Singing Voice Analysis/Synthesis

by

Youngmoo Edmund Kim

B.S Engineering, Swarthmore College, 1993
B.A. Music, Swarthmore College, 1993
M.S. Electrical Engineering, Stanford University, 1996
M.A. Vocal Performance Practice, Stanford University, 1996

Submitted to the Program in Media Arts and Sciences,
School of Architecture and Planning,
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

September, 2003

Author .....
Program in Media Arts and Sciences
August 15, 2003

Certified by .....
Barry L. Vercoe
Professor of Media Arts and Sciences
Thesis Supervisor

Accepted by .....
Andrew B. Lippman
Chairman, Departmental Committee on Graduate Students

Singing Voice Analysis/Synthesis

by Youngmoo Edmund Kim

Submitted to the Program in Media Arts and Sciences,
School of Architecture and Planning, on August 15, 2003,
in partial fulfillment of the requirements
for the degree of Doctor of Philosophy

Abstract

The singing voice is the oldest and most variable of musical instruments. By combining music, lyrics, and expression, the voice is able to affect us in ways that no other instrument can. As listeners, we are innately drawn to the sound of the human voice, and when present it is almost always the focal point of a musical piece. But the acoustic flexibility of the voice in intimating words, shaping phrases, and conveying emotion also makes it the most difficult instrument to model computationally. Moreover, while all voices are capable of producing the common sounds necessary for language understanding and communication, each voice possesses distinctive features independent of phonemes and words. These unique acoustic qualities are the result of a combination of innate physical factors and expressive characteristics of performance, reflecting an individual's vocal identity.

A great deal of prior research has focused on speech recognition and speaker identification, but relatively little work has been performed specifically on singing. There are significant differences between speech and singing in terms of both production and perception. Traditional computational models of speech have focused on the intelligibility of language, often sacrificing sound quality for model simplicity. Such models, however, are detrimental to the goal of singing, which relies on acoustic authenticity for the non-linguistic communication of expression and emotion. These differences between speech and singing dictate that a different and specialized representation is needed to capture the sound quality and musicality most valued in singing.

This dissertation proposes an analysis/synthesis framework specifically for the singing voice that models the time-varying physical and expressive characteristics unique to an individual voice. The system operates by jointly estimating source-filter voice model parameters, representing vocal physiology, and modeling the dynamic behavior of these features over time to represent aspects of expression. This framework is demonstrated to be useful for several applications, such as singing voice coding, automatic singer identification, and voice transformation.

Thesis supervisor: Barry L. Vercoe, D.M.A.
Title: Professor of Media Arts and Sciences

Thesis Committee

Thesis supervisor .....

Barry L. Vercoe
Professor of Media Arts and Sciences
Massachusetts Institute of Technology

Thesis reader .....

Rosalind Picard
Associate Professor of Media Arts and Sciences
Massachusetts Institute of Technology

Thesis reader .....

Perry R. Cook
Associate Professor of Computer Science (also Music)
Princeton University

Acknowledgments

First and foremost, I must thank Professor Barry Vercoe for bringing me into his group and providing such a tremendously supportive environment for my research. Researchers in this field are necessarily of two minds, technical and musical, but are often forced to focus on one to the detriment of the other. Barry, however, has always encouraged my continued development in both areas, allowing me to pursue my study of singing performance unfettered. I consider myself to be extremely fortunate to have landed in such circumstances.

My association with Perry Cook dates back to my studies as a Masters student at Stanford. His dissertation was the inspiration for my own work, and his accomplishments as a researcher and a singer are a model for my own personal goals. I am grateful for his mentoring and his contributions as a reader and critic of this dissertation.

I am indebted to Roz Picard for her skillful teaching of the concepts of pattern recognition and machine learning in addition to her thoughtful comments, criticism, and encouragement provided throughout the thesis writing process. I also had the privilege of serving as her teaching assistant for Signals and Systems during which time I was able to learn from a master teacher first hand.

This dissertation owes a great deal to the members of the Machine Listening/Music, Mind and Machine group, past and present. I have had the great fortune of working with and following in the footsteps of many brilliant and talented people. Prior to my tenure in the group, the work of Dan Ellis, Bill Gardner, and Michael Casey introduced me to approaches to audio analysis beyond signal processing, which was an entirely new concept to me at the time. Among my contemporaries, Keith Martin and Eric Scheirer deserve special mention for their influence as mentors and co-workers who set a remarkably high standard for the rest of us to follow. I am thankful for their willingness to teach, discuss, and share ideas spanning a very broad range of fields and interests. Thanks also to Judy Brown, Paris Smaragdis, Nyssim Lefford, Wei Chai, Ricardo Garcia, Rebecca Reich, and Brian Whitman for their innumerable contributions as co-workers and friends.

I have been assisted and influenced by many other members of the Media Lab community. In particular, I thank Connie Van Rheenen and Kristie Thompson for their support, assistance, and encouragement during my time here.

I am grateful to all of my voice teachers, Lelia Calder, Karen Clark Young, Gregory Wait, and Bill Cutter, for their help and encouragement in my continual attempts to improve as a singer. Their efforts inspired me to learn as much about singing and musicianship as I could.

Special thanks to my friends from the Tanglewood Festival Chorus, Jeni Cameron, Jeramie Hammond, Laura Mennill, and Johanna Schlegel, all fabulous singers who allowed themselves to be recorded and serve as data for my research.

My greatest debt of gratitude is owed to my parents, the Drs. Kim, whose hard work and sacrifices gave me the opportunities I have today. Their unending thirst for knowledge and passion for learning has been a constant source of inspiration, and this dissertation is dedicated to them.

And I am extremely grateful to Charlene Bierl for her love and support.

Table of Contents

1 Introduction 13
- 1.1 Contributions . . . 15
- 1.2 Overview and Organization . . . 16

2 Background 19
- 2.1 Anatomy of the Singing Voice . . . 19
  - 2.1.1 The process of singing . . . 22
- 2.2 Speech vs. Singing . . . 23
- 2.3 Early voice simulations . . . 23
  - 2.3.1 The Vocoder and Voder . . . 24
- 2.4 Advancements in Voice Analysis/Synthesis . . . 25
  - 2.4.1 Improvements to the Vocoder . . . 25
  - 2.4.2 Formant Vocoder . . . 25
  - 2.4.3 Homomorphic Vocoder . . . 25
  - 2.4.4 Linear Predictive Coding . . . 26
  - 2.4.5 Code-Excited Linear Prediction . . . 26
  - 2.4.6 Sinusoidal voice coding . . . 27
  - 2.4.7 High-resolution time-frequency voice analysis/synthesis . . . 27
  - 2.4.8 PSOLA and WSOLA . . . 28
- 2.5 Synthesis of the Singing Voice . . . 28
  - 2.5.1 Formant-based synthesis . . . 28
  - 2.5.2 Formant Wave Functions . . . 29
  - 2.5.3 Waveguide vocal tract physical modeling . . . 29
  - 2.5.4 Sinusoid-based synthesis . . . 29
- 2.6 Related work on Talker ID and Instrument ID . . . 30
- 2.7 Structured Audio . . . 31
  - 2.7.1 Structured Scene Description . . . 32
  - 2.7.2 Singing in the Structured Audio Context . . . 33

3 Source-Filter Parameterization 37
- 3.1 Source-Filter Parameter Estimation . . . 37
  - 3.1.1 KLGLOTT88 model . . . 39
  - 3.1.2 Linear prediction and frequency warping . . . 40
  - 3.1.3 Joint source-filter parameter estimation . . . 43
  - 3.1.4 Parameter estimation for very short periods . . . 49
- 3.2 Glottal Closure Instant and Open-Quotient Determination . . . 52
- 3.3 LF model parameter estimation . . . 54

3.4 Stochastic Component Estimation 58
3.5 Parameter selection and transformation 62
3.6 Summary 63

4 Dynamic Parameter Modeling 65
4.1 Phonetic Segmentation 66
4.1.1 Mel-frequency cepstral coefficients 67
4.1.2 System training 68
4.1.3 Alignment via dynamic programming 69
4.2 Dynamic Parameter Modeling using Hidden Markov Models 70
4.2.1 Specifying the HMM 73
4.2.2 HMM training 74
4.2.3 Estimating state paths 75
4.2.4 Evaluating model likelihood 77
4.3 Summary 78

5 Experiments 79
5.1 Singer Identification 79
5.1.1 Perceptual experiment 81
5.2 Singing Voice Coding 82
5.2.1 Listening experiment 84
5.2.2 Coding efficiency 86
5.3 Voice Transformation 88
5.3.1 Listening Experiment 90
5.4 Summary 91

6 Conclusions 93
6.1 Possible Refinements to the Framework 93
6.2 Fundamental System Limitations 94
6.3 Directions for Further Research 95
6.4 Concluding Remarks 96

Appendix A HMM Tutorial 97
A.1 HMM training 97
A.2 Estimating state paths 100
A.3 Evaluating model likelihood 101

Appendix B Hybrid Coding Scheme 103
B.1 Score-based Parameter Extraction 103
B.1.1 Analysis blocks 104
B.1.2 Pitch Detection 104
B.1.3 Vowel Onset and Release Detection 105
B.1.4 Vowel identification and compression 107
B.1.5 Hybrid Coding Format 107
B.2 Extensions 108

Bibliography 109

List of Figures

1-1 Flow diagram of analysis framework components 16
1-2 Flow diagram of parametric re-synthesis 17

2-1 The anatomy of the voice (after [61]) 20
2-2 Superior view of the vocal folds: (left) abducted, (center) adducted, (right) forced open by breath pressure, as during phonation (after [80]). 21
2-3 A lossless acoustic tube 21
2-4 The frequency response of a lossless acoustic tube 22
2-5 Multiple source encoding using MPEG-4 scene description 33
2-6 The level of structure in various techniques for voice coding and synthesis 35

3-1 Flow diagram of pitch synchronous parameter estimation 38
3-2 The KLGLOTT88 glottal flow derivative model 41
3-3 Varying levels of frequency warping (left) and non-warped and warped linear prediction (right). Note that the warped LP is able to resolve the closely-spaced formants at low frequencies. 42
3-4 Convex vs. non-convex optimization. On the left, function $e_1(x)$ has one minimum x_o, that is both local and global. On the right, $e_2(x)$ has several local minima, x_1, x_2, and x_3, where x_2 is the global minimum. 46
3-5 Joint source-filter parameter estimation of vowel [e] 48
3-6 Top: two periods of singing $s[n]$ sampled at 16 kHz (period approximate 5 msec). Middle: $s[n]$ delayed by 10 samples. Bottom: 10-fold warped all-pass shift operator $d_{10}\{\cdot\}$ applied to $s[n]$ 50
3-7 Top left: KLGLOTT88 model fit to two short periods ($\sim$ 3 msec). Bottom left: Corresponding warped vocal tract filter. Right: Warped z-plane representation of estimated filter. 52
3-8 Top left: KLGLOTT88 model fit to three very short periods ($<$ 2 msec). Bottom left: Corresponding warped vocal tract filter. Right: Warped z-plane representation of estimated filter. 53
3-9 Estimated glottal derivatives for a range of periods and open-quotients. The highlighted box is the fit with the lowest error. 54
3-10 Corresponding estimated vocal tract filters to Figure 3-9. Again, the highlighted box is the lowest error fit. 55
3-11 The best estimated KLGLOTT88 and warped LP parameters for the given range of T and OQ. 56
3-12 The LF glottal derivative wave model and its parameters. 57

3-13 LF model fits of two analysis periods. 58
3-14 Graphical representation of PCA: the variance of the data is best captured by the basis defined by the eigenvectors, v_1 and v_2, which is a rotation of coordinates. 59
3-15 Time-varying source-filter parameters from one vocal segment. 63

4-1 Flowchart of the phonetic segmentation process. 67
4-2 An illustration of dynamic programming 69
4-3 Example of phonetic segmentation (lighter colors indicate shorter distances) 71
4-4 A 4-state unidirectional HMM with transition probabilities 72
4-5 A 3-state fully-connected HMM with transition probabilities 72
4-6 A depiction of an actual parameter trajectory (black line) and the estimated trajectory via state path (gray line) in parameter space. 73
4-7 HMM state log-likelihood and state path for a scale passage on the vowel [e]. 75
4-8 The parameter observations from a scale on the vowel [e]. On the left are line spectrum frequencies derived from the warped LP coefficients. On the right are the LF model parameters. 76
4-9 The reconstructed parameter trajectories from the observation sequence of Figure 4-8. 77

5-1 Interface to singer identification perceptual experiment. 82
5-2 Re-synthesis of one period using different numbers of residual principle components. Left: modeled residual. Center: reconstructed excitation. Right: resulting synthesis. 83
5-3 Mean SNR across nine sound samples using different numbers of residual principle components. 84
5-4 Interface to sound quality perceptual experiment. 85
5-5 Average quality ratings for re-synthesized examples using varying number of codebook components. The dashed line indicates the average quality rating for the original source sounds. 86
5-6 Average listener preference of original vs. re-synthesized samples for varying codebook sizes. 87
5-7 HMM state histogram for two singers for vowel [e]. 89
5-8 Interface to sound similarity perceptual experiment. 90
5-9 Average listener judgment of original vs. transformed samples. 91

B-1 Block diagram of analysis system. 104
B-2 Vowel onset detection of [e] in alleluia. 105
B-3 Vowel release detection of [e] in alleluia. 106
B-4 Sum of formant distances to vowel templates for alleluia. Smaller distance indicates a better match. 107

CHAPTER ONE

Introduction

The singing voice is the oldest musical instrument, but its versatility and emotional power are unmatched. Through the combination of music, lyrics, and expression the voice is able to affect us in ways that no other instrument can. The fact that vocal music is prevalent in almost all cultures is indicative of its innate appeal to the human aesthetic. Singing also permeates most genres of music, attesting to the wide range of sounds the human voice is capable of producing. As listeners we are naturally drawn to the sound of the human voice, and when present it immediately becomes the focus of our attention. This thesis is an exploration of the qualities that make the sound of singing voice so compelling.

Individual voices are highly distinctive and are reflections of the identity of the singer. Once one becomes familiar with a particular singer's voice, one can usually identify that voice in other pieces. Our ability to recognize voices is apparently independent of the music itself. For example, we are quite capable of identifying familiar singers in pieces that we haven't heard before. Also, very little evidence is required for identification: a determination can sometimes be made from just a second or two of sound. And familiarity with a particular voice may be gained after relatively little exposure. After listening to just a verse or a phrase of a song, we often have an idea of the essence of that voice's sound.

Our ability to connect vocal sounds to singer identity rests upon two primary systems: the human auditory system and the physiology of the vocal apparatus. Given the importance and usefulness of vocal communication, it is not surprising that our auditory physiology and perceptual apparatus has evolved to be highly sensitive to the human voice. From the standpoint of evolution, such sensitivity most likely aided the survival and propagation of the species. Perhaps equally important in terms of evolution was the development of an extremely flexible vocal apparatus to facilitate communication. In spite of its extreme flexibility, however, the vocal apparatus is also highly self-consistent. An endless variety of sounds can be produced by a fairly simple physical system of vibration and resonance.

Describing the distinctive character of a voice, however, is difficult without resorting to vague and subjective terms (e.g. "rough" or "squeaky") that have no objective correlates. These qualities are believed to be a combination of physical factors, such as vocal

tract size, and learned expressive factors, such as accent. But quantifying, extracting, and modeling these features has proven to be an extremely difficult task. The standard analysis tools and algorithms of audio signal processing, which have been successful in the analysis of other musical instruments, have fallen far short when it comes to modeling the singing voice.

Similarly, understanding the perceptual features that allow the voice to command such attention, especially in the presence of other instruments or other interfering sounds, has been difficult. Even simply identifying its presence amongst other sounds, a trivial task for a human listener, has been difficult to achieve with computational methods, though the difficulty of this task extends to almost any class of sounds. In many ways, we know less about the perceptually salient features of the voice than we do about modeling the vocal apparatus.

The singing voice has also proven to be very difficult to simulate convincingly, much more so than other musical instruments. The instruments of the orchestra are relatively new when compared with the duration of human evolution, and there are many more years of perceptual fine-tuning to overcome in the case of the singing voice. The voice also presents a challenge because of its greater amount of physical variation compared to other instruments. In order to pronounce different words, a person must move their jaw, tongue, teeth, etc., changing the shape and thus the acoustic properties of the vocal mechanism. This range of acoustic variation is difficult to capture in a low-dimensional model. Since no other instrument exhibits the amount of physical variation of the human voice, synthesis techniques that are well suited to other musical instruments often do not apply well to speech or singing. Comparisons of early work in speech and singing voice synthesis to modern systems demonstrate that progress has been very slow in synthetic voice generation.

Direct synthesis of singing, as opposed to re-synthesis or encoding of an existing signal, adds even more challenges. Most languages can be represented in terms of a limited set of *phonemes* (basic linguistic units of sound), but the rules governing pronunciation and inflection are only guidelines to the actual speaking or singing of that language. The problem of interpreting a musical score in a musically proper and convincing fashion is difficult in the case of any instrument. Direct singing synthesis must overcome both of these hurdles.

Because of the large variability involved with the voice (between different voices and within individual voices themselves), a great deal of speech and singing research has investigated an *analysis/synthesis* approach, where the voice is analyzed (deconstructed) according to some assumed model and a synthesis (or more properly, re-synthesis) formed from parameters established during the analysis. This frames the task as an encoding/decoding problem, and the analysis/synthesis approach has led to tremendous gains in the transmission of a source signal at a reduced information rate. In particular, systems based on *Linear Predictive Coding* (LPC) are the basis of most low-bitrate (speech) coding techniques in use today. The applications of the analysis/synthesis approach, however, are not limited simply to coding. The parameters extracted during analysis can be modified to alter the synthesis. There have been many successful

approaches to voice coding which have also been applied to sound modification and restoration.

This is not to say that meaningful analysis requires synthesis. For example, a voice identification system does not need to synthesize the voices it attempts to identify. Depending on the application, an analysis-only system may require far fewer features than an analysis/synthesis system. The great advantage, however, of analysis/synthesis is the possibility of evaluation of the re-synthesized sounds by human listeners. A system that attempts to capture vocal identity can be evaluated on the basis of whether its re-synthesized output is perceived to preserve that identity. If the synthesis is successful, then the features used in the analysis would seem to accurately capture the essence of voice quality. This is the appeal of analysis/synthesis and it is the primary reason it is the focus of this dissertation.

1.1 Contributions

In this dissertation, I propose a novel framework for analysis and (re-)synthesis of the singing voice using a representation that attempts to accurately preserve the perception of singer identity. This framework is based on the hypothesis that physical characteristics and learned features of expression are the two primary factors responsible for the unique sound of an individual's singing voice, analogous to the contributions of a musical instrument (the physical characteristics) and the technique of an instrumentalist (expression). (Of course, in the case of the voice it is much more difficult to separate the instrument from the instrumentalist!)

To represent the physical characteristics of the voice, the framework simultaneously derives features reflecting the physical configuration of the vocal folds and the vocal tract for each pitch period using a joint-optimization technique proposed by Lu [42]. I have extended this technique to utilize a warped frequency scale for the estimation of the vocal tract filter to more accurately reflect perceptual frequency sensitivity. The joint estimation procedure is further enhanced to include simultaneous estimation of the pitch period and the instants of glottal closure. The signal residual model is an original approach in which a residual codebook is trained specifically for an individual singer.

A major contribution of this thesis is a new dynamic representation of the singing voice that uses Hidden Markov Models to compactly describe the time-varying motion of the estimated vocal fold and vocal tract parameters of an individual singer. The states comprising these models reflect physical characteristics of the singer while the pattern of state-to-state movement is representative of the singer's expressive qualities. This model facilitates the goal of separating the vocal instrument (the model states) from the technique of the instrumentalist (the state path).

This dissertation also includes the results of several perceptual experiments involving human listeners using synthetic sound examples generated using the analysis/synthesis

framework. In general, these experiments validate the proposed models and methods as being central to the representation and preservation of singer identity.

1.2 Overview and Organization

The first stage of the analysis framework entails the identification and extraction of specific computationally derived parameters motivated by the physical features of the voice. These parameters are estimated from sound recordings of singing using classical audio signal processing techniques in combination with numerical optimization theory. In the second stage of analysis, the parameters are modeled dynamically to capture the time-varying nature of the qualities of expression using algorithms from pattern recognition and machine learning. The overall analysis framework is organized according to these two stages of analysis.

In the first stage of analysis the following key parameters are estimated: pitch period, instants of glottal closure, and glottal waveform and vocal tract filter parameters. The second stage encompasses phonetic detection and segmentation of the source signal and dynamic parameter modeling using a Hidden Markov Model. The output of the analysis system is a *state path*, which encapsulates the variance of the parameters and their evolution over time. Figure 1-1 depicts the overall flow diagram linking each component of the framework.

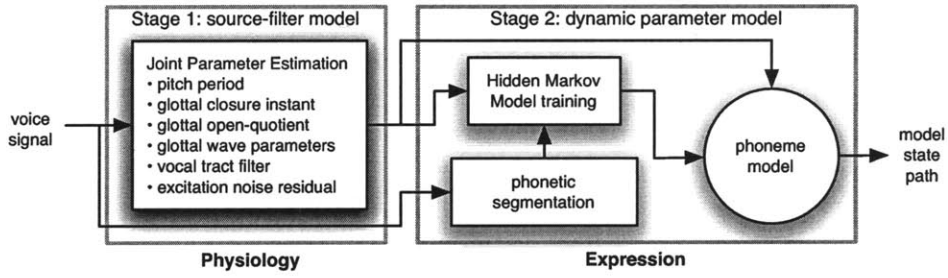


Figure 1-1: Flow diagram of analysis framework components

The system for voice re-synthesis is basically the inverse of the analysis system. The state path is used to drive the phoneme model in order to re-create a time series of source-filter parameters that reflect those of the original voice signal. The regenerated parameters are inputs to the source-filter model components, which output the re-synthesized singing signal. The flowchart describing the re-synthesis process is presented in Figure 1-2.

Accordingly, the remainder of this thesis is organized along the following chapters:

Chapter 2 provides background material related to singing voice analysis/synthesis. This includes an overview of the anatomy and physiology of the vocal apparatus. I

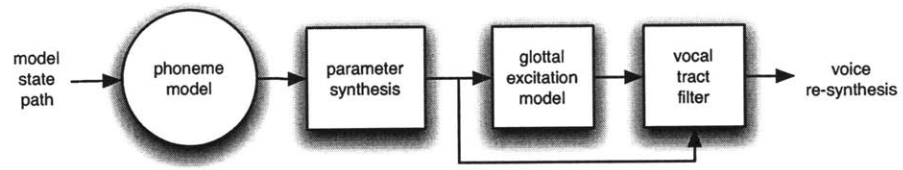


Figure 1-2: Flow diagram of parametric re-synthesis

also present a detailed summary of the theory and methods of prior research related to the analysis, modeling, or simulation of the singing voice.

In Chapter 3, I introduce the source-filter model used to parameterize the physical features of the singing voice. I provide a detailed description of the analysis algorithms used in the extraction of these features from acoustic recordings of the singing voice.

In Chapter 4, I describe the system used to model the evolution of the physical features from the previous chapter over time using a probabilistic Hidden Markov Model. This part of the system is intended to model the expressive qualities of the voice.

Chapter 5 details experiments using the analysis/synthesis framework. The experiments examine the validity of the system in the applications of singing voice coding, singing voice identification, and singing voice transformation.

I conclude with Chapter 6, in which I evaluate the potential of the analysis/synthesis framework and discuss some of the system's inherent limitations. I also suggest potential improvements to the framework as well as some general directions for future singing voice research.

CHAPTER TWO

Background

When attempting to create models of the singing voice, it is useful to understand the mechanics of singing since many modern representations of the voice are simplifications of these mechanics. I will begin with a brief summary of the anatomy of the voice and a short explanation detailing the physiological process of singing. The sections that follow describe previous and current work related to singing voice analysis/synthesis. I will briefly summarize early research on representations for voice transmission and simulation, which are the foundations of systems used today. Following that are overviews of modern systems for voice coding and singing voice synthesis. Other relevant research on certain aspects of machine listening, such as instrument identification and talker identification is also discussed. This chapter concludes with a discussion of structured audio, a concept which has provided the foundation for and motivated the research in this dissertation.

2.1 Anatomy of the Singing Voice

The anatomy of the voice consists of three primary collections of organs: the *respiratory system*, the *larynx*, and the *oropharynx*. These are illustrated in Figure 2-1. The respiratory system consists of the lungs and the diaphragm muscle, which are responsible for storing air and governing breathing, respectively. The movement of the diaphragm compels the lungs to expand and contract, resulting in the air pressure changes necessary for inhalation and exhalation. Sound, of course, is the movement of air molecules and in vocal sounds the release of stored air pressure in the lungs provides the airflow necessary for sound production.

The larynx consists of a skeleton of cartilage (named the thyroid, cricoid, and arytenoid cartilage) enclosing and supporting two structures of muscle and ligaments covered by mucous membranes. These structures are known as the *vocal folds*, which are the primary source for the production of harmonic (pitched) vocal sounds. When the folds are pulled apart, or *abducted*, the air is allowed to pass freely through, as is the case with breathing (Figure 2-2, left). When the folds are pulled together, or *adducted*, the airflow is constricted (Figure 2-2, center), which is the preparatory condition for vibration as we will see shortly. The muscles of vocal folds can alter the shape and stiffness of the

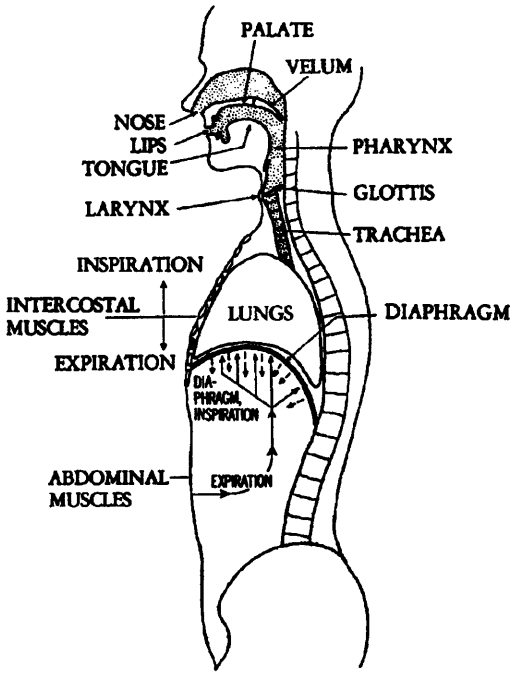


Figure 2-1: The anatomy of the voice (after [61])

folds, resulting in corresponding changes to the acoustic sound output. Vocal fold physiology is believed to be one of the key factors in establishing voice quality.

The oropharynx is the combination of cavities above the larynx comprising the pharynx, oral cavity, and nasal cavity and is also known as the vocal tract. The key characteristic of the vocal tract is its polymorphism, i.e. its ability to assume a wide range of different shapes, which are easily altered by articulating (modifying) the position of the the jaw, tongue, and lips. Since the acoustic properties of an enclosed space follow directly from the shape of that space, the physical flexibility of the vocal tract lends itself to tremendous acoustic flexibility.

To illustrate this, consider an extremely simple model of the vocal tract: a lossless acoustic tube (Figure 2-3). The tube is closed at one end (the vocal folds) and open at the other (the mouth). At the closed end the volume velocity of the air must be zero, forcing all sound velocity waves to have zero amplitude at that point. Consequently, sound waves of certain wavelengths will attain maximum amplitude precisely at the open end of the tube. These particular wavelengths (λ_k) are proportional to the length of the tube:

$$\lambda_k = \frac{4}{k}L, \text{where } k \text{ is odd} \tag{2.1}$$

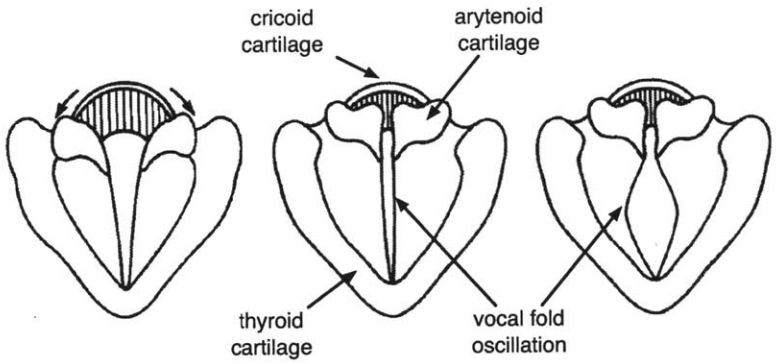


Figure 2-2: Superior view of the vocal folds: (left) abducted, (center) adducted, (right) forced open by breath pressure, as during phonation (after [80]).

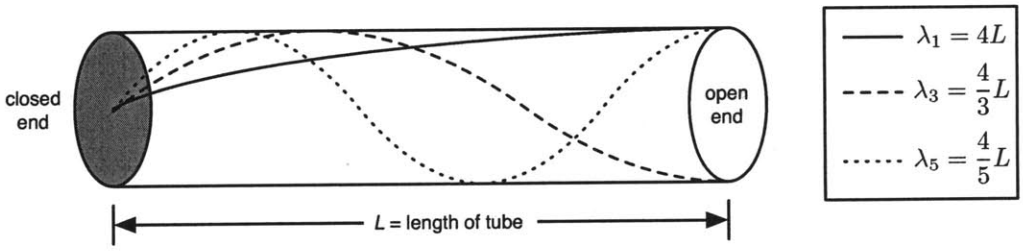


Figure 2-3: A lossless acoustic tube

Since frequency is simply the inverse of wavelength, the corresponding frequencies are also proportional to the length of the tube (Figure 2-4):

$$f_k = \frac{1}{\lambda_k} = \frac{k}{4L}, k \text{ odd} \tag{2.2}$$

Generally these frequencies resulting in maximal endpoint amplitude are called *resonances*, but when specific to the voice they are known as *formants*. Since waves of all other frequencies will not attain maximum amplitude at the open end of the tube, the tube can be viewed as attenuating all other frequencies to varying degrees. From this standpoint, the tube is an acoustic *filter*, altering an input signal according to the tube's physical characteristics. Simply changing the length of the tube changes its frequency response, creating a different filter. Of course in reality the shape of the vocal tract isn't nearly as simple nor is it lossless, but this example illustrates the direct link between the physical length of the vocal tract and acoustic properties.

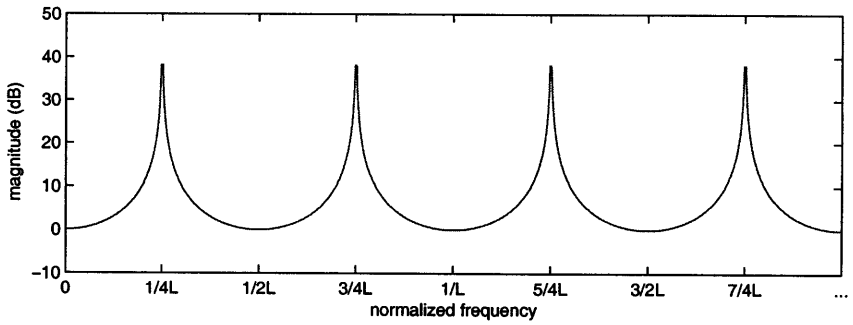


Figure 2-4: The frequency response of a lossless acoustic tube

2.1.1 The process of singing

The process of singing begins with breath pressure produced by the lungs. In the case of *voiced* sounds, the cricoarytenoid muscles initially adduct the vocal folds, but the breath pressure forces them open (Figure 2-2, right). The airflow through the opening is uneven, with the air adjacent to the folds traveling a greater distance than the unimpeded air flowing through the opening. The result is a pressure differential, which causes the vocal folds to be sucked back together by the Bernoulli force. Rapid repetition of this process is called *phonation*, and the frequency of this repetition correlates to our perception of the pitch. In voiced sounds, phonation results in a largely harmonic sound source. For *unvoiced* sounds, the vocal folds remain open and the breath pressure results in free airflow through the larynx into the mouth where it is impeded by a constriction (caused by the tongue, soft palate, teeth, or lips), generating a sound source resulting from air turbulence. Some vocal sounds require both phonation and turbulence as sound sources and are referred to as *mixed* sounds [86].

