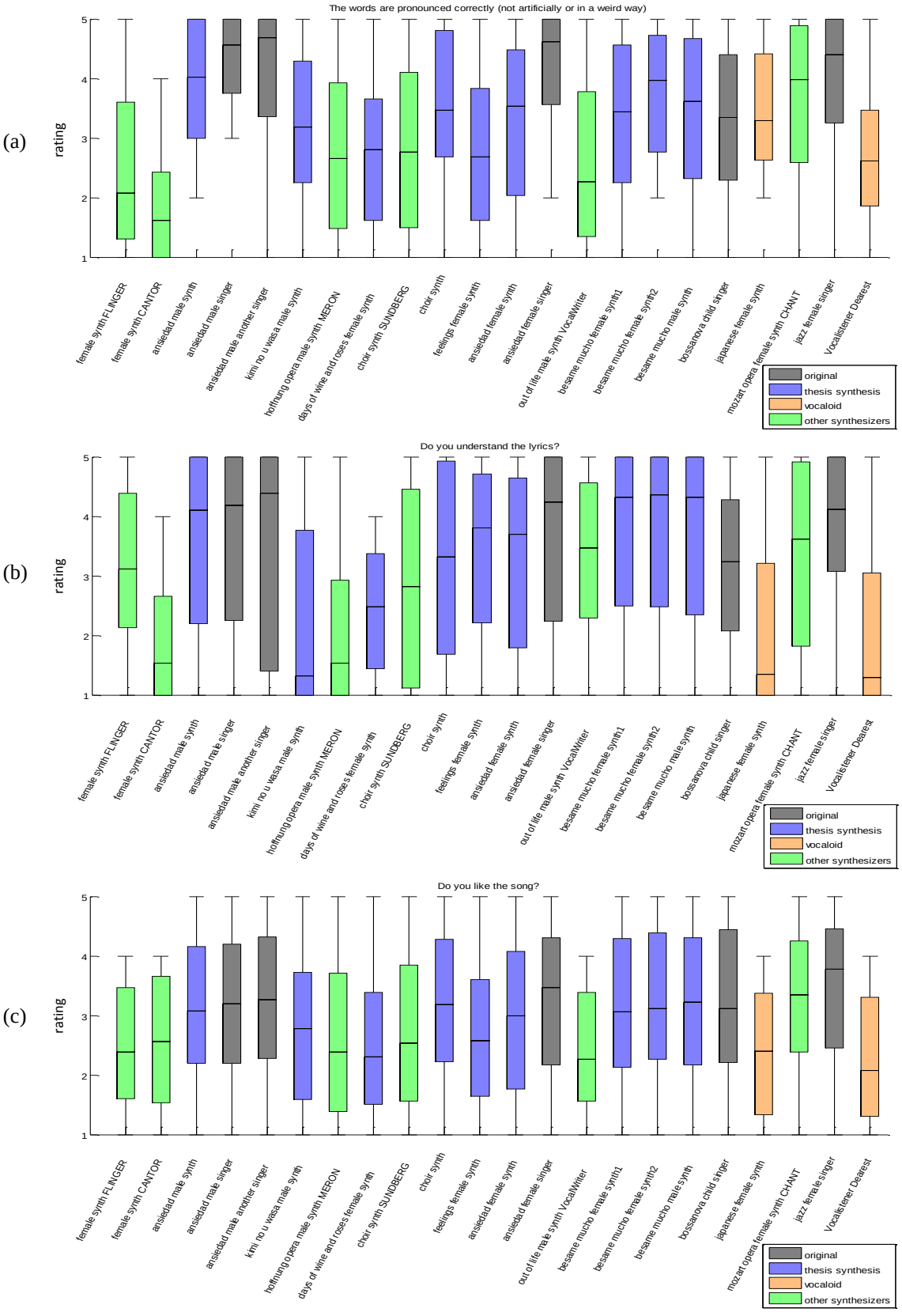


Figure 3.32 Audio ratings statistics for questions 7 to 9.

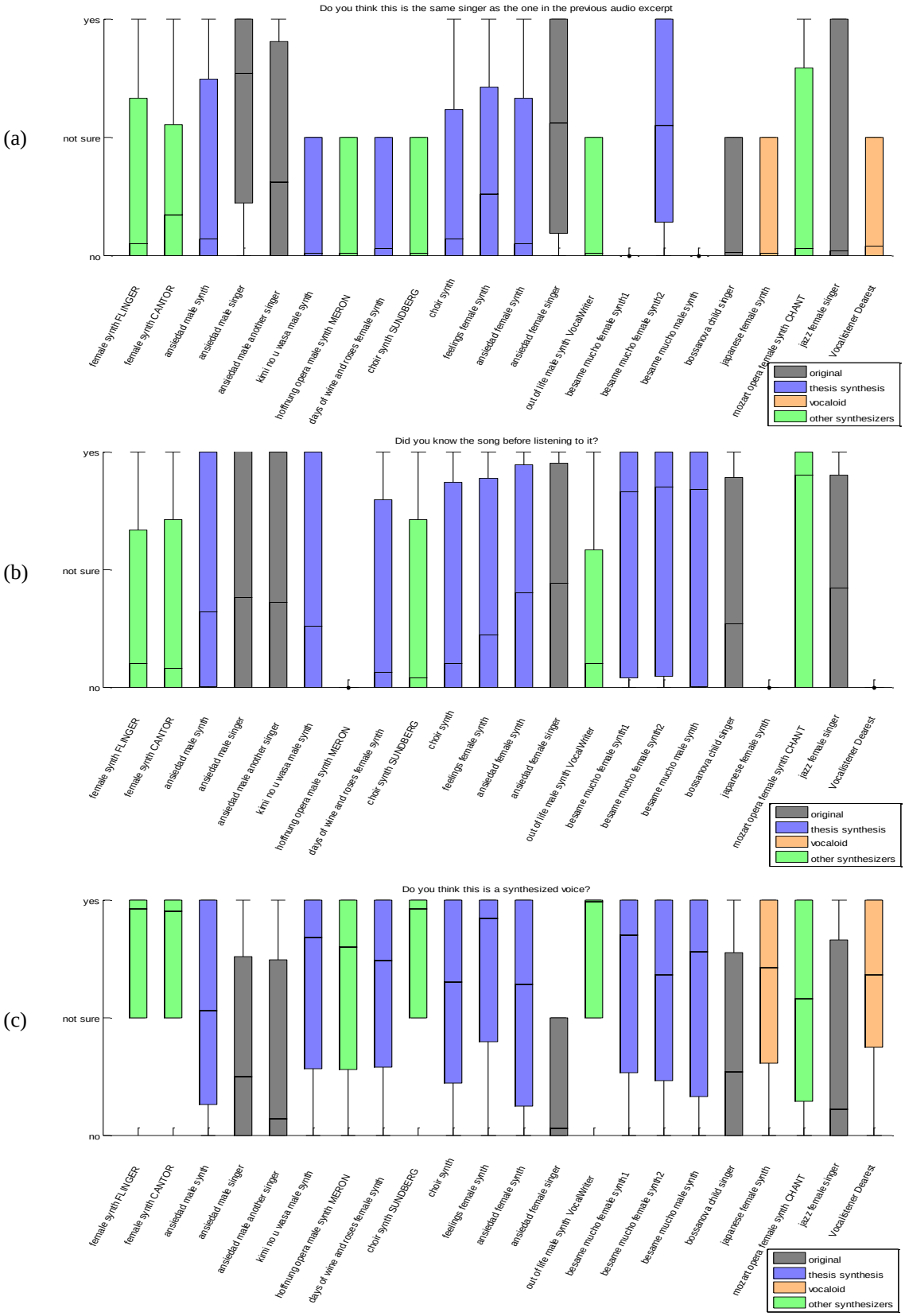


Figure 3.33 Audio answer statistics for questions 10 to 12.

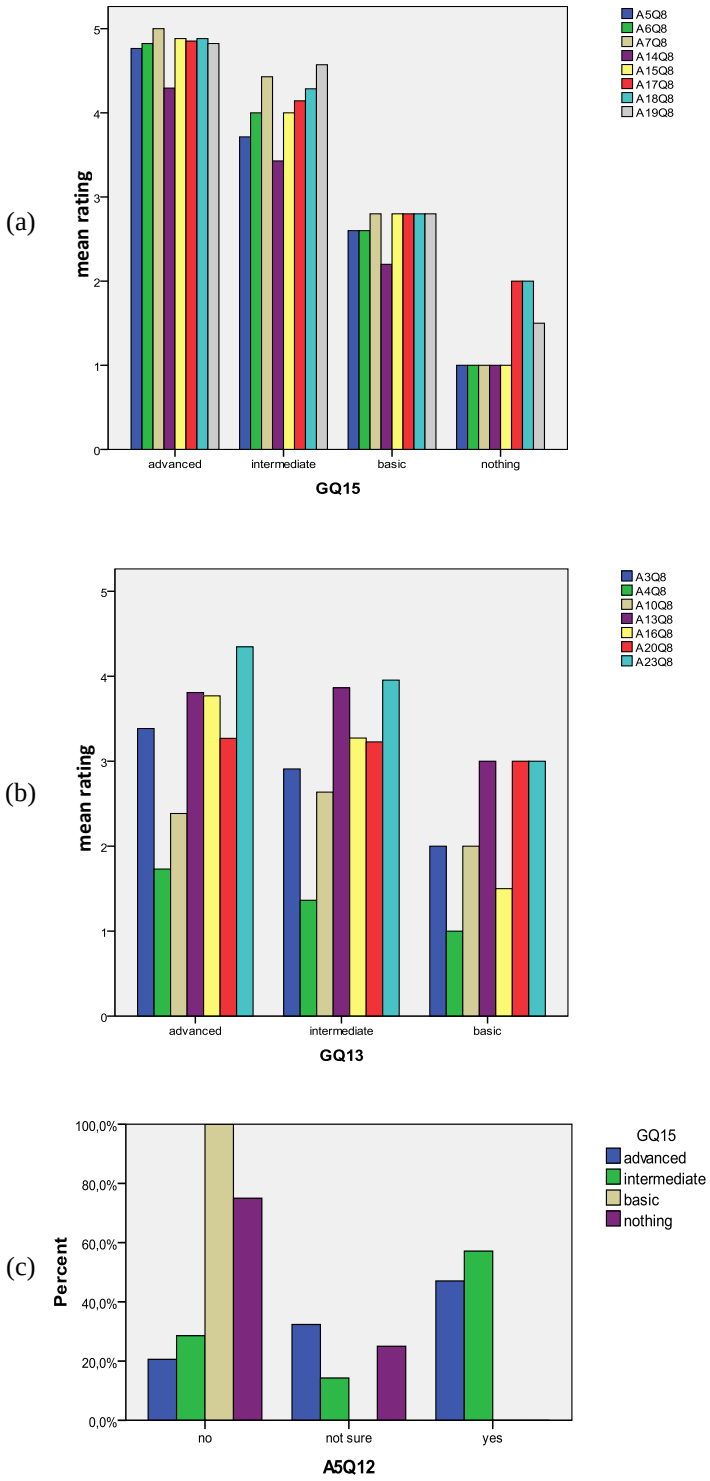


Figure 3.34 Distribution of user ratings for *lyrics understanding* question according to their language understanding level. (a) shows the results for Spanish excerpts, (b) for English excerpts. In the legends, Q8 refers to the question and A*id* to the excerpt (see Table 3.7). (c) shows the Spanish understanding level distribution for the answer to question Q12 (*do you think this is a synthesized voice*) regarding excerpt DS *ansiedad male synth*.

3.7 Conclusions

In this chapter, we have detailed our strategies for synthesizing singing voice. Our aim from the beginning has been to imitate the voice of a particular singer. Therefore, we discarded approaches based on generic models, such as using standard formant parameters for producing different phonemes. Instead, we have based our approach on concatenative synthesis, i.e. on transforming and concatenating prerecorded samples from the target singer.

At the beginning of this chapter, we have introduced the concept of performance based sampling synthesis. We have explained that we are actually modeling the sonic space of a performer-instrument combination, even in the case of a singer, and we have discussed in section 3.1 the different modules required to build such kind of synthesizer. Next, in section 3.2 we have detailed the steps to create a singing performance database. The first thing to do is to study the specific sonic space of the performer-instrument combination we are targeting. We also pointed out that it is better to use sonic space dimensions that are close to perceptual dimensions. We then proposed to divide the singing sonic space in different subspaces, so to capture separately phonetic and expressive aspects. Next, we proposed strategies for defining the sampling grid, taking into consideration recording sessions and database characteristics together with the target quality to achieve. We always tried to propose strategies general enough as to be used for creating a wide range of databases of different singers in different languages. We also pointed out the issues involved during singer recording sessions. In addition, we proposed and implemented several methods to facilitate and automate the database creation, an otherwise intense and time-consuming task.

In section 3.3 we focused on the different aspects involved in the performer modeling task, detailing most common approaches, and discussing which are the performance requirements for synthesizing expressive singing. In the case of our proposed synthesizer we have distinguished two main processes. The former, covered in section 3.4, consists on transforming an input score into a performance trajectory within the sonic space of the target instrument, in our case the singing voice. The latter, detailed in section 3.5, actually generates the output sound by concatenating a sequence of transformed samples that approximates the target performance trajectory.

The Trajectory generation involves computing the optimal sequence of database samples that with minimal transformations better match the target performance trajectory. Sample selection and transformation parameters in concatenative synthesis systems are typically obtained as the optimal solution of a constraint-based problem. Hence, in our case we have proposed a trajectory cost function that considers phonetic timing distance, pitch and loudness transformations and concatenation continuity. We also discussed strategies for off-line and real-time processing scenarios.

The trajectory rendering involves processing samples and smoothing concatenations. It is at this step where all the techniques discussed in the previous chapter 2 are used. We have detailed the different types of sample concatenation discontinuities associated to each synthesis technique, and proposed strategies to deal with them. We have also shown that the phase model proposed in the previous chapter can actually simplify this issue significantly.

Finally, we have performed a listening test intended to compare different synthesizers, our proposed approach, and real performances. We have shown that our system obtains in general better scores than other ones, and it gets closer to the real performances. Moreover, we have generated one example (DS *ansiedad male synth*) that we believe outperforms the most convincing synthesis up to our knowledge, the Mozart's aria synthesized with CHANT. Nevertheless, there is still room for improvements in several aspects since real singer performances are still best rated and distinguishable from synthetic ones.

