The Singing Tree: A Novel Interactive Musical Experience

by

William David Oliver

B.A., University of Rochester (1995)
B.S., University of Rochester (1995)

Submitted to the Department of Electrical Engineering and Computer Science in partial fulfillment of the requirements for the degree of

Master of Science in Electrical Engineering

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

June 1997

Author .... [illegible]
Department of Electrical Engineering and Computer Science
May 28, 1997

Certified by .... [illegible]
Tod Machover
Associate Professor of Music and Media
Thesis Supervisor

Accepted by .... [illegible]
Arthur C. Smith
Chair, Department Committee on Graduate Students

The Singing Tree: A Novel Interactive Musical Experience

by

William David Oliver

Submitted to the Department of Electrical Engineering and Computer Science
on May 28, 1997, in partial fulfillment of the
requirements for the degree of
Master of Science in Electrical Engineering

Abstract

This thesis will discuss the technical and artistic design of the Singing Tree, a novel interactive musical interface which responds to vocal input with real-time aural and visual feedback. A participant interacts with the Singing Tree by singing into a microphone. The participant's voice is analyzed for several characteristic parameters: pitch, noisiness, brightness, volume, and formant frequencies. These parameters are then interpreted and control a music generation engine and a video stream in real-time. This aural and visual feedback is used actively to lead the participant to an established goal, providing a reward-oriented relationship between the sounds one makes and the generated music and video stream one experiences. The Singing Tree is an interesting musical experience for both amateur and professional singers. It is also versitile, working well as a stand-alone interface or as part of a larger interactive experience.

Thesis Supervisor: Tod Machover
Title: Associate Professor of Music and Media

Acknowledgments

The author gratefully acknowledges,

Tod Machover, for his ambition, guidance, and vision which led to both the Singing Tree and the Brain Opera. Tod is a mentor and a friend, who knows what it means to begin again again.....

John Yu, for his tireless work on Sharle and the Singing Tree, and his good nature.

Eric Metois, for his work on the DSP toolkit and formant analysis. I have learned a tremendous amount from working with Eric and reviewing his work.

Ben Denckla, for his attention to detail in helping prepare this work, for the many arguments at the white board, and Nerf-Brain Pacman.

Joe Paradiso, for always making time, coming in an hour early, and leaving an hour late.

Maggie Orth and Ray Kinoshita, for their astronomically many contributions to the physical design of the Singing Tree and the Brain Opera.

Sharon Daniel, for her relentless pursuit of the Singing Tree's beautiful video streams.

Alexis Aellwood, for helping prepare the Brother presentation.

Neil Gershenfeld, for allowing unlimited access to the physics lab during the development of the Singing Tree and the Brain Opera, and for his interesting conversations on stochastic cooling.

Tanya Bezreh, for keeping the Brain Opera crew out of harm's way the world over, and for being able to maintain musical cadence.

Pete Rice, for reading this thesis and alerting the taxi driver when it's time to stop.

Teresa Marrin, Wendi Rabiner, and Beth Gerstein, for reading this thesis, and for assistance in its writing.

Ed Hammond, for the piece of bubble-gum in Tokyo.

Peter Colao and Laird Nolan, for taking the Brain Opera from 'nothing but trouble' to 'nothing but tourable', and for lots of good dessert wines.

Yukiteru Nanase and Masaki Mikami, for making the Tokyo presentation the best.

The Brain Opera Team, for the fun of it!

My Family, for everything.

The author gratefully acknowledges and thanks the Department of Defense Army Research Office for its continuing financial support through the National Defense Science and Engineering Graduate Fellowship program.

The author gratefully acknowledges and thanks the MIT Media Laboratory and the Things That Think Consortium for their financial support of the Brain Opera and the Singing Tree project.

In addition, the author would like to give special recognition to Terry Orlando, Tom Hsiang, Tom Jones, Roman Sobolewski, Phillipe Fauchet, Edward Titlebaum, Alan Kadin, Mariko Tamate, Meg Venn, Ron Yuille, Chieko Suzawa, the Doi Family, Mr. Hakuta, Hiroyuki Fujita, Minoru Ishikawa, Gregg Learned, and the Kanazawa Rotary Club, for their academic inspiration and support, and for their friendship.

This thesis is dedicated to the memory of Harold A. Oliver.

Contents

1 Introduction **13**

1.1 Motivation . . . 13

1.2 The Brain Opera . . . 14

1.3 The Singing Tree . . . 14

1.4 Review of the Literature and Research Activities . . . 15

1.4.1 Voice Analysis and Synthesis . . . 16

1.4.2 Computer Music, Electronic Instruments, and Interactive Music . . . 17

1.4.3 Performance-Driven Interactive Instruments . . . 18

1.4.4 Previous Work from the Media Laboratory . . . 20

1.4.5 The Trouble with Bessel Functions . . . 20

1.5 Fundamental Questions . . . 21

1.6 Organization of this Thesis . . . 22

2 The Singing Tree System **23**

2.1 Design Criteria . . . 23

2.2 Modular Design . . . 24

2.3 Equipment . . . 25

2.3.1 Operational Equipment . . . 25

2.3.2 Aesthetic Equipment . . . 26

2.4 Kurzweil K2500 Sampler/Synthesizer . . . 27

2.4.1 K2500: Overview and Modifications . . . 27

2.4.2 Vocal Samples: Recording and Editing . . . 28

2.4.3 A Short Digression into MIDI . . . 29

3 Singing Tree Implementation: Technical and Artistic Issues **31**

3.1 Technical Decisions in an Artistic Context . . . 31

3.2 Voice Analysis . . . 31

3.2.1 Volume . . . 31

3.2.2 Pitch Estimation . . . 32
3.2.3 Noisiness . . . 37
3.2.4 Brightness . . . 38
3.2.5 Formant Estimation . . . 41
3.2.6 Alternative Methods of Pitch Estimation . . . 60
3.3 Music Generation, Interpretation, and Mappings . . . 65
3.3.1 Sharle . . . 65
3.3.2 General Interpretation and Mapping Objectives . . . 67
3.3.3 Defining Pitch Instability . . . 70
3.3.4 Instrumentation . . . 71
3.3.5 Dynamic Parameter Mapping and Control . . . 73
3.3.6 Examples of Operation . . . 76
3.3.7 Discussion of Design Methodology . . . 77
3.3.8 The Issues of Algorithm, Constraint, and Composition . . . 79
3.3.9 Visual Feedback Control . . . 80