In all three cases, the *source* (phonation, turbulence, or both) is modified by the shape of the vocal tract (throat, mouth, nose, tongue, teeth, and lips). Each shape creates a different acoustic *filter*, further coloring the overall sound. This description of the human voice is the basis of the *source-filter* model (also known as the *excitation-resonance* model), which is the foundation of the majority of voice research including the analysis framework presented in Chapter 3.

All vocal sounds can be categorized by source type: voiced, unvoiced, or mixed. All of the vowels (e.g. [a], [e], [i], [o], [u]) and some consonants are voiced. The voiced consonants in which no turbulence is required ([l], [m], [n,], [r]) are called semivowels, the difference from vowels being that the vocal tract is altered to engage nasal resonance. Phonemes involving air turbulence in the mouth, whether voiced or unvoiced, are known as fricatives ([f] and [s] are unvoiced, whereas [v] and [z] are mixed). Another class of consonants, known as plosives or stops, arise from a sudden explosion of air released after being stopped in the vocal tract, some of which are voiced (e.g. [b], [d], [g]) while others are unvoiced ([p], [t], [k]).

2.2 Speech vs. Singing

Historically, speech and singing research have been closely linked, but there are important differences between the two methods of vocal production. The vast majority of sounds generated during singing are voiced (approximately 90 percent) whereas speech contains a much larger percentage of unvoiced sounds (about 60% voiced and 40% unvoiced for English) [15]. In the most common classical singing technique, known as *bel canto*, singers are taught to sustain vowels as long as possible between other phonemes because they are the most efficient and audible sounds. As a result, singers also learn to develop a high degree of consistency in their pronunciation of vowels, which can make it easier to automatically determine the vowel from analysis of the signal. Classical singers usually employ a technique in which they lower the larynx, creating an additional high-frequency resonance (around 4-5 kHz) not present in other types of vocal production. This resonance, known as the *singer's formant* is especially important for being heard in the presence of other instruments, for example allowing an opera singer to be heard over an entire orchestra [86].

Because of their periodic nature, voiced sounds are often easier to analyze and generate using linear signal processing theory. Even when limiting consideration to voiced sounds, there are important differences between speech and singing. In Western music, the range of fundamental frequencies used in singing is far greater than in speech. Likewise, the singing voice tends to have a much wider dynamic range in terms of amplitude than the speaking voice. Another distinction is that singing in the Western classical tradition is most often an interpretation of a predefined musical score whereas speech is most often spontaneous. The analysis/synthesis framework presented in this dissertation is specific to the classically-trained singing voice and takes advantage of these features that discriminate classical singing from speaking.

One type of speech that has more in common with classical singing is acting. Just as singing is usually performed from a pre-defined score, actor voices are generally the result of a performance from a pre-defined script. Stage actors are trained to project their voices to be heard throughout large auditoriums in a manner similar to opera singers by altering resonance in certain frequency ranges. And like singing, the goal of the voice in acting is often more than just the communication of words, but also the communication of emotion and intent requiring a similar standard of sound quality. Because of these similarities, some of the research presented in this dissertation may also be applicable to actors voices.

2.3 Early voice simulations

Machines capable of mechanically simulating a wide range of speech-like sounds have existed as early as the late 18th Century. A machine built by Wolfgang von Kempelen [20][92], when played by a skilled human operator, could produce intelligible speech. The first electrical machine capable of producing speech sounds was built in 1922 by J. Stewart of the research and development department of AT&T [82]. This device was limited to only a few vowels and consonants. Groundbreaking work in electrical voice

transmission and synthesis was published a decade or so later by Homer Dudley of Bell Laboratories with colleagues R. Riesz and S. Watkins [18]. His devices, the Vocoder and Voder, and their principles of operation formed the foundation of voice analysis and synthesis for many years to come, and even modern speech coding techniques owe a great deal to Dudley's work.

2.3.1 The Vocoder and Voder

The Vocoder was the first analysis/synthesis engine for speech transmission and is based upon a source-filter model of the human voice [18]. An input voice signal is decomposed using a bank of bandpass filters and a pitch detector. The pitch detector is used to control the fundamental frequency of the excitation and to determine whether the source signal is voiced or unvoiced. Together, the bandpass filters provide an approximation of the overall vocal tract filter, and the energy in each bandpass filter is transmitted as a parameter. Because this varies at a much slower rate than the speech itself (10s of Hz as opposed to 100s or 1000s of Hz), the bandwidth required to transmit the speech is significantly reduced [19].

The Voder is essentially the Vocoder with the analysis engine replaced by controls for a human operator [21]. The operator, in controlling the excitation type (voiced vs. unvoiced), the fundamental frequency of excitation, and the resonant bandpass filter responses, is able to synthesize speech or singing. The Voder was demonstrated at the 1939 World's Fair, where a team of specially trained operators was able to create speech and even singing using a simplified keyboard control (similar to a stenograph) to control the filters (and thus the phoneme) as well as a foot pedal to adjust the pitch of the voiced excitations.

Both the Voder and Vocoder employ the same source-filter model, demonstrating how it is applicable to both coding and synthesis of the voice. In fact, Dudley realized that coding and synthesis are essentially the same problem abstracted at different levels. Both are essentially enabling technologies for linguistic communication. In the case of coding, the desire is to preserve as much of the quality of the original sound source (the human speaker) as possible, while in synthesis the goal is to use the symbolic representation of language to create an instantiation that conveys the language accurately. From this perspective, synthesis becomes a (highly abstracted) form of coding. But if the synthesis technique were good enough to accurately render the voice of a particular speaker, the result could be indistinguishable from the real thing. Dudley realized this, and saw that the source-filter model could be used to explore both voice coding and voice synthesis. The original goal of the Vocoder was not only reduced-bandwidth transmission of speech, but also the analysis and synthesis of speech for research investigations, such as the intelligibility and emotional content of voice communication. These insights are directly related to the modern framework of Structured Audio, which is presented in Section 2.7.

2.4 Advancements in Voice Analysis/Synthesis

The following section details advancements on Dudley's work, which has led to the development of today's modern voice coding systems.

2.4.1 Improvements to the Vocoder

Improvements to Dudley's Vocoder, also known as the channel vocoder, continued to be made through the 20th Century. Much of this was spurred by secure communications research conducted during World War II. Advances were made that increased sound quality and reduced the transmission bandwidth required, in addition to digitizing the vocoder for more robust and secure communication [26]. Because the principles of the channel vocoder are not limited to only speech signals, recent computational implementations such as the phase vocoder (based on a Discrete Fourier Transform decomposition of the input signal) have found widespread use in general sound analysis/synthesis, computer music compositions, and sound effects processing [55].

2.4.2 Formant Vocoder

An interesting branch of vocoder research investigated the analysis of formants, or resonances of the voice. The formant vocoder attempted to ascertain the individual formant locations, amplitudes, and bandwidths as well as the excitation source type (harmonic or noise-like) [66]. This resulted in a representation of the voice in a very compact and efficient parameter set. Difficulties were encountered, however, in the extraction of the formant parameters, and the sound-quality of the reconstructed speech was fairly low. Eventually the formant vocoder for speech coding was eclipsed by other coding techniques that provided better sound quality at comparable bitrates. Formant-based techniques have also been explored extensively for parametric voice synthesis, again emphasizing of the close relationship of models for coding and synthesis. In particular, formant-based synthesis of the singing voice has achieved some success, which will be described in further detail in the description of synthesis technique in Section 2.5.1.

2.4.3 Homomorphic Vocoder

The homomorphic vocoder is based on an analysis by Alan Oppenheim in 1966 [58] and was implemented by Tom Stockham and Neil Miller for the separation of voice from orchestra and subsequent sound restoration in recordings of Enrico Caruso [54]. It is an extension of the principles of the channel vocoder using homomorphic transformations to the cepstral (inverse log-Fourier) domain. In this domain, what was multiplication in the frequency domain (and thus convolution in the time domain) becomes a simple addition through the nonlinearity of logarithms. The original voice signal (a convolution of the vocal excitation function and the vocal tract response function—assuming a linear model) is represented as an addition of excitation and source in the cepstral domain and estimation of the separated functions becomes easier. This sepa-

ration is used for more accurate pitch tracking and spectral estimation and can result in very high-quality reconstruction.

2.4.4 Linear Predictive Coding

Perhaps the most influential advance in the history of speech coding after Dudley's work was the development of *Linear Pedictive Coding* (LPC) of speech, first proposed by Atal in 1970. LPC is an analysis/synthesis technique which uses the past samples of a voice signal to adaptively predict future samples using linear least squares estimation [46]. The LPC decomposition can be shown to be equivalent to a source-filter model, where the vocal tract response is modeled using a time-varying all-pole filter function. The LP parameters are calculated using the well-known autocorrelation or covariance methods [66].

When coupled with an appropriate excitation source model (such as an impulse train or noise for voiced and unvoiced segments, respectively), this technique can result in low-bitrate transmission of speech. Vocal tract filter estimation via LPC also has the benefit of being a closed-form computation, requiring no heuristics for the determination of parameters. While computationally intensive, the exponential growth of available computing power has allowed LPC to become the basis of most speech codecs in use today. LPC also has been used on non-voice signals for audio transformations in musical compositions [36].

In the original LPC vocoder implementation, modeling of the excitation source requires a decision on the type of excitation (voiced or unvoiced) and a pitch estimation of voiced segments. Errors in these decisions usually lead to lower sound quality. Modeling of the excitation source as a periodic impulse train for voiced segments also results in reconstructed speech that sounds overly buzzy. This degradation in sound quality led to the development of the CELP algorithm, described in the next section.

2.4.5 Code-Excited Linear Prediction

Code-excited linear prediction (CELP) is a combination of traditional Linear Predictive (LP) modeling of the resonant qualities of the vocal tract coupled with complex excitation modeling. The CELP codec [78] has proven to be quite successful at transmitting toll-quality speech at low-bitrates (down to 4 kbits/sec) and intelligible speech at even lower bitrates. Its use is widespread in today's digital communications devices, such as cellular phones. The general method used in the selection of the excitation is a closed-loop analysis by synthesis, consisting of a search through a codebook of excitation vectors. The residual error from the LP analysis is compared against the entries of the codebook, filtered by the LP parameters. The entry corresponding to the resulting waveform that best matches (determined via correlation) the residual is chosen, along with a corresponding gain. Since the transmitter and receiver share a copy of the codebook, the codebook entry number and gain are the only values that need to be transmitted for the excitation. Codebooks vary according to implementation, but generally contain on the order of hundreds of entries. Variations also exist using multiple codebooks, particularly to model voiced and unvoiced excitations. In this case, how-

ever, no voiced/unvoiced decision of the input signal frames is required. The excitation will be composed of some ratio of the deterministic and stochastic vectors, which is established by the gain parameters calculated.

Further gains in compression have been achieved using *Vector Quantization* (VQ), in which excitation vectors are clustered and quantized so that a single excitation vector is chosen to represent an entire cluster. This attempts to capture the wide variance in excitation vectors while limiting overall system complexity by reducing codebook sizes [47].

2.4.6 Sinusoidal voice coding

Unlike other voice coding techniques, sinusoidal analysis/synthesis for voice coding is not based on a source-filter model. The technique was first demonstrated by McCauley and Quatieri [51]. An input signal is decomposed into a number of sinusoidal partials. For voiced sound segments the partials will be largely harmonic, while for unvoiced segments they will be inharmonic. Careful peak selection, tracking, and phase matching across analysis frames results in high quality transmission at reduced bitrates. The sinusoidal decomposition also lends itself well to modifications, such as time stretching and compression, and has been demonstrated to be very effective on speech and singing voice recordings. Like the channel vocoder, sinusoidal analysis/synthesis makes no assumptions about the input signal and can be applied to arbitrary sound sources and has been put to good use for the synthesis of musical instruments [74] and the coding of general audio [63]. A deficiency of the sinusoidal representation, however, is that it is difficult to relate the sinusoidal parameters to the physical parameters of voice production. High-quality re-synthesis also requires a large number of sinusoids, which requires more parameters and greater bandwidth. The sinusoidal representation has also been used in systems for speech and singing synthesis (Section 2.5.4).

2.4.7 High-resolution time-frequency voice analysis/synthesis

Mellody and Wakefield [52] have proposed a technique for singing voice signal analysis/synthesis using a high-resolution time-frequency distribution, which they call the Modal distribution. By maintaining high-resolution in both time and frequency, they were able to create precise parameterizations of the individual harmonic partials of the voice. These sinusoidal parameters were extracted from recordings of female classically-trained conservatory students and used in experimentally for singer identification and cross-synthesis. The sinusoidal frequencies and magnitudes were used to establish an excitation signal and a frequency magnitude envelope. The identification was performed by long-term averaging the sinusoidal parameters to establish a common magnitude response for each singer. Residuals from the common response were clustered to form a basis for discrimination. Cross-synthesis was achieved by combining the estimated excitation sources from one singer with the magnitude response of another. Because of the high-resolution of the sinusoidal analysis, the sound quality remains high during re-synthesis. Since this approach is essentially a sinusoidal decomposition, however, it similarly lacks a meaningful parameterization in terms of physical process of singing. Separation of component frequencies from magnitudes does not accurately

depict the separate contributions of the vocal folds and the vocal tract and leaves little intuition regarding the components of vocal quality.

2.4.8 PSOLA and WSOLA

Pitch-synchronous overlap-add (PSOLA) [10] and *waveform similarity overlap-add* (WSOLA) [91] are techniques that are commonly used in commercial products for voice processing and modification. Both techniques operate by identifying and manipulating time-domain regions of similarity (by estimating pitch or comparing waveforms). Deletion or replication of segments results in time compression or expansion, respectively. By overlapping and summing the windowed regions, these effects can be achieved while maintaining high sound quality. Pitch manipulation is also possible by altering sampling rates or changing the amount of overlap between windows. The use of devices based on PSOLA/WSOLA is almost ubiquitous in popular music recording, and they are also commonly used to provide pitch correction for live performances as well as karaoke machines.

2.5 Synthesis of the Singing Voice

Following the lead of Dudley with the Vocoder and Voder, systems for direct simulation of the voice have been developed using many of the same models as analysis/synthesis coding systems. Although they have much in common, coding and synthesis are treated today as mostly separate problems. The primary reason for this is that in addition to requiring a model to accurately reproduce the acoustics of the voice, direct synthesis involves translating symbolic input (words, notes, rhythms, etc.) into model control parameters. Voice coding systems have no such translation requirement. As a result, many direct synthesis systems require a great deal of hand tweaking to make realistic sounding vocal lines.

2.5.1 Formant-based synthesis

One of the earliest attempts at a synthesizer designed specifically for singing voice is the Music and Singing Synthesis Equipment (MUSSE) developed at the Royal Institute of Technology (KTH) in Stockholm [37]. Using a source-filter model, the system sends a simulated glottal pulse through a series of resonant filters to model the formants of the vocal tract for a bass/baritone singer. Johan Sundberg and his group at KTH have determined general formant frequencies for all of the vowels and consonants. The most recent version of this synthesizer is MUSSE DIG, a digital implementation of the same structure that includes improvements to the glottal pulse model [6]. Direct synthesis is achieved by using a rule-based system, also developed by Sundberg at KTH, to control the parameters of the synthesizer [87]. The synthesizer can be driven using a live controller.

2.5.2 Formant Wave Functions

Formant wave functions (FOFs, from the French) are time-domain functions with a particular resonance characteristic used to model individual formants [68]. The use of FOFs for singing voice synthesis is implemented in the CHANT system, developed at the Institut de Recherche et Coordination Acoustique/Musique (IRCAM) by Xavier Rodet [69]. It uses five formant wave functions to generate an approximation of the resonance spectrum of the first five formants of a female singer. Each function is repeated at the fundamental period of voicing. The results of the original system for vowel sounds are impressive, but involve a great deal of hand adjustment of parameters. Later work extended the use of FOFs to consonants and unvoiced phonemes [67]. The CHANT system was primarily designed to be a tool for composers.

2.5.3 Waveguide vocal tract physical modeling

Kelly and Lochbaum implemented the first digital physical model of the vocal tract in 1962 [31]. The vocal tract was modeled as series of cylindrical tube sections represented by a digital ladder filter. This model of sound propagation has come to be known as a waveguide digital filter and has been used as the basis for a variety of musical instrument models (e.g. [76]). The model was refined by Liljencrants in 1985 to add a more realistic glottal excitation source [38], which is detailed in Section 3.3.

The Singing Physical Articulatory Synthesis Model (SPASM) was created by Perry Cook at the Center for Computer Research in Music and Acoustics (CCRMA) at Stanford University in 1990 [15]. SPASM uses a digital waveguide to model the physical acoustics of the vocal tract as well as the nasal cavity and throat radiation, and the system is driven by a frequency domain excitation model. An integrated graphical composition system is provided to control the synthesis parameters. In 1994 Välimäki and Karjalainen extended the vocal tract physical model by using variable length conical sections [89]. In general, however, physical modeling is difficult to use in an analysis/synthesis framework as it is difficult to extract the actual physical parameters from an audio signal. Physical modeling is also computationally intensive, though this is becoming less and less of an issue.

2.5.4 Sinusoid-based synthesis

Another approach used in direct synthesis of the singing voice uses sinusoidal models of phonemes, concatenated together to create words. One example is the LYRICOS project, originated by researchers at the Georgia Institute of Technology and Texas Instruments [45]. Its goal is direct synthesis of the singing voice from an appropriately annotated MIDI (digital music score) file. There are robust compositional tools for LYRICOS, but again a great deal of hand-editing is needed to model important features such as vibrato, jitter, and dynamics. Another research project, led by Ken Lomax at Oxford University, trained neural networks on input data from famous singers to control the necessary synthesis parameters [40]. This system was limited to mostly vowels, with a few voiced consonants. Recently, Yoram Meron at the University of Tokyo used Hidden Markov Models (HMMs) to control a mapping from sinusoidal

phoneme models of an individual singer to a target sound, which could be from the same singer or a different singer [53]. This enables a voice transformation in which one singer's voice model (stored phonemes) is driven by another (target sound). Researchers at the Music Technology Group of the Pompeu Fabra University have also been exploring the use of sinusoidal models for singing synthesis and voice morphing in karaoke applications [8], which has been incorporated into a commercial product, the Yamaha Vocaloid [94].

2.6 Related work on Talker ID and Instrument ID

A significant amount of research has been performed on speaker (talker) identification from digitized speech for applications such as verification of identity, and this research may inform approaches to identification of individual singers. For the most part, talker identification systems use features similar to those used in speech recognition (MFCCs—Mel-frequency cepstral coefficients). Many of these systems are trained on pristine data (without background noise) and performance tends to degrade in noisy environments. Since they are trained on spoken data, they perform poorly with singing voice input [48]. Additionally, mel-scale cepstra characterize gross features of the spectrum, which tends to make them more useful for tasks requiring generalization such as speech recognition. Much of the individual distinctiveness in singing, however, is characterized by finer spectral features that are not present in MFCCs.

Also relevant to the task of singer identification is work in musical instrument identification. Our ability to distinguish different voices (even when singing or speaking the same phrase) is akin to our ability to distinguish different instruments (even when playing the same notes). Thus, it is likely that many of the features used in automatic instrument identification systems will be useful for singer identification as well. Work by Martin [49] on solo instrument identification demonstrates the importance of both spectral and temporal features and highlights the difficulty in building machine listening systems that generalize beyond a limited set of training conditions.

In the realm of music information retrieval (MIR) there is a burgeoning amount of interest and work on automatic song, artist, and singer identification from acoustic data. Such systems would obviously be useful for anyone attempting to ascertain the title or performing artist of a new piece of music and could also aid preference-based searches for music. Most of these systems utilize frequency domain features extracted from recordings, which are then used to train a classifier built using one of many machine learning techniques. The features, classification methods, performance, and scope of these MIR systems are relevant and informative to the problem of singer identity and provide some guidelines for expectations of performance of automatic classification systems. At one extreme, robust song identification from acoustic parameters has proven to be very successful (with accuracy greater than 99% in some cases) in identifying songs within very large databases (>100,000 songs) [30], demonstrating that very high performance is attainable using acoustic features in the context of certain limited identification tasks.

Artist identification, however, is a more difficult task than individual song identification, but is closely related to the issue of singer identity. A recent example of an artist identification system is [93], which reports accuracies of approximately 50% in artist identification on a set of popular music albums consisting using MFCC features and a support vector machine (SVM) classifier. The album database, called the Minnowmatch testbed, consisted of 17 different artists and about 250 songs. In many cases, artist identification and singer identification amount to the same thing. To that end, Berenzweig and Ellis [5] implemented a singing detection system to be used as a preprocessing step for artist identification using MFCC features and a hidden Markov model classifier. The classifier achieved a success rate of ~80% in isolating vocal regions within a database of 100 short (~15 second) music segments recorded from FM radio broadcasts. In [4], Berenzweig, Ellis, and Lawrence found that by pre-segmenting the input to focus on voice regions alone, they were able to improve artist identification from ~50% to ~65% on the Minnowmatch testbed using MFCC features and a multilayer perceptron neural network classifier. A singer identification system using LPC features and a SVM classifier on voice-only segments on the same database performed with 45% accuracy [33], demonstrating the utility of voice-coding features for identification, but also betraying a lack of robustness in sound mixtures (due to background instruments) when compared to more general spectral features such as MFCCs. The results of these studies have provided guidance towards appropriate feature selection for the representations singer identity at the center of this dissertation.

2.7 Structured Audio

The term structured audio was suggested by Vercoe *et al* to formalize a broad range of research on the creation, transmission, and rendering of sound representations (i.e. model-based audio) [90]. Representations can range anywhere from encoding schemes for general audio (such as Pulse Code Modulation or encoders based on psychoacoustic masking, e.g. mp3) to low-dimensional musical instrument models (as used in sound synthesizers). Lower dimensional models are obviously more efficient for encoding and transmission of sound. A representation with a small number of properly chosen dimensions (e.g. musically meaningful control parameters such as pitch and volume) lends itself to greater flexibility in sound rendering and is said to be highly structured.

For example, a piano performance might be recorded, encoded, and transmitted, preserving the sound quality to the point where it is almost indistinguishable from the live performance. This representation (waveform encoding), however, contains a low amount of structure since musically salient parameters of the performance are not accessible. Although the reproduction will be accurate, a large amount of bandwidth will generally be required for transmission. Alternatively, the performance could be captured as note and key velocity data from a (properly outfitted) piano keyboard. Transmission of this compact symbolic data requires little bandwidth. The performance could then be re-created using a synthesized piano sound or even a modern player piano. If the synthesizer is of high-quality or if a real piano is used, this too can result in a high-quality performance, nearly indistinguishable from the original. But in this case, the performance could also be rendered with a different sounding piano

or at a different tempo. This representation, which preserves the symbolic control parameters and allows for greater flexibility in reconstruction, is highly structured. Much of the research in structured audio focuses on the creation of highly structured low-dimensional models with musically meaningful control parameters and high sound quality.

Highly structured representations not only hold the promise of greatly reduced bandwidth requirements for the transmission of audio, but ideally treat music not as a static object as in a recording, but as an interpretation or performance of a musical composition. A score is a highly structured representation of a work. It is meant to be interpreted and performed by artists willing to add their own expressivity to the music. This is why musically meaningful control parameters are so desirable. For maximum expressive capacity, control parameters must conform to the vocabulary of musical expression.

Structured audio acknowledges a continuum of representations with varying amounts of structure. Certain representations tend to focus more on accurate reconstruction of sound (coding) while others focus on expressive control parameters (synthesis). Other models contain aspects of both. All are ways of communicating musical meaning and even more basic than that, ways of transmitting information. While the domains of coding and synthesis are not always thought of as being closely related, the structured audio framework demonstrates that they are. This continuum is especially relevant to research in singing voice, which will be addressed in Section 2.7.2.

The concepts of structured audio can be implemented in numerous ways and to varying degrees. Several highly-flexible implementations exist, such as NetSound [9] (based on the Csound language [7]) and MPEG-4 Structured Audio [72], which implements a new language for algorithmic sound description called Structured Audio Orchestra Language (SAOL) as well as a language for score description, Structured Audio Score Language (SASL). These implementations use a formalized language to describe audio in terms of sound sources (represented as variables) and manipulations (signal processing algorithms).

2.7.1 Structured Scene Description

The structured audio concept applies not only to individual sound sources, but to mixtures as well. For example, the MPEG-4 Audio standard allows for scene description, which provides structure at a higher level by allowing the separation of encoding for different sound sources [72]. With a priori knowledge of the source type, specific encoding schemes can be used for maximum compression. For example, speech tracks can be transmitted using a speech codec, such as CELP; acoustic instruments can be encoded using one of the natural audio codecs, such as HILN (harmonic lines plus noise—a sinusoidal coding scheme); and synthetic sounds can be represented using MPEG-4 Structured Audio. It is possible for some complex sound mixtures to be encoded at high-quality at a greatly reduced bitrate. In this context, the applications of a low-dimensional representation specifically for singing voice (transmission or synthesis) become obvious (Figure 2-5).

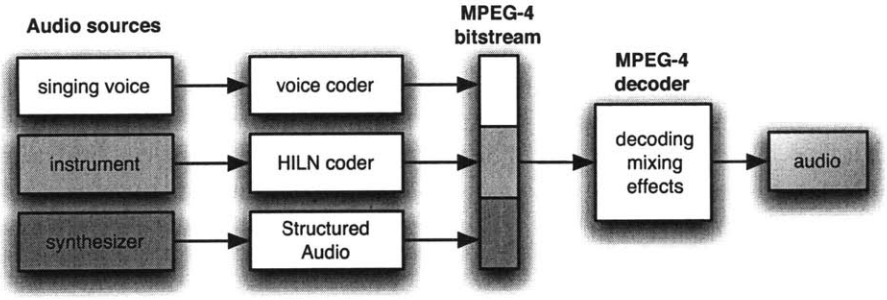


Figure 2-5: Multiple source encoding using MPEG-4 scene description

Descriptive structured audio languages, such as SAOL and Csound, can also be used to implement arbitrary audio coding techniques, including perceptual transform coding (e.g. mp3) and CELP. This technique is called generalized audio coding [73], and can lead to hybrid coding techniques combining aspects of traditional audio coders with the flexibility of synthesis. An example of this kind of hybrid coder, an extension of LPC, is presented in Appendix B. Generalized audio coding also facilitates the rapid development and deployment of new codecs, so that a new codec (e.g. for an specific individual's singing voice) could be downloaded along with a piece of content encoded using that codec, removing dependence on fixed hardware implementations of sound encoding and decoding.

2.7.2 Singing in the Structured Audio Context

Structured audio implementations seek low-dimensional parametric models with high sound quality, and many such models exist for the simulation of musical instrument sounds. As of yet, however, there is no such low-dimensional parametric model with high sound quality for the singing voice. An ideal structured singing voice model would be able to use high-level knowledge about the music itself (the score, lyrics, etc.) for a very compact representation. It would also be desirable for the parameters in such a model to represent intuitively meaningful qualities of the voice and to be easily modified, opening new opportunities for artistic expression. Such a model is the ultimate goal of this research.

To some degree all audio models share what is known as the *encoding problem*, or the estimation of model parameters from an acoustic signal. The difficulty of the parameter estimation varies greatly depending on the representation being used. Reconstruction of an input voice using concatenative synthesis requires the estimation of parameters which are highly meaningful (words, phonemes, durations, and pitch), but are difficult to estimate reliably from an acoustic source. Likewise, the estimation of physical characteristics of the vocal tract (e.g. vocal tract dimensions, shape, and material properties) solely from an acoustic input is quite difficult, though these physical model parameters have a great deal of intuitive meaning. On the other hand, the estimation

of digital filter coefficients using LPC is fairly straightforward, but filter parameters are difficult to directly relate to perceptual features.

The amount of structure of a model is a purely qualitative label and varies subjectively depending on the context of the application. In general, the difficulty of the estimation of a model's parameters appears to be proportional to the model's level of structure (meaningful parameters are more difficult to extract, but greater structure requires meaningful parameters). It is also apparent that different applications are better suited to different models with accordingly different degrees of structure. Models used for direct synthesis require musically meaningful parameters and therefore possess a high degree of structure. Representations best suited for coding (analysis/synthesis) require robust methods of parameter estimation, limiting the complexity of the estimation and most likely the intuitive meaning of the parameters as well. As a result, models used primarily for coding tend to be less structured.

As mentioned in Section 2.3.1, the relationship between coding and synthesis in voice research was made as early as the 1930s by Homer Dudley when he realized his source-filter model could be used for both. While neither the Vocoder nor the Voder retained a particularly high amount of structure (the parameterization beyond the separation of excitation and resonance is not particularly meaningful), the relationship between coding and synthesis was made. The various representations for singing voice coding and synthesis described above fall in different places along the continuum of model structure, as shown in Figure 2-6. The figure illustrates the degree to which the parameters of each model can be related to a high-level construct, such as text or a physical or perceptual feature, which is also proportional to the difficulty of estimating those features.

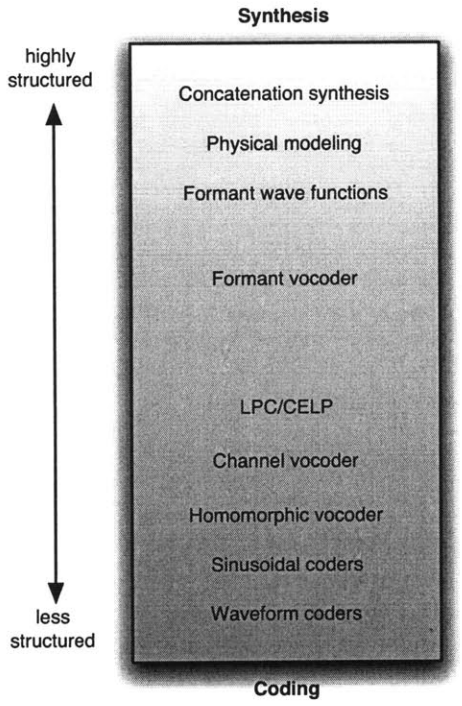
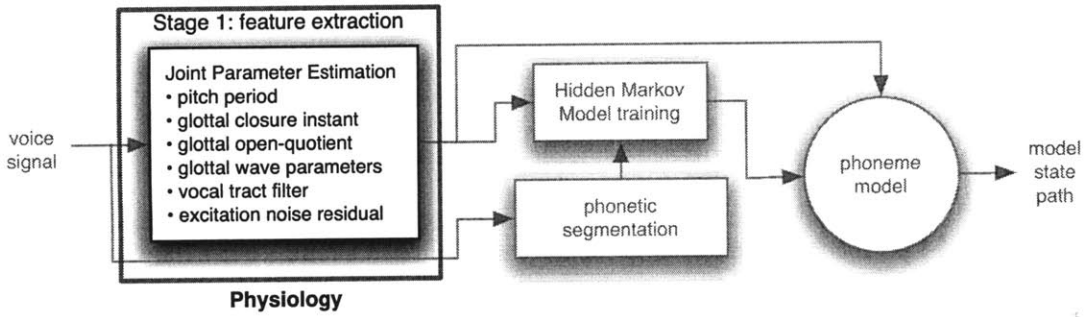


Figure 2-6: The level of structure in various techniques for voice coding and synthesis

CHAPTER THREE

Source-Filter Parameterization



This chapter details the estimation of physically motivated singing voice parameters from acoustic recordings based upon a source-filter model. The source-filter representation is a simple mathematical decomposition reflecting the mechanism of vocal production. The analysis occurs in several stages (Figure 3-1): an initial estimation of several candidate parameter sets, a search for the minimum error estimate, a refinement of the source model parameters, and source residual modeling. The parameter estimation algorithms require a *pitch-synchronous* analysis (time-aligned to each pitch period as defined by the instants of glottal closure). The parameters estimated from the system described in this chapter form the set of observations used in the dynamic modeling described in the following chapter.