Chapter 4

Conclusions

In this dissertation, we have introduced the concept of synthesis based on performance sampling. We have explained that although sampling has been considered a way to capture and reproduce the sound of an instrument, it should be better considered a way to model the sonic space produced by a performer with an instrument. With this aim, we have presented our singing voice synthesizer, pointing out the main issues and complexities emerging along its design.

The singing voice is probably the most complex instrument and the richest one on expressive nuances. After introducing its particular characteristics, we have detailed in Chapter 2 several spectral models we developed during the last few years that specifically tackle them, and we have pointed out the most relevant problems and difficulties we found. Since we use concatenative synthesis as the basis of our approach, we have explored how to transform and smoothly connect recorded samples from the target singer. We studied in depth different approaches based on spectral processing that aimed at transforming those voice samples with the best quality and the highest flexibility. In that sense, we justified the necessity of using spectral voice models that are aware and take advantage of the processes involved in voice production.

Therefore, we defined an amplitude spectral voice model able to distinguish between voice source and vocal tract, and which deals with formants to represent the vocal tract filter. The addition of an amplitude residual envelope allows reconstructing perfectly the original singer's spectrum with all its details and nuances, even in the presence of antiresonances due to the coupling between the nasal and vocal tract cavities. This amplitude spectral model provided the required quality and flexibility for transforming voice samples. In addition, we introduced a spectral phase model able to predict the harmonic phase relationship at voice pulse onsets without significantly altering the perceptual timbrical characteristics. Such phase model has been shown to be essential for simplifying and improving the concatenation of consecutive samples at synthesis.

We have shown that spectral voice models can be considered as high-level models that work on top of low-level voice processing techniques. With the aim of combining in a single technique the advantages of typical time-domain and frequency-domain techniques, we introduced the wide-band voice pulse modeling (WBVPM) technique, which provides enough temporal resolution to transform independently each of the voice pulses while at the same time providing an independent control of each harmonic component. This gives us the required flexibility to perform most types of voice transformations with high quality. WBVPM is able to model voice pulses in frequency domain with a set of sinusoids, which represent both harmonic and noisy components. It provides an independent control of each single pulse, thus allowing pulse sequence transformations with easy. This ability is typical of time-domain methods, but complex to achieve in frequency domain, since it implies dealing with complex subharmonics patterns. At the same time, WBVPM's sinusoidal representation of the

signal allows an independent control of each single harmonic component, this way overcoming typical limitations of time-domain techniques. In this sense, WBVPM combines some of the main pros of both time and frequency-domain methods while avoids some of their main drawbacks.

Voice Pulse Modeling approaches presented in Chapter 2 have been implemented in C++ and incorporated to a wide range of applications, most of them running in real-time, including professional audio effects software for voice manipulation, museum installations and videogames. Those applications show the potential and flexibility of the proposed algorithms.

Next, in Chapter 3, we have introduced the concept of sonic space and performance sampling, emphasizing the fact that we are actually sampling the combination of a performer and an instrument, even in the case of the singing voice. We have discussed the key aspects of the proposed synthesizer and described its different components. We have distinguished two main processes. The former consists of transforming an input score into a performance trajectory within the sonic space of the target instrument, i.e. the singing voice. The latter actually generates the output sound by concatenating a sequence of transformed samples that approximates the target performance trajectory. We have put special emphasis on the issues involved in the creation of the synthesizer's database, starting with the definition of the singing voice sonic space and ending with our efforts in automating the creation process. We have based our system in the approach of concatenative synthesis. Therefore, the basic synthesis procedure is to connect snippets of the singer recordings, and modify each of them so to recreate the target performance. Part of our efforts have focused on connecting those sample units in a transparent way, avoiding discontinuities and unnatural transitions, intending to help hiding the fact that the system is concatenating samples and increasing the sensation of a continuous flow. Along this chapter he have shown how to use and adapt the models and processing techniques discussed in Chapter 2 to the specific target of a singing voice synthesizer. In addition, we have proposed models and templates for synthesizing expressive resources. We have shown that most of them can be implemented in the form of parameterized templates and can be obtained from the actual singer performances.

Moreover, we performed a listening test intended to compare our synthesis approach to real performances and to other synthesizers. The results showed that our approach obtains in general better ratings than the other ones, and that it gets closer to the real performances. Some of our synthesis examples even convinced many of the listeners to be authentic, especially *ansiedad male synth*, which we believe outperforms the most convincing synthetic example up to our knowledge, the Mozart's aria synthesized with CHANT. Yet, there is much research to carry out, since real singer performances are still best rated and distinguishable from most synthetic ones.

We have implemented all the research results in an optimized C++ software application for singing voice analysis, modeling, transformation and synthesis, including tools for database creation. In addition, a significant part of our research results in singing voice synthesis have been incorporated to the Yamaha's virtual singer software Vocaloid[39] (see ANNEX A), and several international patents have been applied for by this company including the author as inventor (see ANNEX D).

We consider our work has contributed to improve singing voice synthesis naturalness and expressiveness. Nevertheless, although the current system is able to generate convincing results in certain situations, there is still much room for improvements in most research areas. However, we believe we are not so far from the day when computer singing will be barely distinguishable from human singing performances.

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

Future perspectives

Regarding future research, one of our main interests it to further develop the concepts involved in a synthesizer based on performance sampling and spectral models, especially in the context of the singing voice. One clear direction is to enrich the sonic space definition with more contexts, for instance with longer phonetic sequences, as often made in text-to-speech systems, or increasing the sampling grid. Another interesting direction is to adopt in the singing synthesizer some of the strategies used in the vowel synthesizer presented in §3.1.2 to model the vowel space.

Regarding the voice processing methods discussed in Chapter 2, we have some proposals on future research directions. Concerning harmonic trajectories modeled with spectral regions, one possible direction to explore is the use of a combination of spectral peak shape descriptors and the harmonic spectral envelope to compute the mapping function between harmonic trajectories and spectral regions. Another interesting idea to explore is the use of different mapping functions for harmonic trajectories and spectral regions. Regarding wide-band voice pulse modeling, future research should address the relationship between harmonic and noisy components, which WBVPM only represents with sinusoids. In quasi-stationary conditions, the noise component is the main responsible of the subtle differences between consecutive voice pulses. This noise component introduces amplitude and phase modulations in the harmonic trajectories, which might be statistically modeled and transformed independently of the harmonic component. We believe this approach would improve the sound quality of transformed signals and, at the same time, open new transformation possibilities.

We perceive that the timbre model is still not realistic enough, so further research is required in this direction. Actually, we believe that a timbre model based on statistical analyses of a singer database might improve the synthesis naturalness, and also increase the feeling of a continuous flow. Different machine learning techniques such as Support Vector Regression (SVR), Neural Networks (NNs) or Gaussian Mixtures Models (GMMs) might be good candidates. For instance, GMMs have been successfully applied to the Voice Conversion task, while NN have been used to model instrument timbres in several synthesizers (Lindemann 2007, Pérez, Bonada, et al. 2007). Another interesting area to research is how to provide to the user subtle pronunciation controls. Most synthesizers provide controls on individual formants, but not on aspects such as more opened or closed vowel pronunciation. Machine learning techniques could be also used for this task.

Regarding expression, we wonder how far we can emulate a given singer by training a system using a set of his/her performances. In this direction, we have already carried out some research on the automatic description of musical articulation gestures used in singing voice performances (Maestre, Bonada and Mayor 2006). Our method characterizes fundamental frequency and energy contours by a set of piece-wise fitting techniques and meaningful parameterization. On the other hand, Hidden Markov Models (HMM) have already achieved promising results in synthesizing expressive speech (Nose, et al. 2007). We believe they could also be used to model the singer's expression by probably considering different layers: low-level HMM models would be in charge of the phonetic intonation, whereas high-level HMM models would deal with expressive gestures. The ideal situation would be to be able to learn the most relevant performing characteristics of a particular singer just from a few recorded performances.

Some future research should also be devoted to take advantage of existing models and databases to better recreate the voices of new virtual singers. Having several available singer databases should open new research paths for improving the synthesis quality. In the field of speech voice conversion, we know of some attempts that are capable of learning a new voice just from a few utterances (Toda, et al. 2007). We imagine that it should be possible, using similar approaches, to create a new singer database out of just a few songs of the target artist, alive or not.

Finally, we should extend and apply the concepts of a synthesizer based on performance sampling and spectral models to other instruments. In fact, we are already working in a promising violin synthesizer based on this approach (Pérez, Bonada, et al. 2008).

Summary of Contributions

This dissertation substantially contributes to the field of voice processing and singing voice synthesis:

a) It critically discusses spectral processing techniques in the context of singing voice modeling, and provides significant improvements to the current state of the art.
b) It applies the proposed techniques to other application contexts such as real-time voice transformations, museum installations or video games.
c) It develops the concept of synthesis based on performance sampling as a way to model the sonic space produced by a performer with an instrument, focusing on the specific case of the singing voice.
d) It proposes and implements a complete framework for singing voice synthesis.
e) It explores the sonic space of the singing voice and proposes a procedure to model it.
f) It discusses the issues involved in the creation of the synthesizer's database and provide tools to automate its generation.
g) It performs a qualitative evaluation of the synthesis results, comparing those to the state of the art and to real singer performance.
h) It implements all the research results into an optimized software application for singing voice analysis, modeling, transformation and synthesis, including tools for database creation.

In addition, the outcomes of this research have been published in the form of several papers in international conferences, journals and book chapters, as listed in annexes C and D. Moreover, a significant part of this research has been incorporated to a commercial singing voice software by Yamaha Corp. and the author is inventor of several patents applied by this same company (see ANNEX D).

ANNEX A

Vocaloid[40] Commercial Software

Quoted from MTG's website (http://www.iua.upf.es/mtg/notas_prensa/Vocaloid.htm), last checked on June 2008.

Universitat Pompeu Fabra's research in Voice Processing

results into YAMAHA's virtual singer software VOCALOID

March 18, 2003 — MTG (Music Technology Group) of the Universitat Pompeu Fabra, a research team placed in Barcelona, Spain, well known for its works on audio processing technologies and their musical and multimedia applications has cooperated with Yamaha Corporation, the world's largest manufacturer of musical instruments and a leader in digital audio, in developing a new software named VOCALOID that allows song writers to generate authentic-sounding singing on their PCs by simply inputting the words and notes of their compositions

VOCALOID has been presented by YAMAHA, at the Musikmesse in Frankfurt am Main, Germany, and at the 114th Audio Engineering Society (AES) Convention in Amsterdam, the Netherlands. MTG has contributed largely to the development of VOCALOID especially in its basic research aspects, namely, singing voice processing in frequency domain

The software runs on Windows-based PCs and synthesizes the sound from "vocal libraries" of recordings of actual singers, retaining the vocal qualities of the original singing voices to reproduce real-sounding vocals. VOCALOID vocal-synthesizing software overcomes a major hurdle composers have faced until now due to the limitations that technology has placed on their ability to freely create songs incorporating singing

By just inputting the melody and words on their PCs, users can produce the vocal parts for their pieces with no further work. The synthesized sound retains the vocal qualities of the original singers' voices because VOCALOID synthesizes the sound from "vocal libraries" of recordings of real people singing. The software also features simple commands that allow users to add expressive effects—such as vibrato and pitch bends—to their synthesized vocals. Currently, VOCALOID can generate singing in Japanese and English. Further development of the range of available vocal libraries will make the production of songs using vocals in a wide range of voice qualities possible using VOCALOID

Quoted from vocaloid website (http://www.vocaloid.com), last checked on June 2008.

VOCALOID is a vocal-synthesizing software that enables song writers to generate authentic-sounding singing on their PCs by simply typing in the lyrics and music notes of their compositions. The software synthesizes the sound from "vocal libraries" of recordings of actual singers, retaining the vocal qualities of the original singing voices to reproduce realistic vocals. The software also features simple commands that enable users to add expressive effects - such as vibrato and pitch bends - to their synthesized vocals. Additional releases to the range of vocal libraries currently available will broaden the range of voices and singing styles that can be generated by VOCALOID. VOCALOID can generate singing in Japanese and English. It runs on Windows 2000/XP.

[40] © Yamaha Corp. http://www.vocaloid.com

VOCALOID software is on sale bundled with VOCALOID libraries from soundware companies under licence from Yamaha. Yamaha is not currently planning to sell the actual VOCALOID software engine as a dedicated product. Below are shown several available Vocaloid vocal fonts, both in English and Japanese languages.

Vocaloid Software

The following screenshot shows the first version of the Vocaloid software released in 2003. Several tracks are displayed together in the piano roll, in a sequencer like manner. The track being edited is emphasized and we can observe the lyrics and their phonetic transcription above and below of each note onset. In addition, some expressive controls are associated to different notes, such as crescendos and vibratos.

Vocaloid has obtained several awards:

Electronic Musician Magazine Editor's choice (2005) award for innovation.

Awarded "Best of What's New 2003" from *Popular Science Magazine*.