4 Results, Observations, and Future Directions 83
4.1 Observations of The Singing Tree in Various Venues . . . 83
4.1.1 The Singing Tree as a Stand-Alone Interface . . . 83
4.1.2 The Singing Tree as Part of the Mind Forest . . . 84
4.2 Critical Evaluation of the Singing Tree . . . 85
4.2.1 Strengths of the Singing Tree System . . . 85
4.2.2 Weaknesses of the Singing Tree System . . . 86
4.3 Future Directions for Applications and Research . . . 87
4.3.1 Interactive Karaoke . . . 87
4.3.2 Fuzzy Systems, Linguistic Variables, and Interactive Systems . . . 88
4.3.3 Higher-Order Spectra Analysis . . . 90
4.3.4 Beyond the Singing Tree . . . 91

5 Conclusions 93
5.1 Synopsis of the Singing Tree . . . 93

A The Novel Interactive Instruments of the Brain Opera 95

B Bayesian Approach to Deriving the Pitch Period Estimator 97

C Additional Examples of Formant Frequency Estimation 99

List of Figures

1-1 The Singing Tree in Tokyo 15

2-1 Cable Schematic of the Singing Tree 26

3-1 Perceived Loudness of Single-Component Sounds as a Function of Frequency 33
3-2 Mechanical Model of Speech Production 42
3-3 Linear, Quasi-Time-Invariant System Model 43
3-4 Expanded View of the Vocal Tract Model 44
3-5 Block Diagram of the Formant Estimation Algorithm 46
3-6 Vowel as a Function of F1 and F2 50
3-7 The Vowel Triangle 51
3-8 Pole-Zero Plot and the DFT in the z-plane 54
3-9 Pole-Zero Plot and the CZT along r=0.75 in the z-plane 55
3-10 Chirp z-Transform over a Smaller Frequency Range for Higher Resolution 55
3-11 Segment of Voice Singing 'aah' and its Power Spectrum, ... 56
3-12 Log-Magnitude and Real Cepstrum 56
3-13 Log-Magnitude and Real Cepstrum of the Glottal and Radiational Component 57
3-14 Approximation of the Real Cepstrum with no Glottal and Radiational Component 57
3-15 Cepstrally Smoothed Log Spectrum 58
3-16 Cepstrally Smoothed Log Spectra in Regions 1 and 2 59
3-17 Cepstrally Smoothed Log Spectra via the Chirp z-Transform 59
3-18 Cepstrally Smoothed Log Spectra of Region 3 (Unit Circle and CZT) 59
3-19 Pitch Estimation via the DFT and CZT for Three Signal Segments 63
3-20 Pitch Estimation via the Real Cepstrum for Three Signal Segments 64

A-1 The Singing Tree (left) and the Rhythm Tree (right) in Tokyo 95
A-2 Harmonic Driving (left) and the Melody Easel (right) 96
A-3 The Gesture Wall (left) and the Mind Forest (right) 96
A-4 The Brain Opera Performance in Tokyo 96

C-1 'aah' Signal at f=139.5 Hz and Relevant Formant Estimation Plots 100
C-2 'ee' Signal at f=296 Hz and Relevant Formant Estimation Plots 101
C-3 'ee' Signal at f=149 Hz and Relevant Formant Estimation Plots 102
C-4 'ooh' Signal at f=290 Hz and Relevant Formant Estimation Plots 103
C-5 'ooh' Signal at f=145 Hz and Relevant Formant Estimation Plots 104

List of Tables

2.1 Singing Tree Equipment List 25

3.1 Formant Frequency Range 49
3.2 Average Formant Frequencies for Several Vowel Sounds 51
3.3 Estimated Formant Frequencies 60
3.4 Instrument Groups Used in the Singing Tree 71

C.1 Estimated Formant Frequencies for the 'aah' Signal at f=139.5 Hz 99
C.2 Estimated Formant Frequencies for the 'ee' Signal at f=296 Hz 100
C.3 Estimated Formant Frequencies for the 'ee' Signal at f=149 Hz 100
C.4 Estimated Formant Frequencies for the 'ooh' Signal at f=290 Hz 101
C.5 Estimated Formant Frequencies for the 'ooh' Signal at f=145 Hz 101

Chapter 1

Introduction

1.1 Motivation

As the scientist is gregarious with her research, so is the composer with his music. Indeed, the objective of sharing one's musical experience with others is the premise of performance. And, be it research results or a musical experience, the benefits are certainly not limited to the creator; distribution and exposure are beneficial to all. The belief that more people should have exposure to the experience of music is not new. Pythagoras and Plato considered music an integral component of education. Mozart bucked tradition to have his operas performed in a truly public forum. Nonetheless, until early this century, performing music in one's home for one's family and friends was not only social pastime, it was a person's primary contact with music [1]. With the technological advances of this century, namely the phonograph, radio, and television, music has become essentially ubiquitous. Whereas 100 years ago, I might have sung the same twelve hymns with my family every night, today I can turn on my radio and listen to any style of music I choose. In this, the technology has done a wonderful job at introducing me and the general public to music from a variety of genres. However, the technology has also shifted the reality of a musical experience from active participation to passive listening [2]. Most people would agree that attending a concert is a superior musical experience to listening to a recording, even if it is a digitally recorded and mastered CD played on the latest system with the well-marketed '1-bit D/A'. In addition, I believe that musicians and composers would add that creating the music oneself can be an even more rewarding experience.

The Opera of the Future and the Physics and Media groups at the MIT Media Laboratory are concerned with using technology to bring the active musical experience to a greater number of people. Novel implementations using computer music algorithms, computer human interfaces, and sensor technologies have led to successful applications such as Vox-Cubed [3], the Hyperstring Trilogy [4], Joystick Music [5], Drum-Boy [6], and the Sensor Chair [7]. Most recently, the Brain Opera was an attempt to collectively involve large numbers of people in the active musical experience from composition to performance. The Singing Tree was developed as one part of the The Brain Opera experience [9].