3.1 Source-Filter Parameter Estimation

As noted in Chapter 2, the source-filter model is used as the basis for a wide variety of voice analysis/synthesis algorithms. The source-filter model is compelling because it reflects the physical mechanism of vocal production. One of the underlying hypotheses of this dissertation is that vocal identity is based in large part upon features related to vocal fold and vocal tract physiology. Therefore, the goal in using the source-filter model here is to extract parameters that are as closely tied to the physical features as

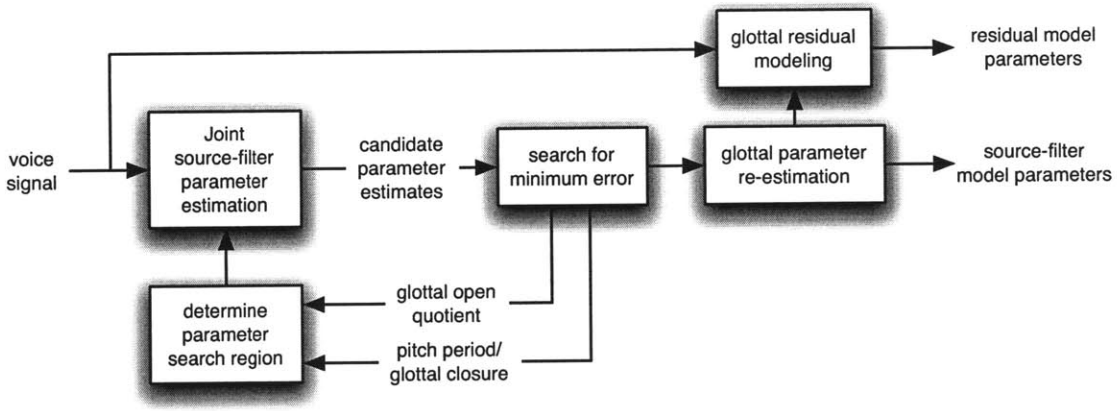


Figure 3-1: Flow diagram of pitch synchronous parameter estimation

possible. The source is represented by glottal airflow, which is directly related to physical qualities of the vocal folds, such as length and stiffness. Likewise, the filter in the model is directly correlated to the physical shape of the vocal tract.

Like most prior research the analysis framework presented here uses a *linear* source-filter model, meaning that the following two properties of the system must hold true: (1) scaling of the source results in a similar scaling of the system output, and (2) the output of the system from additive sources is simply the sum of the outputs resulting from the individual source components. These two properties are known as *scaling* and *superposition*, respectively. As a linear system, the interaction between the source and filter models is entirely additive or multiplicative, which allows for the re-ordering of system components without altering system output behavior.

While there is certainly evidence of non-linear source-tract interaction [2], it is difficult to incorporate such nonlinearities in a parametric computational model. Therefore, the assumption of linearity is a convenient, if not entirely accurate, assumption that also has the benefit of allowing the reordering of model components into a more computationally efficient configuration. This will be taken advantage of in several steps of the joint parameter estimation procedure detailed in this section. The effects arising from nonlinearities that are not accounted for in the linear source-filter model are combined into an additive source that is modeled stochastically (Section 3.4).

Historically, the estimation of source and filter parameters has been performed independently in order to simplify the complexity of the analysis. While this is a convenient approximation, the vocal fold oscillation and vocal tract shape are actually somewhat dependent. For example, singers frequently modify their vowels (vocal tract shape) in order to more easily sing a high pitch [86]. More importantly, the dependencies themselves may be distinctive features of an individual voice. In the framework presented here the model parameters are estimated jointly (simultaneously) from the acoustic data to explicitly account for these dependencies.

Rather than modeling the glottal flow directly, it is convenient to model the derivative of the glottal flow, which we label $g[n]$. The glottal derivative waveform is the result of combining the effect of lip radiation (a differentiation) with the glottal flow, but it also allows the use of a lower-order model. Different waveform periods of course result in varying pitches, and different modes of source articulation (e.g. breathy or pressed) will lead to different waveform shapes. The physical shape of the vocal tract is represented by a filter with impulse response $h[n]$. The sung output $s[n]$ is the *convolution* of the source and filter functions, defined as the sum of the source signal multiplied by shifted copies of the filter impulse response.

$$s[n] = g[n] * h[n] \triangleq \sum_{m=-\infty}^{\infty} g[m]h[n-m] \tag{3.1}$$

Different vocal tract shapes (from movement of the jaw, tongue, lips, etc.) are represented by changing the filter $h[n]$ resulting in different output sounds.

Implementations of the general source-filter representation use assumed models for both the source waveform and the vocal tract filter for simplicity and analytical tractability. By fitting the acoustic data to these assumed models parametrically, we can derive a relatively small and fixed number of features that can be used for establishing singer identity. In this analysis framework, the KLGLOTT88 model [34] is used initially to represent the glottal derivative source, and a fixed-order filter (derived via linear prediction) to model the vocal tract filter. These models lend themselves to a particularly efficient solution for joint parameter estimation, via convex optimization. Using the jointly-derived filter estimates, the excitation is then re-parameterized using the more complex Liljencrants-Fant glottal derivative model, which more accurately reflects the waveshape of the glottal derivative. Effects not represented by these models (such as turbulence) result in a residual noise signal, which is modeled separately. The joint source-filter parameter estimation via convex optimization was first proposed by Lu and Smith in [43], and the overall procedure was refined by Lu in [42].

The analysis system presented here differs from [42] in the following ways: 1) Joint parameter estimation is performed on a warped frequency scale, to more accurately model the frequency sensitivity of human perception, 2) Glottal closure instants are not calculated *a priori*, but are optimized from the data given the assumed models for source and filter, and 3) The residual noise is modeled using a stochastic codebook, individually trained for each singer. These extensions are described in greater detail in the sections that follow.

3.1.1 KLGLOTT88 model

The KLGLOTT88 model [34] is a relatively simple model for describing the derivative glottal wave. We choose to model the derivative glottal wave (rather than the glottal waveform itself) for two reasons: (1) to retain the simplicity of the model (a 2nd-order polynomial) and (2) to efficiently encapsulate the effects of lip radiation; Instead of

differentiating the output, we equivalently apply the differentiation to the glottal wave in accordance with linear systems theory.

The KLGLOTT88 model $\hat{g}[n]$ for the derivative glottal wave is based on a two-piece polynomial representation proposed by Rosenberg [70] which is specified by the following equation:

$$\hat{g}[n] = \begin{cases} 0, & 0 \leq n < n_c \\ 2a(n - n_c) - 3b(n - n_c)^2, & n_c \leq n < T \end{cases} \tag{3.2}$$

In the original KLGLOTT88 representation, this function is then filtered by a first-order IIR low-pass filter for additional control over the spectral tilt of the source waveform. Because of the linearity of the source-filter model, this spectral tilt filter can be separated from the source model and incorporated into the all-pole vocal tract model by simply adding an additional pole (described in Section 3.1.3).

T corresponds to the pitch period (in samples) and n_c represents the duration of the *closed phase* of the glottal glottal cycle, which is best expressed in terms of the open-quotient OQ, the fraction of the period for which the glottis is open:

$$n_c = T - OQ \cdot T \tag{3.3}$$

To maintain an appropriate waveshape, the parameters a and b are always positive values and are further related as follows:

$$a = b \cdot OQ \cdot T \tag{3.4}$$

The model has only two free parameters, a shape parameter a and the open-quotient OQ. Because of its relative simplicity, this model lends itself well to joint parameter estimation with an all-pole vocal tract model, as will be discussed below. An example plot of two glottal cycles as defined by the KLGLOTT88 model is shown in Figure (3-2). The variations in the second period demonstrate the result of increasing the shaping parameter a.

3.1.2 Linear prediction and frequency warping

Linear prediction (LP) estimates a signal $s[n]$, using a linear combination of its p past samples. Here the signal of interest, $s[n]$, is the recorded singing voice. If we assume linear predictability, we obtain the following difference equation relating the glottal source $g[n]$ and the voice output.

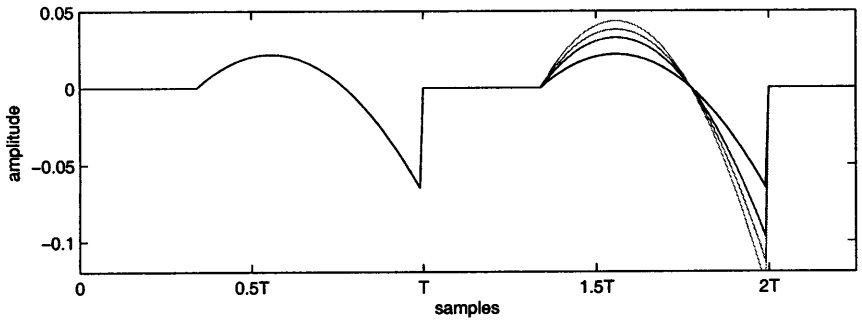


Figure 3-2: The KLGLOTT88 glottal flow derivative model

$$s[n] = \sum_{k=1}^{p} \alpha_k s[n-k] + g[n] \tag{3.5}$$

From Equation (3.5), we derive the transfer function, $H(z)$ relating the voice output to the glottal source in the frequency domain:

$$H(z) = \frac{S(z)}{G(z)} = \frac{1}{1 - \sum_{k=1}^{p} \alpha_k z^{-k}} = \frac{1}{A(z)} \tag{3.6}$$

$H(z)$ is the z-transform of $h[n]$, defined as the vocal tract filter, and equation (3.6) shows that the transfer function is always an all-pole filter. For convenience, we label the denominator polynomial separately as $A(z)$. The goal of LP is to determine the coefficients α_k that provide the best fit for the samples $s[n]$.

To determine the coefficient values the error is minimized between the actual signal $s[n]$ and the predicted values from the previous p samples. The two most common techniques for solving this minimization are the autocorrelation and covariance methods, named for their use of the autocorrelation and covariance matrices, respectively [46]. The analysis is usually restricted to a short window (frame) of samples with a new analysis performed for each advancing frame. The overall result is a time-varying all-pole filter function of order p representing the vocal tract filter for a given frame. The coefficients of $A(z)$ can be factored to determine the pole locations, which generally correspond to the vocal formants. A disadvantage of standard LP is that all frequencies are weighted equally on a linear scale. The frequency sensitivity of the human ear, however, is close to logarithmic. As a result, standard LP analysis sometimes places poles at higher frequencies where the ear is less sensitive and misses closely spaced resonant peaks at lower frequencies where the ear is more sensitive. Using a higher order LPC is one way of compensating for this, but increasing the number of poles increases the

feature dimensionality of each analysis frame, making it difficult to track correlations between frames.

One way of accommodating the nonlinear frequency spacing of human hearing is to warp the frequency domain accordingly for the subsequent signal analysis. The suggestion of a warped frequency is not at all new. Oppenheim, Johnson, and Steiglitz [57] suggested unequal frequency resolution analysis using the Fast Fourier Transform in 1971. *Warped linear prediction* (WLP) was proposed independently by Steiglitz [79] and Strube [85] in 1980 and subsequently used in the simulation of stringed instruments in 1981[36]. Much more work on the formalization of effectiveness and limitations of WLP has been performed recently by Härmä *et al* [29], [28].

We implement a Warped Linear Prediction model by replacing each standard delay with an all-pass filter of the form:

$$z^{-1} \rightarrow D(z) = \frac{z^{-1} - \lambda}{1 - \lambda z^{-1}} \tag{3.7}$$

This has the effect of warping the power spectrum of each frame [28] and can be made to approximate the frequency sensitivity of the ear. A frequency ω is transformed to a warped frequency $\hat{\omega}$ via the following relation:

$$\hat{\omega} = \omega + 2\tan^{-1}\left(\frac{\lambda \sin\omega}{1 - \lambda\cos\omega}\right) \tag{3.8}$$

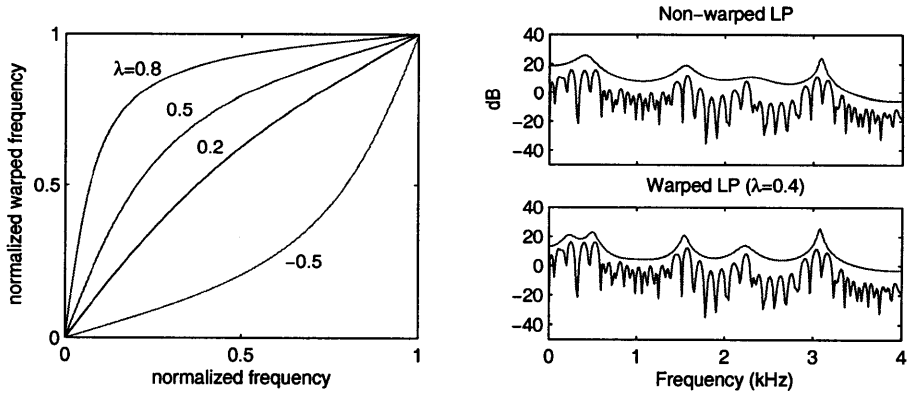


Figure 3-3: Varying levels of frequency warping (left) and non-warped and warped linear prediction (right). Note that the warped LP is able to resolve the closely-spaced formants at low frequencies.

3.1.3 Joint source-filter parameter estimation

The primary assumption of the source-filter model is that the vocal output is the result of a convolution between the glottal excitation $g[n]$ and the impulse response of the vocal tract filter $h[n]$ (Equation 3.1). Therefore, simultaneous estimation of the glottal derivative and vocal tract parameters requires de-convolution of the source and filter functions from the voice signal. This is accomplished by finding the model parameters that minimize the distance between the KLGLOTT88 source model, $\hat{g}[n]$ and linear prediction residual, $g[n]$, over one period. The linear prediction residual can be derived from Equation 3.6:

$$G(z) = \frac{S(z)}{H(z)} = S(z)A(z) = S(z)\left(1 - \sum_{k=1}^{p+1} \alpha_k z^{-k}\right) \tag{3.9}$$

An additional pole has been added in order to incorporate the low-pass filter spectral tilt filter that is part of the KLGLOTT88 source model. Equation (3.9) gives a relation between the glottal source and linear combinations of the output, but the unit delays imply a linear frequency scale. To take advantage of frequency warping, we must replace each delay with the allpass filter $D(z)$ of Equation (3.7).

$$G(z) = S(z)\left(1 - \sum_{k=1}^{p+1} \alpha_k D(z)^k\right) \tag{3.10}$$

If $\delta[n]$ is the impulse response of $D(z)$, then the impulse response of $D(z)^k$ is a generalized shift operator, $d_k\{\cdot\}$, defined as a k-fold convolution of $\delta[n]$ with the signal the operator is applied to [28].

$$\begin{aligned} d_1\{s[n]\} &\equiv \delta[n] * s[n] \\ d_2\{s[n]\} &\equiv \delta[n] * \delta[n] * s[n] \\ d_3\{s[n]\} &\equiv \delta[n] * \delta[n] * \delta[n] * s[n] \\ &\vdots \end{aligned} \tag{3.11}$$

Thus, in the time domain, we obtain the following relation between the glottal derivative and the recorded voice signal:

$$g[n] = s[n] - \sum_{k=1}^{p+1} \alpha_k d_k\{s[n]\} \tag{3.12}$$

We want to determine the parameter values that minimize the distance between the glottal derivative and our KLGLOTT88 model. Subtracting Equations (3.2) and (3.12) we obtain the following expression for the error $e[n]$:

$$\begin{aligned} e[n] &= \hat{g}[n] - g[n] \\ &= \begin{cases} 0 - s[n] + \sum_{k=1}^{p+1} \alpha_k d_k\{s[n]\}, & 0 \le n < n_c \\ 2a(n-n_c) - 3b(n-n_c)^2 - s[n] + \sum_{k=1}^{p+1} \alpha_k d_k\{s[n]\}, & n_c \le n < T \end{cases} \end{aligned} \tag{3.13}$$

Re-ordering of the terms results in:

$$e[n] = \begin{cases} \alpha_1 d_1\{s[n]\} + \cdots + \alpha_{p+1} d_{p+1}\{s[n]\} + 0 - s[n], & 0 \le n < n_c \\ \alpha_1 d_1\{s[n]\} + \cdots + \alpha_{p+1} d_{p+1}\{s[n]\} + 2a(n-n_c) - 3b(n-n_c)^2 - s[n], & n_c \le n < T \end{cases} \tag{3.14}$$

From Equation (3.14) it is apparent that it would be helpful to re-write the error equation in matrix notation. We wish to estimate the parameters α_k, a, and b, so let us define the parameter vector $\mathbf{x}$ as follows:

$$\mathbf{x} = \begin{bmatrix} \alpha_1 & \cdots & \alpha_{p+1} & a & b \end{bmatrix}^T \tag{3.15}$$

The vectors $\mathbf{f}_n$ are defined as the coefficients for the parameter variables.

$$\mathbf{f}_n^T = \begin{cases} \begin{bmatrix} d_1\{s[n]\} & \cdots & d_{p+1}\{s[n]\} & 0 & 0 \end{bmatrix}, & 0 \le n < n_c \\ \begin{bmatrix} d_1\{s[n]\} & \cdots & d_{p+1}\{s[n]\} & 2(n-n_c) & -3(n-n_c)^2 \end{bmatrix}, & n_c \le n < T \end{cases} \tag{3.16}$$

Collecting the vectors $\mathbf{f}_n$ over one period we obtain the following coefficient matrix:

$$\mathbf{F} = \begin{bmatrix} \mathbf{f}_0^T \\ \vdots \\ \vdots \\ \mathbf{f}_{n_c}^T \\ \vdots \\ \vdots \\ \mathbf{f}_{T-1}^T \end{bmatrix} = \begin{bmatrix} d_1\{s[0]\} & \cdots & d_{p+1}\{s[0]\} & 0 & 0 \\ \vdots & & \vdots & \vdots & \vdots \\ \vdots & & \vdots & 0 & 0 \\ d_1\{s[n_c]\} & \cdots & d_{p+1}\{s[n_c]\} & 2(0) & -3(0)^2 \\ \vdots & & \vdots & 2(1) & -3(1)^2 \\ \vdots & & \vdots & \vdots & \vdots \\ d_1\{s[T-1]\} & \cdots & d_{p+1}\{s[T-1]\} & 2(T-n_c-1) & -3(T-n_c-1)^2 \end{bmatrix} \tag{3.17}$$

For notational convenience, we define the error and voice signal over one period as vectors:

$$\mathbf{e} = \begin{bmatrix} e[0] \\ \vdots \\ e[T-1] \end{bmatrix} \qquad \mathbf{s} = \begin{bmatrix} s[n] \\ \vdots \\ s[T-1] \end{bmatrix} \tag{3.18}$$

By combining Equations (3.15), (3.17), and (3.18) we can now re-write Equation (3.14) in matrix notation:

$$\mathbf{e} = \mathbf{F}\mathbf{x} - \mathbf{s} \tag{3.19}$$

The next step is to solve for the parameter estimates $\mathbf{x}$ that minimize Equation (3.19). To do so, we must determine an appropriate minimization criteria. The L_2-norm (sum of squares) of the error vector is a standard optimization metric, and its usage results in successful parameter estimates for this problem.

$$\begin{aligned} \min_{\mathbf{x}} \|\mathbf{e}\|^2 &= \min_{\mathbf{x}} \sum_{n=0}^{T-1} (e[n])^2 \\ &= \min_{\mathbf{x}} \|\mathbf{F}\mathbf{x} - \mathbf{s}\|^2 \end{aligned} \tag{3.20}$$

Convex optimization

As demonstrated by Lu in [42], the relative simplicity of the KLGLOTT88 waveform guarantees that this minimization is a *convex optimization*. Simply speaking, a convex optimization problem is one in which the error function is convex, meaning that its global minimum is guaranteed to be the only local minimum and therefore that point is an optimal solution. An illustration of convexity is shown in Figure 3-4.

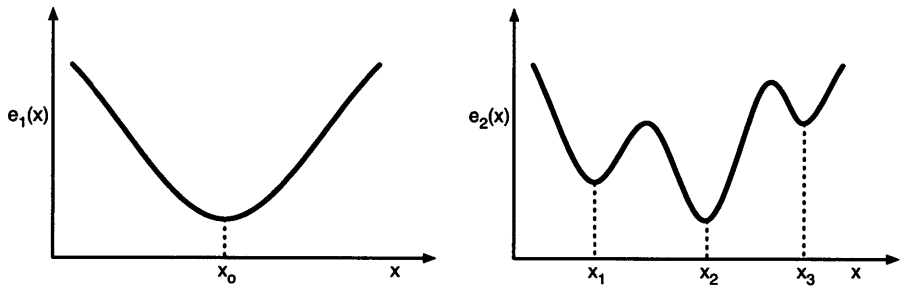


Figure 3-4: Convex vs. non-convex optimization. On the left, function $e_1(x)$ has one minimum x_o, that is both local and global. On the right, $e_2(x)$ has several local minima, x_1, x_2, and x_3, where x_2 is the global minimum.

Quadratic programming

The convex optimization problem of Equation (3.20) can be solved efficiently using *quadratic programming* [24]. A quadratic programming (QP) problem is a minimization problem of the following form:

$$\min_{\mathbf{x}} \; q(\mathbf{x}) = \frac{1}{2}\mathbf{x}^T\mathbf{H}\mathbf{x} + \mathbf{g}^T\mathbf{x} \tag{3.21}$$

$$\text{subject to:} \quad \mathbf{A}\mathbf{x} \geq \mathbf{b} \tag{3.22}$$

$$\mathbf{A}_{eq}\mathbf{x} = \mathbf{b}_{eq} \tag{3.23}$$

To put our problem into this form, we expand Equation (3.20):

$$\begin{aligned} \min_{\mathbf{x}} \|\mathbf{F}\mathbf{x} - \mathbf{s}\|^2 &= (\mathbf{F}\mathbf{x} - \mathbf{s})^T(\mathbf{F}\mathbf{x} - \mathbf{s}) \\ &= \mathbf{x}^T\mathbf{F}^T\mathbf{F}\mathbf{x} - 2\mathbf{s}^T\mathbf{F}\mathbf{x} + \mathbf{s}^T\mathbf{s} \end{aligned} \tag{3.24}$$

The third term $\mathbf{s}^T\mathbf{s}$ is always positive and can be ignored for the purposes of minimization. Thus, we can put our minimization into the QP form of Equation (3.21) using the following definitions:

$$\begin{aligned} \mathbf{H} &= 2\mathbf{F}^T\mathbf{F} \\ \mathbf{g}^T &= -2\mathbf{s}^T\mathbf{F} \end{aligned} \tag{3.25}$$

There are several parameter constraints, some of which are imposed by the KLGLOTT88 source model ($a > 0, b > 0$, and $a = b \cdot OQ \cdot T$). An underlying assumption of the vocal tract model is that it is a resonator (all poles occur in complex conjugate positions, and therefore there are no single poles on the real axis adding any spectral tilt), implying that p is even. We would like the final pole (subsumed from the spectral tilt filter of the KLGLOTT88 model) to have a low-pass characteristic, meaning that the pole itself has a positive real value. The final coefficient α_{p+1} of the polynomial expansion is the product of all of the poles, and we know that poles $1 \ldots p$ will occur in complex pairs, and thus their product will be the product of the pole magnitudes which is real and positive. Therefore, a low-pass characteristic for the final pole can be guaranteed by constraining the value of the final coefficient such that $\alpha_{p+1} > 0$.

We would also like the warped filter coefficients α_k to result in a stable filter. Unfortunately, it is impossible to guarantee a stable filter within the formulation of the quadratic program (such constraints would be nonlinear), and attempting to change the problem formulation to guarantee stability would break the convexity of the problem. Therefore, we make a best effort to preserve stability by putting an upper bound on the final coefficient α_{p+1}. Again, since α_{p+1} is the product of all the pole magnitudes, we choose a maximum magnitude of 0.985 for the p vocal tract poles and 0.9 for the glottal spectral tilt pole as suggested in [42]. These values were found empirically to be helpful in maintaining filter stability when used in the following boundary constraint:

$$\begin{aligned} \alpha_{p+1} &\leq 0.9 \cdot (0.985)^p \quad \text{or} \\ -\alpha_{p+1} &\geq -0.9 \cdot (0.985)^p \end{aligned} \tag{3.26}$$

These boundary constraints are implemented by appropriately defining the weight matrix $\mathbf{A}$ and boundary vector $\mathbf{b}$. Substituting these into Equation (3.22) we get the following vector inequality:

$$\overbrace{\begin{bmatrix} 0 & \cdots & 0 & 0 & 1 & 0 \\ 0 & \cdots & 0 & 0 & 0 & 1 \\ 0 & \cdots & 0 & 1 & 0 & 0 \\ 0 & \cdots & 0 & -1 & 0 & 0 \end{bmatrix}}^{\mathbf{A}} \cdot \overbrace{\begin{bmatrix} \alpha_1 \\ \vdots \\ \alpha_p \\ \alpha_{p+1} \\ a \\ b \end{bmatrix}}^{\mathbf{x}} \geq \overbrace{\begin{bmatrix} 0 \\ 0 \\ 0 \\ -0.9 \cdot (0.985)^p \end{bmatrix}}^{\mathbf{b}} \tag{3.27}$$

There is only one equality constraint which is from the KLGLOTT88 model (Eq. 3.4), so the definitions of $\mathbf{A}_{eq}$ and $\mathbf{b}_{eq}$ in Equation (3.23) are relatively simple.

$$\overbrace{\begin{bmatrix} 0 & \cdots & 0 & 1 & -T \cdot OQ \end{bmatrix}}^{\mathbf{A}_{eq}} \cdot \overbrace{\begin{bmatrix} \alpha_1 \\ \vdots \\ \alpha_{p+1} \\ a \\ b \end{bmatrix}}^{\mathbf{x}} = \overbrace{0}^{\mathbf{b}_{eq}} \tag{3.28}$$

Substituting Equation (3.25) into Equation (3.21) gives us the objective function $q(\mathbf{x})$ with constraints as defined by Equations (3.27) and (3.28). And we have shown that the parameters that minimize $q(\mathbf{x})$ will also minimize $\|\mathbf{e}\|^2$, which is our desired optimization criterion. We now have a properly formed quadratic program that can be solved using any number of well-known iterative numerical algorithms. Most of these fall into a category known as *active set* methods. An active set method operates by iteratively updating the applicable constraints (the active set) at a proposed solution point and using the active constraints to direct the next search step towards the ultimate solution. An extremely comprehensive list of quadratic programming algorithms is available in [27]. For our purposes, the quadratic programming function of the MATLAB Optimization Toolbox [50], which uses an active set algorithm, was sufficient for solving the minimization problem.

The result is a simultaneous estimate of the KLGLOTT88 parameters, a and b, and the warped LP filter coefficients α_k for each period of the source recording $s[n]$. An example of parameter estimation from one period is shown in Figure 3-5. This entire joint estimation process, however, has assumed that the values of T and OQ are known quantities. The following section (3.2) describes the estimation of these parameters.

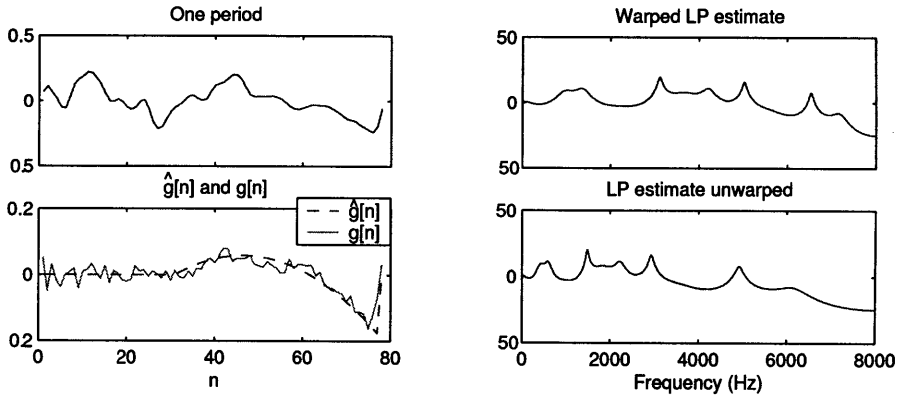


Figure 3-5: Joint source-filter parameter estimation of vowel [e]

All-pole or pole-zero?

The joint parameter estimation procedure results in an all-pole filter in the warped frequency domain. When transformed to the linear frequency domain, however, the reality is that this filter is actually a pole-zero filter. This becomes apparent by substituting the all-pass warping function from Equation (3.7) into the polynomial $A(z)$ in Equation (3.10).

$$\begin{aligned} A(z) &= 1 - \sum_{k=1}^{p} \alpha_k D(z)^k \\ &= 1 - \sum_{k=1}^{p} \alpha_k \left(\frac{z^{-1} - \lambda}{1 - \lambda z^{-1}} \right)^k \end{aligned} \tag{3.29}$$

After this substitution it is clear $A(z)$ becomes a ratio of polynomials and therefore $\frac{1}{A(z)}$, which previously defined the all-pole filter must contain both poles and zeros. This allows the warped filter to make the sharper transitions necessary in providing greater resolution at lower frequencies [29]. Fortunately, by conducting the analysis entirely in the warped domain, the pole-zero representation is never required. This is important because the all-pole representation is critical to the formulation of the minimization problem because the inverse filter has a finite impulse response. If the vocal tract filter was pole-zero, the calculation of the residual would require sequences of infinite length and would no longer be modeled by linear prediction.

3.1.4 Parameter estimation for very short periods

For very short pitch periods (high-pitched singing) there is oftentimes not enough information in a single period to provide an accurate estimation of the KLGLOTT88 and LP filter parameters because the duration of the vocal tract impulse response is significantly longer than the pitch period. The highest soprano notes can have fundamental frequencies of >1 kHz (period <1 msec). Even the high E at the top of the treble clef, a quite reasonable pitch for a soprano, has a period of only $\approx$ 1.5 msec, shorter than the average vocal tract impulse response. Depending on the sampling rate, such short durations may not contain enough samples for a reasonable parameter estimate.

Warped LP is even more sensitive to this problem because each warped delay $d\{\cdot\}$ requires more than one previous linear sample of the signal (z^{-1}). With larger delay values (e.g. z^{-10} vs. $d_{10}\{\cdot\}$) the difference between the signal "shifts" becomes significant (a warped delay is not equivalent to a simple shift, hence the quotation marks); this effect is illustrated in Figure 3-6. Therefore, a "shift" of one period occurs in far fewer warped delays than standard delays. Conversely, this means that a warped LP will require more linear samples than a standard LP of equivalent order.