ANNEX B

Spanish Recording Scripts

SUSTAINS

- 7 pitches
- 4 dynamics (very soft, soft, normal, loud)

[a] paz
[e] pez
[i] pis
[o] por
[u] pus
[B] labio
[l] ala
[L] tallo
[m] ama
[n] nene
[G] lago
[d] dedo
[J] niña

ARTICULATIONS

- 3 pitches (normal, high +5 semitones, low -4 semitones)
- 2 tempos (normal 90 bpm, fast 120bpm)
- 2 dynamics (normal, soft)

001 la señora darling llegó a la ventana
l-a-s-e-J-o-r-a-d-a-r-l-i-n-G-L-e-G-o-a-l-a-b-e-n-t-a-n-a-
adding --> Sil-l s-e e-J J-o o-r r-a a-d d-a a-r r-l l-i i-n n-G G-L L-e e-G G-o o-a a-l l-a a-b b-e e-n n-t t-a a-Sil

002 del archienemigo
d-e-l-a-r-tS-j-e-n-e-m-i-G-o-
adding --> Sil-d r-tS tS-j j-e n-e e-m m-i o-Sil

003 pongamos diez chelines
p-o-n-g-a-m-o-s-d-j-e-T-tS-e-l-i-n-e-s-
adding --> Sil-p n-g g-a a-m m-o o-s s-d d-j e-T T-tS tS-e e-l s-Sil

004 la aurora de nueva york gime
l-a-a-u-r-o-r-a-d-e-n-w-e-B-a-jj-o-r-k-x-i-m-e-
adding --> a-u u-r r-o d-e n-w w-e e-B B-a a-jj jj-o r-k k-x x-i e-Sil

005 los caballos del comendador de la magdalena
l-o-s-k-a-B-a-L-o-s-d-e-l-k-o-m-e-n-d-a-D-o-r-d-e-l-a-m-a-G-D-a-l-e-n-a-
adding --> k-a a-B a-L L-o l-k k-o o-m m-e n-d a-D D-o r-d m-a a-G G-D D-a l-e

006 ¡ qué reloj más raro!
k-e-rr-e-l-o-x-m-a-s-rr-a-r-o-
adding --> Sil-k rr-e l-o o-x x-m a-s s-rr rr-a

007 en zigzag como las oscilaciones de la temperatura
e-n-T-i-G-T-a-G-k-o-m-o-l-a-s-o-s-T-i-l-a-T-j-o-n-e-s-d-e-l-a-t-e-m-p-e-r-a-t-u-r-a-
adding --> Sil-e T-i i-G G-T T-a G-k o-l s-o s-T i-l a-T T-j j-o o-n e-s a-t t-e m-p p-e e-r t-u

008 porque siempre tenía presente la anécdota de ...
p-o-r-k-e-s-j-e-m-p-r-e-t-e-n-i-a-p-r-e-s-e-n-t-e-l-a-a-n-e-k-D-o-t-a-d-e-
adding --> k-e s-j p-r r-e e-t n-i i-a a-p a-a a-n e-k k-D o-t

009 había sido el berdigum romano
a-B-i-a-s-i-D-o-e-l-b-e-r-D-i-G-u-m-rr-o-m-a-n-o-
adding --> Sil-a s-i i-D o-e l-b r-D D-i G-u u-m m-rr rr-o

010 hizo tic tac magníficamente
i-T-o-t-i-k-t-a-k-m-a-G-n-i-f-i-k-a-m-e-n-t-e-
adding --> Sil-i t-i i-k k-t a-k k-m G-n i-f f-i

011 con el zollvereín
k-o-n-e-l-T-o-L-B-e-r-e-i-n-
adding --> l-T T-o o-L L-B B-e n-Sil

012 tras otra a sus dormitorios submarinos
t-r-a-s-o-t-r-a-a-s-u-s-d-o-r-m-i-t-o-r-j-o-s-u-B-m-a-r-i-n-o-s-
adding --> Sil-t t-r s-u u-s d-o r-m i-t t-o r-j u-B B-m r-i

013 y la señora darling junto al fuego
i-l-a-s-e-J-o-r-a-d-a-r-l-i-n-G-x-u-n-t-o-a-l-f-w-e-G-o-
adding --> G-x x-u u-n l-f f-w

014 a hacer tic tac por su cuenta
a-a-T-e-r-t-i-k-t-a-k-p-o-r-s-u-k-w-e-n-t-a-
adding --> T-e r-t k-p p-o r-s u-k k-w

015 oh cuello mío recién degollado
o-k-w-e-L-o-m-i-o-rr-e-T-j-e-n-d-e-G-o-L-a-D-o-
adding --> Sil-o e-L i-o L-a

016 el correr no es sano
e-l-k-o-rr-e-rr-n-o-e-s-a-n-o
adding --> o-rr rr-n

017 que al reloj se le había acabado la cuerda
d-e-k-e-a-l-rr-e-l-o-x-s-e-l-e-a-B-i-a-a-k-a-B-a-D-o-l-a-k-w-e-r-D-a-
adding --> e-a l-rr x-s B-i

018 de pilluelo de parís
d-e-p-i-L-w-e-l-o-d-e-p-a-r-i-s-
adding --> p-i i-L L-w o-d e-p p-a

019 en este tiempo dio el reloj la una después
e-n-e-s-t-e-t-j-e-m-p-o-d-j-o-e-l-rr-e-l-o-x-l-a-u-n-a-d-e-s-p-w-e-s-
adding --> s-t t-j x-l n-a s-p p-w

020 pero el señor darling tenía un carácter demasiado bueno
p-e-r-o-e-l-s-e-J-o-r-d-a-r-l-i-n-G-t-e-n-i-a-u-n-k-a-r-a-k-t-e-r-d-e-m-a-s-j-a-D-o-b-w-e-n-o-
adding --> l-s G-t n-k j-a o-b b-w

021 como si de su huésped recibiese alguna ofensa
k-o-m-o-s-i-d-e-s-u-w-e-s-p-e-D-rr-e-T-i-B-j-e-s-e-a-l-G-u-n-a-o-f-e-n-s-a-
adding --> i-d u-w e-D D-rr i-B B-j l-G a-o o-f f-e

022 por cuyo pie corría un arroyuelo manso
p-o-r-k-u-jj-o-p-j-e-k-o-rr-i-a-u-n-a-rr-o-jj-w-e-l-o-m-a-n-s-o-
adding --> k-u u-jj o-p p-j rr-i a-rr o-jj jj-w

023 con una técnica que había descubierto
k-o-n-u-n-a-t-e-k-n-i-k-a-k-e-a-B-i-a-d-e-s-k-u-B-j-e-r-t-o-
adding --> n-u k-n s-k

024 ni con reloj que sacarse
n-i-k-o-n-rr-e-l-o-x-k-e-s-a-k-a-r-s-e-
adding --> Sil-n n-rr x-k s-a

025 ¡ahí están los árboles otra vez llenos!
a-i-e-s-t-a-n-l-o-s-a-r-B-o-l-e-s-o-t-r-a-b-e-T-L-e-n-o-s-
adding --> n-l r-B B-o T-L

026 que las amistades de la señora darling visitaran el cuarto
k-e-l-a-s-a-m-i-s-t-a-D-e-s-d-e-l-a-s-e-J-o-r-a-d-a-r-l-i-n-G-b-i-s-i-t-a-r-a-n-e-l-k-w-a-r-t-o-
adding --> i-s D-e G-b b-i w-a

027 la señora darling dijo inmediatamente
l-a-s-e-J-o-r-a-d-a-r-l-i-n-G-d-i-x-o-i-m-m-e-D-j-a-t-a-m-e-n-t-e-
adding --> G-d d-i i-x x-o o-i i-m m-m D-j

028 sin piedad llevabais al matadero
s-i-n-p-j-e-D-a-D-L-e-B-a-B-a-i-s-a-l-m-a-t-a-D-e-r-o-
adding --> Sil-s n-p D-L a-i l-m

029 las cinco llagas de cristo la luz juega el ajedrez
l-a-s-T-i-n-k-o-L-a-G-a-s-d-e-k-r-i-s-t-o-l-a-l-u-T-x-w-e-G-a-e-l-a-x-e-D-r-e-T-
adding --> G-a k-r l-u u-T T-x x-w a-e a-x x-e D-r T-Sil

030 con edgardo azelingo al encuentro de guillermo
k-o-n-e-D-G-a-r-D-o-a-T-e-l-i-n-g-o-a-l-e-n-k-w-e-n-t-r-o-d-e-G-i-L-e-r-m-o-
adding --> D-G g-o G-i

031 que tan fácilmente se dejaba subyugar
k-e-t-a-n-f-a-T-i-l-m-e-n-t-e-s-e-d-e-x-a-B-a-s-u-B-jj-u-G-a-r-
adding --> n-f f-a e-d e-x x-a B-jj jj-u u-G r-Sil

032 para que el rey de Jarlem cante
p-a-r-a-k-e-e-l-rr-e-i-d-e-x-a-r-l-e-m-k-a-n-t-e-
adding --> e-e e-i m-k

033 jamás volveréis a oír el tam-tam
x-a-m-a-s-b-o-l-B-e-r-e-i-s-a-o-i-r-e-l-t-a-m-t-a-m-
adding --> Sil-x s-b b-o l-B i-r l-t m-t m-Sil

034 a sucumbir bajo el peso de su misma abnegación
a-s-u-k-u-m-b-i-r-b-a-x-o-e-l-p-e-s-o-d-e-s-u-m-i-s-m-a-a-B-n-e-G-a-T-j-o-n-
adding --> m-b r-b b-a l-p s-m B-n

035 de modo que cuando entren con aire imponente la señora darling pueda no darle ni siquiera un beso
d-e-m-o-D-o-k-e-k-w-a-n-d-o-e-n-t-r-e-n-k-o-n-a-i-r-e-i-m-p-o-n-e-n-t-e-l-a-s-e-J-o-r-a-d-a-r-l-i-n-G-p-w-e-D-a-n-o-d-a-r-l-e-n-i-s-i-k-j-e-r-a-u-n-b-e-s-o-
adding --> o-D o-k G-p n-o k-j n-b

036 los piratas escucharon ceñudos
l-o-s-p-i-r-a-t-a-s-e-s-k-u-tS-a-r-o-n-T-e-J-u-D-o-s-
adding --> u-tS tS-a n-T J-u u-D

037 a tercero día hacíamos san juan
a-t-e-r-T-e-r-o-d-i-a-a-T-i-a-m-o-s-a-n-x-w-a-n-
adding --> r-T

038 sed generoso
s-e-D-x-e-n-e-r-o-s-o
adding --> D-x

039 que se bañan en las lunas perdieron sus castañuelas
k-e-s-e-b-a-J-a-n-e-n-l-a-s-l-u-n-a-s-p-e-r-D-j-e-r-o-n-s-u-s-k-a-s-t-a-J-w-e-l-a-s-
adding --> e-b a-J J-a s-l n-s J-w
040 llevan tomajaucs
L-e-B-a-n-t-o-m-a-x-a-u-k-s-
adding --> Sil-L

041 que desconfiasen de la potestad civil
k-e-d-e-s-k-o-m-f-j-a-s-e-n-d-e-l-a-p-o-t-e-s-t-a-D-T-i-B-i-l-
adding --> m-f f-j D-T l-Sil

042 seguiremos siendo respetuosos súbditos del rey
s-e-G-i-r-e-m-o-s-j-e-n-d-o-rr-e-s-p-e-t-w-o-s-o-s-u-B-D-i-t-o-s-d-e-l-rr-e-i-
adding --> t-w w-o B-D i-Sil

043 en el reloj de cuco
e-n-e-l-rr-e-l-o-x-d-e-k-u-k-o-
adding --> x-d

044 de eclipsarse una obra tan perfecta
d-e-e-k-l-i-p-s-a-r-s-e-u-n-a-o-B-r-a-t-a-n-p-e-r-f-e-k-t-a-
adding --> k-l i-p p-s e-u o-B B-r r-f

045 y anciana con la nariz ganchuda
jj-a-n-T-j-a-n-a-k-o-n-l-a-n-a-r-i-T-g-a-n-tS-u-D-a-
adding --> Sil-jj i-T T-g n-tS tS-u

046 como de obligación absoluta y puramente individual
k-o-m-o-d-e-o-B-l-i-G-a-T-j-o-n-a-B-s-o-l-u-t-a-i-p-u-r-a-m-e-n-t-e-i-n-d-i-B-i-D-w-a-l-
adding --> e-o B-l B-s u-t p-u D-w

047 los tristes pronósticos de doña beatriz fueron cumpliéndose muy aprisa
l-o-s-t-r-i-s-t-e-s-p-r-o-n-o-s-t-i-k-o-s-d-e-d-o-J-a-b-e-a-t-r-i-T-f-w-e-r-o-n-k-u-m-p-l-j-e-n-d-o-s-e-m-u-jj-a-p-r-i-s-a-
adding --> o-J T-f p-l l-j m-u jj-a

048 si bien templada por una benignidad grandísima
s-i-b-j-e-n-t-e-m-p-l-a-D-a-p-o-r-u-n-a-b-e-n-i-G-n-i-D-a-D-g-r-a-n-d-i-s-i-m-a-
adding --> b-j r-u D-g g-r

049 que era mitad poesía mitad fantasía reflexiva
k-e-e-r-a-m-i-t-a-D-p-o-e-s-i-a-m-i-t-a-D-f-a-n-t-a-s-i-a-rr-e-f-l-e-G-s-i-B-a-
adding --> D-p D-f e-f f-l G-s

050 bajo el Tomajauc del terrible Pantera
b-a-x-o-e-l-t-o-m-a-x-a-u-k-d-e-l-t-e-rr-i-B-l-e-p-a-n-t-e-r-a-
adding --> Sil-b k-d e-rr

051 y deambulan intactas las lluvias bailarinas
i-d-e-a-m-b-u-l-a-n-i-n-t-a-k-t-a-s-l-a-s-L-u-B-j-a-s-b-a-i-l-a-r-i-n-a-s-
adding --> b-u u-l s-L L-u

052 y de antorchas nupciales los blandones
i-d-e-a-n-t-o-r-tS-a-s-n-u-p-T-j-a-l-e-s-l-o-s-b-l-a-n-d-o-n-e-s-
adding --> s-n u-p p-T b-l

053 sobre su estilo problemático este apólogo esdrújulo-enigmático
s-o-B-r-e-s-u-e-s-t-i-l-o-p-r-o-B-l-e-m-a-t-i-k-o-e-s-t-e-a-p-o-l-o-G-o-e-s-D-r-u-x-u-l-o-e-n-i-G-m-a-t-i-k-o-
adding --> u-e o-G s-D u-x G-m

054 y don alonso entonces intimó a su hija su última e irrevocable resolución
i-d-o-n-a-l-o-n-s-o-e-n-t-o-n-T-e-s-i-n-t-i-m-o-a-s-u-i-x-a-s-u-u-l-t-i-m-a-e-i-rr-e-B-o-k-a-B-l-e-rr-e-s-o-l-u-T-j-o-n-
adding --> u-i u-u i-rr

055 a cuyos pies discurre un riachuelo
a-k-u-jj-o-s-p-j-e-s-d-i-s-k-u-rr-e-u-n-rr-j-a-tS-w-e-l-o-
adding --> u-rr rr-j a-tS tS-w

056 y huido el alguacil de un racimo
i-w-i-D-o-e-l-a-l-g-w-a-T-i-l-d-e-u-n-rr-a-T-i-m-o-
adding --> l-g g-w l-d

057 la interrumpió doña beatriz- que no huiré
l-a-i-n-t-e-rr-u-m-p-j-o-d-o-J-a-b-e-a-t-r-i-T-k-e-n-o-w-i-r-e-
adding --> rr-u T-k o-w w-i

058 y iba hecho zufaina
jj-i-B-a-e-tS-o-T-u-f-a-i-n-a-
adding --> e-tS tS-o o-T T-u u-f