1.2 The Brain Opera

The Singing Tree is one of six novel interfaces used in the Mind Forest of the Brain Opera, an interactive opera composed and developed by Tod Machover and the Brain Opera Team at the MIT Media Laboratory. The Brain Opera, based in part on Marvin Minsky's book, *Society of Mind* [8], is divided into three parts [9]:

The Mind Forest: an interactive space in which the audience explores and creates music related to the Brain Opera via six novel interfaces.

Net Music: a virtual interactive space in which Internet participants explore and create music related to the Brain Opera via Java applets.

The Performance: the Brain Opera performance in which three performers use novel interfaces to simultaneously play written music and introduce the audience and Internet contributions to the piece.

Late last century and early this century, psychologists and psychiatrists hypothesized that the human conscious is somehow 'managed' by one (or a small number of) highly intelligent 'control center(s)'. The worlds of 'human thought' and 'consciousness' are so unlike any other phenomena in nature, that many believed the mind to be scientifically unexplainable. In *Society of Mind* [8], Marvin Minsky proposes the theory that the human mind has no such 'control center', but, rather, that intelligent thought is actually an assembly or 'Society' of individually mindless 'agents'. Minsky creates a metaphor between the human brain and forests of these agents, and herein lies the concept of the Brain Opera's Mind Forest and its interfaces: the Singing Tree, the Speaking Tree, the Rhythm Tree, Harmonic Driving, the Melody Easel, and the Gesture Wall. The participants are the agents who interact with the Brain Opera through these six interfaces. Pictures of these interfaces are presented in Appendix A.

The Brain Opera debuted last July at the 1996 Lincoln Center Festival in New York City, and has since been performed at Linz, Austria; Copenhagen, Denmark; Tokyo, Japan; and West Palm Beach, Florida. It is scheduled to continue touring through 1998, at which time it will likely find a permanent home as an installation in an interactive music center or science museum.

1.3 The Singing Tree

A participant interacts with the Singing Tree by singing any pitch into the Singing Tree microphone, as illustrated in Figure 1-1. While singing, the voice is analyzed for several characteristic parameters. These parameters are then interpreted as control parameters and mapped to the input variables of a music generation algorithm called Sharle and a video stream. In effect, the participant's vocal parameters determine the musical and visual feedback in real-time. The participant is surrounded in a musical and visual 'aura' which is designed to work as reward-oriented feedback, leading the participant to an established goal. In the

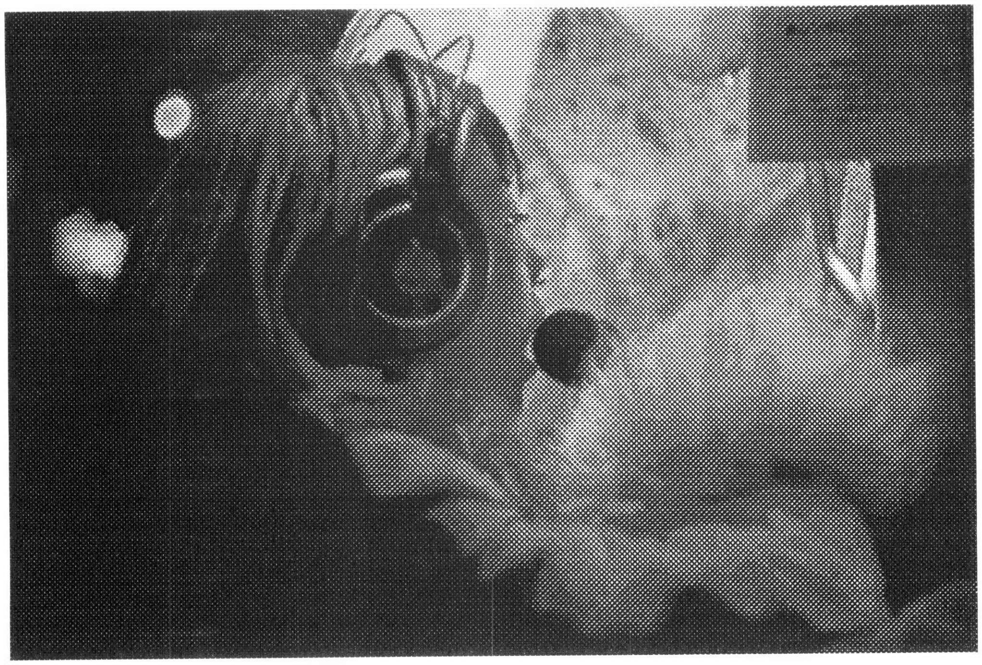

Figure 1-1: The Singing Tree in Tokyo

Brain Opera, the goal was determined to be a pure and steadily sung pitch. The aural and visual feedback is designed to reward participants who can maintain such a pitch within an established tolerance [10].

In a larger sense, the Singing Tree is an attempt to design an interface which responds in real-time to information contained in the human voice using algorithms that are computationally inexpensive [10]. Furthermore, the Singing Tree contributes to the efforts of Tod Machover's group in the fields of interpretation and musical mapping. It attempts to identify and interpret musically meaningful information in the singing voice for use in creating a meaningful musical and visual experience. In addition to the Brain Opera context, The Singing Tree is also an example of a human-computer interface which can seamlessly extract useful information from the human voice. These types of human-computer interfaces will play an increasingly important role in 'smart' applications.

1.4 Review of the Literature and Research Activities

The Singing Tree and the Brain Opera are innovations based the works of many in the admittedly broad fields of voice analysis and synthesis, music analysis and synthesis, computer music, computer human interfaces, and control systems. In this section, particularly relevant works which established the precedent for interactive music projects, including the Singing Tree and the Brain Opera, are presented. The purpose is to provide an introduction to the foundation and context of this work. Readers who wish to further research these areas are referred to the appropriate references.

1.4.1 Voice Analysis and Synthesis

The Singing Tree derives its statistics for music generation from characteristics in the human singing voice such as pitch period, brightness, and vowel formants. Much of the work in this field started in the 1950's, with significant advances in the 1960's and 1970's paralleling the advancements in the computational efficiency of computers [53]. The 1980's and 1990's have seen applications of these advances in the fields of medical diagnostics, speech recognition systems, and voice identification. In particular, the works of Johan Sundberg [11], Kenneth Stevens [14], Gerald Bennett [65], Xavier Rodet[15], and David Wessels [25] are fundamental to the techniques used in the Singing Tree voice analysis algorithms.