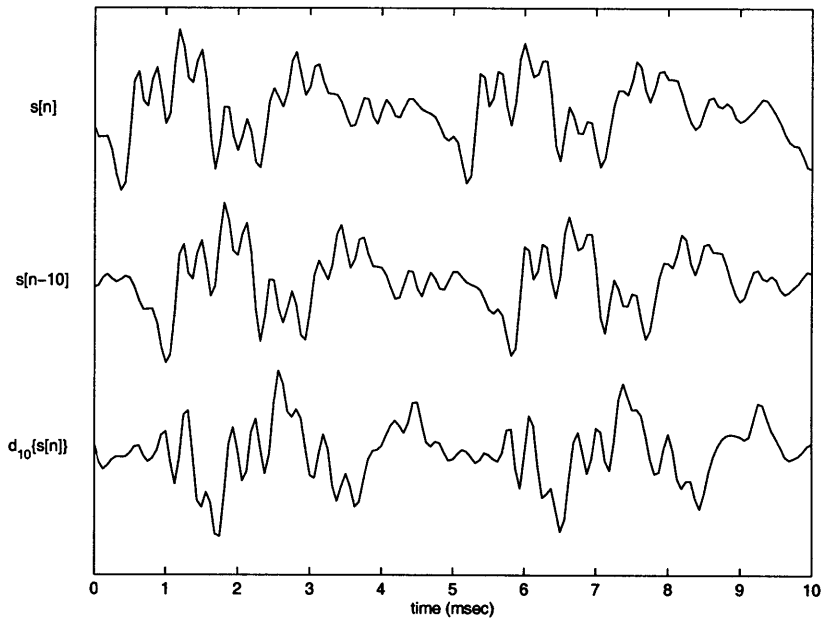


Figure 3-6: Top: two periods of singing $s[n]$ sampled at 16 kHz (period approximate 5 msec). Middle: $s[n]$ delayed by 10 samples. Bottom: 10-fold warped all-pass shift operator $d_{10}\{\cdot\}$ applied to $s[n]$

The problem in analyzing short periods can be overcome by estimating KLGLOTT88 parameters over multiple periods with a single warped LP filter. This solution assumes that the rate of vocal tract articulation remains fairly constant and is not bound to the fundamental frequency of the glottal source. The previous framework can be simply extended to include multiple pitch periods. Even when modeling multiple periods, the relative simplicity of the KLGLOTT88 model ensures that the parameter estimation remains a convex optimization problem, allowing us to independently optimize the glottal shape parameters for each period.

For example, if we wish to simultaneously model two glottal periods the parameter vector becomes:

$$\mathbf{x} = \begin{bmatrix} \alpha_1 & \cdots & \alpha_{p+1} & a_1 & b_1 & a_2 & b_2 \end{bmatrix}^T \tag{3.30}$$

The coefficient matrix then becomes:

$$\mathbf{F} = \begin{bmatrix} d_1\{s[0]\} & \cdots & d_p\{s[0]\} & 0 & 0 & 0 & 0 \\ \vdots & & \vdots & \vdots & \vdots & & \\ d_1\{s[n_c]\} & \cdots & d_p\{s[n_c]\} & 2(0) & -3(0)^2 & \vdots & \vdots \\ \vdots & & \vdots & \vdots & \vdots & & \\ d_1\{s[T-1]\} & \cdots & d_p\{s[T-1]\} & 2(T-n_c-1) & -3(T-n_c-1)^2 & 0 & 0 \\ d_1\{s[T]\} & \cdots & d_p\{s[T]\} & 0 & 0 & 0 & 0 \\ \vdots & & \vdots & & & \vdots & \vdots \\ d_1\{s[T+n_c]\} & \cdots & d_p\{s[T+n_c]\} & \vdots & \vdots & 2(0) & -3(0)^2 \\ \vdots & & \vdots & & & \vdots & \vdots \\ d_1\{s[2T-1]\} & \cdots & d_p\{s[2T-1]\} & 0 & 0 & 2(2T-n_c-1) & -3(2T-n_c-1)^2 \end{bmatrix} \tag{3.31}$$

Similarly, we must extend our boundary and equality constraints accordingly:

$$\overbrace{\begin{bmatrix} 0 & \cdots & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & \cdots & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & \cdots & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & \cdots & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & \cdots & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & \cdots & 0 & -1 & 0 & 0 & 0 & 0 \end{bmatrix}}^{\mathbf{A}} \cdot \overbrace{\begin{bmatrix} \alpha_1 \\ \vdots \\ \alpha_p \\ \alpha_{p+1} \\ a_1 \\ b_1 \\ a_2 \\ b_2 \end{bmatrix}}^{\mathbf{x}} \geq \overbrace{\begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0.9 \cdot -(0.985)^p \end{bmatrix}}^{\mathbf{b}} \tag{3.32}$$

$$\overbrace{\begin{bmatrix} 0 & \cdots & 0 & 1 & -T \cdot OQ & 0 & 0 \\ 0 & \cdots & 0 & 0 & 0 & 1 & -T \cdot OQ \end{bmatrix}}^{\mathbf{A}_{eq}} \cdot \overbrace{\begin{bmatrix} \alpha_1 \\ \vdots \\ \alpha_{p+1} \\ a_1 \\ b_1 \\ a_2 \\ b_2 \end{bmatrix}}^{\mathbf{x}} = \overbrace{\begin{bmatrix} 0 \\ 0 \end{bmatrix}}^{\mathbf{b}_{eq}} \tag{3.33}$$

Substituting these new values into the QP formulas (Eqns. 3.21-3.23) leads to the desired result: a single set of warped LP parameters and independent glottal shape parameters for each period. The values of T and OQ, however, must remain fixed over all periods in the analysis frame. This approach can be extended to include any number of periods, though seldom is more than two or three periods necessary. Examples of two- and three-period parameter estimation are shown in Figures 3-7 and 3-8.

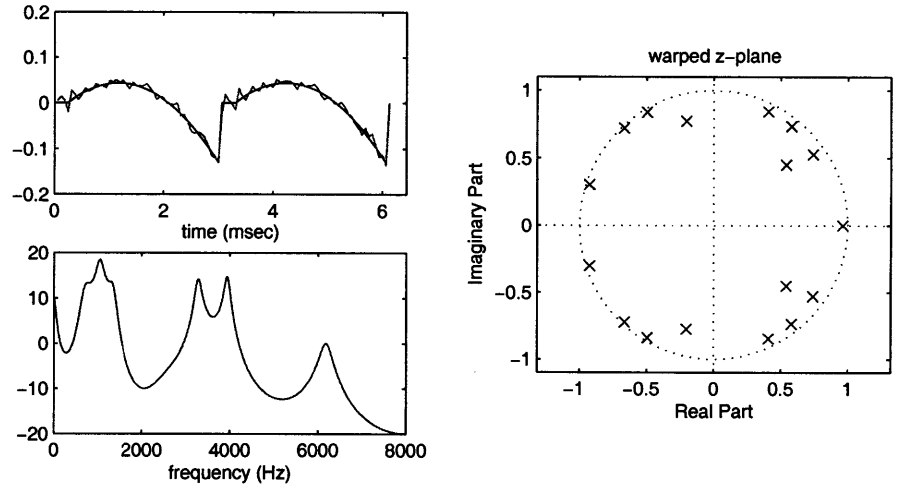


Figure 3-7: Top left: KLGLOTT88 model fit to two short periods ($\sim$ 3 msec). Bottom left: Corresponding warped vocal tract filter. Right: Warped z-plane representation of estimated filter.

3.2 Glottal Closure Instant and Open-Quotient Determination

The parameter estimation detailed in the previous section is *pitch-synchronous*, meaning each analysis frame is time-aligned with each period of the output waveform. This requires the estimation of period boundaries as defined by the *glottal closure instants* (GCIs), which also define the endpoints of the glottal derivative model. The instants of glottal closure are often the moments of strongest excitation, and several proposed techniques for GCI detection are based upon this assumption. Strube [84] suggested a technique using the peaks of the log-determinant of a sliding autocovariance window, which indicate the moments of least linear predictability. Ma *et al.* [44] use a similar method based on the Frobenius norm, the square root of the sum of the squared singular values of a matrix (formed from the windowed signal and its delays). Cheng and O'Shaughnessy [11] use a maximum likelihood method based on the Hilbert transform. In [77], Smits and Yegnanayarana propose a method that utilizes the zero-crossings of the group delay of a windowed signal, based on the assumption that the vocal tract is minimum-phase and thus a window of speech will move from negative to positive phase slope as it crosses the moment of excitation (zero-phase). In a pitch-synchronous analysis of glottal parameters for speaker identification, Plumpe [62]) uses an initial fixed-length LP analysis and identifies the local maxima of the residual as the GCIs.

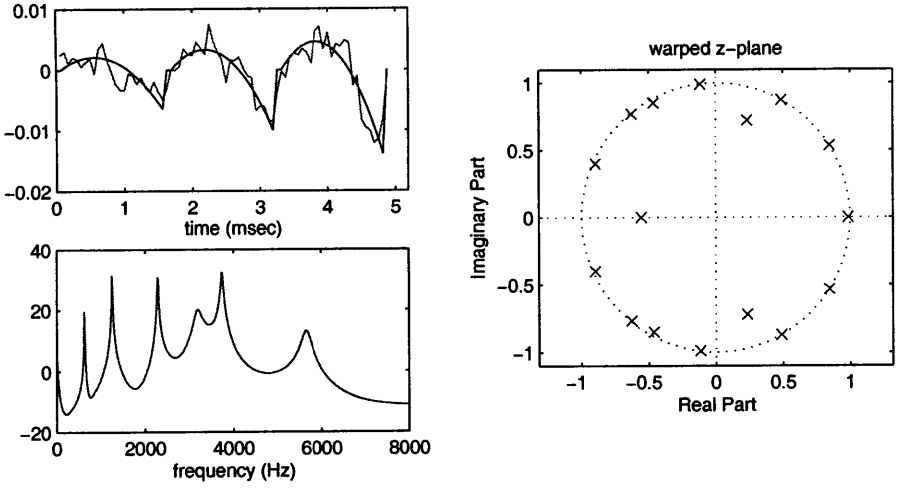


Figure 3-8: Top left: KLGLOTT88 model fit to three very short periods (< 2 msec). Bottom left: Corresponding warped vocal tract filter. Right: Warped z-plane representation of estimated filter.

Each of these techniques has some deficiencies which may result in GCI detection errors. The joint parameter estimation technique is very sensitive to proper period alignment because of the assumed underlying glottal derivative model, so accurate and robust GCI estimation is critical for reasonable parameter estimates. Instead of attempting to calculate the GCIs *a priori*, a search is performed for the period that results in the best model fit (minimizing the overall error). Likewise, it is necessary to search for the appropriate value of the open quotient, OQ. Therefore, we simultaneously search for the optimal values of T and OQ that will minimize the overall error (Eq. 3.20).

The procedure assumes that each analysis frame is contiguous to the previous frame, thus the current period begins one sample after the end of the previous period. Since neither T nor OQ will vary greatly from one period to the next, it is only necessary to search a small range (within a few samples) of values around the T and OQ of the previous frame. The combination of T and OQ that result in the lowest error from the joint paramester estimation (Equation 3.20) are chosen to be the period and open-quotient for the current frame. An example of this search procedure is shown in Figures 3-9 - 3-11.

The search is initialized by locating several initial GCI estimates from the local maxima of a smoothed fixed-length LP residual. These estimates provide an initial starting point for the first analysis frame as well as an initial value of T. The initial analysis frame performs a search for values of T around the initial GCI-derived estimate and performs a wider search over the entire range of reasonable values of OQ (which has been found empirically to vary from 0.4 to 1 [34]).

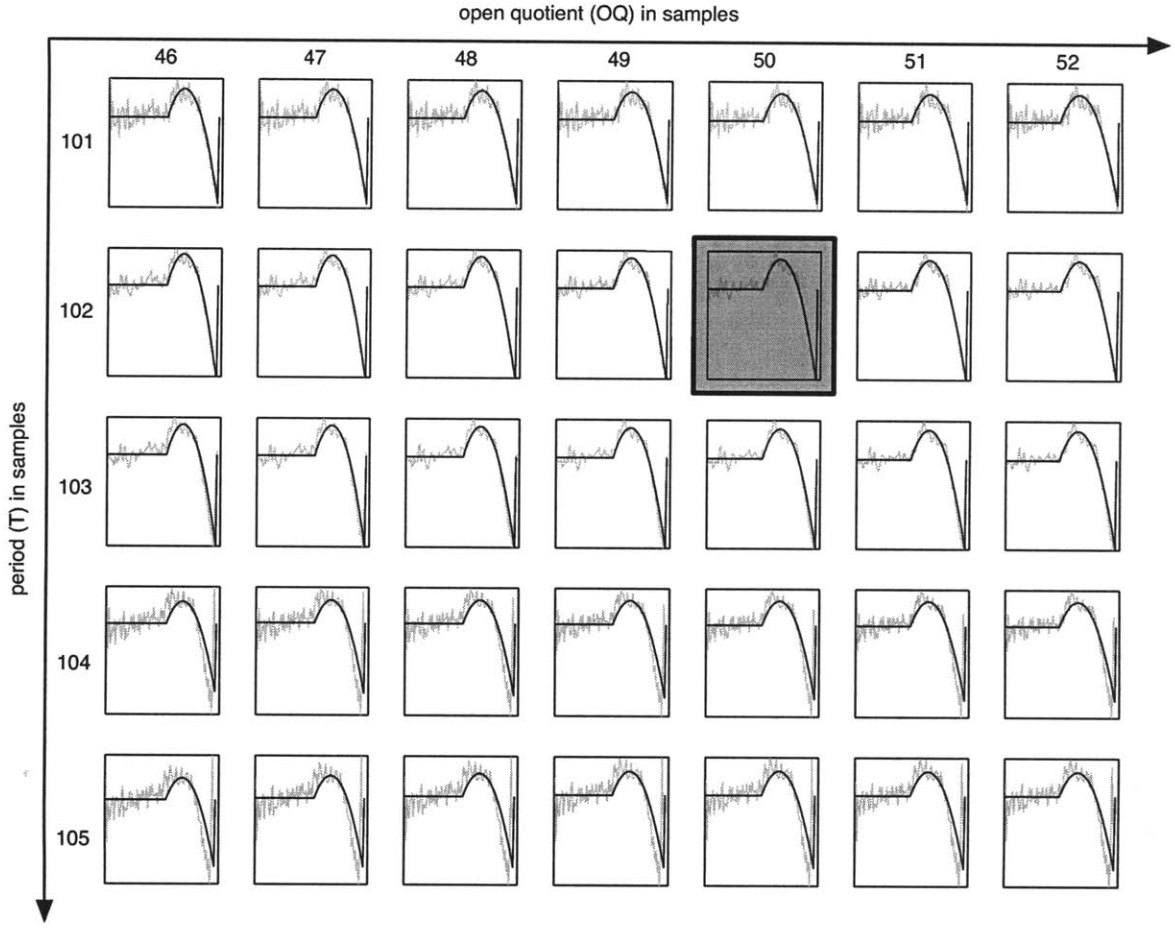


Figure 3-9: Estimated glottal derivatives for a range of periods and open-quotients. The highlighted box is the fit with the lowest error.

Since the search range is highly restricted, the values of T and OQ will not vary much from frame to frame. Further constraints can be added by adding penalties to the overall error for greater deviations of T and OQ.

3.3 LF model parameter estimation

The Liljencrants-Fant (LF) model [23] is an alternative model for the glottal flow derivative that has gained wide acceptance. It more accurately reflects the glottal derivative wave than the KLGLOTT88 model while employing a reasonably small number of parameters that are fairly easy to fit to measured data. The LF model, however, is precluded from being integrated in the joint source-filter parameter estimation algorithm because its use of nonlinear elements (exponentials and sinusoids) prevents any guar-

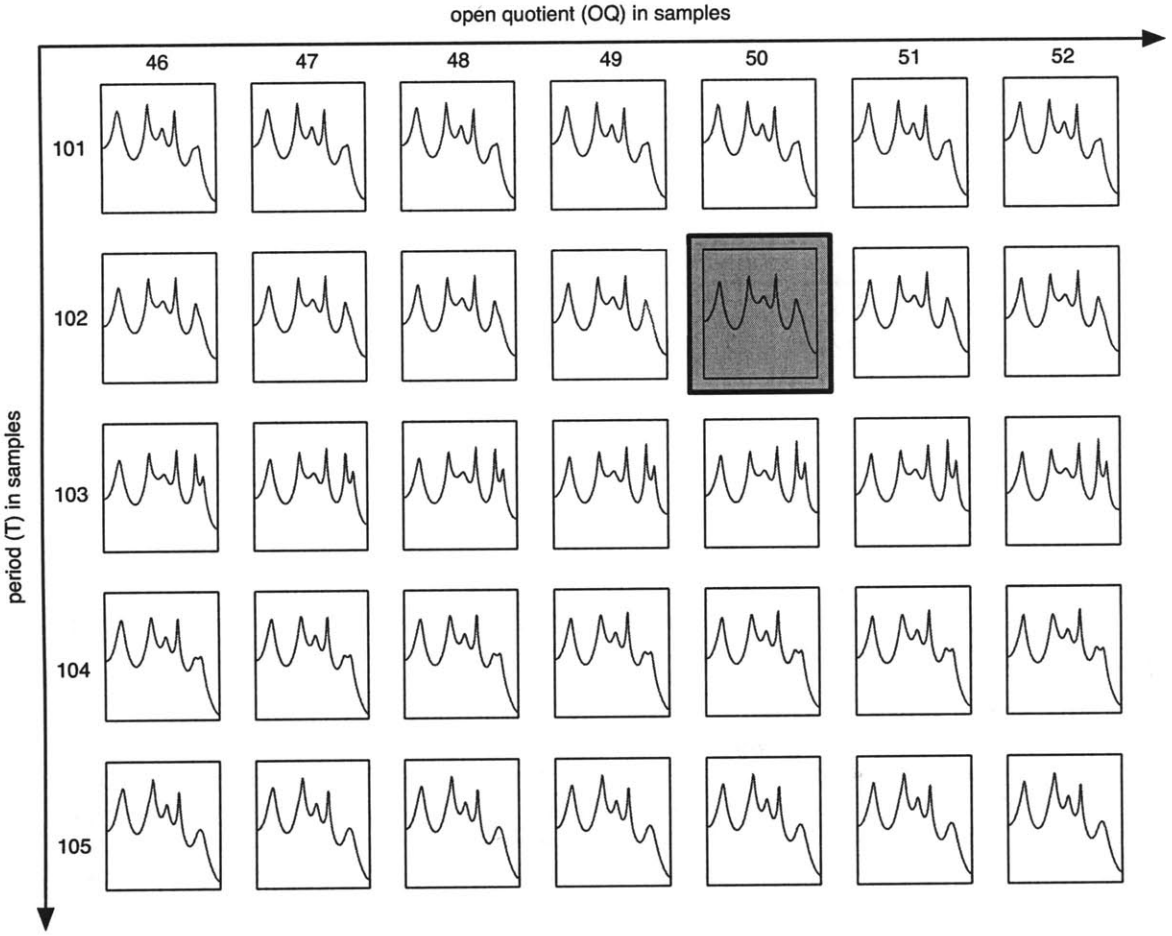


Figure 3-10: Corresponding estimated vocal tract filters to Figure 3-9. Again, the highlighted box is the lowest error fit.

antees of convexity. Instead, the residual $g[n]$ of the jointly estimated warped vocal tract filter can be used to re-parameterize the glottal flow derivative according to the LF model using nonlinear optimization [83]. The greater accuracy of the LF model allows us to reduce the overall system error, thus improving the sound quality of the analysis/synthesis procedure.

The LF model for a single period is shown in Figure 3-12 and is described by the following set of equations:

$$\tilde{g}[n] = \begin{cases} 0, & 0 \leq nT_s < T_o \\ E_o e^{\alpha(nT_s - T_o)} \sin\left[\omega_o(nT_s - To)\right], & T_o \leq nT_s < T_e \\ -\frac{E_e}{1-e^{-\beta(T_c-T_e)}}\left[e^{-\beta(nT_s-T_e)} - e^{-\beta(T_c-T_e)}\right], & T_e \leq nT_s < T_c \end{cases} \quad (3.34)$$

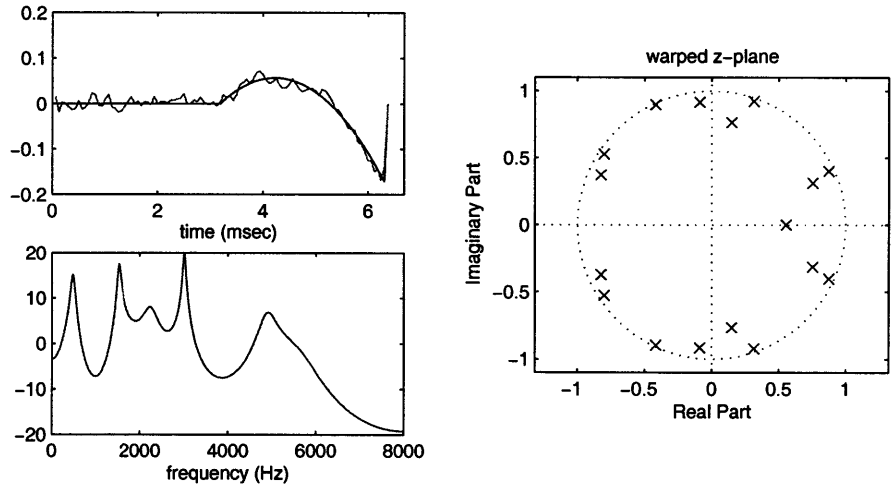


Figure 3-11: The best estimated KLGLOTT88 and warped LP parameters for the given range of T and OQ.

T_s represents the sampling period. T_o, T_e, and T_c are the timing parameters representing the instants of glottal opening, maximum negative value, and glottal closure (also the end of the glottal period), respectively. E_o, α, and ω_o control the waveshape during the open phase (from T_o to T_e, while E_e and β control the shape during the return phase (T_e to T_c). To ensure continuity, the following relation between E_o and E_e must be true at time T_e:

$$E_e = E_o e^{\alpha(T_e - T_o)} \sin\left[\omega_o(T_e - T_o)\right] \tag{3.35}$$

Since E_e (the minimum of the glottal derivative) is derived more easily during analysis than E_o, it is usually estimated first and E_o derived from it. As the result of this relation, there are only four waveshape parameters (disregarding the timing parameters), and consequently the LF model is sometimes referred to as a four-parameter model. The three timing parameters, however, are integral to the model as well, so there are truly seven parameters.

It is also convenient to define an additional time point, T_m (shown in Figure 3-12), which is the location of the downward zero crossing in the open phase of the glottal derivative wave. This point also corresponds to the peak of the glottal flow wave. Because of the sinusoidal shape of the initial part of the open phase, it is clear that the shaping parameter ω_o, the frequency of the sinusoidal component, is dependent on $T_m - T_o$:

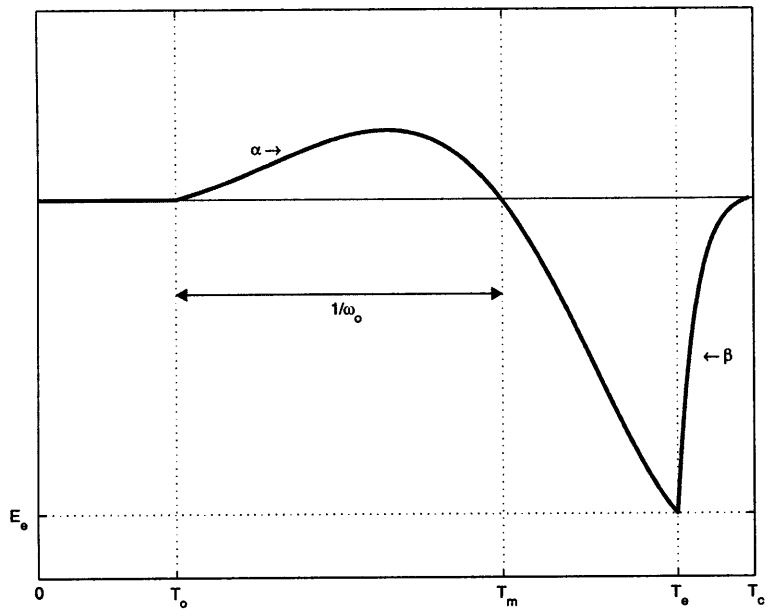


Figure 3-12: The LF glottal derivative wave model and its parameters.

$$\omega_o = \frac{1}{T_m - T_o} \tag{3.36}$$

The values of T_e, E_e, and T_m are fairly easily estimated by inspection of the warped LP residual $g[n]$. T_c is simply the period T from the previous joint parameter estimation step. The initial estimate of T_o is determined by a point to the left of the peak of $g[n]$ at which the value drops below a threshold close to zero. The frequency parameter ω_o can then be estimated according to Equation (3.36). It is also helpful to define β in terms of another timing parameter, T_a:

$$\beta T_a = 1 - e^{-\beta(T_c - T_e)} \tag{3.37}$$

T_a is the time constant for the return phase of the wave and can be determined uniquely from β and vice versa. The relation, however, is nonlinear, but is easily obtained using numerical methods. An initial value for T_a is chosen to be $\frac{1}{2}(T_c - T_e)$.

All of the parameters are then further refined from their initial estimates using constrained nonlinear minimization. Fixing T_e, E_e, and T_c *a priori* reduces the complexity of the search, allowing the optimization to converge fairly quickly. The minimization function is the again the L_2 norm or sum of the squared error at each point. It must

be noted that since this problem is *not* convex, there is the distinct possibility of optimizing to a local minimum instead of the global minimum. By providing good initial parameter estimates we hope to avoid such a situation. At the very least, the initial parameter estimates are an attempt to ensure that the final parameter values do not vary wildly from frame to frame. At the end of the optimization we have values for 6 parameters ($T_o, T_m, T_e, T_a, \alpha$, and E_e) plus the period T of the wave. Figure 3-13 shows the LF wave fit for two distinctly different residual signals.

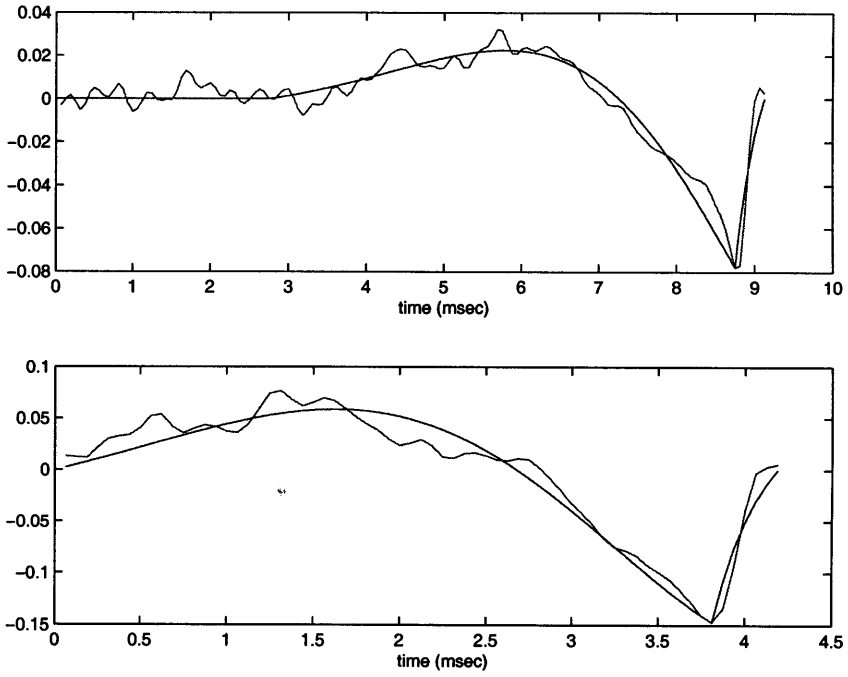


Figure 3-13: LF model fits of two analysis periods.

3.4 Stochastic Component Estimation

There are several stochastic sources that contribute to the overall vocal output, such as glottal aspiration noise and air turbulence in the vocal tract. These other noise-like sources prevent a perfect match to the vocal wave using the LF and WLP parameterization. Glottal aspiration noise is strongly correlated to the instants of glottal opening and closure, and certain vocal tract shapes are also more susceptible to air turbulence. Therefore, the stochastic components must also be modeled in a specific way to reflect these dependencies. Vocal noise components have been previously modeled analytically using using LP-derived filters for noise shaping [15] [74] and statistically using wavelet de-noising [42].

The analysis framework described in this dissertation uses a stochastic codebook approach in which the glottal derivative residuals are used collectively to train a codebook that is representative of the stochastic characteristics of an individual voice. The codebook is determined via *Principal Components Analysis* (PCA) [17] of the glottal derivative residuals. PCA involves the calculation of the eigenvectors and eigenvalues of the covariance matrix of the data. The lowest eigenvectors (those corresponding to the highest eigenvalues) capture the greatest amount of variance, statistically, of the data set. These eigenvectors form an orthogonal basis that is effectively a rotation of the original cartesian basis. A simple graphical representation of PCA is shown in Figure 3-14. PCA has been shown to be useful in reducing the dimensionality of audio signals [75].

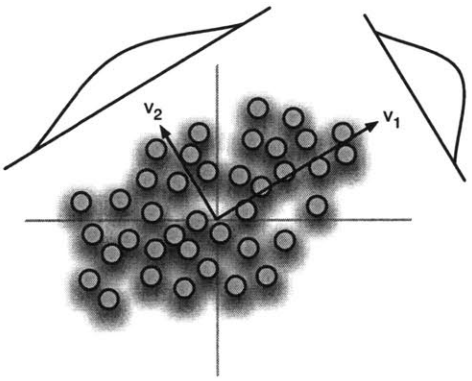


Figure 3-14: Graphical representation of PCA: the variance of the data is best captured by the basis defined by the eigenvectors, v_1 and v_2, which is a rotation of coordinates.