059 dijo el bribonzuelo de juan- no estar aquí
d-i-x-o-e-l-b-r-i-B-o-n-T-w-e-l-o-d-e-x-w-a-n-o-e-s-t-a-r-a-k-i-
adding --> b-r T-w

060 pollo con nueces
p-o-L-o-k-o-n-n-u-e-T-e-s
adding --> n-n

061 cuando obtuvo un premio por cultura general
k-w-a-n-d-o-o-B-t-u-B-o-u-n-p-r-e-m-j-o-p-o-r-k-u-l-t-u-r-a-x-e-n-e-r-a-l-
adding --> o-o B-t o-u m-j

062 y se dieron cuenta de que seis le parecía una cantidad bastante grande
i-s-e-d-j-e-r-o-n-k-w-e-n-t-a-d-e-k-e-s-e-i-s-l-e-p-a-r-e-T-i-a-u-n-a-k-a-n-t-i-D-a-D-b-a-s-t-a-n-t-e-g-r-a-n-d-e-
adding --> D-b e-g

063 a la luz difusa varios objetos
a-l-a-l-u-T-d-i-f-u-s-a-b-a-r-j-o-s-o-B-x-e-t-o-s-
adding --> T-d f-u B-x

064 sus armaduras de y estrellas de nariz rota
s-u-s-a-r-m-a-D-u-r-a-s-d-e-jj-e-s-t-r-e-L-a-s-d-e-n-a-r-i-T-rr-o-t-a-
adding --> D-u e-jj jj-e T-rr

065 tu
t-u
adding --> u-Sil

066 de sus facciones y su voz trémula
d-e-s-u-s-f-a-k-T-j-o-n-e-s-i-s-u-b-o-T-t-r-e-m-u-l-a-
adding --> s-f k-T u-b T-t

067 del chiquillo
d-e-l-tS-i-k-i-L-o-
adding --> l-tS tS-i k-i

068 de que esto sentimos júzguelo cada uno
d-e-k-e-e-s-t-o-s-e-n-t-i-m-o-s-x-u-T-G-e-l-o-k-a-D-a-u-n-o-
adding --> s-x T-G G-e

069 nocturno
n-o-k-t-u-r-n-o
adding --> r-n

070 anotad todo
a-n-o-t-a-D-t-o-D-o
adding --> D-t

071 cuantas lágrimas he podido enjugar esas he enjugado
k-w-a-n-t-a-s-l-a-G-r-i-m-a-s-e-p-o-D-i-D-o-e-n-x-u-G-a-r-e-s-a-s-e-e-n-x-u-g-a-D-o-
adding --> G-r n-x u-g

072 hubo gran alegría cuando Peter llegó
u-B-o-g-r-a-n-a-l-e-G-r-i-a-k-w-a-n-d-o-p-e-t-e-r-L-e-G-o-
adding --> Sil-u o-g r-L

073 tu niñez ya fábula de fuentes
t-u-n-i-J-e-T-jj-a-f-a-B-u-l-a-d-e-f-w-e-n-t-e-s-
adding --> i-J J-e T-jj a-f B-u

074 que inmediatamente trajo al noble huésped
k-e-i-m-m-e-D-j-a-t-a-m-e-n-t-e-t-r-a-x-o-a-l-n-o-B-l-e-w-e-s-p-e-D-
adding --> l-n e-w D-Sil

075 con lo cual llevaba mejor el verme engullir
k-o-n-l-o-k-w-a-l-L-e-B-a-B-a-m-e-x-o-r-e-l-b-e-r-m-e-e-n-g-u-L-i-r-
adding --> l-L g-u u-L

076 ranas y grillos hacen la glorieta
rr-a-n-a-s-i-g-r-i-L-o-s-a-T-e-n-l-a-g-l-o-r-j-e-t-a-
adding --> Sil-rr i-g a-g g-l

077 de lemus en monforte iban componiendo ya una hueste poderosa
d-e-l-e-m-u-s-e-n-m-o-m-f-o-r-t-e-i-B-a-n-k-o-m-p-o-n-j-e-n-d-o-jj-a-u-n-a-w-e-s-t-e-p-o-D-e-r-o-s-a
adding --> n-m f-o n-j a-w

078 y la esterilidad y la viudez vendrán juntas sobre ti
i-l-a-e-s-t-e-r-i-l-i-D-a-D-i-l-a-b-j-u-D-e-T-b-e-n-d-r-a-n-x-u-n-t-a-s-o-b-r-e-t-i-
adding --> j-u T-b d-r

079 la habilidad de su adversario
a-l-a-a-B-i-l-i-D-a-D-d-e-s-u-a-D-B-e-r-s-a-r-j-o-
adding --> D-d u-a D-B

080 cuando sabes que de mi voz la dulce melodía nunca ha tenido igual
k-w-a-n-d-o-s-a-B-e-s-k-e-d-e-m-i-b-o-T-l-a-d-u-l-T-e-m-e-l-o-D-i-a-n-u-n-k-a-a-t-e-n-i-D-o-i-G-w-a-l-
adding --> i-b T-l d-u G-w

081 mientras el dodo le ofrecía solemnemente…
m-j-e-n-t-r-a-s-e-l-d-o-D-o-l-e-o-f-r-e-T-i-a-s-o-l-e-m-n-e-m-e-n-t-e
adding --> Sil-m f-r m-n

082 del óxido de hierro de los grandes puentes
D-e-l-o-G-s-i-D-o-d-e-jj-e-rr-o-d-e-l-o-s-g-r-a-n-d-e-s-p-w-e-n-t-e-s-
adding --> Sil-D s-g

083 de voluptuosidad sensual alambicada
d-e-b-o-l-u-p-t-w-o-s-i-D-a-D-s-e-n-s-w-a-l-a-l-a-m-b-i-k-a-D-a-
adding --> p-t D-s s-w

084 con esto desvanecido y hecho dueño
k-o-n-e-s-t-o-d-e-s-B-a-n-e-T-i-D-o-i-e-tS-o-d-w-e-J-o-
adding --> s-B i-e d-w

085 con una hija llamada margaret y…
k-o-n-u-n-a-i-x-a-L-a-m-a-D-a-m-a-r-G-a-r-e-D-i-
adding --> r-G

086 todas estas desdichas exacerbaron su orgullo ofendido
t-o-D-a-s-e-s-t-a-s-d-e-s-D-i-tS-a-s-e-G-s-a-T-e-r-B-a-r-o-n-s-u-o-r-g-u-L-o-o-f-e-n-d-i-D-o-
adding --> i-tS u-o r-g

087 en realidad no tenía ni idea
e-n-rr-e-a-l-i-D-a-D-n-o-t-e-n-i-a-n-i-i-D-e-a-
adding --> D-n i-i

088 a su vez por rizos
a-s-u-b-e-T-p-o-r-rr-i-T-o-s-
adding --> T-p r-rr

089 no estaba esperando ni un grito ni un graznido
n-o-e-s-t-a-B-a-e-s-p-e-r-a-n-d-o-n-i-u-n-g-r-i-t-o-n-i-u-n-g-r-a-T-n-i-D-o-
adding --> i-u T-n

090 como el ruibarbo y soltaba gruñidos
k-o-m-o-e-l-rr-w-i-B-a-r-B-o-i-s-o-l-t-a-B-a-g-r-u-J-i-D-o-s-
adding --> rr-w u-J J-i

091 respondió el abad con vehemencia en contribuir
r-e-s-p-o-n-d-j-o-e-l-a-B-a-D-k-o-n-b-e-e-m-e-n-T-j-a-e-n-k-o-n-t-r-i-B-w-i-r-
adding --> Sil-r D-k B-w

092 la noche una vez más
l-a-n-o-tS-e-u-n-a-b-e-T-m-a-s-a-k-o-s-t-a-D-o-s-
adding --> o-tS T-m

093 que proclamaba que todos los chicos yacían muertos
k-e-p-r-o-k-l-a-m-a-B-a-k-e-t-o-D-o-s-l-o-s-tS-i-k-o-s-jj-a-T-i-a-n-m-w-e-r-t-o-s-
adding --> s-tS s-jj m-w

094 cogiendo un farol y alzando el garfio
k-o-x-j-e-n-d-o-u-n-f-a-r-o-l-jj-a-l-T-a-n-d-o-e-l-g-a-r-f-j-o-
adding --> x-j l-jj

095 cuando por fin llegó de verdad la maté
k-w-a-n-d-o-p-o-r-f-i-n-L-e-G-o-d-e-b-e-r-D-a-D-l-a-m-a-t-e-
adding --> n-L D-l

096 por última vez sus perros admiraron
p-o-r-u-l-t-i-m-a-b-e-T-s-u-s-p-e-rr-o-s-a-D-m-i-r-a-r-o-n-
adding --> T-s D-m

097 lo cual es probablemente la mejor prueba
l-o-k-w-a-l-e-s-p-r-o-B-a-B-l-e-m-e-n-t-e-l-a-m-e-x-o-r-p-r-w-e-B-a-
adding --> r-p r-w

098 que soplara en su dirección y esto supuso un cambio tan agradable
k-e-s-o-p-l-a-r-a-e-n-s-u-d-i-r-e-k-T-j-o-n-jj-e-s-t-o-s-u-p-u-s-o-u-n-k-a-m-b-j-o-t-a-n-a-G-r-a-D-a-B-l-e-
adding --> u-d n-jj

099 en inglés se llama tinker bell
e-n-i-n-G-l-e-s-e-L-a-m-a-t-i-n-k-e-r-b-e-L-
adding --> G-l

100 Jorge Darling
x-o-r-x-e-d-a-r-l-i-n-G-
adding --> r-x G-Sil

101 capitán del Jalisco Yin
k-a-p-i-t-a-n-d-e-l-x-a-l-i-s-k-o-L-i-n-
adding --> l-x L-i

102 luego siguió haciendo tic tac
l-w-e-G-o-s-i-G-j-o-a-T-j-e-n-d-o-t-i-k-t-a-k-
adding --> G-j k-Sil

103 fuímonos a acostar y en toda la noche…
f-w-i-m-o-n-o-s-a-a-k-o-s-t-a-r-jj-e-n-t-o-D-a-l-a-n-o-tS-e
adding --> Sil-f r-jj

104 gritó y luego disparó y clara cayó revoloteando
g-r-i-t-o-i-l-w-e-G-o-d-i-s-p-a-r-o-i-k-l-a-r-a-k-a-jj-o-rr-e-B-o-l-o-t-e-a-n-d-o-
adding --> Sil-g l-w

105 aquí y allá surgía una cabeza
a-k-i-jj-a-L-a-s-u-r-x-i-a-u-n-a-k-a-B-e-T-a-
adding --> i-jj

106 por desgracia nunca se despertaba
p-o-r-d-e-s-G-r-a-T-j-a-n-u-n-k-a-s-e-d-e-s-p-e-r-t-a-B-a-
adding --> s-G

107 viva la alegría y una buena soga
B-i-B-a-l-a-a-l-e-G-r-i-a-jj-u-n-a-b-w-e-n-a-s-o-G-a-
adding --> Sil-B

108 gracias
G-r-a-T-j-a-s-
adding --> Sil-G

109 ñoña
J-o-J-a-
adding --> Sil-J

110 western
w-e-s-t-e-r-n
adding --> Sil-w

111 ciertamente no fingían tener sueño
T-j-e-r-t-a-m-e-n-t-e-n-o-f-i-n-x-i-a-n-t-e-n-e-r-s-w-e-J-o-
adding --> Sil-T

112 chap
tS-a-p-
adding --> Sil-tS p-Sil

113 Dupond
d-u-p-o-n-d
adding --> d-Sil

114 uf
u-f-
adding --> f-Sil

115 reloj
rr-e-l-o-x-
adding --> x-Sil

116 futbol match
f-u-t-b-o-l-m-a-tS
adding --> tS-Sil

ANNEX C

Publications by the author related to the dissertation research

[P1] Bonada, J. "Desenvolupament d`un entorn gráfic per a l`análisi, transformació i síntesi de sons mitjanant models espectrals." *Master Thesis, Catalunya Polytechnic University (UPC).* Barcelona, 1997.

[P2] Serra, X., J. Bonada, P. Herrera, and R. Loureiro. "Integrating Complementary Spectral Models in the Design of a Musical Synthesizer." *Proceedings of International Computer Music Conference.* Thessaloniki,Greece, 1997.

[P3] Amatriain, X., J. Bonada, and X. Serra. "METRIX: A Musical Data Definition Language and Data Structure for a Spectral Modeling Based Synthesizer." *Proceedings of COST G6 Conference on Digital Audio Effects.* Barcelona, Spain, 1998.

[P4] Herrera, P., and J. Bonada. "Vibrato Extraction and Parameterization in the Spectral Modeling Synthesis framework." *Proceedings of COST G6 Conference on Digital Audio Effects.* Barcelona, Spain, 1998.

[P5] Serra, X., and J. Bonada. "Sound Transformations Based on the SMS High Level Attributes." *Proceedings of COST G6 Conference on Digital Audio Effects.* Barcelona, Spain, 1998.

[P6] Cano, P., A. Loscos, and J. Bonada. "Score-Performance Matching using HMMs." *Proceedings of International Computer Music Conference.* Beijing, China, 1999.

[P7] Loscos, A., P. Cano, and J. Bonada. "Low-Delay Singing Voice Alignment to Text." *Proceedings of International Computer Music Conference.* Beijing, China, 1999.

[P8] Bonada, J. "Automatic Technique in Frequency Domain for Near-Lossless Time-Scale Modification of Audio." *Proceedings of the International Computer Music Conference.* Berlin, Germany, 2000.

[P9] de Boer, M., J. Bonada, P. Cano, A. Loscos, and X. Serra. "Singing Voice Impersonator Application for PC." *Proceedings of International Computer Music Conference.* Berlin, Germany, 2000.

[P10] de Boer, M., J. Bonada, and X. Serra. "Using the Sound Description Interchange Format within the SMS Applications." *Proceedings of International Computer Music Conference.* Berlin, Germany, 2000.

[P11] Cano, P., A. Loscos, J. Bonada, M. de Boer, and X. Serra. "Voice Morphing System for Impersonating in Karaoke Applications." *Proceedings of International Computer Music Conference.* Berlin, Germany, 2000.

[P12] Amatriain, X., J. Bonada, A. Loscos, and X. Serra. "Spectral Modeling for Higher-level Sound Transformation." *Proceedings of MOSART Workshop on Current Research Directions in Computer Music.* Barcelona, Spain, 2001.