Formant Analysis of the Speaking and Singing Voice

Johan Sundberg's research in the 1960's and 1970's on the analysis of the human singing voice has introduced the concepts of 'singing formant' and its important role in operatic singing [11], [12],[13]. In short, opera singers can significantly shift their vowel formant structure to make their words more comprehendible, tune their pitch to the orchestra accompaniment, and 'cut through' the orchestra's sound. In addition, Kenneth Stevens' research in the fundamentals of speech and vowel formation also introduced similar concepts of formant shifting and tuning in his analysis of the unusual chanting modes of Tibetan Lamas [14]. The Singing Tree utilized formant frequencies as a control parameter for music generation.

Voice Synthesis Using Formant Frequencies

The works of Xavier Rodet, Gerald Bennett, and Johan Sundberg in the field of singing voice synthesis include examples of using formant frequencies to re-create a singing voice. In particular, the CHANT project at IRCAM in Paris, France introduced a synthesis-by-rule technique to create a human singing voice from the first five formant frequencies. The project is important to the Singing Tree, because it specifically used the vocal apparatus as the physical model for speech production in its analysis. Furthermore, it is an example of using this model as a the basis for generating sound. In the case of CHANT, the sound it generated was the voice it was intending to model, while in the Singing Tree, the sound generated was literally music. The music generation algorithm of the Singing Tree was controlled by mapping specific characteristics of the voice as derived from a vocal model, and it included the formant frequencies as stated previously. CHANT also was an introduction to the concept of rule-based knowledge structures and schemes, in which one represents a formalization of a decision, a gesture, or musical organization in terms of 'rules' [15], [65].

Speech and Voice Therapy and Training

Much of the research in speech modeling and voice analysis was directed toward diagnostic applications and speech therapy. Speech-language pathologists, otolaryngologists, voice teachers, and singers often use software which can identify the acoustic features of sustained or sung vowels for clinical assessment or vocal training. These programs allow one to view a time-domain or frequency-domain representation of one's voice to characterize one's hoarseness, harshness, breathiness, formant structure, glottal attack, and fundamental frequency contour to name a few [16]. This area of research takes voice parameter information and displays it on a screen so that the user can adjust his vocal input to match a desired pattern. This feedback mechanism is analogous to the feedback model used in the Singing Tree.

1.4.2 Computer Music, Electronic Instruments, and Interactive Music

Interactive music is often, by nature of the implementation, a 'computer-based music', but that does not necessarily imply that it is 'computer music' in the traditional sense of the term. While much of the research in computer music has focussed on providing compositional tools for the composer, the work in interactive music has focussed heavily on the user (i.e., the interactive 'instrumentalist'), the user interface, and control issues [17], [10]. In addition, while interactive music systems often use an 'electronically-based interface' which creates a musical experience, that certainly does not qualify them as electronic instruments. Nor does it imply any notion of interactivity in the functionality of an electronic instrument. Nonetheless, the three fields do overlap in an evolutionary sense, and the present research directions and implementations found in the field of interactive music systems are strongly rooted in the works of electronic instrument and computer music pioneers. There are many examples of precedent to the Brain Opera and its instruments, at least in motivation and methodology. These include the works of Leon Theremin [7], John Cage [18], Paul Lansky [19], Max Mathews [20], Karlheinz Stockhausen [21], [22], Morton Subotnick [23], Pierre Boulez [24], David Rokeby [25], George Lewis [26], Richard Teitelbaum [25], Barry Vercoe [25], Tod Machover [25], and Robert Rowe [25]. In addition, although his contribution is not specifically in the fields of computer or interactive music, Heinrich Schenker's contribution to music analysis carries implications in these fields.

The Theremin

The theremin was invented by Leon Theremin in 1920, and is an early example of a truly electronic instrument [7]. The idea behind the theremin is that the frequency of an oscillator can be adjusted by changing the value of the capacitance in a tank circuit. The theremin allows the user to change the capacitance by moving her hands closer to or further from an antenna. In application, there are two frequencies, of which the user can adjust one. The frequency of the beating between the two signals determines the pitch one hears. Another antenna is used to control volume. The result is an instrument which is played without ever touching anything. The user moves her arms to create a pitch of a certain volume. The Brain Opera uses a similar technology in the Sensor Chair and the Gesture Wall [7]. The difference is in the role of the user. In the theremin, the user acts as a capacitive shunt by moving closer to or further from the antenna, while in the Sensor Chair and the Gesture Wall, the user is the antenna. Both the Sensor Chair and the Gesture Wall have four receivers mounted in front of the user in the configuration of a square (one receiver in each corner of the square). The user stands (Gesture Wall) or sits (Sensor Chair) on a metal transmitter which couples a low-frequency RF signal into the body. The user then moves her hand (now an antenna) around in front of her body, and a computer determines the spatial position of her hand by the differential signal strengths at the receivers. Considering positions over time, velocity and acceleration metrics can also be extracted. These measurements are then interpreted and mapped to a sampler (or other sound source) to create music. In addition to the Brain Opera, the Sensor Chair technology has been used in performances by Penn and Teller, the magicians, and by the artist formerly known as Prince [7].

"Imaginary Landscape No. 4"

John Cage's piece, "Imaginary Landscape No. 4" (for 12 Radios), is an early example of a musical performance which combines human and a non-traditional electronic 'instrument'. The piece was scored for 12 radios and 24 people. For each radio, one person would control the volume and another would control the frequency. The result was a musical piece in which one, several, or all radios would be playing different stations at different volumes simultaneously. The word 'instrument' is in quotes above, because, despite the fact that both radios and instruments play music, the radio is not an instrument in its functional sense. The difference is the level of control and the variety of music that can be produced. A radio has a very high level of control, but a wide variability in musical sound; while one cannot dictate the characteristics of the specific piece of music being broadcast on any particular channel, one can certainly change channels and hear remarkably different styles of music. On the other hand, an instrument allows its user to control every nuance of the sound it produces, but cannot fundamentally changes its sound. The Brain Opera and the Singing Tree attempt to fall somewhere in between these two models, allowing some lower-level control and some higher-level control [18].