In our case we want to find the eigenvectors of our set of residual noise vectors, $r_m[n]$, the difference between the LF parameterized function and the vocal tract residual from the WLP inverse filter for each analysis period m.

$$r_m[n] = \tilde{g}_m[n] - g_m[n] \tag{3.38}$$

PCA requires that all input vectors be the same length for statistical analysis, but the length of each residual noise vector $r[n]$ varies according to the period. This can be overcome by transforming each residual to the frequency domain using equal-sized *Fast Fourier Transforms* (FFTs) of N points to obtain:

$$
\begin{aligned}
R_m[\omega_k] &= \mathcal{F}\{r_m[n]\}, \text{ where} \\
\omega_k &= \frac{2\pi k}{N} \\
k &= 0, \ldots, N-1
\end{aligned}
\tag{3.39}
$$

Since $r_m[n]$ is real, we need only the first $\frac{N}{2}+1$ values of $R[\omega_k]$ to avoid any data loss, although the FFT values will generally be complex numbers. We define the transformed residual as a complex vector, $\mathbf{r}_m$:

$$
\mathbf{r}_m = R_m[\omega_k], \quad k = 0, \ldots, \frac{N}{2} \tag{3.40}
$$

The transformed residual vectors $\mathbf{r}_m$ are collected in a complex matrix $\mathbf{R}$:

$$
\mathbf{R} = \begin{bmatrix} | & | & | & \\ \mathbf{r}_1 & \mathbf{r}_2 & \mathbf{r}_3 & \cdots \\ | & | & | & \end{bmatrix} \tag{3.41}
$$

The next step is to find the principal components of matrix $\mathbf{R}$, which are the eigenvectors of the covariance matrix $\mathbf{R}\mathbf{R}^T$. The eigenvectors $\mathbf{w}$ of a matrix $\mathbf{M}$ are defined as all vectors for which the following relation holds true:

$$
\mathbf{M}\mathbf{w} = \lambda\mathbf{w} \tag{3.42}
$$

In this case, λ is the associated eigenvalue. The eigenvalues of $\mathbf{M}$ are all λ which satisfy this relation:

$$
\det(\mathbf{M} - \lambda\mathbf{I}) = 0 \tag{3.43}
$$

The eigenvectors and eigenvalues have the property of *diagonalizing* a matrix, which is a rotation of the matrix to one that has only nonzero values along the diagonal. Collecting the eigenvectors in the columns of matrix $\mathbf{W}$ results in the diagonalization equations.

$$
\mathbf{W}^{-1}\mathbf{M}\mathbf{W} = \mathbf{\Lambda} = \begin{bmatrix} \lambda_1 & & 0 \\ & \ddots & \\ 0 & & \lambda_n \end{bmatrix} \tag{3.44}
$$

$$\mathbf{M} = \mathbf{W}\Lambda\mathbf{W}^{-1} \tag{3.45}$$

Attempting to calculate all of the eigenvalues of $\mathbf{R}\mathbf{R}^T$ using Equation (3.43) in order to calculate the eigenvectors (3.42) would be tedious and computationally time consuming. Fortunately, the eigenvectors of $\mathbf{R}\mathbf{R}^T$ can be calculated efficiently using *Singular Value Decomposition* (SVD), which performs a matrix factorization of the form:

$$\mathbf{R} = \mathbf{U}\Sigma\mathbf{V}^T \tag{3.46}$$

$\mathbf{U}$ and $\mathbf{V}$ are *orthogonal* matrices (the inverse of the matrix is its transpose, $\mathbf{U}^T\mathbf{U} = \mathbf{I}$ and $\mathbf{V}^T\mathbf{V} = \mathbf{I}$), and Σ is a diagonal matrix composed of singular values σ_k. From Equation (3.46), it follows that:

$$\begin{aligned}
\mathbf{R}\mathbf{R}^T &= (\mathbf{U}\Sigma\mathbf{V}^T)(\mathbf{U}\Sigma\mathbf{V}^T)^T \\
&= \mathbf{U}\Sigma\mathbf{V}^T\mathbf{V}\Sigma^T\mathbf{U}^T \\
&= \mathbf{U}\Sigma\Sigma^T\mathbf{U}^T \\
&= \mathbf{U}\begin{bmatrix} \sigma_1^2 & & 0 \\ & \ddots & \\ 0 & & \sigma_n^2 \end{bmatrix}\mathbf{U}^T
\end{aligned} \tag{3.47}$$

This has the same form as Equation (3.45), substituting $\mathbf{U}$ for $\mathbf{W}$ (remember $\mathbf{U}^{-1} = \mathbf{U}^T$. This means that $\mathbf{U}$ is the matrix of eigenvectors of the covariance matrix $\mathbf{R}\mathbf{R}^T$, which is our desired result. Note also that Equation (3.47) is also in the same basic form of an SVD (Eq. 3.46). Thus, $\mathbf{U}$ can be calculated directly from an SVD of $\mathbf{R}\mathbf{R}^T$. In this case $\mathbf{R}$ is $m \times n$ where $m = \frac{N}{2} + 1$ and n is the number of observed periods ($m \ll n$). This means that $\mathbf{R}\mathbf{R}^T$ is only $n \times n$ and the cost of the SVD calculation will be greatly reduced.

It should be noted that because the data in matrix $\mathbf{R}$ is complex valued, the transposed matrix $\mathbf{U}^T$ becomes a conjugate or *Hermetian* transposed matrix $\mathbf{U}^H$, defined as follows:

$$\mathbf{U}^H(i,j) \triangleq \overline{\mathbf{U}}(j,i) \tag{3.48}$$

$\mathbf{U}$ is a square matrix of $\frac{N}{2} + 1$ eigenvectors of length $\frac{N}{2} + 1$, which comprise the stochastic codebook. For each transformed noise vector, $\mathbf{r}_m$, we obtain a weighting vector $\mathbf{w}_m$ corresponding to the contribution of each of the eigenvectors in codebook $\mathbf{U}$:

$$\mathbf{w}_m = \mathbf{U}\mathbf{r} \tag{3.49}$$

The first c eigenvectors, along with the first c values of $\mathbf{w}_m$, are used to calculate $\tilde{\mathbf{r}}_m$, an estimate of $\mathbf{r}_m$. The time-domain glottal residual estimate $\tilde{r}_m[n]$ is then the inverse FFT of $\tilde{\mathbf{r}}_m$. If $c = \frac{N}{2} + 1$, $r_m[n]$ can be reconstructed without any loss. Then, in conjunction with the LF glottal model and the warped all-pole vocal tract filter, the input signal can be perfectly reconstructed. Reducing c, however, can still provide a fair estimate of $r_m[n]$. Since $r_m[n]$ should represent a relatively small fraction of the glottal source, even a rough approximation of $r_m[n]$ can result in a high quality voice re-synthesis. The consequences and benefits of a reduced representation will be explored more thoroughly in Chapter 5.

3.5 Parameter selection and transformation

Each set of estimated source-filter parameters provides a representation of one period of voice data. Since many periods are required for a vocal utterance, these parameters will necessarily vary over time. The modeling of the time-varying component is the subject of the next chapter, but the source-filter parameters from each period form the observations upon which the dynamic model will be based. Instead of using the parameter values directly, it is sometimes helpful to transform them to another value that will be easier to model over time.

One particularly useful parameter transformation is the transformation of the all-pole filter coefficients α_k into ordered *line spectrum frequencies* (LSFs) [59]. For time-series modeling, the LSF representation is preferable to the filter coefficients for several reasons. First of all, interpolation between two sets of LSF values results in a stable filter while interpolation between sets of α_k does not. Also, the LSF values are always ordered in frequency, which makes drawing correlations between frames trivial. Line spectrum frequencies are also limited in range (0 to 2π) and do not have as much variance as the filter coefficients. For this reason, the vocal tract filter is transformed to an LSF representation for the dynamic model. They are calculated using the following relations:

$$A(z) = 1 - \alpha_1 z^{-1} - \cdots - \alpha_N z^{-N} \tag{3.50}$$

$$A_1(z) = A(z) + z^{-N-1}A(z^{-1}) \tag{3.51}$$

$$A_2(z) = A(z) - z^{-N-1}A(z^{-1}) \tag{3.52}$$

The roots of both $A_1(z)$ and $A_2(z)$ will be in complex pairs on the unit circle, and can thus be described by a single frequency value. Moreover, for a minimum phase system, the roots of $A_1(z)$ will alternate with the roots $A_2(z)$, providing an enforced order-

ing of parameters. The original polynomial $A(z)$ can be recovered with the following equation:

$$A(z) = \frac{A_1(z) + A_2(z)}{2} \tag{3.53}$$

As mentioned previously, from the LF model we obtain 6 parameters (the 4 timing parameters, α, and E_e. Of course, the duration of the period T is also necessary. Finally, we also record the energy from each analysis period, E_T. All of the parameters are collected into a single observation vector. Figure 3-15 demonstrates the variation of some of the parameters over time.

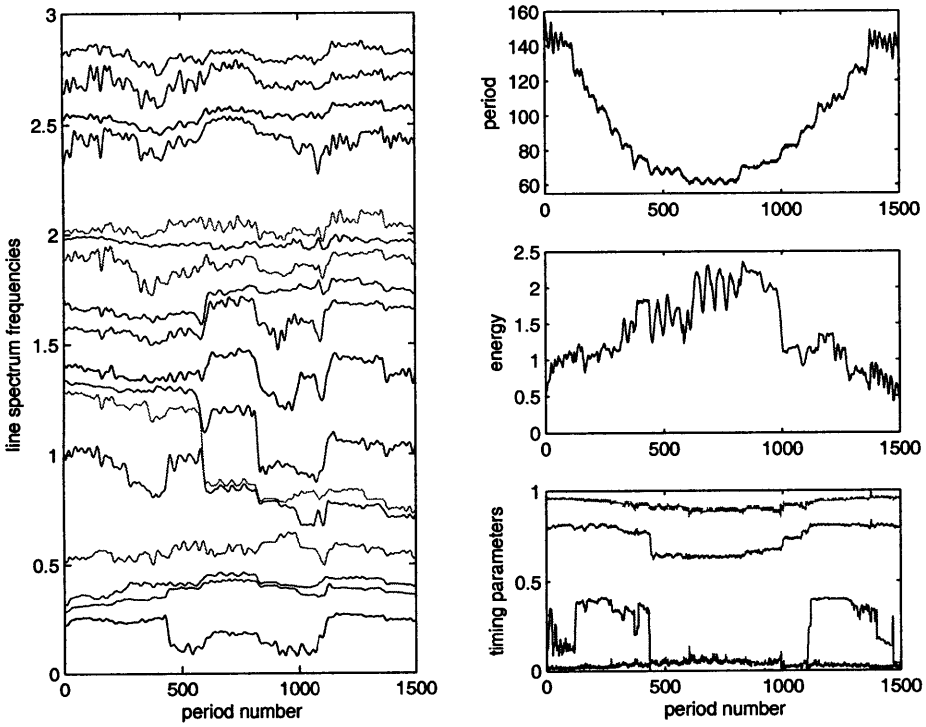


Figure 3-15: Time-varying source-filter parameters from one vocal segment.

3.6 Summary

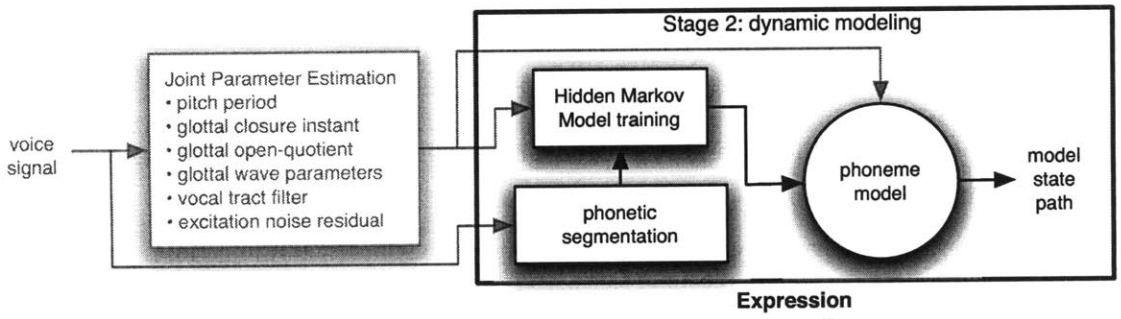
This chapter has described in detail all of the components of the source-filter parameterization. To recap, the analysis is conducted pitch-synchronously, aligned to the period of the singing signal. Initially, source-filter model parameters are jointly estimated

based on the KLGLOTT88 glottal flow derivative model and a frequency warped all-pole filter derived via warped linear prediction. The period and glottal open-quotient are estimated through a limited search over a small area of likely values. The warped LP residual is then re-parameterized using the Liljencrants-Fant glottal flow derivative model for greater accuracy. Finally, the stochastic glottal residual is modeled via a codebook constructed using Principal Components Analysis. The model parameters calculated during this phase of the analysis allow for perfect reconstruction of the input signal.

The source-filter parameter estimation described in this chapter is intended to represent features related to physical characteristics of the singer at a given time. As such, this analysis is limited to one period (or in the case of very short periods, a small number of periods) of singing data. Obviously, the instantaneous physical features of a singer do change (sometimes quite rapidly) over time. Although a few of the source-filter parameters are constrained between analysis frames, as of yet we have not discussed a dynamic representation to model the evolution of parameters during singing. This subject is investigated in the following chapter.

CHAPTER FOUR

Dynamic Parameter Modeling



The previous chapter dealt with the estimation of short-time features motivated by the physiology of the vocal production system. The focus now becomes describing the evolution of those features over time and modeling their behavior in such a way as to capture identifying aspects of vocal expression. Because most singing is structurally based on linguistic communication, the basic structure for time-varying vocal analysis used here provided by phonetics: the segregation of sound into acoustic units based on their linguistic function. Conceptually, phonemes provide a logical and convenient method of classifying and ordering vocal utterances. The perceptual salience of a purely phonetic description, however, leaves much to be desired. This is because the acoustic properties of phonemes vary tremendously depending on context. And since its definition is purely conceptual, it is difficult to determine a ground truth for the definitive characteristics of a phoneme.

In other tasks, such as speech recognition, a variety of strategies have been used to overcome the limitations of a purely phonetic representation. Markov chains (a linked progression of random processes) have been successfully used to model the evolution of the acoustic properties of phonemes as an ordered succession of parameter states over short durations. These phoneme models are usually implemented as unidirectional *Hidden Markov Models* (HMMs). An enhancement to this approach is to combine consecutive phoneme sets into diphones and triphones, acknowledging the fact that the sound of a phoneme may be affected by the preceeding and succeeding

phonemes. These techniques have been fairly successful as analysis tools in performing phoneme estimation for speech recognition systems, but their use in systems for natural-sounding concatenative speech synthesis has been somewhat less successful. The problem is still the enormous acoustic variation of the voice. Even modeling diphones and triphones does not adequately encompass the possible variations in vocal output. Another limitation of diphones and triphones is that a very large amount of training data is required to adequately populate a comprehensive model. While there are large corpuses of high-quality labeled speech training data, there is a paltry amount of singing voice data.

As mentioned in Chapter 2, singing is quite different from speech. In classical singing voiced sounds dominate, particularly vowels. A well-trained singer is able to sustain vowels for long durations without sounding static by constantly varying the sound to suit his or her expressive intentions. Because of the vowel emphasis in singing and the limited amount of available training data, the system presented here uses a single phoneme (as opposed to multiphone) representation. But the phoneme model diverges from the traditional Markov chain representation by using a much larger number of *fully-connected* states (the states are not restricted to occuring in a particular order). In using a larger number of fully-connected states, this model is designed to better encapsulate the range of acoustic variations that occur in an individual's singing voice.

Since the system deals with individual phonemes, the first step is to perform a phonetic segmentation of the input signal, so that data can be applied to the appropriate phoneme model. Once the input is segmented, the source-filter parameters derived in the previous chapter are used as observations to train the individual phoneme models. Once training is complete, new data can be analyzed with the model and distilled into an HMM state path, which (it is hoped) contains the time-varying expressive information of a signal segment. The goal of this chapter is to present each of the dynamic modeling steps in greater detail.

4.1 Phonetic Segmentation

The phonetic segmentation system presented here assumes that a word transcript of the singing input is available *a priori*, a restriction currently necessary for accurate segmentation. This assumption does not seem unreasonable, given that most classical singing is an interpretation of a pre-defined score. A phonetic transcript can be derived from a word transcript in a straightforward manner using a phonetic dictionary. Figure 4.1 shows the major components of the phonetic segmentation system.

In order to automatically segment vowels from other phonemes, a straightforward approach is to train a segmentation system on recordings for which the individual phoneme regions have been annotated. This type of phonetic transcription can be done by hand, but the task is quite tedious and error-prone. Fortunately, there exists an ample amount of accurate phonetically-labeled speech data that is primarily used to train speech recognition systems. With the selection of appropriate features, a

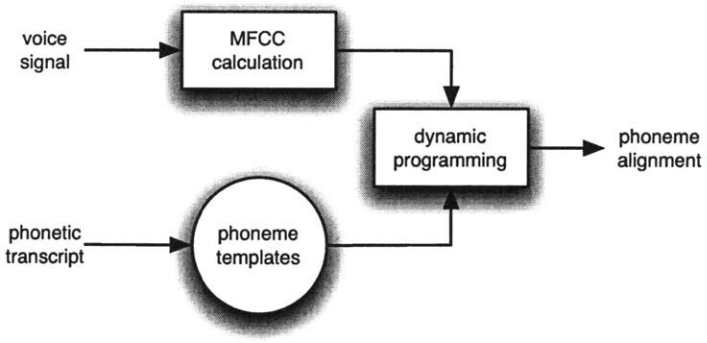


Figure 4-1: Flowchart of the phonetic segmentation process.

database of speech phonemes can be used as a starting point for phonetic segmentation of singing.

There are many features in both the time and frequency domains that can aid in the discrimination of phonemes. Different phonemes tend to have fairly distinct formant patterns, though the specific locations vary a great deal from person to person and even between utterances from the same person. Human auditory perception, of course, is adept at generalizing these variations into broad phoneme classes. Computationally, features derived from the general spectral envelope have proven to be fairly successful when used in systems for identifying phonemes, particularly ones which perform some degree of perceptual frequency warping. The most commonly utilized features are the *Mel-frequency cepstral coefficients* (MFCCs) of the audio signal.

4.1.1 Mel-frequency cepstral coefficients

The *cepstrum* C_k of a time-domain signal $s[n]$ is the inverse Fourier transform of the log-magnitude spectrum of the signal.

$$C_k = \mathcal{F}^{-1}\{\log \|\mathcal{F}\{s[n]\}\|\} \tag{4.1}$$

Since $s[n]$ is real, its Fourier transform $S(\omega)$ will be an even function. Taking the log-magnitude of $S(\omega)$ then results in a real and even function. Thus in practice, the cepstrum is normally computed using the FFT and the *Discrete Cosine Transform* (DCT). The cosine transform of an even function is equivalent to its Fourier transform and requires only half the number of calculations to compute. The individual cepstral coefficients are the values of C_k for $k = 0, 1, \ldots$.

The *mel scale* is an attempt to model frequency from a purely perceptual standpoint. The mel scale was first proposed by Stevens and Volkmann [81], based on psychoacoustic experiments examining people's perception of different frequencies relative to

a 1000 Hz tone at 40 dB. Thus, a frequency at x mels was judged to be twice as high as the frequency corresponding to $\frac{x}{2}$ mels. The overall result is a scale that is roughly linear up to 1000 Hz and approximately logarithmic above it.

MFCCs are computed by redistributing the linearly-spaced bins of the log-magnitude FFT into mel-spaced bins followed by a DCT. The redistribution can be viewed as an interpolation of the log-spectrum values or equivalently as multiplication by a mel-spaced filterbank. A relatively small number of coefficients (13 are commonly used) provide a general approximation of the spectral envelope at a resolution that is a fair compromise between the specificity needed to distinguish different phonemes and the generality necessary for grouping like phonemes from different sources. This is especially critical for our task since the training and testing data come from different domains (speech and singing, respectively).

An additional benefit of MFCCs is that they have a decorrelating effect on the spectral data, maximizing the variance of the coefficients (similar to the effect of PCA) [39]. This is obviously beneficial to applications involving classification, such as phoneme identification. MFCCs have gained wide acceptance as front-end features for speech recognition systems [65]. They have also been used as features in other audio applications such as music description [39], musical instrument identification [16][22], and music similarity measurement [3].

4.1.2 System training

TIMIT is a large database of speech samples from 630 different speakers combining data collected by Texas Instruments (TI) and MIT that is distributed by the National Institute of Standards and Technology (NIST) [56]. The speech samples are phonetically balanced and have been phonetically labeled by hand. All sample segments were segregated by phoneme to form a training data set for each of the 40 English phonemes.

Reference templates were created for each phoneme by averaging 13 MFCCs calculated from short-time (16 msec) windowed overlapping frames. The simplicity of this representation has obvious limitations. First of all, any time-varying features are lost in the averaging. It is also possible that averaging the features over such a wide variety of speakers may reduce the overall discriminatory effect of the features. Since vowels comprise the vast majority of classical singing and are the primary focus of this dissertation, some of these limitations become lesser concerns. Unlike some other phonemes, the identification of vowels does not depend heavily on time-varying characteristics. Likewise, vowels are spectrally quite distinct from one another and the discriminability of the templates will likely be less affected by averaging. Additionally, the high computational requirements of the segmentation algorithm benefit from a simple representation.

4.1.3 Alignment via dynamic programming

The same features (13 MFCCs calculated over 16 msec frames) are calculated directly from the time-domain singing input signal to be segmented, and the order of expected

phonemes is determined from the transcript of the singing input available beforehand. The goal of the system is to best align the extracted features with the expected phonemes in the correct order so that exact boundaries between phonemes can be determined. Towards this goal, a weighted L_2 norm is calculated between each frame of the new data and the reference templates of each of the expected phonemes. The result is a distance matrix in which the rows are in the order of the expected phonemes and each column is one analysis frame of the singing input.

The technique of *dynamic programming* (DP) is used to find the optimal alignment between the extracted features and the ordered phonemes. DP is an efficient technique for determining the least-cost path between multiple possibilities and given certain constraints on the movement of the path [65]. In this particular case, the goal is to find the shortest overall distance path traversing the MFCC-to-phoneme reference template distance matrix given the constraint of phoneme order (Figure 4-2). This application of DP is often called *dynamic time warping.*

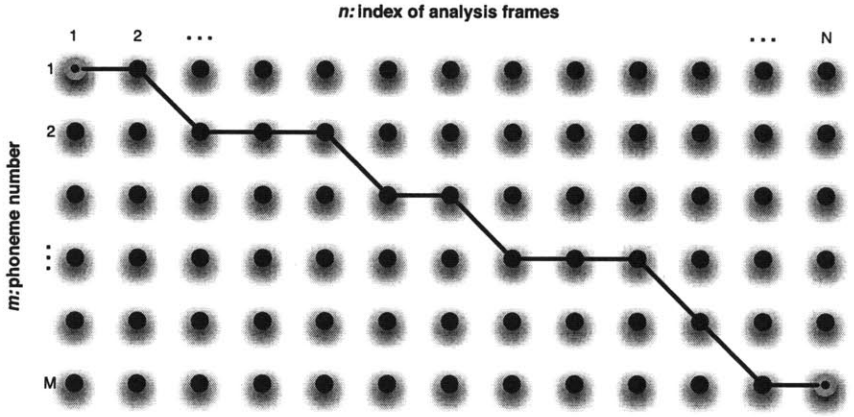


Figure 4-2: An illustration of dynamic programming

The DP algorithm is based on the principle of optimality, which is described recursively: each point on an optimal (least-cost) path itself is the result of an optimal path to the initial starting point. This suggests a recursive strategy that evaluates all possible paths to the desired endpoint starting from an initial condition. If $\phi_m(1, l)$ is the minimum cost from the initial condition (1) to point l in m decisions, then the minimum cost to point n in $m + 1$ decisions (frames) is

$$\phi_{m+1}(1, n) = \min_l [\phi_m(1, l) + d_{m+1}(n)] \tag{4.2}$$

where $d_{m+1}(n)$ is the cost of moving to point n in frame $m+1$. This recursive equation provides a method for incrementally searching for the overall optimum path. Once the final desired state is reached (point N in M decisions), the optimal path can be

determined by backtracking through the minimum cost function $\phi_m(1,n)$ for $m = M, M-1, \ldots, 1$ starting at $n = N$.

In the specific case of phoneme alignment, m is the index of the analysis frames while n is the index of phonemes (1 being the first and N being the last). The cost function, $d_m(n)$, is the weighted L_2 norm of the 13 MFCCs calculated from analysis frame m ($C_k(m)$) and the reference template of phoneme n ($P_k(n)$), where l is the phoneme at frame $m-1$. The weights w_k are specified to reduce the influence the first cepstral coefficient C_0 which is highly correlated to signal energy and spans a wider range of values than the other coefficients.

$$d_m(l,n) = \sum_{k=0}^{12} [(C_k(m) - P_k(n))w_k]^2 \tag{4.3}$$

The constraints to the path optimization problem are that the phonemes occur in a specific order and therefore path movement is limited to at most one point (phoneme) between analysis frames. In terms of Equation (4.2), $n - l \leq 1$ for all m. These constraints eliminate many path possibilities in the exhaustive space of all paths, reducing the overall computational requirements of the path optimization while still considering all relevant path possibilities.

The end result is a path that indicates the optimal transitions between phonemes. Since the frame boundaries most likely will not correspond to the period boundaries, the transitions are shifted slightly to the nearest period boundary. An example of a distance matrix and the resulting phonetic segmentation is shown in Figure 4-3.

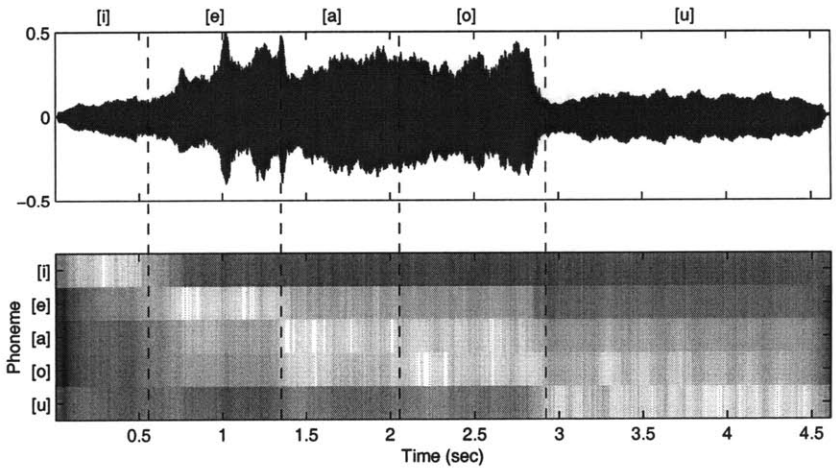


Figure 4-3: Example of phonetic segmentation (lighter colors indicate shorter distances)

If the estimated alignment is valid, the phoneme segments from the sung input can be used to update the reference phoneme templates. Since the original templates were derived entirely from speech data, integrating additional sung training data is likely to improve the effectiveness of the system. Given enough training data, specialized templates could potentially be built for each individual singer.

4.2 Dynamic Parameter Modeling using Hidden Markov Models

A *Markov chain* is a representation for linked stochastic processes in which the links themselves are also stochastic processes. The links, known as *transitions*, are normally represented by a simple probability model while the linked processes or *states* generally involve more complex stochastic models. In an *observable* Markov model, the states correspond to random processes whose outcomes are directly observable (e.g. the outcome of a coin flip directly observed as either heads or tails). In a *hidden* Markov model (HMM), the states are not directly observable, but are indirectly observed through another random process, hence the "hidden" nature of the states. In many HMMs, these states do not necessarily correspond to a tangible or easily described aspect of the underlying process, but are simply clusters of observations that have been grouped together statistically. The number of states in the model, however, must be determined *a priori*.

In a discrete-time HMM, the model is considered to be in exactly one state at each time sample. With the passing of each time sample, the model may transition to another state or may remain in the same state. HMMs can be used to model a wide range of time-varying processes, and they have been quite successful in tasks such as automatic speech recognition. In speech HMMs, the states are determined indirectly from the MFCC observations and most commonly represent phonemes and subphonemes. Phoneme and word models are then defined as a consecutive series of states. The ordering constraint is implemented by limiting state transitions only to consecutive states. A simple 4-state example of this model topology is given in Figure 4-4. This type of model is appropriate for speech recognition because of the short duration of phonemes, their fairly rapid rate of change, and an underlying foundation of language based upon phonetic ordering.

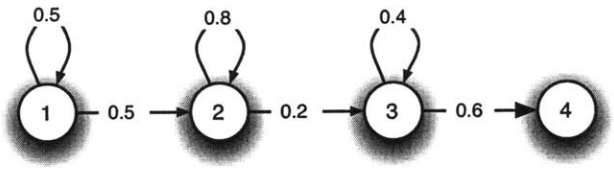


Figure 4-4: A 4-state unidirectional HMM with transition probabilities

For singing voice analysis/synthesis, however, the goal goes beyond the understanding of language and includes the communication of less well-defined extralinguistic information such as musical expression and emotion. Pursuant to this goal, the states of the HMM vowel models in this system are intended to represent something quite different from the states used for speech. Vowels in singing are longer (sometimes much longer) and have a wider acoustic variability than spoken vowels, which is reflected in the source-filter model parameters. The acoustic and parameter variation normally takes place smoothly in subtle gradations over time. Modeling these variations entails accounting for a more densely populated feature space in terms of the location and number of states. To better encompass this acoustic space, we use a fairly large number of states (10). Morever, we use a topology in which the states are *fully-connected*, i.e. transitions may occur from any state to any state, also known as an *ergodic* HMM. Figure 4-5 shows an example of a simple fully-connected HMM. The working hypothesis is that through better description and encapsulation of the physical feature space, an HMM will preserve and model the key aspects of expression that are necessary for preserving singer identity.

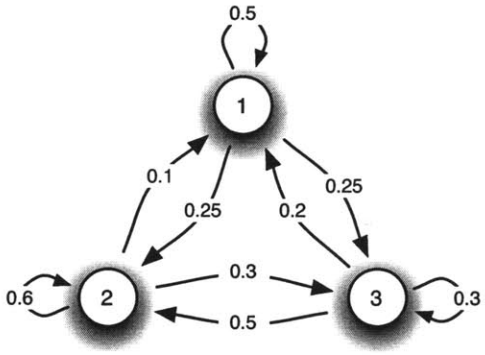


Figure 4-5: A 3-state fully-connected HMM with transition probabilities

The primary benefit of the HMM representation is the ability to model general parameter changes over time through the changes in state. In an ideal case, the trained states are concentrated clusters of data in the parameter space. Hence, the state path is an approximation to the actual parameter trajectory, in which the estimated trajectory passes through the mean of each state. The accuracy of the approximation will depend upon the variance of the states. A diagram of this general behavior is given in Figure 4-6.

The observation vectors for the vowel models are the source-filter model parameter estimates of the previous chapter, in particular the line spectrum frequencies representing the vocal tract filter, the LF parameters for the glottal derivative wave, the period T, and the energy of each analysis frame. Since the observations are taken every pitch period each state will correspond to one period, and state transitions will occur at this

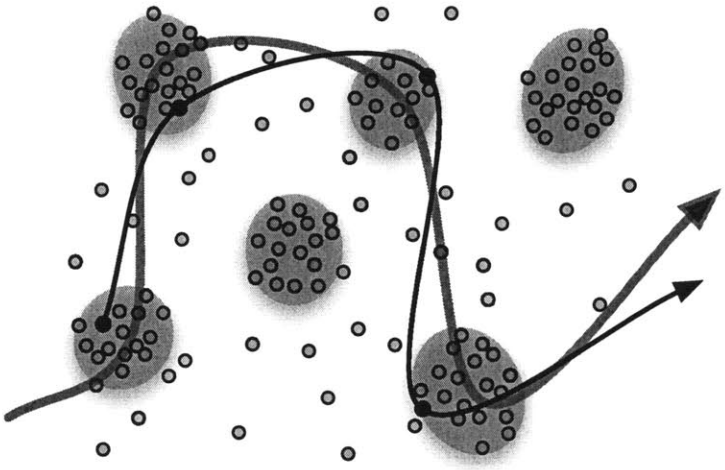


Figure 4-6: A depiction of an actual parameter trajectory (black line) and the estimated trajectory via state path (gray line) in parameter space.

rate. As mentioned previously, each model consists of 10 states. The number of states was determined empirically by evaluating models with 5, 10, 12, and 15 states in terms of sound quality of the state-path-only re-synthesis and model log-likelihood (calculations of these criteria are described in sections 4.2.3 and 4.2.4, respectively). 10-state models were chosen qualitatively as a compromise between the divergent goals of encompassing the wide variability of each vowel's observed features and limiting the computational complexity of the model. Training the HMM involves defining the states, determining the observation probability distribution of each state, and determining the transition probabilities from each state to each of the other states, including the probability of remaining in the current state.