[P13] Bonada, J., O. Celma, A. Loscos, J. Ortolà, and X. Serra. "Singing Voice Synthesis Combining Excitation plus Resonance and Sinusoidal plus Residual Models." *Proceedings of International Computer Music Conference.* Havana, Cuba, 2001.

[P14] Bonada, J., A. Loscos, P. Cano, X Serra, and H. Kenmochi. "Spectral Approach to the Modeling of the Singing Voice." *Proceedings of the 111th AES Convention.* New York, USA, September, 2001.

[P15] Amatriain, X., J. Bonada, A. Loscos, and X. Serra. "Spectral Processing." In *DAFX: Digital Audio Effects*, Editor Udo Zölzer, pp. 373-438. John Wiley & Sons Publishers, 2002.

[P16] Amatriain, X., J. Bonada, A. Loscos, J. Arcos, and V. Verfaille. "Content-based Transformations." *Journal of New Music Research*, 2003: vol 32, nº 1.

[P17] Bonada, J., A. Loscos, and H. Kenmochi. "Sample-based Singing Voice Synthesizer by Spectral Concatenation." *Proceedings of the Stockholm Music Acoustics Conference.* Stockholm, Sweden, 2003.

[P18] Bonada, J., A. Loscos, O. Mayor, and H. Kenmochi. "Sample-based Singing Voice Synthesizer using Spectral Models and Source-Filter Decomposition." *Proceedings of 3rd Intl. Workshop on Models and Analysis of Vocal Emissions for Biomedical Applications (MAVEBA).* Firenze, Italy, 2003.

[P19] Gouyon, F., L. Fabig, and J. Bonada. "Rhythmic expressiveness transformations of audio recordings: swing modifications." *Proceedings of 6th International Conference on Digital Audio Effects.* London, UK, 2003.

[P20] Bonada, J. "High Quality Voice Transformations based on Modeling Radiated Voice Pulses in Frequency Domain." *Proc. of the 7th Int. Conference on Digital Audio Effects.* Naples, Italy, October, 2004.

[P21] Loscos, A., and J. Bonada. "Emulating Rough And Growl Voice in Spectral Domain." *Proceedings of 7th Int. Conference on Digital Audio Effects.* Naples, Italy, October, 2004.

[P22] Bonada, J. "Voice Solo to Unison Choir Transformation." *Proceedings of 118th Audio Engineering Society Convention.* Barcelona, Spain, 2005.

[P23] Gómez, E., and J. Bonada. "Tonality visualization of polyphonic audio." *Proceedings of International Computer Music Conference.* Barcelona, Spain, 2005.

[P24] Bonada, J. "Esophageal Voice Enhancement by Modeling Radiated Pulses in Frequency Domain." *Proceedings of 121st Convention of the Audio Engineering Society.* San Francisco, CA, USA, 2006.

[P25] Bonada, J., M. Blaauw, A. Loscos, and H. Kenmochi. "Unisong: A Choir Singing Synthesizer." *Proceedings of 121st Convention of the Audio Engineering Society.* San Francisco, CA, USA, 2006.

[P26] Bonada, J., A. Loscos, and M. Blaauw. "Improvements to a Sample-Concatenation Based Singing Voice Synthesizer." *Proceeding of the 121st AES Convention.* San Francisco, USA, October, 2006.

[P27] Janer, J., J. Bonada, and M. Blaauw. "Performance-driven control for sample-based singing voice synthesis." *Proceedings of 9th International Conference on Digital Audio Effects.* Montreal, Canada, 2006.

[P28] Janer, J., J. Bonada, and S. Jordà. "Groovator - an implementation of real-time rhythm transformations." *Proceedings of 121st Convention of the Audio Engineering Society.* San Francisco, CA, USA, 2006.

[P29] Maestre, E., J. Bonada, and O. Mayor. "Modeling musical articulation gestures in singing voice performances." *Proceedings of 121st Convention of the Audio Engineering Society.* San Francisco, CA, USA, 2006.

[P30] Mayor, O., J. Bonada, and A. Loscos. "The Singing Tutor: Expression Categorization and Segmentation of the Singing Voice." *Proceedings of 121st Convention of the Audio Engineering Society.* San Francisco, CA, USA, 2006.

[P31] Vinyes, M., J. Bonada, and A. Loscos. "Demixing Commercial Music Productions via Human-Assisted Time-Frequency Masking." *Proceedings of 120st Convention of the Audio Engineering Society*. Paris, France, 2006.

[P32] Bonada, J., and X. Serra. "Synthesis of the Singing Voice by Performance Sampling and Spectral Models." *IEEE Signal Processing Magazine*, January 2007: vol. 24, no. 1.

[P33] Guaus, E., Bonada, J., Perez, A., Maestre, E., Blaauw, M. "Measuring the bow pressing force in a real violin performance". *Proceedings of International Symposium on Musical Acoustics,* Barcelona, Spain, *2007.*

[P34] Perez, A., Bonada, J., Maestre, E., Guaus, E., Blaauw, M. "Combining Performance Actions with Spectral Models for Violin Sound Transformation". *Proceedings of 19th International Congress on Acoustics,* Madrid, Spain, 2007.

[P35] Maestre, E., Bonada, J., Blaauw, M., Perez, A., Guaus, E. "Acquisition of violin instrumental gestures using a commercial EMF device". *Proceedings of International Computer Music Conference*, Copenhagen, Denmark, 2007.

[P36] Bonada, J. "Wide-Band Harmonic Sinusoidal Modeling". *Proceedings of International Conference on Digital Audio Effects*¸Helsinki, Finland, 2008.

[P37] Perez, A., Bonada, J., Maestre, E., Guaus, E., Blaauw, M. "Score Level Timbre Transformations of Violin Sounds". *Proceedings on International Conference on Digital Audio Effects*, Helsinki, Finland, 2008.

[P38] Perez, A., Bonada, J., Maestre, E., Guaus, E., Blaauw, M. "Measuring Violin Sound Radiation for Sound Equalization". *Proceedings of Acoustics08*, Paris, France, 2008.

[P39] Gouyon, F., Herrera, P., Gómez, E., Cano, P., Bonada, J., Loscos, A., Amatriain, X., Serra, X. "Content Processing of Music Audio Signals". In *Sound to Sense, Sense to Sound: A State of the Art in Sound and Music Computing*, Polotti, P. and Rocchesso, D. Editors, Logos Verlag Berlin GmbH (ISBN 978-3-8325-1600-0), pp. 83-160, 2008.

[P40] Coleman, G., Bonada, J. "Sound Transformation by Descriptor Using an Analytic Domain". *Proceedings of International Conference on Digital Audio Effects*, Helsinki, Finland, 2008.

[P41] Gómez, E., Bonada, J. "Automatic Melodic Transcription of Flamenco Singing". *Proceedings of Fourth Conference on Interdisciplinary Musicology (CIM08)*, Thessaloniki, Greece, 2008.

ANNEX D

Patents by the author related to the dissertation research

[Pat1] EP0982713 *Voice converter with extraction and modification of attribute data.* Inventors: Bonada, J., Kayama, H., Yoshioka, Y., Serra, X. *Applicant: Yamaha Corp. Publication date: 2000-03-01.*

[Pat2] JP2000003197 *Voice transforming device, voice transforming method and storage medium which records voice transforming program.* Inventors: Bonada, J. and Yoshioka, Y. *Applicant: Yamaha Corp. Publication date: 2000-01-07.*

[Pat3] JP2001117564 *Device and method for processing musical sound. Inventors: Bonada, J., Kawashima, T., and Serra, X. Applicant: Yamaha Corp. Publication date: 2001-04-27.*

[Pat4] JP2001117597 *Device and method for voice conversion and method of generating dictionary for voice conversion.* Inventors: Bonada, J., Yoshioka, Y., Serra, X. and Shiimentsu, M. *Applicant: YAMAHA Corp. and Univ. Pompeu Fabra. Publication date: 2000-03-01.*

[Pat5] JP2001116780 *Signal analyzer and signal analysis method.* Inventors: Bonada, J. and Yoshioka, Y. *Applicant: YAMAHA Corp. and Univ. Pompeu Fabra. Publication date: 2001-04-27.*

[Pat6] JP2001117578 *Device and method for adding harmony sound.* Inventors: Bonada, J., Cano, P., Kondo, T. and Loscos, A. *Applicant: YAMAHA Corp. and Univ. Pompeu Fabra. Publication date: 2001-04-27.*

[Pat7] JP2001117600 *Device and method for aural signal processing.* Inventors: Bonada, J., Kayama, H. and Serra, X. *Applicant: YAMAHA Corp. and Univ. Pompeu Fabra. Publication date: 2001-04-27.*

[Pat8] JP2001184099 *Device and method for voice conversion.* Inventors: Bonada, J., Shiimentsu, M. and Kawashima, T. *Applicant: YAMAHA Corp. and Univ. Pompeu Fabra. Publication date: 2001-07-06.*

[Pat9] EP0982713 *Voice converter with extraction and modification of attribute data.* Inventors: Bonada, J., Kayama, H., Serra, X. and Yoshioka, Y. *Applicant: YAMAHA Corp. Publication date: 2000-03-01.*

[Pat10] EP1220195 *Singing voice synthesizing apparatus, singing voice synthesizing method, and program for realizing singing voice synthesizing method.* Inventors: Bonada, J., Kenmochi, H. and Serra, X. *Applicant: YAMAHA Corp. Publication date: 2002-07-03.*

[Pat11] EP1688911 *Voice synthesizing apparatus.* Inventors: Bonada, J. and Hisaminato, Y. *Applicant: YAMAHA Corp. Publication date: 2006-08-09.*

[Pat12] JP2005275420 *Voice analysis and synthesizing apparatus, method, and program* Inventors: Bonada, J. and Yoshioka, Y. *Applicant: YAMAHA Corp. Publication date: 2005-10-06.*

[Pat13] JP2006145867 *Voice processor and voice processing program.* Inventors: Bonada, J. and Kenmochi, H. *Applicant: YAMAHA Corp. Publication date: 2006-06-08.*

[Pat14] WO2006046761 *Pitch converting apparattus.* Inventors: Bonada, J. and Fujishima, T. *Applicant: YAMAHA Corp. Publication date: 2006-05-54.*

[Pat15] JP2004077608 A*pparatus and method for chorus synthesis and program.* Inventors: Bonada, J. and Kenmochi, H. *Applicant: Yamaha Corp. Publication date: 2004-03-11.*

[Pat16] JP2003255998 *Singing synthesizing method, device, and recording medium.* Inventors: Bonada, J., Loscos, A., Kenmochi, H. *Applicant: Yamaha Corp. Publication date: 2003-09-10.*

[Pat17] JP2006215204 *Voice synthesizer and program.* Inventors: Bonada, J., Kenmochi, H. *Applicant: Yamaha Corp. Publication date: 2006-08-17.*

[Pat18] JP2006119655 *Voice synthesizer.* Inventors: Bonada, J., Hisaminato, Y. *Applicant: Yamaha Corp. Publication date: 2006-05-11.*

ANNEX E

Audio references[41]

VOICE SOURCE

[1] fry utterance of an adult male

HARMONICS AS SPECTRAL REGIONS

[2] singing male voice

[3] audio [2] transposed one octave below, identity harmonic mapping

[4] audio [2] transposed by 1.75 ratio, identity harmonic mapping

[5] audio [2] transposed one octave below, closest frequency harmonic mapping

[6] audio [2] transposed by 1.75 ratio, closest frequency harmonic mapping

[7] female singing the word "gardens"

[8] audio [7] modified with a timbre scaling of $T_{timbre} = 1.18$, closest frequency harmonic mapping

[9] audio [7] modified with a timbre scaling of $T_{timbre} = 1.18$, joint formant-pitch-modification harmonic mapping

[10] audio [7] modified with a pitch transposition of $T_{pitch} = 0.75$, joint formant-pitch-modification harmonic mapping

[11] audio [7] modified with a pitch transposition of $T_{pitch} = 0.75$, joint formant-pitch-modification harmonic mapping, neighbor selection using the amplitude difference function $d_a(h)$