"Groove"

Groove is an early example of interactive music designed by Max Mathews and F. Moore at ATT Bell Laboratories in the late 1960's. Groove allows a user to control 14 'functions of time' simultaneously in real time. These 'functions of time' are used to control various parameters of a song such as speed, volume, and waveform. The music one hears is the sensory feedback which closes the loop and allows the user to make further adjustments in real-time [27]. In this, GROOVE is conceptually a predecessor to the Singing Tree and other instruments used in the Brain Opera.

1.4.3 Performance-Driven Interactive Instruments

George Lewis, Richard Teitelbaum, and Tod Machover are examples of musicians/composers who have used interactive music as a performance tool. This is typically accomplished by analyzing a metric of the instrument, such as pitch, and using the information to drive a music generation algorithm, sampler, or sequencer which augments the instruments sound. George Lewis' approach is to augment the music in an improvisatory manner. For example, he uses a pitch follower to convert a trombone's pitch to MIDI, which in turn drives a software algorithm designed to improvise with the trombone. As Lewis is a jazz trombonist, the improvisatory style is typically a jazz style. Lewis' music generation algorithms are probabilistic, and driven entirely by the player (i.e., there are no sequences) [25]. Richard Teitelbaum is a pianist, who uses a MIDI interface on an acoustic piano to send MIDI note information to a computer. The computer acts as a complex musical 'transformer', able to add delay, transposition, repetitions, and even play other solenoid-driven pianos. An interesting concept in Teitelbaum's approach, is that the pianist has full control of the computers functions; there is no ambiguity. The user tells the computer through a switch (not the piano) that he wants a particular transposition, and the computer responds by implementing that transposition [25]. Tod Machover has also designed and composed several pieces for interactive instrument, or 'hyperinstrument', including the opera, *Valis* for piano, mallet percussion, and computer; *Towards the Center,* a piece for flute, clarinet, violin, violon-cello, keyboard, percussion, and computer; *Bug-Mudra,* for acoustic guitar, electric guitar, dextrous hand master (a hyper-conducting-glove which measures the angle of three joints on each finger), and computer; and the Hyperstring Trilogy for hyperviolin, hyperviola, and hypercello, which are described in greater detail in Section 1.4.3 [25].

Schenker Analysis

The music analysis technique developed by Heinrich Schenker takes a complex piece of music (or tonal event) and, in stages, reduces it to a more structurally fundamental representation of the basic musical (tonal) progression [28]. In this, there is an attempt to separate the fundamental musical idea from the embellishment and ornamentation. The lower-level details of the music are represented in a compact, higher-level notation. Thus, there is in music theory an analogy to the representation attempted in many of the Brain Opera's interactive instruments; the user controls the interactive instruments by manipulating a higher-level form of the music. What is heard, however, is the music in its detailed, lower-level form. The success or failure of the experience lies, in part, in the designer's ability to make a meaningful connection between the higher-level representation of the music and its actual, lower-level form. One finds a similar situation in mathematics and physics, in which one tries to find an 'ideal' representation for a matrix or wavefunction by projecting onto an appropriate set of basis vectors or basis functions. A judicious choice will lead to a remarkable simplification, allowing one to easily manipulate (exactly or approximately, as the case may be) the more complex version of the matrix or wavefunction by operating on its simplified form. Literature offers an analogy describing the difference between a 'judicious representation' and a poorly chosen one. The vocabulary of a Hemingway novel may be at the same reading-level as a modern day, 7th grade English text on Shakespeare. However, the former is considered artistically ingenious for its simplicity in representing the complexities of the human condition, while the latter is merely a 'dumbed-down' version of an English classic, with much of the original work's complexity and meaning lost in the 'translation'. While interactive music systems often present hardware and software implementation challenges which seemingly become the obstacles to success, one should not underestimate the God-like intervention that an ill choice of musical representation (and all that accompanies it, including parameter interpretation and mapping) can have on a project's mortality.

1.4.4 Previous Work from the Media Laboratory

There are several predecessors to the Brain Opera and the Singing Tree from the Media Laboratory. Most notable are the Hyperinstruments, Joystick Music, and Drum Boy. The Hyperinstruments Project began in 1986 as an attempt to provide virtuosi with added degrees of expressivity in their instruments by augmenting the instruments' sounds. Of the many Hyperinstrument Projects described in Section 1.4.2, the author had the opportunity to work on the Hyperstrings Trilogy Performance at the Lincoln Center Festival in July, 1996. The specific instruments were a hypercello, hyperviola, and hyperviolin. The augmentation was accomplished through the addition of sensors to the instrument to gather information on the player's performance and technique. For example, bow position and bow pressure were measured. This information was analyzed to determine musical meaning, and then mapped to a sequencer or sampler to play back additional sounds along with the instrumentalist in real time. The result was a 'new' instrument. The virtuosi knew how to play this instrument in the traditional sense, but had to experiment with it to learn how it would behave in its augmented condition. Virtuosi, including Yo-Yo Ma, Paul Silverthorn, Ani Kavafian, and Matt Haimovitz, have used these hyperinstruments to play Tod Machover's Hyperstrings Trilogy: *Begin Again Again* for hypercello, *Song of Penance* for hyperviola, and *Forever and Ever* for hyperviolin.

Drum Boy and Joystick Music, on the other hand, are examples of an attempt to give enhanced musical expressivity to amateur musicians. In Drum Boy, a user was able to control complex drum patterns using relatively simple, higher-level controls such as 'complex vs. simple', 'energetic vs. calm', and 'mechanical vs. graceful'. The control mechanisms were literally 'adjectival transformation functions', which would change the drum pattern in a manner which was appropriate for the particular adjective. An extension of the Drum Boy research was Joystick Music. In Joystick Music, the user would control the rhythmic and melodic evolution of a simple musical motif, or 'seed', via two joysticks. The joysticks had higher-level controls associated with each direction, and, typically, one joystick was used to control the rhythm while the other controlled the melody [5]. A further extension of Joystick Music was the music generation algorithm used in the Singing Tree called Sharle [36].

1.4.5 The Trouble with Bessel Functions

This brief review of the literature closes with a quote from Paul Lansky. Lansky was asked what he thought the most unhealthy aspects of computer music are. His answer makes a strong point about priorities in creating computer music, and can be extended to those who create interactive musical experiences or instruments [19].