4.2.1 Specifying the HMM

A Hidden Markov Model is defined by the number of states N and three sets of probability distributions. The states are simply labeled as $\{1, 2, \ldots, N\}$ and the state at time t is denoted as q_t. The first set of probability distributions consists of the state-to-state transition probabilities $\mathbf{A} = \{a_{ij}\}$ where

$$a_{ij} = P[q_{t+1} = j | q_t = i], \quad 1 \leq i, j \leq N. \tag{4.4}$$

The second set of distributions are the probability distributions of the observations for a given state j: $\mathbf{B} = \{b_j(\mathbf{o})\}$, where $\mathbf{o}$ is the vector of observations. The distributions are modeled as a mixture of K continuous observation densities.

$$b_j(\mathbf{o}) = \sum_{k=1}^{K} c_{jk} \mathcal{N}_k(\mathbf{o}, \mu_{jk}, \mathbf{U}_{jk}), \quad 1 \leq j \leq N \tag{4.5}$$

The coefficients c_{jk} are the mixture weights of the component densities and must therefore sum to 1. $\mathcal{N}$ refers to the the Gaussian or normal density, with mean μ_{jk} and covariance $\mathbf{U}_{jk}$. If the length of the observation vector $\mathbf{o}$ is M, this is more explicitly written as

$$\mathcal{N}_k(\mathbf{o}, \mu_{jk}, \mathbf{U}_{jk}) = \frac{e^{-\frac{1}{2}(\mathbf{o}-\mu_{jk})^T \mathbf{U}_{jk}^{-1}(\mathbf{o}-\mu_{jk})}}{(2\pi)^{\frac{M}{2}} |\mathbf{U}_{jk}|^{\frac{1}{2}}}. \tag{4.6}$$

The third probability distribution needed to define an HMM is the probability distribution of the initial state $\pi = \{\pi_i\}$, also known as the *prior.*

$$\pi_i = P[q_1 = i], \quad 1 \leq i \leq N \tag{4.7}$$

Since the distributions $\mathbf{A}, \mathbf{B}$, and π along with the number of states N completely specify an HMM, for convenience the model is denoted as λ, where

$$\lambda \sim (\mathbf{A}, \mathbf{B}, \pi). \tag{4.8}$$

There are three tasks commonly associated with the use of HMMs: (1) *Model training*—the determination of the HMM parameters from observations of data; (2) *State-path determination*—the calculation of the most likely sequence of HMM states for new observations given a trained model; and (3) *Model likelihood evaluation*—calculating the likelihood of a sequence of data given a trained model. There are well-established solutions to each of these problems, which we will make use of in the following sections. More detailed descriptions and derivations of these methods are available in Appendix A.

4.2.2 HMM training

For determine a dynamic model for each vowel, we use the source-filter parameter observation sequences taken across the duration of a vowel to determine the model parameters **A, B**, and π. The most common HMM training technique is an iterative method known as the *Baum-Welch* or *Expectation-Maximization* (EM) algorithm (see Appendix A.1 for details). The vowel phoneme models are trained on source-filter parameter observation sequences from data that has been phonetically segmented using the system from Section 4.1. Multiple observation sequences are helpful for achieving a model that generalizes well over a wide range of parameter variations. Each vowel model consists of 10 states, with a single Gaussian observation density with a full co-

variance matrix (not diagonal) for each state. A single Gaussian (rather than a mixture of Gaussians) was chosen due to the relative sparsity of training data (2-10 training sequences for each vowel model).

The EM training algorithm determines the model states according to the density of the observations in parameter space. The meaning of the states themselves is not easily defined from a perceptual standpoint, but they are common parameter combinations derived from the voice signal relating to aspects of the physical state of the singer at a given time. The state-to-state motion captures some of the enormous variety of physical configurations involved in singing. Determining these state-to-state dynamics is the focus of the next section.

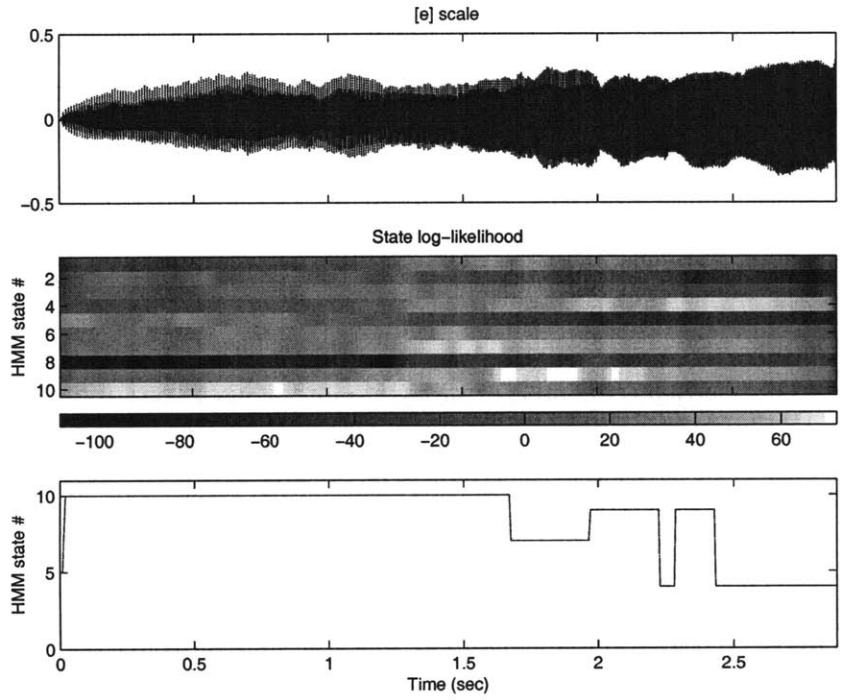


Figure 4-7: HMM state log-likelihood and state path for a scale passage on the vowel [e].

4.2.3 Estimating state paths

The state-path representation is central to this framework, as it compactly represents the essential dynamics of the source-filter parameters. The variation of parameters over time is reflective of the expressive characteristics of the singer. The state path representation is also the foundation for the application of this framework to singing voice coding. The hypothesis is that this representation will be rich enough to fully encode the time-varying aspects of the voice with a result that is perceptually sufficient.

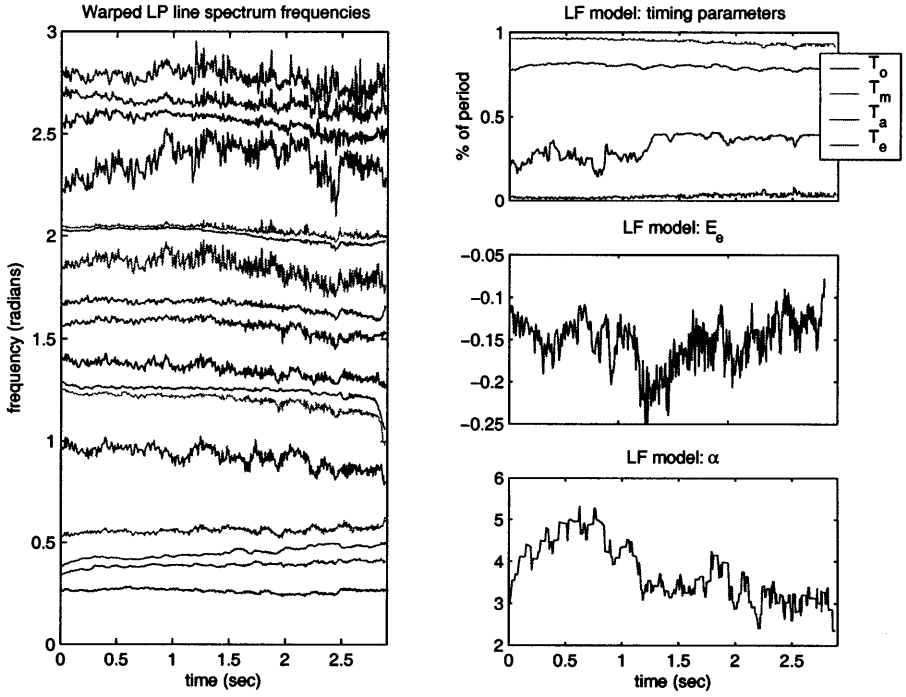


Figure 4-8: The parameter observations from a scale on the vowel [e]. On the left are line spectrum frequencies derived from the warped LP coefficients. On the right are the LF model parameters.

The state path is calculated by finding the most likely (least cost) succession of states, given the model probabilities **A**, **B**, and π. An efficient algorithm for performing this minimization is the *Viterbi* algorithm, detailed in Appendix A.2. An example state path for a 10-state HMM for a scale on the vowel [e] is shown in Figure 4-7.

The state path can then be used to estimate the parameter trajectories from the state means. The parameter observation sequence (line spectrum frequencies and LF model paramters) for the signal of Figure 4-7 is shown in Figure 4-8. Figure 4-9 shows the estimated parameter trajectories reconstructed via the state path.

It is apparent that the reconstructed parameter trajectories follow the general motion of the original parameter sequence. For sound re-synthesis, smoothing of the quantized parameter trajectories via a low-pass filter is generally required to avoid abrupt discontinuities in the output signal.

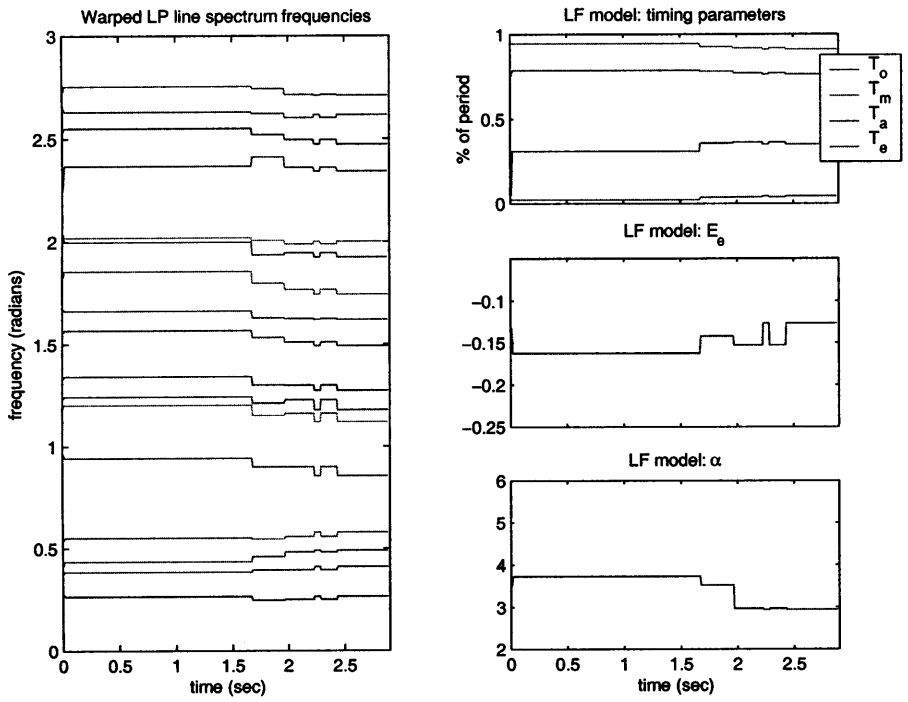


Figure 4-9: The reconstructed parameter trajectories from the observation sequence of Figure 4-8.

4.2.4 Evaluating model likelihood

Evaluating model likelihood is useful when attempting to fit an observation sequence to one of several candidate models. This is the method used most often in classification problems involving HMMs, such as speech recognition (finding the best phoneme, multi-phone, word, or language model to fit to the observed data). The model that best fits the observed sequence will be the model with the highest log-likelihood. This value is calculated efficiently using the *forward-backward* algorithm (Appendix A.3).

In the task of singer identification, the likelihoods of parameter observation sequences from one phoneme are calculated against the trained HMMs of that phoneme from the candidate singers. The observations will have the greatest likelihood for the model of the singer whose states best encapsulate the distributions of the observed parameters, which hopefully will be that of the same singer who produced the observed data.

4.3 Summary

This chapter has presented the components of the singing voice analysis framework designed to model the time-varying qualities of the voice. The general structure for time-varying analysis is provided by the linguistic decomposition of singing into ordered phonemes. Accordingly, this chapter describes a proposed method for phonetic segmentation to divide the signal into areas for more detailed analysis.

The dynamic analysis is concentrated on the modeling of vowels, which represent the vast majority of singing. A system built upon hidden Markov models is used to describe the wide range of parameter (and corresponding acoustic) variations for each vowel that occur over time in singing. The model states reflect a set of physically-derived parameters at an instant in time. Because of the wide variance of the physical features involved in singing, numerous states are needed to adequately represent the parameter space. The key feature of the HMM, however, is its ability to document the state changes over time via the state path.

The analysis framework presented thus far in this dissertation incorporates the derivation of physically-motivated features and the modeling of their dynamic behavior, both of which have been hypothesized as being necessary to represent singer identity. The next chapter discusses a series of experiments designed to evaluate whether the analysis system, which is designed around this set of features, is able to model and preserve the distinctive characteristics of an individual voice.

CHAPTER FIVE

Experiments

The previous two chapters have detailed the components of the singing voice analysis/synthesis framework at the core of this dissertation. This chapter explores three different applications of the framework, and details the experiments used to evaluate the performance of the framework in these tasks. As mentioned in the introduction, the advantage of an analysis/synthesis system is that the synthesized sounds can be used in human listening experiments to validate the system perceptually, which is arguably the most important metric. Thus, for each of the objective experiments performed with the system, a corresponding listening experiment is presented to ascertain whether the perceived results correlate with the objective results.

The three applications investigated in this chapter are singer identification, singing voice coding, and singing voice transformation. Clearly, each of these applications centers on the concept of singer identity at the core of this framework and this dissertation. One way or another, these experiments will provide significant evidence as to the ability of the analysis/synthesis framework to model and preserve the perceptual features of vocal identity. The sound examples used in these experiments are available at http://sound.media.mit.edu/~moo/thesis.

5.1 Singer Identification

The first application is an analysis-only system for automatically identifying the singer whose voice is captured in a sound recording. The experiment is limited to high-quality voice-only recordings (without any accompanying instruments), which rules out commercial recordings. Even the few *a cappella* commercial recordings available usually have some amount of sound effects (such as reverberation) applied, which have been known to enhance the perceived quality of the voice. These effects, however, are difficult to account for in a general manner and are not at all accounted for in this analysis framework. Therefore, the data set for all of the experiments is limited to sound recordings made specifically for this research. Recordings were made in a sound-proof recording studio with the amplified microphone output recorded directly to disk at a sampling rate of 48 kHz. All data was later downsampled to a sampling rate of 16 kHz to reduce the computational requirements.

The data set consisted of recordings from 4 conservatory-trained classical singers (two sopranos, one tenor, and one bass-baritone). Each singer performed a variety of vocal exercises (such as scales and arpeggios) approximately 5-10 seconds in length emphasizing the 5 major vowels ([i], [e], [a], [o], and [u]) in addition to one entire piece from the classical repertoire. The vowel exercises comprised the training set, and source-filter parameters were extracted for each recording using the parameter estimation system of Chapter 3. The recordings were further segmented by vowel using the phonetic segmentation system, which was provided a phonetic transcript of each exercise. The source-filter parameter estimates from each of the vowel segments formed an observation sequence used to train the vowel HMMs specific to each singer. The observation vectors consisted of the line spectral frequency values (describing the vocal tract filter) and the LF timing and shape parameters. The parameters for the stochastic modeling of the source residual were not used in this experiment. Models of each of the primary vowels were trained for each of the four singers.

Passages from the recorded classical pieces were used as the testing data to evaluate the performance of the system. As before, source-filter parameters were estimated for the passages, and vowel segments were identified and extracted with the aid of a phonetic transcript. The relevant source-filter parameters from those segments were used as test observation sequences. The likelihoods of each test sequence were calculated for each singer's corresponding vowel model (as provided by the phonetic transcript) using the forward algorithm detailed in Appendix A.3. The singer corresponding to the model with the greatest likelihood was labeled the singer of the segment, while the singer with the most vowel HMM matches in the overall passage was deemed to be the source performer. The testing set consisted of five 5-7 second passages from each singer.

The singer identification system performed with an overall accuracy of >90% when operating over entire excerpts (only one of the 12 excerpts was mis-identified), as shown in Table 5.1. Though this result is for an admittedly small data set, the result is quite promising. There were a total of 69 vowel segments in all of the excerpts, and 32 segments (~46%) were identified correctly by having the highest vowel HMM log-likelihood. A confusion matrix showing the identification results for the individual vowel segments is shown in Table 5.2.

Table 5.1: Confusion results of singer identification experiment over whole phrases.

Singer No.	1	2	3	4
1 (bass)	1.00			
2 (tenor)		1.00		
3 (soprano)			0.75	0.25
4 (soprano)				1.0

As can be seen from the table, most of the confusion occurred between the two soprano singers, which is to be expected because of the similarities of their voice classification. This is also an initial indication that the analysis/synthesis framework could possibly be

Table 5.2: Confusion results of singer identification experiment over individual vowels.

Singer No.	1	2	3	4
1 (bass)	0.53	0.10	0.05	0.32
2 (tenor)	0.35	0.50	0.00	0.15
3 (soprano)	0.21	0.00	0.42	0.37
4 (soprano)	0.17	0.17	0.33	0.33

used as a metric for voice similarity, since confusions tend to occur more often between similar voices than dissimilar voices.

5.1.1 Perceptual experiment

A corresponding perceptual experiment was performed using the same data set used in the singer identification system in order to compare the performance of the system to human performance. The testing conditions were kept as close as possible to the conditions used to evaluate the automatic system. A series of "test" samples, the same passages used in the machine listening experiment, were presented to each listener along with two reference samples for each singer to serve as "training" data. The training samples consisted of two vocal exercise passages from each singer, and were simply labeled "Singer 1-4 A/B". Both training and testing samples were presented simultaneously in a simple user interface, though the training samples remained constant throughout the experiment. For each presentation of test data, subjects were asked to select which singer (1-4) they thought was the singer of the test passage. Listeners were allowed to audition the training samples and the test sample as many times as they wished before making a selection. Nine subjects with varying music backgrounds participated in this experiment as well as the others detailed in this chapter.

Table 5.3: Confusion results of singer identification perceptual experiment.

Singer No.	1	2	3	4
1 (bass)	0.96	0.04		
2 (tenor)	0.04	0.96		
3 (soprano)			0.33	0.67
4 (soprano)			0.56	0.44

The overall accuracy across all of the participants was 65%. This relatively low number is due almost entirely to confusions between the two soprano singers (Table 5.3). Grouping the sopranos together results in a performance of 98%. Discrimination between the bass and tenor was high (96%) and male/female discrimination was 100%. The results from this limited experiment are consistent with the basic categorization due to voice part. Identification between different voice parts is very accurate, but there is a great deal of confusion within a category. This data set is far too limited to draw any more general conclusions. We do see, however, that the performance of the

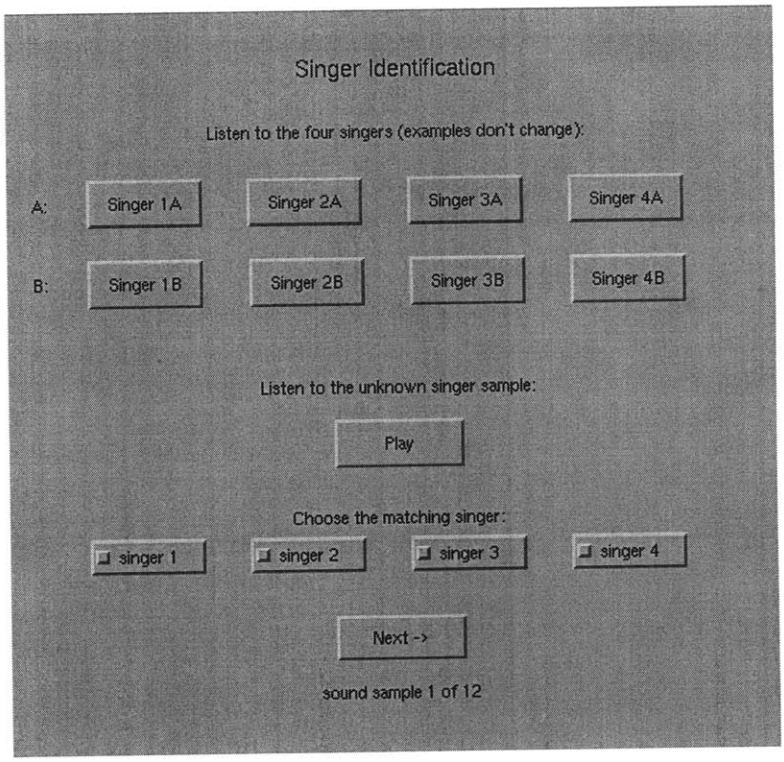


Figure 5-1: Interface to singer identification perceptual experiment.

machine listening system is comparable to human performance on the same data set. In fact, the system exceeds human performance in some respects, particularly in its performance in discriminating between the two soprano voices. The results of these two experiments provide some evidence that the analysis/synthesis is able to capture some features relating to vocal identity.

5.2 Singing Voice Coding

The second experiment investigates the application of the analysis/synthesis framework as a low-bitrate encoder/decoder (codec) for singing voice transmission. As detailed in Chapter 2, there are many well-established codecs for speech most of which are based upon certain assumptions about the signal (such as ratio of voicing) making them less applicable to singing. Additionally, all low-bitrate ($<$ 12kbps) speech codecs limit the frequency bandwidth to 4 kHz, which is suitable for speech intelligibility but leaves much to be desired when applied to the singing voice.

The key aspect of the analysis/synthesis framework in this application is the HMM state path representation. Each point of the path basically represents an entire period of singing data with a single state number, meaning that a great deal of data compression

is possible. Of course, the sound quality of the system depends upon how well the states are able to model the acoustic variation of the voice.

This experiment focuses entirely on the vowel-only passages that formed the training data set for the previous experiment. Again the vowel segments are analyzed to form the observation sequences used to train the analysis HMMs for the system. In this experiment, however, the vowel exercise segments also form the test data set for the analysis/synthesis. The overlap of the training and testing sets here doesn't seem all that unreasonable since it is not intended to be a real-time coding scheme. Given that a sound to be transmitted can be analyzed in advance, it seems appropriate that the results of the analysis should be able to be incorporated into the signal model.

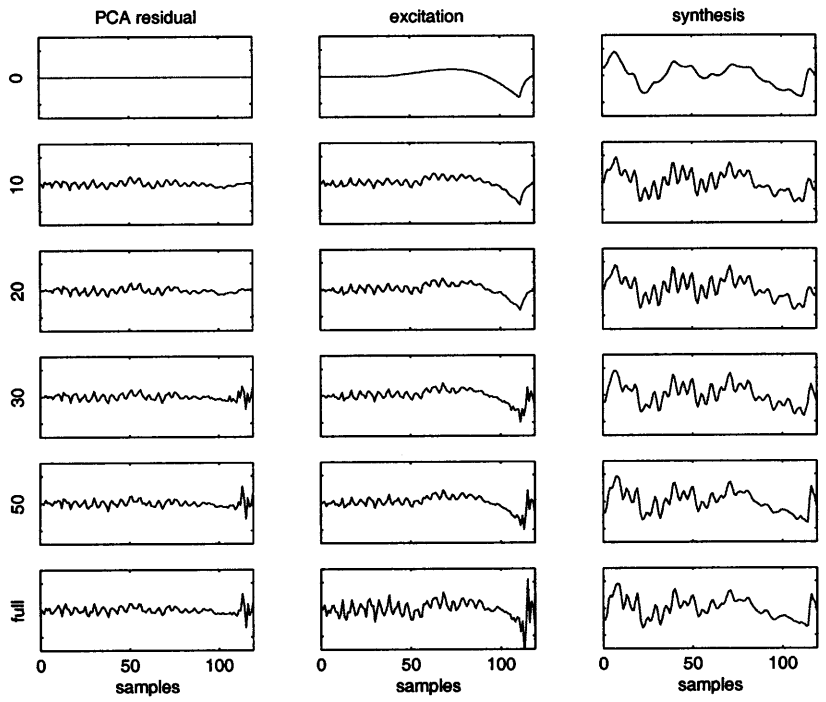


Figure 5-2: Re-synthesis of one period using different numbers of residual principle components. Left: modeled residual. Center: reconstructed excitation. Right: resulting synthesis.

Each segment of the data set is analyzed via the appropriate vowel HMM of the singer, resulting in a state path for the segment. We also record the stochastic parameters for each analysis period. In combination, the state path and the stochastic information are used to re-create the glottal derivative waveform, the vocal tract filter, and the singing waveform itself. Glottal residual reconstruction and the resulting synthesis is shown in Figure 5-2 for varying numbers of residual components.

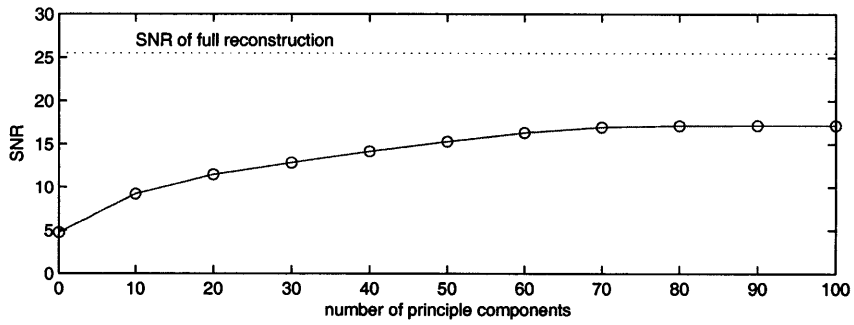


Figure 5-3: Mean SNR across nine sound samples using different numbers of residual principle components.

Since the analysis/re-synthesis occurs on a period-by-period basis, we can directly compare the original and re-synthesized output waveforms. The *signal-to-noise-ratio* (SNR) is an objective measure that describes the difference between two waveforms. If $s[n]$ is the original signal and $\hat{s}[n]$ is the synthesis, the SNR can be calculated by the following equation:

$$SNR = 10 \log \left(\frac{\sum \hat{s}^2[n]}{\sum (s[n] - \hat{s}[n])^2} \right) \tag{5.1}$$

The maximum SNR is theoretically infinite (if $\hat{s}[n] = s[n]$ exactly), but in practice quantization of the signal values and filter coefficients inject some noise into the system. The SNR values using varying numbers of principle components (from 0 to 100) for glottal residual reconstruction are presented in Table 5.4 for nine of the re-synthesized vowel passages. The codebook used was trained using the glottal residuals of all of the nine source signals, consisting of 512 vectors, each of length 513 (based on FFTs of length 1024). The SNR value of the full reconstruction (using the exact glottal residual) is provided in the final column of the table to give an indication of the minimum amount of noise in the analysis/re-synthesis procedure. The SNR values averaged across the nine sound examples is plotted against the number of codebook components used for reconstruction in Figure 5-3.

5.2.1 Listening experiment

While there are many objective measures with which to assess the performance of an encoding/decoding scheme, none is truly able to specify the perceived quality or loss of quality. Some encoding schemes can produce signals with very high signal-to-noise ratios, the resulting signals may still contain considerable audible distortion and perceived as having low sound quality. The inverse situation can also occur (a signal with

Table 5.4: SNR values for nine re-synthesized vowel exercises using varying numbers of residual principle components.

no.	0	10	20	30	40	50	60	70	80	90	100	Full
1	5.32	7.58	9.54	11.13	12.56	14.00	14.91	15.41	15.51	15.50	15.50	18.81
2	5.72	9.18	11.21	12.64	14.50	16.07	17.15	17.36	17.43	17.44	17.44	26.89
3	3.23	10.48	11.38	11.98	12.78	13.26	13.64	13.85	13.88	13.88	13.88	25.42
4	5.40	10.62	14.04	15.17	15.88	16.50	16.63	16.87	16.98	16.98	16.98	24.58
5	5.78	9.41	11.11	12.37	13.37	14.19	15.23	15.78	15.92	15.92	15.92	17.29
6	4.88	7.20	10.32	11.26	12.67	14.00	15.26	16.52	17.11	17.41	17.41	31.04
7	3.89	9.46	10.66	12.88	14.91	16.88	19.04	20.03	20.33	20.34	20.34	30.10
8	3.62	9.47	13.10	13.91	14.40	14.87	15.13	15.32	15.30	15.30	15.30	24.58
9	4.96	9.51	12.02	14.16	16.13	17.84	19.89	21.39	21.63	21.63	21.63	30.85

a relatively low SNR value, but with high perceived quality). Therefore, a true assessment of the resulting sound quality of a codec can only be made through perceptual listening experiments.

Re-synthesized signals from the state path representation were used as the basis of a listening experiment in which listeners were asked to judge the sound quality of the compressed signal relative to the original signal. Listeners were presented with the original and a coded version of each of the nine sound samples and asked to indicate which sample sounded better. The coded examples were re-synthesized using varying numbers of residual codebook components (from 0 to 100). Subjects were also asked to rate the sound quality of the example they indicated as being lesser quality on a scale from 1 to 5 (with larger values representing higher quality). The interface for the experiment is shown in Figure 5-4.

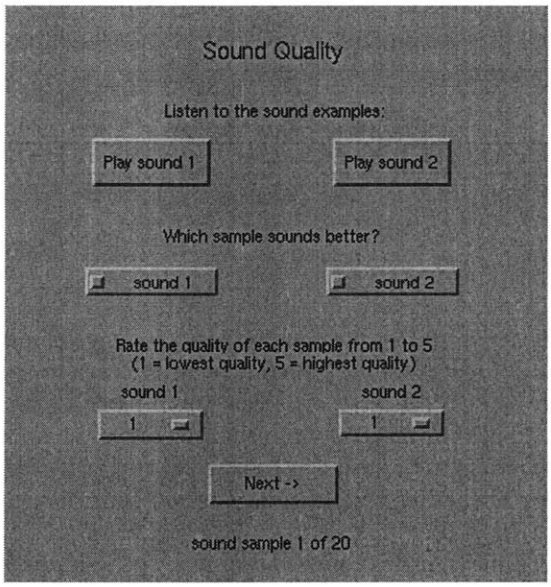


Figure 5-4: Interface to sound quality perceptual experiment.

The results of the listening experiment demonstrate a correlation between the quality ratings and the number of codebook components used in re-synthesis. Though the average quality rating for the re-synthesized sounds is generally lower than the average quality of the original source sounds for most component levels (Figure 5-5), there were many instances when subjects selected the re-synthesized sound as preferable to the original sound (Table 5.5 and Figure 5-6).

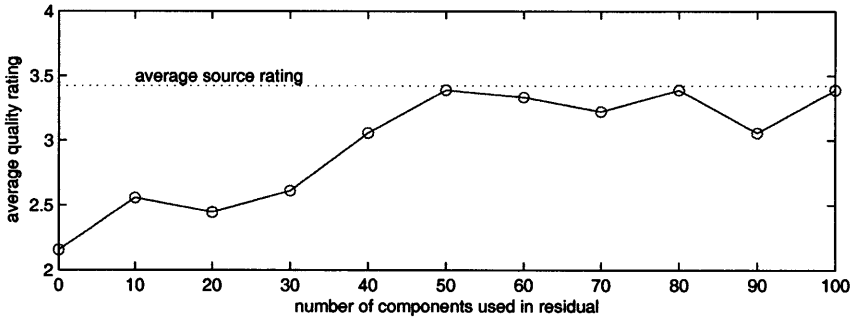


Figure 5-5: Average quality ratings for re-synthesized examples using varying number of codebook components. The dashed line indicates the average quality rating for the original source sounds.

Table 5.5: Average listener preference of original vs. re-synthesized samples for varying codebook sizes.

	0	10	20	30	40	50	60	70	80	90	100
original	**0.71**	0.43	**1**	**0.71**	**0.71**	0.43	0.29	**0.57**	**0.71**	**0.71**	0.43
re-synthesis	0.29	**0.57**	0	0.29	0.29	**0.57**	**0.71**	0.43	0.29	0.29	**0.57**

5.2.2 Coding efficiency

Previous work has demonstrated the compression advantage of encoding vowels using pre-stored (static) templates [32]. Results with the current system demonstrate a higher quality re-synthesis using the 10-state vowel models than with static vowel templates. Disregarding the overhead of transmitting the vowel models and codebook, each period can be reconstructed using just a vowel identifier, the HMM state number, the amplitude, the pitch period, and the codebook component weights.