[12] female singing $f_0^{range} \approx 350 - 386\,\text{Hz}$

[13] audio [12] transformed $T_{pitch} = 1.75$, basic technique

[14] audio [12] transformed $T_{pitch} = 1.75$, strategy 1

[15] audio [12] transformed $T_{pitch} = 1.75$, strategy 2

[16] audio [12] transformed $T_{pitch} = 1.75$, strategy 3

[17] audio [12] transformed $T_{pitch} = 1.75$, strategy 4

[18] female *scat* fast singing $f_0^{range} \approx 127 - 318\,\text{Hz}$

[19] audio [18] transformed $T_{pitch} = 1.75$, basic technique

[20] audio [18] transformed $T_{pitch} = 1.75$, strategy 1

[21] audio [18] transformed $T_{pitch} = 1.75$, strategy 2

[22] audio [18] transformed $T_{pitch} = 1.75$, strategy 3

[23] audio [18] transformed $T_{pitch} = 1.75$, strategy 4

[24] male *slow* singing $f_0^{range} \approx 100 - 291\,\text{Hz}$

[25] audio [24] transformed $T_{pitch} = 1.75$, basic technique

[26] audio [24] transformed $T_{pitch} = 1.75$, strategy 1

[27] audio [24] transformed $T_{pitch} = 1.75$, strategy 2

[41] all files can be downloaded from http://www.mtg.upf.edu/~jbonada/thesis

[28] audio [24] transformed $T_{pitch} = 1.75$, strategy 3
[29] audio [24] transformed $T_{pitch} = 1.75$, strategy 4
[30] male *soul* singing $f_0^{range} \approx 98 - 211\,\text{Hz}$
[31] audio [30] transformed $T_{pitch} = 1.75$, basic technique
[32] audio [30] transformed $T_{pitch} = 1.75$, strategy 1
[33] audio [30] transformed $T_{pitch} = 1.75$, strategy 2
[34] audio [30] transformed $T_{pitch} = 1.75$, strategy 3
[35] audio [30] transformed $T_{pitch} = 1.75$, strategy 4
[36] male speech $f_0^{range} \approx 59 - 100\,\text{Hz}$
[37] audio [36] transformed $T_{pitch} = 1.75$, basic technique
[38] audio [36] transformed $T_{pitch} = 1.75$, strategy 1
[39] audio [36] transformed $T_{pitch} = 1.75$, strategy 2
[40] audio [36] transformed $T_{pitch} = 1.75$, strategy 3
[41] audio [36] transformed $T_{pitch} = 1.75$, strategy 4
[42] female speech $f_0^{range} \approx 190 - 410\,\text{Hz}$
[43] audio [42] transformed $T_{pitch} = 1.75$, basic technique
[44] audio [42] transformed $T_{pitch} = 1.75$, strategy 1
[45] audio [42] transformed $T_{pitch} = 1.75$, strategy 2
[46] audio [42] transformed $T_{pitch} = 1.75$, strategy 3
[47] audio [42] transformed $T_{pitch} = 1.75$, strategy 4
[48] male *breathy* singing $f_0^{range} \approx 270 - 396\,\text{Hz}$
[49] audio [48] transformed $T_{pitch} = 1.75$, basic technique
[50] audio [48] transformed $T_{pitch} = 1.75$, strategy 1
[51] audio [48] transformed $T_{pitch} = 1.75$, strategy 2
[52] audio [48] transformed $T_{pitch} = 1.75$, strategy 3
[53] audio [48] transformed $T_{pitch} = 1.75$, strategy 4
[54] male *gospel* singing $f_0^{range} \approx 145 - 357\,\text{Hz}$
[55] audio [54] transformed $T_{pitch} = 1.75$, basic technique
[56] audio [54] transformed $T_{pitch} = 1.75$, strategy 1
[57] audio [54] transformed $T_{pitch} = 1.75$, strategy 2
[58] audio [54] transformed $T_{pitch} = 1.75$, strategy 3
[59] audio [54] transformed $T_{pitch} = 1.75$, strategy 4
[60] audio [12] transformed $T_{pitch} = 0.65$, basic technique
[61] audio [12] transformed $T_{pitch} = 0.65$, strategy 1
[62] audio [12] transformed $T_{pitch} = 0.65$, strategy 2
[63] audio [12] transformed $T_{pitch} = 0.65$, strategy 3
[64] audio [12] transformed $T_{pitch} = 0.65$, strategy 4
[65] audio [18] transformed $T_{pitch} = 0.65$, basic technique
[66] audio [18] transformed $T_{pitch} = 0.65$, strategy 1
[67] audio [18] transformed $T_{pitch} = 0.65$, strategy 2
[68] audio [18] transformed $T_{pitch} = 0.65$, strategy 3
[69] audio [18] transformed $T_{pitch} = 0.65$, strategy 4
[70] audio [24] transformed $T_{pitch} = 0.65$, basic technique
[71] audio [24] transformed $T_{pitch} = 0.65$, strategy 1
[72] audio [24] transformed $T_{pitch} = 0.65$, strategy 2

[73] audio [24] transformed $T_{pitch} = 0.65$, strategy 3
[74] audio [24] transformed $T_{pitch} = 0.65$, strategy 4
[75] audio [30] transformed $T_{pitch} = 0.65$, basic technique
[76] audio [30] transformed $T_{pitch} = 0.65$, strategy 1
[77] audio [30] transformed $T_{pitch} = 0.65$, strategy 2
[78] audio [30] transformed $T_{pitch} = 0.65$, strategy 3
[79] audio [30] transformed $T_{pitch} = 0.65$, strategy 4
[80] audio [36] transformed $T_{pitch} = 0.65$, basic technique
[81] audio [36] transformed $T_{pitch} = 0.65$, strategy 1
[82] audio [36] transformed $T_{pitch} = 0.65$, strategy 1
[83] audio [36] transformed $T_{pitch} = 0.65$, strategy 3
[84] audio [36] transformed $T_{pitch} = 0.65$, strategy 4
[85] audio [42] transformed $T_{pitch} = 0.65$, basic technique
[86] audio [42] transformed $T_{pitch} = 0.65$, strategy 1
[87] audio [42] transformed $T_{pitch} = 0.65$, strategy 2
[88] audio [42] transformed $T_{pitch} = 0.65$, strategy 3
[89] audio [42] transformed $T_{pitch} = 0.65$, strategy 4
[90] audio [48] transformed $T_{pitch} = 0.65$, basic technique
[91] audio [48] transformed $T_{pitch} = 0.65$, strategy 1
[92] audio [48] transformed $T_{pitch} = 0.65$, strategy 2
[93] audio [48] transformed $T_{pitch} = 0.65$, strategy 3
[94] audio [48] transformed $T_{pitch} = 0.65$, strategy 4
[95] audio [54] transformed $T_{pitch} = 0.65$, basic technique
[96] audio [54] transformed $T_{pitch} = 0.65$, strategy 1
[97] audio [54] transformed $T_{pitch} = 0.65$, strategy 2
[98] audio [54] transformed $T_{pitch} = 0.65$, strategy 3
[99] audio [54] transformed $T_{pitch} = 0.65$, strategy 4

SHAPE VARIANCE

[100] male speech
[101] audio [100] modified with a pitch transposition of $T_{pitch} = 0.5$, shape invariant using estimated voice pulse onset locations
[102] audio [100] modified with a pitch transposition of $T_{pitch} = 0.5$, shape invariant using an arbitrary position within the pulse period

WBVPM

[103] dance style male voice utterance
[104] synthetic signal with amplitude and frequency modulations
[105] residual obtained from standard sinusoidal modeling
[106] residual obtained from wide-band sinusoidal modeling
[107] female singing the word ‘yacht’ with vibrato.
[108] male speech with very low fundamental frequency around 70Hz.
[109] female expressive singing
[110] male singing ‘aiaiaiaia’ at different notes without vibrato

PHASE MODEL FROM HARMONIC ENVELOPE

[111] alienAndMonster
[112] resynthesis of audio [111] with phase model
[113] audio [111] transformed T_{pitch}=0.7 , T_{timbre}=1.05
[114] audio [111] transformed with phase model T_{pitch}=0.7 , T_{timbre}=1.05
[115] audio [111] transformed T_{pitch}=1.25 , T_{timbre}=0.95
[116] audio [111] transformed with phase model T_{pitch}=1.25 , T_{timbre}=0.95
[117] DeepenedVoice
[118] resynthesis of audio [117] with phase model
[119] audio [117] transformed T_{pitch}=0.7 , T_{timbre}=1.05
[120] audio [117] transformed with phase model T_{pitch}=0.7 , T_{timbre}=1.05
[121] audio [117] transformed T_{pitch}=1.25 , T_{timbre}=0.95
[122] audio [117] transformed with phase model T_{pitch}=1.25 , T_{timbre}=0.95
[123] Duyos
[124] resynthesis of audio [123] with phase model
[125] audio [123] transformed T_{pitch}=0.7 , T_{timbre}=1.05
[126] audio [123] transformed with phase model T_{pitch}=0.7 , T_{timbre}=1.05
[127] audio [123] transformed T_{pitch}=1.25 , T_{timbre}=0.95
[128] audio [123] transformed with phase model T_{pitch}=1.25 , T_{timbre}=0.95
[129] SpokenStation
[130] resynthesis of audio [129] with phase model
[131] audio [129] transformed T_{pitch}=0.7 , T_{timbre}=1.05
[132] audio [129] transformed with phase model T_{pitch}=0.7 , T_{timbre}=1.05
[133] audio [129] transformed T_{pitch}=1.25 , T_{timbre}=0.95
[134] audio [129] transformed with phase model T_{pitch}=1.25 , T_{timbre}=0.95
[135] Laroche
[136] resynthesis of audio [135] with phase model
[137] audio [135] transformed T_{pitch}=0.7 , T_{timbre}=1.05
[138] audio [135] transformed with phase model T_{pitch}=0.7 , T_{timbre}=1.05
[139] audio [135] transformed T_{pitch}=1.25 , T_{timbre}=0.95
[140] audio [135] transformed with phase model T_{pitch}=1.25 , T_{timbre}=0.95
[141] LayeredVocals
[142] resynthesis of audio [141] with phase model
[143] audio [141] transformed T_{pitch}=0.7 , T_{timbre}=1.05
[144] audio [141] transformed with phase model T_{pitch}=0.7 , T_{timbre}=1.05
[145] audio [141] transformed T_{pitch}=1.25 , T_{timbre}=0.95
[146] audio [141] transformed with phase model T_{pitch}=1.25 , T_{timbre}=0.95
[147] FemaleSpeech
[148] resynthesis of audio [147] with phase model
[149] audio [147] transformed T_{pitch}=0.7 , T_{timbre}=1.05
[150] audio [147] transformed with phase model T_{pitch}=0.7 , T_{timbre}=1.05
[151] audio [147] transformed T_{pitch}=1.25 , T_{timbre}=0.95
[152] audio [147] transformed with phase model T_{pitch}=1.25 , T_{timbre}=0.95

[153] d-dialog2301
[154] resynthesis of audio [153] with phase model
[155] audio [153] transformed T_{pitch}=0.7 , T_{timbre}=1.05
[156] audio [153] transformed with phase model T_{pitch}=0.7 , T_{timbre}=1.05
[157] audio [153] transformed T_{pitch}=1.25 , T_{timbre}=0.95
[158] audio [153] transformed with phase model T_{pitch}=1.25 , T_{timbre}=0.95
[159] Oldsctman099
[160] resynthesis of audio [159] with phase model
[161] audio [159] transformed T_{pitch}=0.7 , T_{timbre}=1.05
[162] audio [159] transformed with phase model T_{pitch}=0.7 , T_{timbre}=1.05
[163] audio [159] transformed T_{pitch}=1.25 , T_{timbre}=0.95
[164] audio [159] transformed with phase model T_{pitch}=1.25 , T_{timbre}=0.95
[165] Randy
[166] resynthesis of audio [165] with phase model
[167] audio [165] transformed T_{pitch}=0.7 , T_{timbre}=1.05
[168] audio [165] transformed with phase model T_{pitch}=0.7 , T_{timbre}=1.05
[169] audio [165] transformed T_{pitch}=1.25 , T_{timbre}=0.95
[170] audio [165] transformed with phase model T_{pitch}=1.25 , T_{timbre}=0.95
[171] Scat
[172] resynthesis of audio [171] with phase model
[173] audio [171] transformed T_{pitch}=0.7 , T_{timbre}=1.05
[174] audio [171] transformed with phase model T_{pitch}=0.7 , T_{timbre}=1.05
[175] audio [171] transformed T_{pitch}=1.25 , T_{timbre}=0.95
[176] audio [171] transformed with phase model T_{pitch}=1.25 , T_{timbre}=0.95
[177] Dance
[178] resynthesis of audio [177] with phase model
[179] audio [177] transformed T_{pitch}=0.7 , T_{timbre}=1.05
[180] audio [177] transformed with phase model T_{pitch}=0.7 , T_{timbre}=1.05
[181] audio [177] transformed T_{pitch}=1.25 , T_{timbre}=0.95
[182] audio [177] transformed with phase model T_{pitch}=1.25 , T_{timbre}=0.95
[183] Kuk13
[184] resynthesis of audio [183] with phase model
[185] audio [183] transformed T_{pitch}=0.7 , T_{timbre}=1.05
[186] audio [183] transformed with phase model T_{pitch}=0.7 , T_{timbre}=1.05
[187] audio [183] transformed T_{pitch}=1.25 , T_{timbre}=0.95
[188] audio [183] transformed with phase model T_{pitch}=1.25 , T_{timbre}=0.95
[189] A-singw0104growl
[190] resynthesis of audio [189] with phase model
[191] audio [189] transformed T_{pitch}=0.7 , T_{timbre}=1.05
[192] audio [189] transformed with phase model T_{pitch}=0.7 , T_{timbre}=1.05
[193] audio [189] transformed T_{pitch}=1.25 , T_{timbre}=0.95
[194] audio [189] transformed with phase model T_{pitch}=1.25 , T_{timbre}=0.95
[195] EBit113Growl
[196] resynthesis of audio [195] with phase model
[197] audio [195] transformed T_{pitch}=0.7 , T_{timbre}=1.05

[198] audio [195] transformed with phase model T_{pitch}=0.7 , T_{timbre}=1.05
[199] audio [195] transformed T_{pitch}=1.25 , T_{timbre}=0.95
[200] audio [195] transformed with phase model T_{pitch}=1.25 , T_{timbre}=0.95
[201] LouisArmstrong
[202] resynthesis of audio [201] with phase model
[203] audio [201] transformed T_{pitch}=0.7 , T_{timbre}=1.05
[204] audio [201] transformed with phase model T_{pitch}=0.7 , T_{timbre}=1.05
[205] audio [201] transformed T_{pitch}=1.25 , T_{timbre}=0.95
[206] audio [201] transformed with phase model T_{pitch}=1.25 , T_{timbre}=0.95
[207] Brenda
[208] resynthesis of audio [207] with phase model
[209] audio [207] transformed T_{pitch}=0.7 , T_{timbre}=1.05
[210] audio [207] transformed with phase model T_{pitch}=0.7 , T_{timbre}=1.05
[211] audio [207] transformed T_{pitch}=1.25 , T_{timbre}=0.95
[212] audio [207] transformed with phase model T_{pitch}=1.25 , T_{timbre}=0.95
[213] C-Jazz1503
[214] resynthesis of audio [213] with phase model
[215] audio [213] transformed T_{pitch}=0.7 , T_{timbre}=1.05
[216] audio [213] transformed with phase model T_{pitch}=0.7 , T_{timbre}=1.05
[217] audio [213] transformed T_{pitch}=1.25 , T_{timbre}=0.95
[218] audio [213] transformed with phase model T_{pitch}=1.25 , T_{timbre}=0.95
[219] Tch7
[220] resynthesis of audio [219] with phase model
[221] audio [219] transformed T_{pitch}=0.7 , T_{timbre}=1.05
[222] audio [219] transformed with phase model T_{pitch}=0.7 , T_{timbre}=1.05
[223] audio [219] transformed T_{pitch}=1.25 , T_{timbre}=0.95
[224] audio [219] transformed with phase model T_{pitch}=1.25 , T_{timbre}=0.95
[225] Cathydry
[226] resynthesis of audio [225] with phase model
[227] audio [225] transformed T_{pitch}=0.7 , T_{timbre}=1.05
[228] audio [225] transformed with phase model T_{pitch}=0.7 , T_{timbre}=1.05
[229] audio [225] transformed T_{pitch}=1.25 , T_{timbre}=0.95
[230] audio [225] transformed with phase model T_{pitch}=1.25 , T_{timbre}=0.95
[231] L-Licksp15
[232] resynthesis of audio [231] with phase model
[233] audio [231] transformed T_{pitch}=0.7 , T_{timbre}=1.05
[234] audio [231] transformed with phase model T_{pitch}=0.7 , T_{timbre}=1.05
[235] audio [231] transformed T_{pitch}=1.25 , T_{timbre}=0.95
[236] audio [231] transformed with phase model T_{pitch}=1.25 , T_{timbre}=0.95
[237] D-Singw2502
[238] resynthesis of audio [237] with phase model
[239] audio [237] transformed T_{pitch}=0.7 , T_{timbre}=1.05
[240] audio [237] transformed with phase model T_{pitch}=0.7 , T_{timbre}=1.05
[241] audio [237] transformed T_{pitch}=1.25 , T_{timbre}=0.95
[242] audio [237] transformed with phase model T_{pitch}=1.25 , T_{timbre}=0.95