> I have trouble with Bessel functions at this point. I'm very impressed with how far it (frequency modulation) can be extended. But very simple frequency modulation wears me down. ...It is the extent to which there is an obsession with the machine rather than with what it produces.

I have heard too many discussions among computer music people who were only concerned with software and hardware without even considering what kinds of pieces they were producing. ...Good machines and better software certainly make life easier, but there is not a one-to-one correlation between the quality of pieces and the tools used to make them. Often, systems which are too easy to use encourage thoughtlessness. *Paul Lansky*

Bessel functions represent the spectral envelope in frequency modulation, which was a major development in the field of computer music. Lansky's point, however, is well taken. Be it computer music or interactive music, the technology should drive the artistry to new areas. It is certainly interesting and fruitful to develop new technologies in their own right, but one should not lose sight of the artistry. If the artistic goals are maintained in the face of technological innovation, interactive music systems will not be 'thoughtless', nor will they seem 'simplistic'.

1.5 Fundamental Questions

The Singing Tree project introduced several fundamental issues, many of which remain largely unresolved. This is not a failure of the research necessarily, but a result of the fundamental and philosophical nature of the issues at hand. They are mentioned here, because they are issues which remain a catalyst for debate among the author, his colleagues, and the critics of interactive musical projects.

Why are we building a Singing Tree, or a Brain Opera for that matter? Indeed, the motivation behind the projects is to introduce more people to the 'musical experience', but who is to dictate what that is? In fact, some might say these efforts are rather arrogant and condescending. Does this imply an attempt to bring the musical experience to those who cannot otherwise experience it? Others might argue that attempts to take an instrument's natural control space and transform it into something easier that anyone can master is, in effect, a 'dumbing down' of the instrument's capabilities and the music it plays. While the previous is a rather harsh interpretation of the goals of the group's research, they are valid questions which should be reviewed. To the author, this work is merely an attempt to develop new and interesting musical experiences. Anyone can enjoy playing a musical instrument or singing without being a virtuoso. Musical prodigy or otherwise, people understand their own musical tastes, the music to which they enjoy listening, and, simply stated, what sounds good to them. If a person can design a musical experience which simply sounds good and is enjoyable to use, then it is the author's belief that the work is a success.

Analysis of the singing voice singing one note is a particularly rich and, yet, tractable proposition, because it poses the analysis problem in differential or perturbative reference to one note. The Singing Tree project is an attempt to analyze a complex system, the singing voice, and measure or derive a finite number of statistics. These statistics are used to drive a music generation algorithm, and in that, there is an implication that these statistics somehow contain the musical intention of the participant. That is a rather difficult implication to accept. The musical 'gesture' that one person uses to represent a musical intention may be very different from the musical 'gesture' that another uses for the same intention. Are there orthogonal bases that span the instrument space? What are the bases of a human voice? a cello? the world? [29] How many do we need to be assured of 'the right answer'? These questions, of course, cannot be answered explicitly. While one can only conjecture through experience, imagination, and experimentation which measurements are significant, and how many are sufficient, there is reason to believe that a judicious choice of statistics or bases can do a pretty good job without worrying too much about the details [29], [30].

The Singing Tree project is a prime example of a system whose behavior is best specified using linguistic variables. In other words, it is much easier to describe in words how the Singing Tree output should behave as a function of its input than it is to describe quantitatively. In fact, many of the projects in Tod Machover's group are of this type. The methodology used in creating the music generation algorithm, Sharle, and the musical mappings of the Singing Tree, is in principle similar to that used in fuzzy control and the fuzzy interpretation of a linguistic variable [31]. The Singing Tree does not intentionally use fuzzy set theory, but the author believes the circumstantial similarities to be compelling and further consideration should be given to this area [32]. More on this topic will follow in Section 4.3.

1.6 Organization of this Thesis

After a brief introductory statement in Chapter 1, the foundation of the thesis begins in Chapter 2 with a discussion of the Singing Tree at the system level. The design criteria, design approach, equipment, and vocal sample preparation are covered from a 'phenomenological' perspective. Chapter 3 develops the physical models and mathematical approaches used in the voice analysis, interpretation, and mapping algorithms. While there are many equations and graphs, the exposition reveals the technology's role as a tool by which the artistic goals of the Singing Tree project were realized. Chapter 4 discusses the author's observations of the Singing Tree in use, and it includes relevant comments from notable users. The chapter continues with a look at the next steps one could take in both research and development for the Singing Tree and other interactive experiences. Finally, Chapter 5 summarizes the work.

The thesis is a full discussion of the Singing Tree's development and operation. This does not imply that the author built the Singing Tree by himself. To the contrary, the Singing Tree project was a culmination of the works and efforts of many. Those who made the most significant contributions are mentioned with their work, while others are mentioned in the acknowledgments. There are research projects which focus on one particular question in great detail, and then there are projects, such as the Singing Tree, which investigate and incorporate concepts spanning a wide spectrum of disciplines. To the credit of all who were involved in this project, the Singing Tree project proved to be an excellent learning experience for the author because of its breadth.

Chapter 2

The Singing Tree System

2.1 Design Criteria

The Singing Tree design criteria are summarized in the following [9], [33].

> Each singing-forest station will be for one person only. Each person will be close miked and wearing headphones; maximum sound isolation. Each station (ca. 5 in all) will concentrate on a single pitch (either this pitch is invariant and "user" adjusts to it, or we ask the person to sing their "favorite" note which assumedly would be one comfortable in their range, and adjust the rest to that....we will decide this later). The person is told to sing that note as calmly, simply, and "concentratedly" as possible, and that computer will "reward" this: i.e. the calmer the singing (on that one note) the more "beautiful" the aura; the more "nervous" the singing the more "agitated" the aura. ... The goal is not to make a "control" instrument so that the "performer" can consciously vary the sonic result by, for instance, producing a specific consonant (which would be hard). Rather, we are designing a very sensitive analysis engine that does a very reasonable job of recognizing changes in these parameters as reliably as possible, and with the fewest possible "glitches"/discontinuities. ... Once we have experimented with measuring all of these things, and tuning the sensitivity of our algorithms, we will decide how to combine the parameters into higher-level mappings. ... We are working to produce a kind of rolling arpeggiation spreading out above and below the reference pitch, and capable of moving from very calm and harmonious to quite agitated and inharmonic. Individual notes will be faded in and out very gracefully, so the result will never sound like arpeggios with accented notes, but rather like light patterns constantly changing on the surface of rippling water. ... I want it to sound like a choir of 127 angels singing at the same time, reacting to every perturbation in the voice. ... *Tod Machover*