Since the vowel identifier remains constant over many periods, the vowel information can be efficiently encoded using run-length encoding [25] making its contribution to the overall bitrate per period negligible (distinguishing between 5 vowel models requires <3 bits per $\sim$100s of pitch periods or $\sim$0.01 bits per period). The same holds true for the state number, though to a slightly lesser degree. The average duration of a

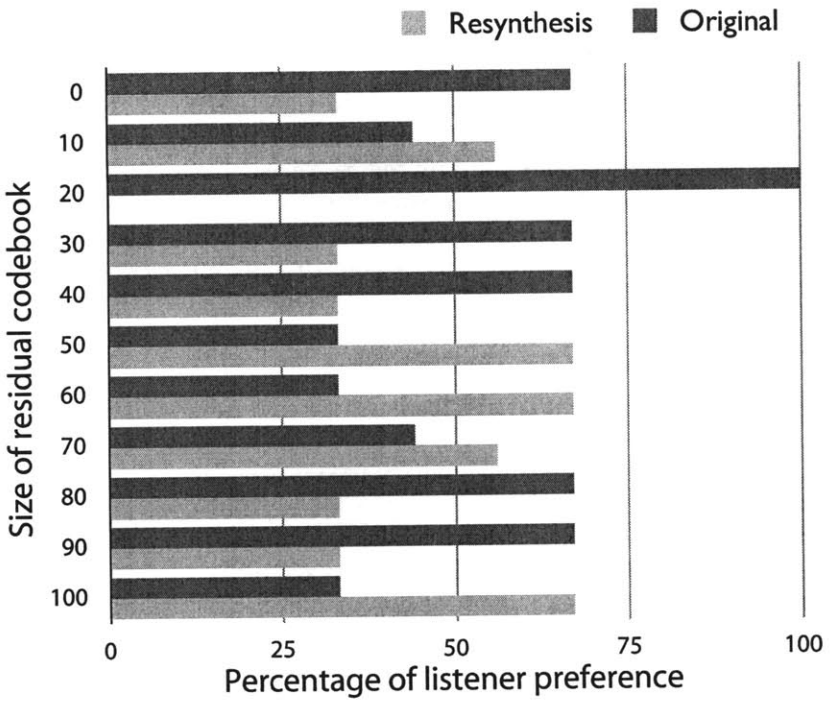


Figure 5-6: Average listener preference of original vs. re-synthesized samples for varying codebook sizes.

path segment is $\sim$ 50 periods, so a 10-state model requires 4 bits per $\sim$50 pitch periods or $<$0.1 bits per period.

Therefore, the primary sources of bit usage are the period, amplitude, and the codebook component weights. The period duration changes very little between adjacent periods and can be encoded as a differential using $<$2 bits per period. The amplitude can be efficiently encoded in 8 bits using a non-uniform quantization such as μ-law encoding. Additionally, the amplitude varies slowly compared to the period and can be downsampled by a factor of 10, reducing the bit usage to $<$1 bit per period.

Given an average pitch period of 70 samples (at a sampling rate of 16 kHz) with 16-bit samples (1120 bits per period), we could achieve a possible compression of $>$250:1 during vowel segments ($\sim$4 bits per period) without the use of a residual codebook. Of course, adding any number of residual codebook components will require additional bits to specify the weight of each component. By distributing bits efficiently (more bits for the low-numbered eigenvectors of the codebook), an average of 3 bits per code vector is not unreasonable. So a 10-vector codebook would require $\sim$30 additional bits per period, resulting in a compression of $>$30:1 (still significantly better than the $\sim$10:1 compression achieved with common speech coders). The framework could conceivably be extended to include models for all voiced phonemes. Of course, significant

modifications in the parameter estimation step would be needed to accommodate unvoiced phonemes.

In lieu of consonant phoneme models, the analysis/synthesis framework could be used for vowel-only encoding in conjunction with traditional coding schemes for the other phonemes. One such possibility, a hybrid coding scheme that uses prior knowledge of the musical score to switch between pre-defined templates for vowels and standard LPC for other phonemes, is presented in Appendix B to demonstrate how a system for vowel-only encoding, similar in terms of limitations to the framework presented here, can still be used advantageously on all singing voice sources.

5.3 Voice Transformation

Voice transformation is a nebulous term that is used to describe general modifications applied to the voice in order to enact a change in voice quality (such as gender or age alteration) [88]. A good deal of research has been performed in this general area, including work specifically on the glottal source [12] and the vocal tract filter [13]. In this dissertation, however, voice transformation is more specifically defined as the alteration of one voice so that it is perceived to have characteristics of another specific target voice. Not surprisingly, this sort of transformation has proven to be an elusive goal. The method proposed here attempts to modify both glottal source and vocal tract filter parameters estimated from one singer to reflect those of another, but the form this mapping should take is not obvious. Each singer has his or her own unique dependencies between the source and filter parameters, which are also affected by fundamental frequency and amplitude, indicating that a direct linear mapping would be inappropriate.

The HMM states, however, provide common point of reference between differing voice models. In mapping from the state model of one singer to another, the prosody of the transformed result would still be determined by the motion of the input state path. In re-synthesizing one singer's state path using another singer's state model, it is hoped that the source-filter parameter dependencies encapsulated in the states of the target singer will evoke the vocal quality of that singer in the sound output.

Because the states themselves represent clusters of significant concentration in each singer's parameter space, they may be related between singers in terms of likelihood of occurrence. Thus, one possible mapping is to reorder each HMM's state labels according to their frequency of occurrence (state 1 becomes the most often occurring state, followed by state 2, etc.), enforcing a semi-statistical relationship between the state numbers (Figure 5-7). From this mapping, the revised state path from one singer can be used to drive another singer's HMM for re-synthesis.

This mapping proved to be more successful than the histogram-based map in transforming the vocal quality of one singer to another, but it is difficult to quantify the resulting sound objectively. An SNR calculation for this transformation would be meaningless, since the output waveform is clearly intended to be much different from the

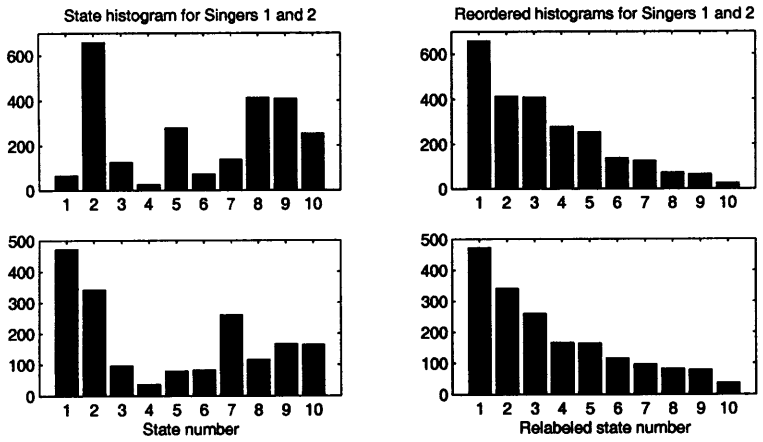


Figure 5-7: HMM state histogram for two singers for vowel [e].

Table 5.6: Distance matrix of state glottal parameters and mapping between states of vowel [e] of two singers.

state no.	1	2	3	4	5	6	7	8	9	10
1	11.60	16.33	14.19	11.39	13.21	**13.17**	13.37	11.64	11.51	14.07
2	11.20	15.71	13.67	10.99	12.71	12.70	12.89	**11.22**	11.11	13.55
3	16.57	23.59	**20.39**	16.26	19.00	18.88	19.19	16.65	16.44	20.22
4	13.49	19.18	16.58	13.25	**15.51**	15.37	15.61	13.57	13.39	16.46
5	16.82	**23.99**	20.72	16.52	19.33	19.18	19.50	16.92	16.69	20.55
6	14.61	20.68	17.93	14.34	16.67	16.62	**16.88**	14.66	14.49	17.77
7	7.51	10.44	9.10	**7.38**	8.54	8.48	8.60	7.53	7.46	9.04
8	9.91	13.92	12.10	9.73	11.28	11.23	11.40	9.94	**9.83**	12.00
9	**10.45**	14.65	12.75	10.26	11.87	11.84	12.02	10.47	10.37	12.64
10	16.01	22.85	19.72	15.72	18.43	18.26	18.56	16.11	15.89	**19.57**

input. The evaluation of the success of this transformation is therefore left to a perceptual experiment.

The stochastic codebooks were not used in the voice transformation experiments because there is no clear method of mapping from one codebook to another. The calculation of the glottal residual (which determines component selection and weight determination) requires a reference for the target sound which does not exist in this case. Not surprisingly, simply applying the code vector weights calculated from the source singer's codebook to the target singer's codebook resulted in a greatly distorted sound. A different parameterization is likely needed to provide a more suitable mapping of the stochastic excitation component.

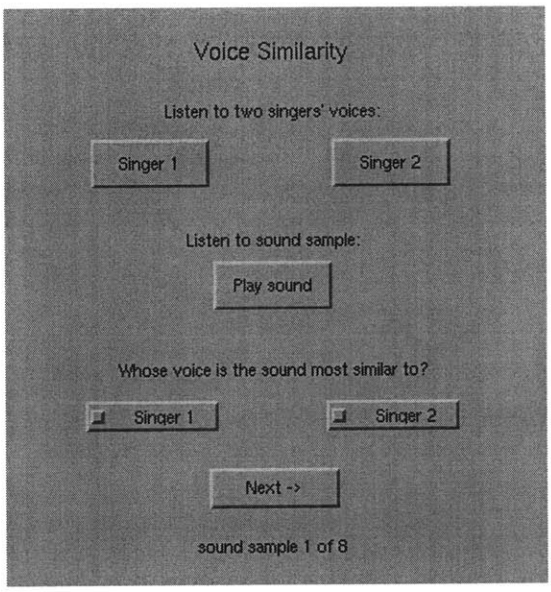


Figure 5-8: Interface to sound similarity perceptual experiment.

5.3.1 Listening Experiment

Judging the success of voice transformation is difficult to do quantitatively. We therefore turn to subjective judgments deduced from listening experiments. Listeners were presented with encoded samples from two different voices, and an additional sample which was a re-synthesized voice transformation from one singer to the other. Transformed examples were generated in both directions (singer A to singer B and singer B to singer A). The reference samples for the two singers were encoded without the use of the stochastic codebook, in order to achieve greater consistency in sound quality across the samples. Subjects were asked to simply judge whether they thought the re-synthesized sample sounded more similar to singer A or B. Participants were presented with 8 transformed sound examples. The interface for the experiment is shown in Figure 5-8.

The summary of results of the perception experiment averaged across listeners is given in Table 5.7 and Figure 5-9. While not all of the sound examples were convincing to listeners, most of were judged by some listeners to be most similar to the target singer, lending the transformation some perceptual validity.

Table 5.7: Average perceived judgments of voice transformation sound examples.

judgment	1	2	3	4	5	6	7	8
target:	0.5	0.17	**0.66**	**0.83**	**0.83**	**1**	**0.83**	0.33
source:	0.5	**0.83**	0.33	0.17	0.17	0	0.17	**0.66**

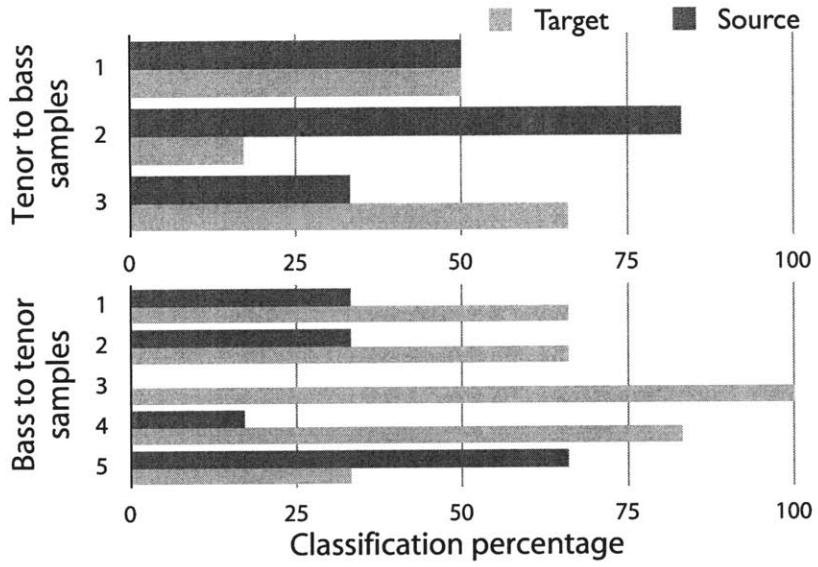


Figure 5-9: Average listener judgment of original vs. transformed samples.

5.4 Summary

This chapter has detailed the application of the analysis/synthesis framework to three different tasks: singing voice identification, singing voice coding, and singing voice transformation. The utility of the proposed framework to each of these applications has been validated to some degree through three corresponding perceptual listening experiments, demonstrating that the dynamic modeling of source-filter parameters that form the core of the system play a substantial role in the perception of singer identity. The concluding chapter will present some of the limitations of the analysis/synthesis framework in greater detail, and will propose some possible directions for further research..

CHAPTER SIX
Conclusions

In this dissertation I have presented a framework for analysis and re-synthesis of the singing voice, attempting to model the physical and expressive factors involved in the perception of singer identity. The model parameters are based on specific computationally derived parameters motivated by the physical features of the voice, which are subsequently modeled dynamically to represent the time-varying nature of the expressive qualities of the voice. The utility of the framework has been demonstrated in the applications of singing voice coding, singing voice identification, and singing voice transformation. These results have been validated through perceptual listening experiments involving re-synthesized sound examples generated using the analysis/synthesis framework. In this concluding chapter, I will briefly discuss some limitations of the presented framework and research questions that warrant further investigation.

6.1 Possible Refinements to the Framework

The joint parameter estimation procedure detailed in Chapter 3 introduces several sources of error. Quantizing the time-based parameters (period, glottal open quotient, glottal closure instant, LF-model timing parameters) to integer sample numbers introduces aliasing, which can cause audible distortion in re-synthesized sound samples. For long pitch periods, the error from quantization to integer values is proportionally small. For shorter pitch periods, however, the quantization noise becomes more of an issue. The current implementation deals with this problem by highly smoothing the estimated parameter values to reduce the amount of aliasing in the reconstructed signal.

Another approach in addressing this problem would be to simply use data sampled at a higher rate, further reducing the error between the time-quantized parameter values and the actual parameter values. The cost of this approach is the greater number of computations needed to perform the analysis for larger numbers of samples. Another problem with this approach is the likely necessity of higher vocal tract filter orders to adequately represent the spectral shape of the filter at higher bandwidths, meaning an increase in size of the parameter vector for each period. HMM training becomes increasingly difficult with longer parameter vectors as there is likely to be less correlation between successive frames.

One alternative would be to perform the parameter estimation at a lower sampling rate, while performing the reconstruction at a higher rate. This would require some method of estimating of the vocal tract filter response outside of the original bandwidth of the analysis sampling rate. This type of spectrum estimation has been developed in the domain of perceptual audio coding. Known as *spectral band replacement*, it has been incorporated into a revision of the MPEG-4 Advanced Audio Coding (AAC) standard [1]. It is possible that a similar technique could be applied to improve the sound quality of the singing voice analysis/synthesis framework.

Fractional delay methods [35] may also provide a way of reducing the distortion due to aliasing of the timing parameters. These techniques, however, in addition to requiring additional computation, may be incompatible with the formulation of the parameter estimation problem as a convex optimization. Fractional delays could be used in the already nonlinear estimation of the LF model parameters without ill-effect, but whether the results would exceed those of simple parameter smoothing is as of yet undetermined.

The stochastic codebook approach is convenient for analysis/synthesis applications, such as coding, where a target waveform is available. This representation of the residual, however, is not parametric. It is possible that an analysis of the codebook weights could yield a more meaningful parameterization so that the stochastic component could be incorporated into the voice transformation procedure.

6.2 Fundamental System Limitations

Throughout this dissertation, I have focused exclusively on vowels because they are the predominant sounds of classical singing. The framework presented, however, could be extended relatively easily to include all voiced consonants. Accounting for unvoiced sounds, however, would require an alternative parameterization of the source model, since the KLGLOTT88 and LF models would no longer be applicable. Additionally it would be necessary to segregate voiced sounds from non-voiced sounds. While much research has been performed on systems for voicing detection in speech (see [78] for a comprehensive list), this topic is beyond the scope of this dissertation.

All systems for singing voice analysis have difficulty dealing with high-pitched singing, such as soprano voices. As mentioned in Section 3.1.4, it is difficult to obtain vocal tract filter estimates when the pitch period is much less than the natural duration of the vocal tract impulse response. Even using multiple periods oftentimes doesn't result in accurate vocal tract estimates because the overlapped responses obscure the true response. Alternative methods may be more accurate in deconvolving the source and filter functions at high fundamental frequencies.

A glaring limitation of the system is the requirement of a phoneme or word transcript of the singing passages in order to perform the phonetic segmentation prior to model training. Although many systems exist for automatic speech recognition (ASR), they perform poorly on singing data because of the drastic differences in rhythm and

prosody between speech and singing. It may be possible to train an ASR system entirely on singing voice data, which could provide automatic lyric transcription, but a major hurdle exists in collecting and labeling the vast amount of training data that would be required.

Similarly, another potential deficiency of the analysis/system system is that it requires a great deal of training data in order to build accurate HMMs for each singer. Since the parameter estimation relies on very clean solo voice recordings, it is difficult to build models for a large number of singers. There have been some attempts to extract voice features from commercial recordings, such as [33] in which a system is presented for performing singer identification from commercial music recordings by first attempting to isolate regions of singing. It would be difficult to use the framework presented in this dissertation on such sound mixtures since it would be required to account for external factors such as background instruments and audio effects, which are not accounted for by the underlying source and filter models.

6.3 Directions for Further Research

The extension of the framework to include the other voiced phonemes raises the issue of the phoneme-to-phoneme transitions. The phonetic segmentation used in this dissertation has mostly been for analytical convenience, providing a method for dividing the problem into semantically meaningful sub-problems. However, the boundaries between phonemes are not well-defined and warrant further study.

It is possible that the phonetic segmentation could be improved by building a classifier upon the source-filter features as opposed to MFCCs. The phoneme detection could be combined with a unidirectional HMM phoneme model to perform the segmentation. Such a system would again require a great deal of singing voice training data, which is not readily available.

This dissertation has focused entirely on analysis/synthesis and has not addressed how to synthetically generate high-quality sounds from purely semantic representations (such as a score). The generation of realistic time-varying model parameters has been studied, primarily using rule-based systems [87]. It would be interesting to see whether such systems could be applied to create model parameters evocative of the style of a particular singer.

Another interesting investigation would be to determine whether the source-filter parameterization itself could be further parameterized. The different vowels have been considered independently, but there is likely significant overlap between the different models. Titze [88] has proposed that the various vowels are simply transformations of a base neutral configuration of the vocal tract, corresponding to the schwa vowel [ə]. One could consider a parameterization of the source-filter framework along those lines, using a long-term average [14] as the base representation, parameterized using principle components analysis.

The source data for this dissertation was collected from unaccompanied singers recorded in a fairly acoustically absorptive room. Most singing performances, however, occur under a vastly different set of circumstances (e.g. other instruments, more acoustically reflective surroundings, different physiological states), which may result in different behavior and thus different parameter estimates. In speech, the alteration of one's voice due to noise is known as the *Lombard effect* [41]. It would be possible to study the corresponding changes in singing parameters by altering the recording conditions. For example, singers could be asked to sing along with accompaniment in their headphones. Artificial reverberation of their singing could be added to their headphones as well.

In this framework, the expressive characteristics of a singer are represented solely by the time-varying motion of the model parameters, but there are clearly more features of expression than are depicted here. It would be interesting to couple the expressive parameters of the singing voice framework with biometric sensor data used to study affect [60], for both singers and listeners.

6.4 Concluding Remarks

In this dissertation, I have attempted to address a question that does not have a consistent answer: What defines the unique sound of a singer? The limited success of the framework I have proposed in addressing this question is equally difficult to quantify. On the one hand, with a very limited set of training data, the system has proven to be fairly adept at drawing distinctions between singers and capturing some of the features of voice identity. On the other hand, the limitations of the training data make it impossible to generalize these results to the greater population of singing voices, and I do not claim that the analysis/synthesis framework is able to do so yet.

The greatest advances in the computational understanding of audio have come using techniques that are primarily data driven. Correspondingly, the most broadly successful applications are ones that have been trained with the greatest variance of training data. In particular, speech-related applications have benefited from a tremendous amount of meticulously annotated training data. Such a corpus is currently not available for machine listening applications focusing solely on singing voice. While this limitation has driven much ingenuity, most successes have been limited in scope. The wider dissemination of musical source materials and recordings will hopefully lead to even greater successes in this area of research.

APPENDIX A

HMM Tutorial

This appendix is intended as an introduction to solutions to the three primary problems of Hidden Markov Models: model training, highest likelihood path determination, and model likelihood evaluation. For more details, the reader is encouraged to consult one of the many papers textbooks on the subject, such as [65].

A.1 HMM training

The training of the model parameters ($\mathbf{A}, \mathbf{B}$, and π) is difficult to optimize globally. Common HMM training algorithms instead focus on iterative estimation of maximum likelihood (ML) model parameters, given an observation sequence. That is,

$$\max_{\lambda} P(\mathbf{O}|\lambda), \quad \text{where} \tag{A.1}$$

$$\mathbf{O} = \begin{bmatrix} \mathbf{o}_1 & \mathbf{o}_2 & \cdots & \mathbf{o}_T \end{bmatrix}. \tag{A.2}$$

The ML model parameters are only guaranteed to represent a local maximum, implying that parameter initialization is quite important. The most common of the iterative ML techniques is the Baum-Welch algorithm, also known as expectation-maximization (EM) [64]. The EM algorithm is a well-known technique also used for function fitting and data clustering. In fact, the HMM training problem can be viewed as primarily fitting the observation densities to the data clusters. The training algorithm is now presented in summary.

In order to maximize the likelihood of the training observation sequence given the model, it is convenient to define the probabilities $\gamma_t(i)$ and $\xi_t(i,j)$:

$$\gamma_t(i) = P(q_t = i|\mathbf{O}, \lambda) \tag{A.3}$$

$$\xi_t(i,j) = P(q_t = i, q_{t+1} = j|\mathbf{O}, \lambda). \tag{A.4}$$

Given the observations and the model, $\gamma_t(i)$ is the probability of state i at time t, and $\xi_t(i)$ is the joint probability of state i at time t and state j at time $t+1$. It is worth mentioning that $\gamma_t(i)$ can be calculated by summing $\xi_t(i,j)$ over all possible states j:

$$\gamma_t(i) = \sum_{j=1}^{N} \xi_t(i,j). \tag{A.5}$$

The summation of $\xi_t(i,j)$ over all possible transition times of the observation sequence $(t = 1, \ldots, T-1)$ gives us the expected number of transitions from state i to state j. Similarly, summing $\gamma_t(i)$ over time (excluding the final observation $t = T$) produces the expected number of transitions from state i for the sequence, while including time $t = T$ in the summation of $\gamma_t(i)$ simply reveals the expected number of occurrences of state i. These expected values provide a logical update rule for the model parameters.

The priors π_i should reflect the expected probability of state i at time $t = 1$. The updated parameter π_i' is then estimated simply as:

$$\pi_i' = \gamma_1(i) \tag{A.6}$$

The state-to-state transition probabilities a_{ij} are updated as the expected number of transitions from state i to state j, normalized by the expected number of transitions from state i. These expected values are the result of the summations mentioned above:

$$a_{ij}' = \frac{\sum_{t=1}^{T-1} \xi_t(i,j)}{\sum_{t=1}^{T-1} \gamma_t(i)}. \tag{A.7}$$

Re-estimation of the observation distribution parameters is a bit more complex. The mean and variance of the Gaussian model densities for each state are clearly related to the expected occurrences of that state. In cases where the observation density is modeled using a single Gaussian ($K = 1$ in Eq. 4.5), the updated mean and covariance parameters are calculated directly from the observation sequence and are simply scaled by the expected state frequencies:

$$\mu_j' = \frac{\sum_{t=1}^{T} \gamma_t(j) \cdot \mathbf{o}_t}{\sum_{t=1}^{T} \gamma_t(j)}, \tag{A.8}$$

$$\mathbf{U}_j' = \frac{\sum_{t=1}^{T} \gamma_t(j) \cdot (\mathbf{o}_t - \mu_j)(\mathbf{o}_t - \mu_j)^T}{\sum_{t=1}^{T} \gamma_t(j)}. \tag{A.9}$$

When more than one Gaussian component is used to model the observation distribution, the state frequencies are altered to include the probability of the observation being accounted for by a particular mixture component. This alteration is relatively straightforward, but will not be detailed here.

Iterating Equations (A.6) - (A.9) improves the HMM parameter estimates until the maximum likelihood parameters for the observation sequence are found. Calculating each iteration requires calculating $\gamma_t(i)$ and $\xi_t(i,j)$, which are expensive to calculate directly from their definitions (Eqns. A.3 and A.4). The computational requirements can be greatly reduced by taking advantage of the following derivation beginning with Bayes' rule.

$$\begin{aligned}\xi_t(i,j) &= \frac{P(q_t = i, q_{t+1} = j, \mathbf{O}|\lambda)}{P(\mathbf{O}|\lambda)} \\ &= \frac{P(\left[\mathbf{o}_1 \quad \cdots \quad \mathbf{o}_t\right], q_t = i|\lambda) P(\mathbf{o}_{t+1}, q_{t+1} = j|q_t = i, \lambda) P(\left[\mathbf{o}_{t+2} \quad \cdots \quad \mathbf{o}_T\right]|q_{t+1} = j, \lambda)}{P(\mathbf{O}|\lambda)}\end{aligned} \tag{A.10}$$

At this point it is helpful to separate out the following probabilities:

$$\alpha_t(i) = P(\left[\mathbf{o}_1 \quad \mathbf{o}_2 \quad \cdots \quad \mathbf{o}_t\right], q_t = i|\lambda) \tag{A.11}$$

$$\beta_t(i) = P(\left[\mathbf{o}_{t+1} \quad \cdots \quad \mathbf{o}_T\right]|q_t = i, \lambda). \tag{A.12}$$

$\alpha_t(i)$ and $\beta_t(i)$ are known as the *forward* and *backward* probabilities, respectively. These variables are particularly helpful because they can be calculated via induction:

$$\alpha_t(j) = \left[\sum_{i=0}^{N} \alpha_{t-1}(i) a_{ij}\right] b_j(\mathbf{o}_t), \quad 2 \leq t \leq T, \ 1 \leq j \leq N \tag{A.13}$$

$$\beta_t(i) = \sum_{j=1}^{N} a_{ij} b_j(\mathbf{o}_{t+1}) \beta_{t+1}(j), \quad T - 1 \geq t \geq 1, \ 1 \leq i \leq N. \tag{A.14}$$

The other term in the numerator of Equation (A.10) is simply

$$P(\mathbf{o}_{t+1}, q_{t+1} = j|q_t = i, \lambda) = a_{ij} b_j(\mathbf{o}_{t+1}). \tag{A.15}$$

The variable $\xi_t(i,j)$ can now be written in terms of $\alpha_t(i)$ and $\beta_t(i)$. Continuing with Equation (A.10):

$$\begin{aligned}\xi_t(i,j) &= \frac{\alpha_t(i)a_{ij}b_j(\mathbf{o}_{t+1})\beta_{t+1}(j)}{P(\mathbf{O}|\lambda)} \\ &= \frac{\alpha_t(i)a_{ij}b_j(\mathbf{o}_{t+1})\beta_{t+1}(j)}{\sum_{i=1}^{N}\sum_{j=1}^{N}\alpha_t(i)a_{ij}b_j(\mathbf{o}_{t+1})\beta_{t+1}(j)}\end{aligned} \tag{A.16}$$

The denominator of Equation (A.16) results from summing the numerator, originally $P(q_t = i, q_{t+1} = j, \mathbf{O}|\lambda)$, over all possible states i and j to obtain $P(\mathbf{O}|\lambda)$. $\gamma_t(i)$ can be computed from $\xi_t(i,j)$ using Equation (A.5), but can also be written in terms of $\alpha_t(i)$ and $\beta_t(i)$. Starting with Equation (A.3) we can derive the following:

$$\begin{aligned}\gamma_t(i) &= \frac{P(\mathbf{O}, q_t = i|\lambda)}{P(\mathbf{O}|\lambda)} \\ &= \frac{\alpha_t(i)\beta_t(i)}{\sum_{i=1}^{N}\alpha_t(i)\beta_t(i)}.\end{aligned} \tag{A.17}$$

The end result of this is an efficient means of computing probabilities $\gamma_t(i)$ and $\xi_t(i,j)$, which in turn are used to iteratively update the model parameters $\lambda' \sim (\mathbf{A}', \mathbf{B}', \pi')$ via EM. The forward and backward variables, $\alpha_t(i)$ and $\beta_t(i)$ will also be useful in the sections that follow for estimating the state paths and overall model likelihood.

The discussion of parameter estimation thus far has dealt with only a single observation sequence $\mathbf{O}$ for training. In practice, model training uses as many sequences as possible. Fortunately, multiple training sequences are easily incorporated into the iterative estimation if we assume that the sequences are independent. Because the parameter estimates are based on the weightings of the various state frequencies, $\gamma_t(i)$ and $\xi_t(i,j)$, the expected values calculated from different sequences can simply be summed together, which will optimize the parameters over all of the observation sequences.

A.2 Estimating state paths

Once a model is trained it can be used to determine the most likely sequence of states resulting from a sequence of observations. If $\mathbf{q} = [q_1 \quad q_2 \quad \cdots \quad q_T]$ is a state sequence (path) the optimal state path is defined as the $\mathbf{q}$ that maximizes the $P(\mathbf{q}|\mathbf{O}, \lambda)$, the *a posteriori* probability of the path, given the observations and the model. This optimization problem can be solved using the *Viterbi algorithm*, which is a dynamic programming method similar to the one used for phonetic segementation in Section 4.1.3.

First, we note that by Bayes' rule:

$$P(\mathbf{q}|\mathbf{O},\lambda) = \frac{P(\mathbf{q},\mathbf{O}|\lambda)}{P(\mathbf{O}|\lambda)}. \tag{A.18}$$

Since the denominator will be constant for all $\mathbf{q}$, maximizing the numerator will maximize the expression. Therefore, the algorithm maximizes $P(\mathbf{q},\mathbf{O}|\lambda)$, which reduces the computational requirements of the calculation. To implement the Viterbi algorithm, we define a function $\delta_t(j)$ as follows:

$$\delta_t(j) = \max_{q_1,\ldots,q_{t-1}} P(\left[q_1 \quad \cdots \& q_{t-1} \quad q_t = j\right]^T, \left[\mathbf{o}_1 \quad \cdots \quad \mathbf{o}_t\right]|\lambda). \tag{A.19}$$

$\delta_t(j)$ is the highest scoring (most probable) path ending at state j at time t. Conveniently, $\delta_t(j)$ can also be defined recursively from prior time steps.

$$\delta_t(j) = [\max_i \ \delta_{t-1}(i)a_{ij}]b_j(\mathbf{o}_{t-1}) \tag{A.20}$$

For an HMM with N states, the initial conditions are determined from the distribution of the priors.

$$\delta_1(i) = \pi_i b_i(\mathbf{o}_1) \tag{A.21}$$

The optimal state path can then be calculated recursively by iterating Equation (A.20) over $2 \leq t \leq T$. The path is determined by keeping track of the individual states that maximize $\delta_t(i)$ for each time step t.

A.3 Evaluating model likelihood

The likelihood of an observation sequence can also be evaluated against a trained model, $P(\mathbf{O}|\lambda)$, which facilitates the classification of observed data. In this application, an observation sequence is evaluated against several models to determine which one provides the best fit (highest probability). This is the technique used in speech recognition systems to identify phonemes and words, and it is the method used in this dissertation for the identification of singing voices.