[243] TC-Track5
[244] resynthesis of audio [243] with phase model
[245] audio [243] transformed $T_{pitch}=0.7, T_{timbre}=1.05$
[246] audio [243] transformed with phase model $T_{pitch}=0.7, T_{timbre}=1.05$
[247] audio [243] transformed $T_{pitch}=1.25, T_{timbre}=0.95$
[248] audio [243] transformed with phase model $T_{pitch}=1.25, T_{timbre}=0.95$

ADDITIVE VOWEL SYNTESIZER
[249] Synthesis example using the additive vowel synthesizer

TEMPLATES
[250] synthesis using *normal* note attack template
[251] synthesis using *smooth* note attack template
[252] synthesis using *smooth long* note attack template
[253] synthesis using *strong accent* note attack template
[254] synthesis using *low exaggerated* note attack template
[255] synthesis using *sexy* note attack template

SINGING SYNTHESIS EVALUATION
[256] SP Antonio Carlos Jobim – fascination rhythm
[257] OS springsong NTT
[258] DS female synth FLINGER
[259] DS female synth CANTOR
[260] DS ansiedad male synth
[261] SP ansiedad male singer
[262] SP ansiedad male another singer
[263] DS kimi no u wasa male synth
[264] OS hoffnung opera male synth MERON
[265] DS days of wine and roses female synth
[266] OS choir synth Sundberg
[267] DS choir synth
[268] DS feelings female synth
[269] DS ansiedad female synth
[270] SP ansiedad female singer
[271] OS out of life male synth VocalWriter
[272] DS besame mucho female synth1
[273] DS besame mucho female synth2
[274] DS besame mucho male synth
[275] SP bossanova child singer
[276] VS Japanese female synth
[277] OS Mozart opera female synth CHANT
[278] SP jazz female singer
[279] VS Vocalistener Dearest

Bibliography

Abatzoglou, T. "Fast maximum likelihood joint estimation of frequency and frequency rate." *ICASSP*, 1986: vol 2, pp. 1409-1412.

Abe, M., and J.O. Smith. "AM/FM rate estimation for time-varying sinusoidal modeling." *Proc. of Int. Conf. on Acoustics, Speech and Signal Processing.* Vol 3, pp. 201-204, 2005.

Alonso, M. "Expressive Performance Model for a Singing Voice Synthesizer." *Master Thesis, Enginyeria Superior en Informàtica, Universitat Pompeu Fabra.* Barcelona, 2004.

Althoff, R., F. Keiler, and U. Zölzer. "Extracting sinusoids from harmonic signals." *Proceedigns of the 2nd COST G-6 Workshop on Digital Audio Effects (DAFx99).* Trondheim, 1999.

—. "Extracting sinusoids from harmonic signals." *Proceedigns of the 2nd COST G-6 Workshop on Digital Audio Effects (DAFx99).* Trondheim, 1999.

Amatriain, X., J. Bonada, A. Loscos, and X. Serra. "Spectral Modeling for Higher-level Sound Transformation." *Proceedings of MOSART Workshop on Current Research Directions in Computer Music.* Barcelona, Spain, 2001.

Amatriain, X., J. Bonada, A. Loscos, and X. Serra. "Spectral Processing." In *DAFX: Digital Audio Effects*, by Editor Udo Zölzer, pp. 373-438. John Wiley & Sons Publishers, 2002.

Amatriain, X., J. Bonada, A. Loscos, J. Arcos, and V. Verfaille. "Content-based Transformations." *Journal of New Music Research*, 2003: vol 32, nº 1.

Amatriain, X., J. Bonada, and X. Serra. "METRIX: A Musical Data Definition Language and Data Structure for a Spectral Modeling Based Synthesizer." *Proceedings of COST G6 Conference on Digital Audio Effects.* Barcelona, Spain, 1998.

Arfib, D., F. Keiler, and U. Zölzer. "Source-Filter Processing." In *DAFX - Digital Audio Effrects*, by U. Zölzer, pp. 299-372. Chichester: J. Wiley & Sons, 2002.

Auger, F., and P. Flandrin. "Improving the Readability of Time-Frequency and Time-Scale Representations by the Reassignment Method." *IEEE Transactions on Signal Procesing*, 1995: vol. 43, no. 5, pp. 1068-1089.

Black, A. "Perfect Synthesis for All of the People All of the Time." *IEEE TTS Workshop* . Santa Monica, USA, 2002.

Bonada, J. "Automatic Technique in Frequency Domain for Near-Lossless Time-Scale Modification of Audio." *Proceedings of the International Computer Music Conference.* Berlin, Germany, 2000.

—. "Desenvolupament d`un entorn gráfic per a l`análisi, transformació i síntesi de sons mitjanant models espectrals." *Master Thesis, Catalunya Polytechnic University (UPC).* Barcelona, 1997.

—. "High Quality Voice Transformations based on Modeling Radiated Voice Pulses in Frequency Domain." *Proc. of the 7th Int. Conference on Digital Audio Effects.* Naples, Italy, October, 2004.

—. "Voice Solo to Unison Choir Transformation." *Proceedings of 118th Audio Engineering Society Convention.* Barcelona, Spain, 2005.

—. "Wide-Band Harmonic Sinusoidal Modeling." *Proc. of the 11th Int. Conference on Digital Audio Effects.* Helsinki, Finland, 2008.

Bonada, J., A. Loscos, and H. Kenmochi. "Sample-based Singing Voice Synthesizer by Spectral Concatenation." *Proceedings of the Stockholm Music Acoustics Conference.* Stockholm, Sweden, 2003.

Bonada, J., A. Loscos, and M. Blaauw. "Improvements to a Sample-Concatenation Based Singing Voice Synthesizer." *Proceedings of the 121st AES Convention.* San Francisco, USA, October, 2006.

Bonada, J., A. Loscos, O. Mayor, and H. Kenmochi. "Sample-based Singing Voice Synthesizer using Spectral Models and Source-Filter Decomposition." *Proceedings of 3rd Intl. Workshop on Models and Analysis of Vocal Emissions for Biomedical Applications (MAVEBA).* Firenze, Italy, 2003.

Bonada, J., A. Loscos, P. Cano, X Serra, and H. Kenmochi. "Spectral Approach to the Modeling of the Singing Voice." *Proceedings of the 111th AES Convention.* New York, USA, September, 2001.

Bonada, J., and X. Serra. "Synthesis of the Singing Voice by Performance Sampling and Spectral Models." *IEEE Signal Processing Magazine*, January 2007: vol. 24, no. 1.

Bonada, J., M. Blaauw, A. Loscos, and H. Kenmochi. "Unisong: A Choir Singing Synthesizer." *Proceedings of 121st Convention of the Audio Engineering Society.* San Francisco, CA, USA, 2006.

Bonada, J., O. Celma, A. Loscos, J. Ortolà, and X. Serra. "Singing Voice Synthesis Combining Excitation plus Resonance and Sinusoidal plus Residual Models." *Proceedings of International Computer Music Conference.* Havana, Cuba, 2001.

Cano, P. "Fundamental Frequency Estimation in the SMS analysis." *Proceedings of first DAFx conference.* Barcelona, 1998.

Cano, P., A. Loscos, and J. Bonada. "Score-Performance Matching using HMMs." *Proceedings of International Computer Music Conference.* Beijing, China, 1999.

Cano, P., A. Loscos, J. Bonada, M. de Boer, and X. Serra. "Voice Morphing System for Impersonating in Karaoke Applications." *Proceedings of International Computer Music Conference.* Berlin, Germany, 2000.

Cheveigné, A., and H. Kawahara. "Comparative evaluation of F0 estimation algorithms." *7th European Conference on Speech Communication and Technology, EUROSPEECH-2001, 2451-2454.* Denmark, 2001.

Childers, D. G. "Measuring and Modeling Vocal Source-Tract Interaction." *IEEE Transactions on Biomedical Engineering*, July 1994: vol. 41, no. 7, pp. 663-671.

Childers, D.G. "Speech Processing and Synthesis for Assessing Vocal Disorders." *Engineering in Medicine and Biology Magazine, IEEE.* 1990. vol. 9, no. 1, pp.69-71.

Cook, P. "SPASM: A Real-Time Vocal Tract Physical Model Editor/Controller and Singer; The Companion Software Synthesis System." *Computer Music Journal*, 1992: vol. 17, no. 1, pp. 30-44.

Cooper, C., D. Murphy, D. Howard, and A. Tyrrell. "Singing Synthesis with an Evolved Physical Model." *IEEE Transactions on Audio, Speech and Language Processing*, July 2006: vol. 14, no. 4.

Cox, M. G. "An algorithm for approximating convex functions by means of first-degree splines." *Computer Music Journal*, 1971: vol. 14 pp. 272-275.

de Boer, M., J. Bonada, and X. Serra. "Using the Sound Description Interchange Format within the SMS Applications." *Proceedings of International Computer Music Conference.* Berlin, Germany, 2000.

de Boer, M., J. Bonada, P. Cano, A. Loscos, and X. Serra. "Singing Voice Impersonator Application for PC." *Proceedings of International Computer Music Conference.* Berlin, Germany, 2000.

Depalle, P., and X. Rodet. "Synthèse additive par FTT inverse." Rapport Interne IRCAM, Paris, 1990.

Desainte-Catherine, M., and S. Marchand. " High Precision Fourier Analysis of Sounds using Signal Derivatives." *LaBRI Research Report 120498. University of Bordeaux.* 1998.

Desainte-Catherine, M., and S. Marchand. "High-Precision Fourier Analysis of Sounds using Signal Derivatives." *Journal of Audio Engineering Society*, 2002: vol. 48, no. 7/8.

DiFederico, R. "Waveform Preserving Time Stretching and Pitch Shifting for Sinusoidal Models of Sound." *Proc. of the 1st Int. Conference on Digital Audio Effects.* Barcelona, Spain, November, 1998.

Dutoit, T. "High Quality Text-to-Speech Synthesis of the French Language." *PhD thesis, Polytechnique de Mons.* 1993.

Fabig, L., and J. Janer. "Transforming Singing Voice Expression - The Sweetness Effect." *Proc. of the 7th Int. Conference on Digital Audio Effects.* Naples, Italy, 2004.

Fant, G. "Glottal flow: models and interaction." *Journal of Phonetics*, 1986: vol. 14, pp 393-399.

Fitz, K.R. "The Reassigned Bandwith-Enhanced Method." *PhD thesis, University of Illinois.* 1999.

Friberg, A. "A Quantitative Rule System for Musical Performance." *PhD Thesis, Department of Speech, Music and Hearing, Royal Institute of Technology.* Stockholm, 1995.

Frydén, L., J. Sundberg, and A. Askenfelt. "What Tells you the Player Is Musical? An Analysis-by-Synthesis Study of Music Performance." *Publication issued by the Royal Swedish Academy of Music, (39):61-75.* 1983.

García, G. "Analyse des signaux sonores en termes de partiels et de bruit: Extraction automatique des trajets fréquentiels par des modèles de Markov cachés." *Master's thesis, DEA Automatique et Traitement des signaux, Orxay*, 1992.

Garcia, R.A., and K.M. Short. "Accurate Low-Frequency Magnitude and Phase Estimation in the Presence of DC and Near-DC Aliasing." *Proceedings of the 121st AES Convention.* San Francisco, USA, October, 2006.

Garnier, M., N. Henrich, M. Castellengo, D. Sotiropoulos, and D. Dubois. "Characterisation of Voice Quality in Western Lyrical Singing: from Teacher's Judgements to Acoustic Descriptions." *Journal of Interdisciplinary Music Studies*, 2007: vol 2., pp 62-91.

Gómez, E., and J. Bonada. "Tonality visualization of polyphonic audio." *Proceedings of International Computer Music Conference.* Barcelona, Spain, 2005.

Goodwin, M. "Adaptative Signal Models: Theory, Algorithms and Audio Applications." *PhD thesis.* University of California, Berkeley, 1997.

Gouyon, F., et al. "Content processing of musical audio signals." In *Sound to sense sense to sound: A state-of-the-art.* Book to publish in 2007.

Gouyon, F., L. Fabig, and J. Bonada. "Rhythmic expressiveness transformations of audio recordings: swing modifications." *Proceedings of 6th International Conference on Digital Audio Effects.* London, UK, 2003.

Hamon, C., E. Moulines, and F. Charpentier. "A diphone synthesis system based on time-domain prosodic modifications of speech." *Acoustics, Speech, and Signal Processing ICASSP.* Glasgow, UK, 1989. 238-241.

Herrera, P., and J. Bonada. "Vibrato Extraction and Parameterization in the Spectral Modeling Synthesis framework." *Proceedings of COST G6 Conference on Digital Audio Effects.* Barcelona, Spain, 1998.

Huang, X., F. Alleva, H. Hon, M. Hwang, and R. Rosenfeld. "The SPHINX-II Speech Recognition System: an Overview"." *Computer Speech andLanguage, vol. 7, no. 2, pp. 137-148*, 1993.

Janer, J., J. Bonada, and M. Blaauw. "Performance-driven control for sample-based singing voice synthesis." *Proceedings of 9th International Conference on Digital Audio Effects.* Montreal, Canada, 2006.

Janer, J., J. Bonada, and S. Jordà. "Groovator - an implementation of real-time rhythm transformations." *Proceedings of 121st Convention of the Audio Engineering Society.* San Francisco, CA, USA, 2006.

Juslin, P.N., and P. Laukka. "Communication of Emotions in Vocal Expression and Music Performande: Different Channels, Same Code?" *Psychological Bulletin, 129(5):770-814*, 2003.

Kaegi, Werner, and Tempelaars. "VOSIM – a New Sound Synthesis System." *Journal of the Audio Engineering Society*, 1978: vol. 26, no. 6, pp. 418-425.