Additional design criteria were later included. The musical interaction experience must be accessible to both amateur and professional singers. The Singing Tree must work as a stand-alone interactive interface and as part of a larger interactive experience. There will be a unique video stream [34] for each of the trees, and these videos will also represent the state of the participant's ability to meet the established goal. The software and design should be flexible, allowing new goals, new music, and new video to be incorporated easily. Due to funding restrictions, a total of three Singing Tree systems were produced (rather than the originally planned 5 systems). As described in the remainder of the thesis, the resulting Singing Trees closely matched these design criteria. Differences will be specifically noted.

2.2 Modular Design

The first issue in designing a system is to determine how to divide the project into sub-systems. This was particularly relevant in the case of the Singing Tree, because it was to be both a stand-alone interactive interface and a component of a larger interactive experience. In addition, the other interactive interfaces and the protocol by which to coordinate them were also in development at the same time as the Singing Tree. The Singing Tree Team saw modularity as the means to organize our own efforts, while also minimizing the amount of work that would be required later when coordination protocol was firmly established.

The Singing Tree was modularized into three categories:

1. Vocal Parameter Analysis

2. Interpretation and Mapping

3. Music Generation and Video Display

This was the most practical categorization for several reasons. First, Eric Metois had already developed a DSP toolkit [30] for most of the vocal analysis that needed to be used. Learning how the algorithms work, adding additional algorithms where needed, and porting the code from UNIX to the Windows platform [35] was far better than 're-inventing the wheel'. Second, John Yu was nearly finished with his music generation engine called Sharle [36]. By utilizing work already accomplished, development time was greatly reduced, and more importantly, the remaining tasks to be completed were clearly defined. To complete the project and meet the Lincoln Center deadline, a formant analyzer would need to be added to the DSP toolkit, interpretation and mapping algorithms would need to be developed, the specific equipment and wiring diagram would need to be established, the video would need to be created, the musical response needed to be determined, and the specific vocal samples to be used as the chorus would need to be recorded [37] and prepared.

One can make the argument that choosing to use established work or clearly defining tasks too early in a project can stifle creativity. While the author acknowledges this line of reasoning, he is more compelled to believe that a project subjected to this philosophy for too long would suffer from a lack of direction.

Table 2.1: Singing Tree Equipment List

Company	Model Number	Equipment
IBM	PC750	computer, 133 Mhz Pentium, 64 Mb RAM
Kurzweil	K2500	rack mount sampler, 64 Mb RAM, Fujitsu hard drive
Mackie	1202	12 channel audio mixer
ART	SC2	2 channel compressor/limiter/gate
ART	Effects Network	effects processor
ART	Pro MPA	microphone preamplifier
Samson	Servo 150	studio amplifier, 75W stereo
Samsung		LCD screen and driver
ETA	PD8C	power conditioner, EMI/RFI spike surge protection
KRK		100W speaker
Sure	58	microphone

Creativity is not destroyed by establishing one's design parameters, rather, the creative arena is simply clarified. In the case of the Singing Tree, the interpretation and mapping algorithms allowed a wide range of artistic creativity in the aural design of the experience [10].

2.3 Equipment

The equipment used to create the Singing Tree can be categorized into two groups: equipment required to make the Singing Tree function, and equipment used to manufacture the physical station used in the Brain Opera. Equipment in the former group represents the equipment necessary for the Singing Tree to operate and is listed in Table 2.1. Equipment in the latter group includes anything used to create the aesthetic experience in the Mind Forest Space.

2.3.1 Operational Equipment

As shown in Figure 2-1, a participant sings into the microphone, the signal first travels to the preamplifier, then on to the compressor/limiter which prevents clipping (level overload), and finally to the mixer. From the mixer, the voice is channeled through the main output to the computer, and also through an auxilliary output to the effects processor which returns processed voice to the mixer. The computer samples and analyzes the voice, determines the music and video to be output, sends MIDI commands to the Kurtzweil sampler to play music, and displays the correct video bitmap on the LCD screen. The Kurzweil audio out is split with part going to the amplifier and the external speaker, and the other part going to the mixer. Finally, the processed voice and Kurzweil are mixed and channeled to the headphones. The result is that the participant will hear both her own singing voice and the musical accompaniment, while the public will hear only the generated music. This design decision was intended to help alleviate feelings of insecurity regarding public singing.