Given a state path $\mathbf{q}$ and hidden Markov model λ, the probability of an observation sequence $\mathbf{O}$, assuming that the observations are independent, can be written as

$$P(\mathbf{O}|\mathbf{q},\lambda) = b_{q_1}(\mathbf{o}_1)b_{q_2}(\mathbf{o}_2)\cdots b_{q_T}(\mathbf{o}_T), \tag{A.22}$$

and the probability of the state path itself is

$$P(\mathbf{q}|\lambda) = \pi_{q_1} a_{q_1 q_2} a_{q_2 q_3} \cdots a_{q_{T-1} q_T}. \tag{A.23}$$

Again using Bayes' rule and summing over all possible state paths $\mathbf{q}$, we obtain the following expression of the desired observation likelihood:

$$\begin{aligned} P(\mathbf{O}|\lambda) &= \sum_{\forall \mathbf{q}} P(\mathbf{O}|\mathbf{q}, \lambda) P(\mathbf{q}|\lambda) \\ &= \sum_{q_1, \ldots, q_T} \pi_{q_1} b_{q_1}(\mathbf{o}_1) a_{q_1 q_2} b_{q_2}(\mathbf{o}_2) \cdots a_{q_{T-1} q_T} b_{q_T}(\mathbf{o}_T). \end{aligned} \tag{A.24}$$

Direct computation of this sum over all possible state paths of length T would be extremely expensive and time consuming. Predictably, there is a more efficient method for calculating the likelihood known as the *forward-backward* algorithm [64]. Recall that the forward and backward probabilities were defined in Equations (A.11) and (A.12).

Considering the forward probabilities, we know that $\alpha_t(i)$ can be determined from previous time values using induction (Eq. A.13). This again suggests a recursive technique for calculating $P(\mathbf{O}|\lambda)$. The initial probabilities for $t = 1$ are functions of the priors and the observation distributions.

$$\alpha_1(i) = \pi_i b_i(\mathbf{o}_1), \quad 1 \leq i \leq N \tag{A.25}$$

The probability of the each observation ending in a particular state j is determined by following the induction through to time T. The likelihood of the observation sequence is then calculated by summing the inducted probabilities over all possible final states:

$$P(\mathbf{O}|\lambda) = \sum_{i=1}^{N} \alpha_T(i) \tag{A.26}$$

This procedure reduces the overall number of calculations from an order of TN^T to TN^2. Alternatively, the induction can be performed backwards, starting at time T. For the backwards induction, the probabilities $\beta_t(i)$ are initialized to be equal to 1 and the induction occurs in reverse time from $t = T - 1, \ldots, 1$. The forward and backward algorithms produce the same likelihood calculation, the result being that a model corresponding well to the observation sequence will have a higher overall likelihood than a poorly matching model. This reveals a metric for comparing the relative performance of different HMMs.

APPENDIX B

Hybrid Coding Scheme

In this appendix, a technique is introduced for detecting vowel sounds in a singing-voice signal by anticipating the vowel, pitch, and duration indicated in the musical score. As the onset and release timings of vowels are detected, the LPC filter parameters during the vowel duration can be replaced by a single filter matched to the desired vowel. The resulting parameterization is more compact than LPC and still maintains comparable sound. The transitions between vowels (generally consonants) are parameterized using traditional LPC. The resulting technique is a hybrid voice coder that is both more efficient than LPC and in some ways more flexible.

The use of score-based analysis in this appendix is inspired by previous work by Scheirer [71] that used prior knowledge of a piano score to extract expressive performance information from recordings. Scheirer's system tracked keyboard onsets and releases based on predictions made from the score. The approach used here is based upon that earlier system, with significant modifications for the acoustic properties of the singing voice versus those of the piano. In particular, no timbral model was required in the case of the piano, whereas one is needed for voice in order to identify different vowels.

B.1 Score-based Parameter Extraction

The analysis model presented here takes a digitized source signal from a human singer (singing from a pre-defined score) and outputs the standard LPC parameters of pitch, gain, and filter coefficients. For simplicity, the data used for this experiment was the phrase Alleluia, Amen, performed by a trained singer. This phrase consists of only four vowel sounds, the International Phonetic Alphabet symbols for which are [a], [e], [u], and (briefly) [i] and three liquid voiced consonants: [l], [m], and [n]. While this is a small subset of the possible phonetic choices, the techniques for vowel identification and analysis may be extended to include other vowels.

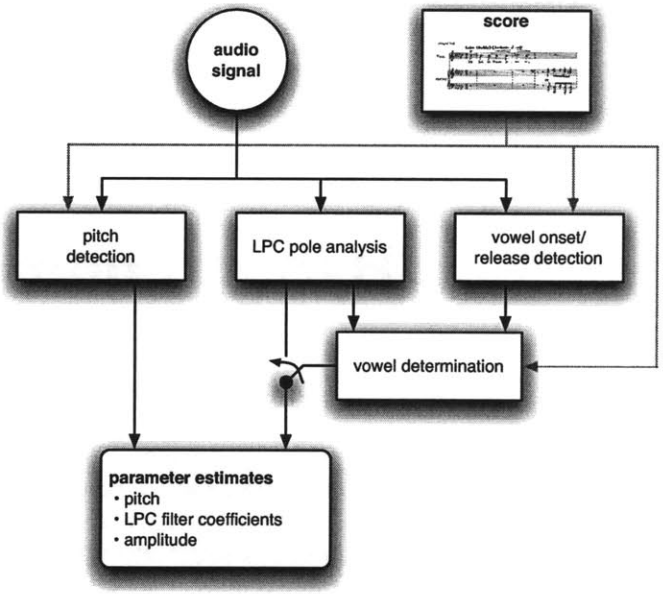


Figure B-1: Block diagram of analysis system.

B.1.1 Analysis blocks

The singing sources used in this experiment were digitized at a sampling rate of 11025 Hz. The system uses prior knowledge of the musical score to aid in the determination of the current pitch and vowel being sung. The parameters are estimated on a per-frame basis, where each frame is approximately 45ms in length (500 samples). Frames were overlapped by 50% and windowed using a Hanning window prior to processing.

The information presented in the score included the time signature, the tempo (beats per minute), the onset and offset of each note (in beats within a measure), and the primary vowel of each note. The score format was designed to present the same amount of information as standard musical notation.

B.1.2 Pitch Detection

Pitch extraction in this system is performed by finding the peak of an autocorrelation of the windowed signal that is targeted within a specific range defined by the score (from the current pitch period to the next pitch period). In this way, most errors are avoided, such as octave errors common to pitch detection by simple autocorrelation. This method was chosen for computational simplicity, and because autocorrelation is also used in each frame for the calculation of the LPC coefficients.

B.1.3 Vowel Onset and Release Detection

Training data for vowel identification was collected by having the singer sing each of the vowels [a], [e], and [u] at seven different pitches. The LPC filter parameters were calculated and averaged for each vowel to obtain a vowel template. In a low-order LP analysis, the pole angles generally correspond to the formant frequencies. An order of p=8 was used so that each pole pair would locate one of the four largest formants. The gain parameter in each frame was calculated from the energy in the error prediction signal.

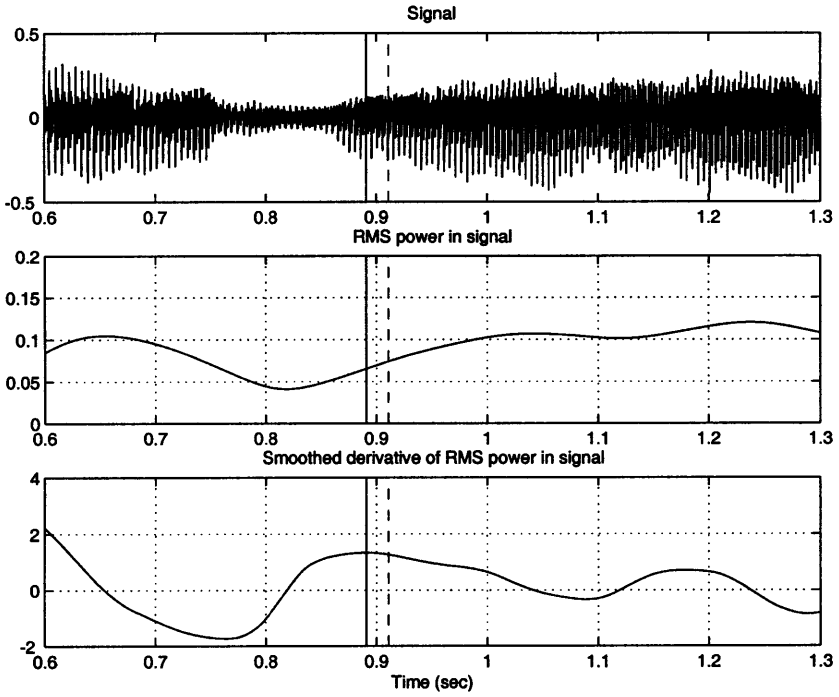


Figure B-2: Vowel onset detection of [e] in alleluia.

The system first looks for vowel onsets by examining the energy of the input signal. An onset location prediction is calculated from the score and the indicated tempo and is used to locate a detection window. The detection window spans from halfway between the predicted onset and the previous calculated onset to the midpoint between the predicted onset and the next predicted onset. The current system is restricted to sounds without diphthongs (consecutive vowel sounds), so vowel onset will occur either at a note onset or after a consonant. The energy of a vowel is likely to be greater than the energy of a consonant because the vocal tract is open for vowels and closed for consonants. Thus, the location of the vowel onset is taken to be the local maximum

derivative of the closest to the predicted onset, which accounts for both cases in which the vowel is preceded by a consonant and cases in which the vowel is preceded by silence. An example of a detected onset is shown in Figure B-2. Calculated onsets are used to readjust the tempo estimate, which adjusts the next predicted onset.

The vowel releases are located after all the onsets have been found. A release detection window spans from halfway between a note's predicted release and its calculated vowel onset to the calculated vowel onset of the next note or the end of the file. Again a consonant or silence follows each vowel, so the energy of the signal is used to determine the release location. The vowel release is taken to be the point at which the energy falls below 60% of the maximum energy in the note (between the times of consecutive onsets), as shown in Figure B-3. The space between the onset and offset is the vowel duration. Space outside the vowel duration (silence and note transitions usually indicative of consonants) is encoded using standard LPC parameterization.

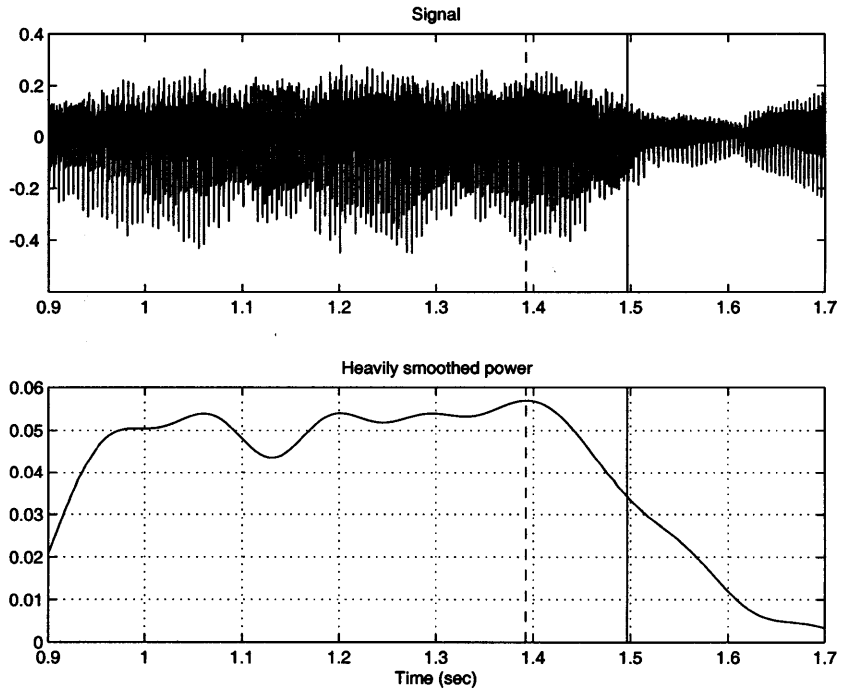


Figure B-3: Vowel release detection of [e] in alleluia.

B.1.4 Vowel identification and compression

Vowels within the vowel duration are identified by the locations of the formants. The formants are ordered by frequency and compared to formant locations calculated for different vowels from training data. The vowel with the smallest sum of absolute distances between ordered formants is taken to be the vowel. The frames calculated LPC filter coefficients are then replaced with generic coefficients for the given vowel, which are also calculated from averaged training data. Since the majority of analysis frames will consist of vowels, the data required to represent a note can be greatly reduced. Of course, this is at the expense of sound quality, but the resulting re-synthesis is perceptually close to the regularly LPC coded re-synthesis.

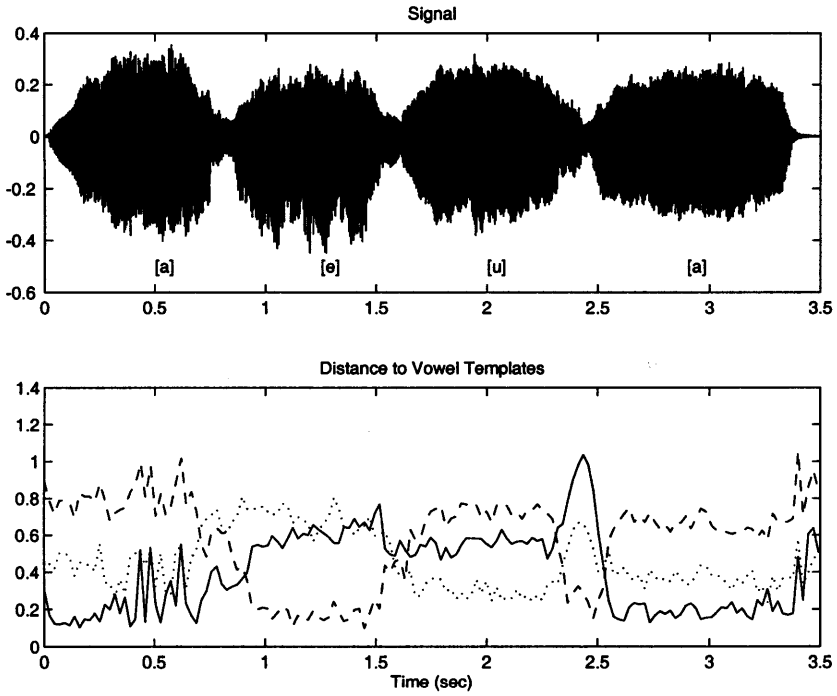


Figure B-4: Sum of formant distances to vowel templates for alleluia. Smaller distance indicates a better match.

B.1.5 Hybrid Coding Format

As with standard LPC, pitch and gain parameters are transmitted for each analysis frame. For frames in which vowels occur, no filter coefficients are needed, since they are replaced with values from the vowel templates. Thus, a codebook containing these vowel templates must also be included in the transmission. Ignoring the overhead of

the codebook, this coding scheme results in a bitstream for the given sound example that is about $\frac{1}{4}$ the size of a LPC encoded bitstream with comparable sound quality.

B.2 Extensions

An obvious improvement to the current system would be to add support for the detection and synthesis of other vowels and voiced and unvoiced consonants. This would require making a voiced/unvoiced determination; there are well-documented techniques for doing that in the LPC literature [66]. The increased number of choices would lead to more confusion in the detection, so a better heuristic (other than simple formant distance) for phonetic matching may be needed.

The current system could also be easily extended using different orders of LPC analysis for the vowel matching analysis and the audio analysis/re-synthesis. The current system uses a small number of poles (eight) to make formant selection, and thus vowel detection, easier. A low order LPC analysis could be used for formant detection, and a higher order could be used for the actual coding. The replacement vowel templates would also need to be recalculated at the higher order. The greater number of poles in the re-synthesis would result in better sound quality.

The techniques presented in this appendix are not exclusively limited to LPC. LPC was chosen because it allows the formant frequencies to be tracked easily. Other analysis/re-synthesis methods, particularly the analysis/synthesis framework detailed in this dissertation, could be used as the primary coding engine. Since vowel onset and release timing is calculated using the time-domain energy of the signal, it is independent of the coding technique. When using an alternative coding scheme, vowel tracking could be accomplished using the phonetic segmentation algorithm of Section 4.1.

Bibliography

[1] *ISO/IEC 14496-3:2001/Amd 1:2003 (amendment to the MPEG-4 audio standard).* International Organization for Standardization, http://mpeg.tilab.com/working_documents/mpeg-04/audio/amd1.zip, 2003.

[2] T. V. Ananthapadmanabha and G. Fant. Caclulation of true glottal flow and its components. *Speech Communication*, 1(3-4):167–184, December 1982.

[3] J.-J. Aucoututrier and F. Pachet. Music similarity measures: What's the use? In *Proc. International Symposium on Music Information Retrieval.* ISMIR, October 13-17 2002.

[4] A. Berenzweig, D. P. Ellis, and S. Lawrence. Using voice segments to improve artist classification of music. In *Proc. AES-22 Intl. Conf. on Virtual, Synthetic, and Entertainment Audio*, 2002.

[5] A. L. Berenzweig and D. P. W. Ellis. Locating singing voice segments within music signals. In *Proc. IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*, pages 119–123, New Paltz, NY, 2001.

[6] G. Berndtsson and J. Sundberg. The MUSSE DIG singing synthesis. *Proceedings of the Stockholm Music Acoustics Conference (SMAC)*, pages 279–281, 1993.

[7] R. Boulanger, editor. *The Csound Book.* MIT Press, Cambridge, MA, 2000.

[8] P. Cano, A. Loscos, J. Bonada, M. de Boer, and X. Serra. Voice morphing system for impersonating in karaoke applications. In *Proc. International Computer Music Conference*, 2000.

[9] M. Casey and P. Smaragdis. Netsound. In *Proc. of the International Computer Music Conference*, 1996.

[10] F. J. Charpentier and M. G. Stella. Diphone synthesis using an overlap-add technique for speech waveforms concatenation. In *Proc. International Conference on Acoustics, Speech, and Signal Processing*, 1986.

[11] Y. M. Cheng and D. O'Shaughnessy. Automatic and reliable estimation of glottal closure instant and period. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 37(12):1805–1815, December 1989.

[12] D. G. Childers. Glottal source modeling for voice conversion. *Speech Communication*, 16:127–138, 1995.

[13] D. G. Childers, K. Wu, D. M. Hicks, and B. Yegnanarayana. Voice conversion. *Speech Communication*, 8(2):147–158, June 1989.

[14] T. F. Cleveland, J. Sundberg, and R. E. Stone. Long-term-average spectrum characteristics of country singers during speaking and singing. *Speech, Music, and Hearing Quarterly Progress and Status Report (TMH-QPSR)*, 40(2-3/2000):89–94, 2000.

[15] P. R. Cook. *Identification of Control Parameters in an Articulatory Vocal Tract Model, with Applications to the Synthesis of Singing.* PhD thesis, Stanford University, 1990.

[16] G. DePoli and P. Prandoni. Sonological models for timbre characterization. *Journal of New Music Research*, 26(2), 1997.

[17] R. Duda, P. Hart, and D. Stork. *Pattern Classification.* John Wiley & Sons, New York, 2nd edition, 2000.

[18] H. Dudley. Synthesizing speech. *Bell Laboratories Record*, 15:98–102, 1936.

[19] H. Dudley. Remaking speech. *Journal of the Acoustical Society of America*, 11(2):169–177, 1939.

[20] H. Dudley. The speaking machine of Wolfgang von Kempelen. *Journal of the Acoustical Society of America*, 22(2):151–166, 1950.

[21] H. Dudley, R. R. Riesz, and S. S. A. Watkins. A synthetic speaker. *Journal of the Franklin Institute*, 2227(6):739–764, 1939.

[22] A. Eronen and A. Klapuri. A musical instrument recognition using cepstral coefficients and temporal features. In *Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing.* IEEE, June 5-9 2000.

[23] G. Fant, J. Liljencrants, and Q.-g. Lin. A four parameter model of glottal flow. *Speech Transmission Laboratory Quarterly Progress and Status Report (STL-QPSR)*, (4):1–13, 1985.

[24] R. Fletcher. *Practical Methods of Optimization, 2nd ed.* John Wiley & Sons, May 2000.

[25] A. Gersho and R. M. Gray. *Vector Quantization and Signal Compression.* Kluwer Academic Publishers, 1992.

[26] B. Gold and C. M. Rader. The channel vocoder. *IEEE Transactions on Audio and Electroacoustics*, AU-15(4):148–161, 1967.

[27] N. Gould and P. Toint. A quadratic programming bibliography. Technical Report 2000-1, RAL Numerical Analysis Group Internal Report, 2001.

[28] A. Härmä. A comparison of warped and conventional linear predictive coding. *IEEE Transactions on Speech and Audio Processing*, 9(5):579–588, 2001.

[29] A. Härmä, M. Karjalainen, L. Savioja, V. Välimäki, U. K. Laine, and J. Huopaniemi. Frequency-warped signal processing for audio applications. *Journal of the Audio Engineering Society*, 48(11):1011–1031, November 2000.

[30] J. Herre, E. Allamanche, and O. Hullmuth. Robust matching of audio signals using spectreal flatness features. In *Proc. IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*, 2001.

[31] J. L. Kelly and C. C. Lochbaum. Speech synthesis. In *Proc. Fourth International Congress on Acoustics*, pages 1–4, 1962.

[32] Y. E. Kim. Structured encoding of the singing voice using prior knowledge of the musical score. In *Proc. IEEE Workshop on the Applications of Signal Processing to Audio and Acoustics*, pages 47–50, 1999.

[33] Y. E. Kim. Singer identification in popular music recordings using voice coding features. In *Proc. International Symposium on Music Information Retrieval*, Paris, 2002.

[34] D. H. Klatt and L. C. Klatt. Analysis, synthesis, and perception of voice quality variations among female and male talkers. *JASA*, 82(3):737–793, 1990.

[35] T. Laakso, V. Välimäki, M. Karjalainen, and U. K. Laine. Splitting the unit delay–tools for fractional delay filter design. *IEEE Signal Processing Magazine*, 13(1):30–60, January 1996.

[36] P. Lansky and K. Steiglitz. Synthesis of timbral families by warped linear predicition. *Computer Music Journal*, 5(3):45–49, 1981.

[37] B. Larsson. Music and singing synthesis equipment (MUSSE). *Speech Transmission Laboratory Quarterly Progress and Status Report (STL-QPSR)*, (1/1977):38–40, 1977.

[38] J. Liljencrants. *Speech Synthesis with a Reflection-Type Line Analog.* PhD thesis, Royal Institute of Technology, Stockholm, Sweden, 1985.

[39] B. Logan. Mel frequency cepstral coefficients for music modeling. In *Proc. International Symposium on Music Information Retrieval.* ISMIR, October 23-25 2000.

[40] K. Lomax. *The Analysis and Synthesis of the Singing Voice.* PhD thesis, Oxford University, 1997.

[41] E. Lombard. Le signe de l'elévation de la voix. *Ann. Maladies Oreille, Larynx, Nez, Pharynx*, 37:101–119, 1911.

[42] H.-L. Lu. *Toward a High-Quality Singing Synthesizer with Vocal Texture Control.* PhD thesis, Stanford University, 2002.

[43] H.-L. Lu and J. O. Smith. Joint estimation of vocal tract filter and glottal closure waveform via convex optimization. In *Proc. IEEE Workshop on Applications of Signal Proessing to Audio and Acoustics*, October 1999.

[44] C. Ma, Y. Kamp, and L. F. Willems. Frobenius norm approach to glottal closure detection from the speech signal. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 2(2):258–264, 1994.

[45] M. W. Macon, L. Jensen-Link, J. Oliverio, M. A. Clements, and E. B. George. Concatenation-based midi-to-singing voice synthesis. *Presented at 103rd Meeting of the Audio Engineering Society, AES Preprint 4591*, 1997.

[46] J. Makhoul. Linear prediction: A tutorial review. *Proceedings of the IEEE*, 63:1973–1986, 1975.

[47] J. Makhoul, S. Roucos, and H. Gish. Vector quantization in speech coding. *Proceedings of the IEEE*, 73(11):1551–1588, 1985.

[48] R. Mammone, X. Zhang, and R. P. Ramachandran. Robust speaker recognition: A feature-based approach. *IEEE Signal Processing Magazine*, 13(5):58–71, 1996.

[49] K. Martin. *Sound-Source Recognition: A Theory and Computational Model.* PhD thesis, Massachusetts Institute of Technology, 1999.

[50] The Mathworks. *MATLAB Optimization Toolbox*, 2002.

[51] R. J. McAulay and T. Quatieri. Speech analysis/synthesis based on a sinusoidal representation. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 34:744–754, 1986.

[52] M. Mellody. *Signal Analysis of the Female Singing Voice: Features for Perceptual Singer Identity.* PhD thesis, University of Michigan, 2001.

[53] Y. Meron. *High Quality Singing Synthesis using the Selection-based Synthesis Scheme.* PhD thesis, University of Tokyo, 1999.

[54] N. J. Miller. *Filtering of Singing Voice Signal from Noise by Synthesis.* PhD thesis, University of Utah, 1973.

[55] J. A. Moorer. The use of the phase vocoder in computer music applications. *Journal of the Audio Engineering Society*, 26(1):42–45, 1978.

[56] N. I. of Standards and T. (NIST). The DARPA TIMIT acoustic-phonetic continuous speech corpus, 1990.

[57] A. V. Oppenheim, D. H. Johnson, and K. Steiglitz. Computation of spectra with unequal resolution using the fast fourier transform. *Proceedings of the IEEE*, 59:299–301, February 1971.

[58] A. V. Oppenheim and R. W. Schafer. *Discrete-Time Signal Processing.* Prentice-Hall, Englewood Cliffs, NJ, 1989.

[59] D. O'Shaughnessy. *Speech Communication.* Addison-Wesley, 1987.

[60] R. Picard. *Affective Computing.* MIT Press, 1997.

[61] J. M. Pickett. *The Sounds of Speech Communication.* University Park Press, 1980.

[62] M. D. Plumpe, T. F. Quatieri, and D. A. Reynolds. Modeling of the glottal flow derivative waveform with application to speaker identification. *IEEE Transactions on Speech and Audio Processing*, 7(5):569–586, Sept. 1999.

[63] H. Purnhagen and N. Meine. HILN - the MPEG-4 parametric audio coding tools. *Proceedings of the IEEE International Symposium on Circuits and Systems*, 2000.

[64] L. R. Rabiner. A tutorial on hidden markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257–286, February 1989.

[65] L. R. Rabiner and B.-H. Juang. *Fundamentals of Speech Recognition.* Prentice Hall, 1993.

[66] L. R. Rabiner and R. W. Schafer. *Digital Processing of Speech Signals.* Prentice-Hall, Englewood Cliffs, NJ, 1978.

[67] G. Richard, C. d'Alessandro, and S. Grau. Unvoiced speech synthesis using poissonian random formant wave functions. In *Signal Processing VI: European Signal Processing Conference*, pages 347–350, 1992.

[68] X. Rodet. Time-domain formant-wave-function synthesis. *Computer Music Journal*, 8(3):9–14, 1984.

[69] X. Rodet, Y. Potard, and J.-B. Barrière. The CHANT project: From the synthesis of the singing voice to synthesis in general. *Computer Music Journal*, 8(3):15–31, 1984.

[70] A. Rosenberg. Effect of glottal pulse shape on on the quality of natural vowels. *Journal of the Acoustical Society of America*, 49:583–590, 1971.

[71] E. D. Scheirer. Using musical knowledge to extract expressive information from audio recordings. In D. F. Rosenthal and H. G. Okuno, editors, *Computational Auditory Scene Analysis.* Erlbaum, Mahweh, NJ, 1998.

[72] E. D. Scheirer. Structured audio and effects processing in the MPEG-4 multimedia standard. *Multimedia Systems*, 7(1):11–22, 1999.

[73] E. D. Scheirer and Y. E. Kim. Generalized audio coding with MPEG-4 structured audio. In *Proceedings of the AES 17th International Conference*, pages 189–204, 1999.

[74] X. Serra and J. O. Smith. Spectral modeling synthesis: A sound analysis/synthesis system based on a deterministic plus stochastic decomoposition. *Computer Music Journal*, 14(4):12–24, 1990.

[75] P. Smaragdis. *Redundancy Reduction for Computational Audition, a Unifying Approach.* PhD thesis, Massachusetts Institute of Technology, 2001.

[76] J. O. Smith. *Techniques for Digital Filter Design and System Identification with Application to the Violin.* PhD thesis, Stanford University, 1985.

[77] R. Smits and B. Yegnanayarana. Determination of instants of significant excitation in speech using group delay function. *IEEE Transactions on Speech and Audio Processing*, 3:325–333, 1995.

[78] A. Spanias. Speech coding: A tutorial review. *Proceedings of the IEEE*, 82:1539–1582, 1994.

[79] K. Steiglitz. A note on variable recursive digital filters. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 28(1):111–112, February 1980.

[80] K. N. Stevens. *Acoustic Phonetics.* MIT Press, 1998.

[81] S. S. Stevens, J. Volkmann, and E. B. Newman. A scale for the meaurement of the psychological magnitude of pitch. *Journal of the Acoustical Society of America*, 8:185–190, 1937.

[82] J. Q. Stewart. An electrical analogue of the vocal organs. *Nature*, 110:311–12, 1922.

[83] H. Strik, B. Cranen, and L. Boves. Fitting a lf-model to inverse filter signals. In *Proceedings of the Third European Conference on Speech Communication and Technology (EUROSPEECH-93)*, 1993.

[84] H. W. Strube. Determination of the instant of glottal closure from the speech wave. *Journal of the Acoustical Society of America*, 56(5):1625–1629, 1974.

[85] H. W. Strube. Linear prediction on a warped frequency scale. *Journal of the Acoustical Society of America*, 68(4):1071–1076, 1980.

[86] J. Sundberg. *The Science of the Singing Voice.* Northern Illinois University Press, Dekalb, IL, 1987.

[87] J. Sundberg. Synthesis of singing by rule. In M. Mathews and J. R. Pierce, editors, *Current Directions in Computer Music Research*, pages 45–55. MIT Press, Cambridge, 1989.

[88] I. Titze, D. Wong, B. Story, and R. Long. Considerations in voice transformation with physiologic scaling principles. *Speech Communication*, 22:113–123, 1997.

[89] V. Välimäki and M. Karjalainen. Improving the Kelly-Lochbaum vocal tract model using conical tube sections and fractional delay filtering techniques. In *Proc. 1994 International Conference on Spoken Language Processing (ICSLP)*, 1994.

[90] B. L. Vercoe, W. G. Gardner, and E. D. Scheirer. Structured audio: Creation, transmission, and rendering of parametric sound representations. *Proceedings of the IEEE*, 86(5):922–940, 1998.

[91] W. Verhelst and M. Roelands. An overlap-add technique based on waveform similiarity (WSOLA) for high-quality time-scale modifications of speech. In *Proc. International Conference on Acoustics, Speech, and Signal Processing*, 1993.

[92] W. von Kempelen. *Mechanismus der menschlichen Sprache nebst der Beschreibung seiner sprechenden Maschine.* 1791.

[93] B. Whitman, G. Flake, and S. Lawrence. Artist detection in music with minnowmatch. In *Proc. IEEE Workshop on Neural Networks for Signal Processing*, pages 559–568, Falmouth, MA, 2001.

[94] Yamaha Corporation Advanced System Development Center. *New Yamaha VOCALOID Singing Synthesis Software Generates Superb Vocals on a PC*, 2003. http://www.global.yamaha.com/news/20030304b.html.