Kawahara, T., et al. "Free Software Toolkit for Japanese Large Vocabulary Continuous Speech Recognition." *Proceedings of the European Conference on Speech Communication and Technology (EUROSPEECH).* pp.1691-1694, 2001.

Keiler, F., and S. Marchand. "Survey on Extraction of Sinusoids in Stationary Sounds." *Proceedings of the 5th Intl. Conference on Digital Audio Effects.* Hamburg, Germany, September, 2002.

Kelly, J., and C. Lochbaum. "Speech Synthesis." *Proc. of the 4th Int. Congr. Acoustics.* 1962. pp. 1-4.

Kim, Y. E. "Singing Voice Analysis/Synthesis." *PhD Thesis, Massachussetts Institute of Technology, USA*, 2003.

Klatt, D.H. "Software for a cascade/parallel formant synthesizer." *Journal Acoustics of American Society*, 1980: pp. 971-995.

Kob, M. "Physical Modeling of the Singing Voice." *PhD thesis, Institute of Technical Acoustics, Aachen University.* Germany, 2002.

Laroche, J. "Frequency-Domain Techniques for High-Quality Voice Modification." *Proc. of the 6th Int. Conference on Digital Audio Effects.* London, UK, September, 2003.

Laroche, J., and M. Dolson. "About this phasiness business." *Proc. Intern. Computer Music Conf. ICMC.* Thessaloniki, 1997.

Laroche, J., and M., Dolson. "Improved phase-vocoder: Time-Scale Modification of Audio." *IEEE Transactions on Speech and Audio Processing*, 1999: vol. 7, no. 3, pp. 323-332, May.

Lars, F., and J. Janer. "Transforming Singing Voice Expression - The Sweetness Effect." *Proc. of the 7th Int. Conference on Digital Audio Effects.* Naples, Italy, October, 2004.

Larsson, B. "Music and Singing Synthesis Equipment (MUSSE)." *Speech Transmission Laboratory Quaterly Progress and Status Report (STL-QPSR), 1/1977, 38-40.* 1977.

Laurson, M., V. Norilo, and M. Kuuskankare. "PWGLSynth: A Visual Synthesis Language for Virtual Instrument Design and Control." *Computer Music Journal*, 2005: vol. 29, pp. 29-41.

Lindemann, E. "Music Synthesis with Reconstructive Phrase Modeling." *IEEE Signal Processing Magazine*, 2007: vol. 24, no. 1, January.

Lindqvist Gauffin, J. "Inverse Filtering. Instrumentation and Techniques." *STL-QPSR, KTH 4:1-4*, 1964.

Llisterri, J., and J.B. Mariño. "Spanish Adaptation of SAMPA and Automatic Phonetic Transcription." *ESPRIT PROJECT 6819 (SAM-A Speech Technology Assessment in Multilingual Applications).* 1993.

—. "Spanish Adaptation of SAMPA and Automatic Phonetic Transcription." *ESPRIT PROJECT 6819 (SAM-A Speech Technology Assessment in Multilingual Applications).* 1993.

Loscos, A., and J. Bonada. "Emulating Rough And Growl Voice in Spectral Domain." *Proceedings of 7th Int. Conference on Digital Audio Effects.* Naples, Italy, October, 2004.

—. "Esophageal Voice Enhancement by Modeling Radiated Pulses in Frequency Domain." *Proceedings of 121st Convention of the Audio Engineering Society.* San Francisco, CA, USA, 2006.

Loscos, A., P. Cano, and J. Bonada. "Low-Delay Singing Voice Alignment to Text." *Proceedings of International Computer Music Conference*. Beijing, China, 1999.

LTD, Creative Tech. Synthesis of time-domain signals using non-overlapping transforms. US Patent US6311158. October 30, 2001.

Macon, M. W. "Speech Synthesis Based on Sinusoidal Modeling." *PhD thesis, Georgia Institute of Technology, USA*, 1996.

Macon, M. W., L. Jensen-Link, J. Oliverio, M. Clements, and E. B. George. "A system for singing voice synthesis based on sinusoidal modeling." *Proc. of International Conference on Acoustics, Speech, and Signal Processing*. 1997. vol. 1, pp. 435-438.

Macon, M., L. Jensen-Link, J. Oliverio, M. Clements, and E. George. "A Singing Voice Synthesis System Based on Sinusoidal Modeling." *Proc. Intl. Conference on Acoustics, Speech, and Signal Processing ICASSP*. 1997. vol. 1, pp. 435-438.

Maestre, E., J. Bonada, and O. Mayor. "Modeling musical articulation gestures in singing voice performances." *Proceedings of 121st Convention of the Audio Engineering Society*. San Francisco, CA, USA, 2006.

Maestre, E., J. Bonada, M. Blaauw, A. Pérez, and E. Guaus. "Acquisition of Violin Instrumental Gestures Using a Commercial EMF Tracking Device." *Proceedings of the International Computer Music Conference*. Copenhagen, 2007.

Makhoul, J. "Linear prediction: a tutorial review." *Proceedigns of the IEEE*, 1975: vol. 63(4), pp. 561-580.

Marchand, S. "Improving Spectral Analysis Precision with an Enhanced Phase Vocoder using Signal Derivatives." *Proceedings of the workshop on Digital Audio Effects DAFX*. Barcelona, Spain, 1998.

Markel, J. D., and A. H. Gray. *Linear Prediction of Speech*. Springer-Verlag Ed., 1975.

Mayor, O., J. Bonada, and A. Loscos. "The Singing Tutor: Expression Categorization and Segmentation of the Singing Voice." *Proceedings of 121st Convention of the Audio Engineering Society*. San Francisco, CA, USA, 2006.

Mayor, O., J. Bonada, and J. Janer. "Kaleivoicecope: Voice Transformation from Interactive Installations to Video-Games." *AES 35th International Conference*. London, UK, 2009.

McAulay, R., and T. Quatieri. "Speech Analysis/Synthesis based on a Sinusoidal Representation." *IEEE Trans. Acoust., Speech, Signal Processing*, 1986: vol 34, no. 4, pp. 744-754.

Meron, Y. "High Quality Singing Synthesis using the Selection-Based Synthesis Scheme." *PhD thesis, Dept. of Information and Communication Engineering, University of Tokyo*. Tokyo, Japan, 1999.

Moulines, E., C. Hamon, and F. Charpentier. "High-quality prosodic modifications of speech using time-domain overlap-add synthesis." *Twelfth GRETSI Colloquium*. Juan-les-Pins, France, 1989.

Moulines, E., F. Charpentier, and C. Hamon. "A diphone synthesis system based on time-domain prosodic modifications of speech." *Proceedings ICASSP*. 1989. 238-241.

Mullen, J., D. M. Howard, and D.T. Murphy. "Waveguide Physical Modeling of Vocal Tract Acoustics: Flexible Formant Bandwidth Control from Increased Model Dimensionality." *IEEE Transactions on Audio, Speech and Language Processing*, May 2006: vol. 14, no. 3, pp. 964-971.

Mullen, J., D.M. Howard, and D.T. Murphy. "Acoustical Simulations of the Human Vocal Tract Using the 1D and 2D Digital Waveguide Software Model." *Proc. of the 4th Int. Conference on Digital Audio Effects*. Naples, Italy, 2004. pp. 311-314.

Nose, T., J. Yamagishi, T. Masuko, and T. Kobayashi. "A Style Control Technique for HMM-based Expressive Speech Synthesis." *IEICE Transactions on Information and Systems*, 2007: vol. 90, n.9, pp. 1406-1413.

Peeters, G. "Modèles et modification du signal sonore adaptés à ses caractéristiques locales." *PhD Thesis, spécialité Acoustique Traitement du signal et Informatique Appliqués à la Musique, Universite Paris 6*, 2001.

Pérez, A., J. Bonada, E. Maestre, E. Guaus, and M. Blaauw. "Combining Performance Actions with Spectral Models for Violin Sound Transformation." *19th Intl. Congress on Acoustics.* Madrid, 2007.

—. "Combining Performance Actions with Spectral Models for Violin Sound Transformation." *19th International Congress on Acoustics (ICA).* Madrid, 2007.

—. "Score Level Timbre Transformations of Violin Sounds." *Proc. of the 11th Intl. Conference on Digital Audio Effects (DAFx-08).* Espoo, Finland, 2008.

Puckette, M. "Phase-locked Vocoder." *IEEE ASSP Conference on Applications of Signal Processing to Audio and Acoustics.* Mohonk, 1995.

Rabiner, L.R., and B. Juang. *Fundamentals of Speech Recognition.* Prentice-Hall, Inc., 1993.

Röbel, A. "A new approach to transient processing in the phase vocoder." *Proceedings DAFx.* London, 2003.

—. "Frequency slope estimation and its application for non-stationary sinusoidal parameter estimation." *Proceedings of the 10th Intl. Conference on Digital Audio Effects (DAFx-07).* Bordeaux, France, 2007.

Röbel, A., and X. Rodet. "Efficient Spectral Envelope Estimation and its Application to Pitch-Shifting and Envelope Preservation." *Proceedings of teh 8th Intl. Conference on Digital Audio Effects, pp. 30-35.* Madrid, 2005.

Rodet, X. "Synthesis and Processing of the Singing Voice." *1st IEEE Benelux Workshop on Model Based Processing and Coding of Audio.* Leuven, 2002.

Rodet, X., Y. Potard, and J.B. Barrière. "The CHANT Project: from the Synthesis of the Singing Voice to Synthesis in General." *Computer Music Journal*, 1984: 8(3):15-31.

Ross, J., and J. Sundberg. "Syllable and Tone Boundaries in Singing." *4th Pan European Voice Conference.* Stockholm, Sweden, August, 2001.

Sakakibara, L.-I., L. Fuks, H. Imagawa, and N. Tayama. "Growl Voice in Ethnic and Pop Styles." *Proceedings of the Intl. Symposium on Musical Acoustics ISMA.* Nara, Japan, April, 2004.

Saunders, C., D. Hardoon, J. Shawe-Taylor, and G. Widmer. "Using String Kernels to Identify Famous Performers from their Playing Style." *Proceedings of the 15th European Conference on Machine Learning.* Pisa, Italy, 2004.

Schoentgen. "Stochastic Models of Jitter." *Journal of Acoustic Society of America.* 2001. vol. 109, no. 4,pp. 1631-1650.

Schwarz, D. "Corpus-based Concatenative Synthesis." *IEEE Signal Processing Magazine*, 2007: vol. 24, no. 1, January .

Sedgewick, R. *Algoritmhs.* Addison-Wesley, eds., 1998.

Serra, X. "A System for Sound Analysis-Transformation-Synthesis based on a Deterministic plus Stochastic Decomposition." *PhD thesis, CCRMA, Dept. of Music, Stanford University.* USA, 1989.

Serra, X., and J. Bonada. "Sound Transformations Based on the SMS High Level Attributes." *Proceedings of COST G6 Conference on Digital Audio Effects.* Barcelona, Spain, 1998.

Serra, X., J. Bonada, P. Herrera, and R. Loureiro. "Integrating Complementary Spectral Models in the Design of a Musical Synthesizer." *Proceedings of International Computer Music Conference.* Thessaloniki,Greece, 1997.

Smith, J.O. "Physical Modeling Using Digital Waveguides." *Computer Jusic Journal*, 1992: vol. 16, no. 4, pp. 74-87.

Smits, R., and B. Yegnanarayana. "Determination of Instants of Significant Excitation in Speech using Group Delay Function." *IEEE Transactions on Speech and Audio Processing*, 1995.

Story, B.H. "Using Imaging and Modeling Techniques to Understand the Relation Between Vocal Tract Shape to Acoustic Characteristics." *Proc. Stockholm Music Acoustics Conf. SMAC-03.* 2003. pp. 435-438.

Story, B.H., I.R. Titze, and E.A., Hoffman. "Vocal Tract Area Functions from Magnetic Resonance Imaging." *Journal of Acoustics Society of America*, 1996: vol. 104, no. 1, pp. 471-487.

Strawn, J. "Approximation and syntactic analysis of amplitude and frequency functions for digital sound synthesis." *Computer Music Journal*, 1995: vol. 4, no. 3, pp. 678-689.

Sundberg, J. "The KTH Synthesis of Singing." *Advances in Cognitive Psychology*, 2006: vol. 2, no. 2-3, pp. 131-143.

Sundberg, J. "The science of the Singing Voice." Northern Illinois University Press, 1987.

Titze, I.R. "Workshop on Acoustic Voice Analysis, Summary Statement." *Nat. Center for Voice and Speech.* Denver, Colorado, 1994.

Toda, T., Y. Ohtani, and K. Shikano. "One-to-Many and Many-to-One Voice Conversion based on Eigenvoices." *Proceedings of ICASSP.* vol. 4, pp. 1249-1252, Honolulu, HI, 2007.

Verma, T.S., S.N. Leving, and T. Meng. "Transient Modeling Synthesis: a Flexible Analysis/Synthesis Tool for Transient Signals." *Proceedings of the Intl. Computer Music Conference.* Greece, 1997.

Vine, D.S.G., and R. Sahandi. "Synthesis of Emotional Speech Using RP-PSOLA." *IEEE Colloquium on the State of the Art in Speech Synthesis.* London, 2000.

Vinyes, M. "On time localization of sinusoids from the DFT of their sum." *Internal Report, Music Technology Group, Universitat Pompeu Fabra.* Barcelona, 2006.

Vinyes, M., J. Bonada, and A. Loscos. "Demixing Commercial Music Productions via Human-Assisted Time-Frequency Masking." *Proceedings of the 120th AES Convention.* Paris, France, 2006.

Widmer, G., and A. Tobudic. "Playing Mozart by Analogy: Learning Multi-Level Timing and Dynamics Strategies." *Journal of New Music Research*, 2003: 32, pp. 259-268.

Widmer, G., and W. Goebl. "Computational Models of Expressive Music Performance: The State of the Art." *Journal of New Music Research*, 2004: vol. 33, no. 3, pp. 203-216.

Yegnanarayana, B., and R. Veldhuis. "Extraction of Vocal-Tract System Characteristics from Speech Signal." *IEEE Transactions on Speech and Audio Processing*, 1998: vol. 6, no. 4, pp. 313-327.

Zivanovic, M., A. Röbel, and X. Rodet. "Adaptive Threshold Determination for Spectral Peak Classification." *Proc. of the 10th Int. Conference on Digital Audio Effects (DAFx-07).* Bordeaux, France, 2007.