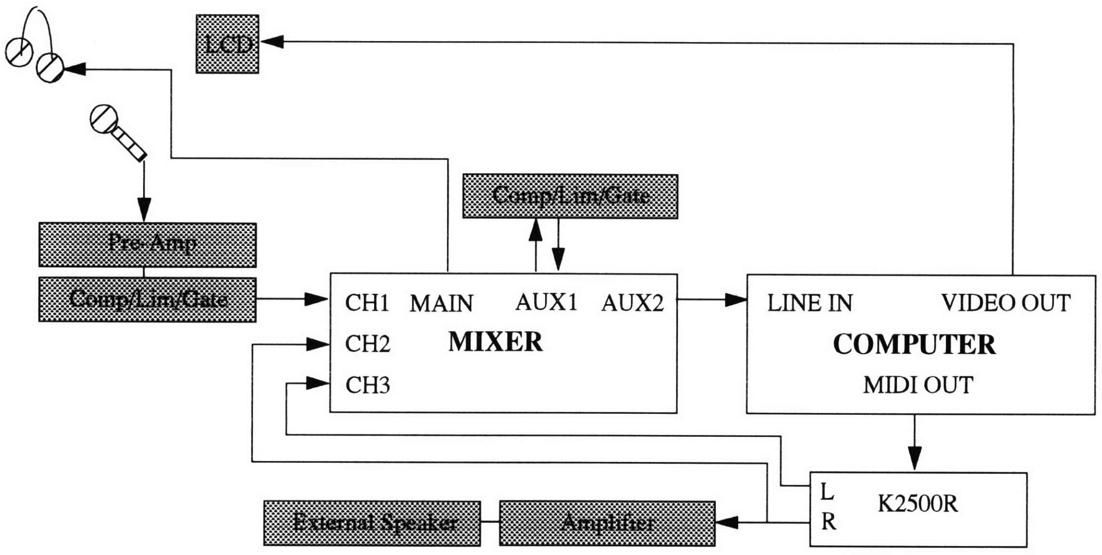

Figure 2-1: Cable Schematic of the Singing Tree

2.3.2 Aesthetic Equipment

Many decisions made during the physical design of the Brain Opera were a collaboration of ideas between the technologists and the visual artists [34], [38], [39]. The Singing Tree, as one sees it today in the Brain Opera Mind Forest, has the microphone, LCD screen, and headphones mounted inside an organic-looking, white hood. The hood resembles an ear, and the participant enters this 'ear' to sing into the microphone. The hood's height is adjustable, and it provides an enclosed surrounding to reduce the participants' feelings of self-consciousness about public singing. The Singing Tree as a unit has the appearance of a tree, large and round at the top with a slender set of three supports comprising the 'tree trunk'. The hardware for the Singing Tree is located on a platform far above the experience and out of sight of the participant. The reader is referred to Figure A-3 for a picture of the Mind Forest which contains several Singing Tree hoods.

The original design for the Singing Tree hood called for the use of a swinging microphone arm which would come to the participant's head when she entered the hood. Small personal speakers would be mounted inside the hood and aimed at the approximate position of the participant's ears. Headphones were not included in the original design, because it was thought that, with the number of people using the Singing Tree, the headphones would get greasy and participants would avoid the experience. There were two problems with this design. First, there were the inherent mechanical difficulties in implementing such a system. Second, it was projected that the Mind Forest would be a very loud space, and the small speakers in the hood would have to compete with the sound entering the hood from the Mind Forest. It was suggested that the microphone be mounted, and that headphones be used rather than small speakers. The headphones would also be mounted, so that the participant would literally enter the headphones when she entered the hood. While this new design was an improvement, the hood's shape turned out to be such that the microphone was 30cm from the participant's head. As stated previously, the Singing Tree was located in a loud interactive space, and it was extremely important to have the participant as close to the microphone as possible. The design was changed again, and, in the final version, the microphone is positioned about 5 cm from the participant's mouth.

Two other designs did not make it to the Lincoln Center debut. The first was an additional umbrella shaped hood which came down over the 'ear' shaped hood and the participant once he was inside. This was another attempt at privacy and sound isolation. The problem was that the two-hooded system was large and mechanically difficult to operate. In addition, the umbrella shaped hood was cut such that it had an opening precisely at the point where the external speaker was located. Instead of isolating the microphone from external sound sources, the hood acted to channel the Singing Tree's generated music back into the microphone, creating an unwanted feedback. The umbrella hood was removed. The other design that did not make it as a Singing Tree accessory because it was unnecessary, but was utilized as a Speaking Tree accessory, was the Sensor Mat. The Sensor Mat was a flat on/off switch connected to an output pin held at +5V and an input pin on the computer's serial port (DB9). When a participant stepped on the switch, the input pin went to +5V and the computer detected that a person was standing on the mat at the experience. Originally designed to accommodate the swinging microphone arm, the floor mat was instead used at the Talking Tree stations to initiate those experiences. [40]

2.4 Kurzweil K2500 Sampler/Synthesizer

The K2500 sampler/synthesizer plays an extremely important role in the Singing Tree operation. In fact, the K2500 plays an important role in the entire Brain Opera, which utilizes a total of 17 units in the Mind Forest and Performance. The K2500 stores the instrument sounds and the recorded vocal samples used to play the music generated by the computer. The work involved in preparing the K2500 for its use in the Singing Tree and the preparation of the samples is described in this section. In addition, a brief introduction to the Musical Instrument Digital Interface (MIDI) protocol is given.

2.4.1 K2500: Overview and Modifications

The K2500R (rack mount) version is used for the Singing Tree. It is a sampler with a proprietary Variable Architecture Synthesis Technology (VAST) which allows one to use samples and internally generated waveforms. It includes several DSP functions to modify the samples, and it has an on-board effects processor. The K2500R is 48-note polyphonic, which is of particular interest to the Singing Tree application, because it requires a sampler with the ability to play many different instruments simultaneously. It comes with 200 preset sound patches and 2 Mb of RAM. It has 8 audio outs, SCSI ports, and MIDI in/out/thru [41].

As shipped, the K2500R did not have sufficient RAM, nor an internal hard drive. Both would be needed for the Singing Tree experience. The author installed 64 Mb RAM and a Fujitsu 540 Mb internal hard drive in each of the three Singing Tree K2500R units. An interesting implementation in the Kurzweil design is that the SCSI connector pins on the internal SCSI board are backwards with respect to conventional designs. In other words, attaching the internal hard drive via a conventional SCSI cable (SCSI1 cable for internal applications) did not work. The design was likely implemented to require users to buy a proprietary Kurzweil SCSI1 cable. Our solution was to cut off the guide key from a conventional cable such that it could be inserted backwards.

2.4.2 Vocal Samples: Recording and Editing

The vocal samples used for the Singing Tree were performed by the Boston Camerata, who specialize in the singing styles of Renaissance and pre-Renaissance music. If one listens to interpretations of the works of Leonin and Perotin (ca. 11th-12th century), the singing style is a very pure tone with no vibrato. This was the 'ideal' style of voice for the Singing Tree.

The vocal samples were to represent a chorus in the final version of the Singing Tree, and so a soprano, alto, tenor, and bass were sampled to cover the range of pitches a choir would cover. The specific motifs they sang were composed by Tod Machover and divided into three categories:

1. Basic Motif: one pitch held for approximately two measures; singing style is pure
2. Intermediate Motif: one to several pitches sung for approximately two measures, singing style to vary from pure to slightly agitated, consistent vowel and rhythmic structure.
3. Frenetic Motif: one to several pitches sung for approximately two measures, singing style is agitated, vowel is frenetically changing, inconsistent rhythmic structure

Conceptually, the Basic Motif represents the singing style of the chorus when the participant sings purely; the Intermediate Motif represents the singing style when the participant is singing consistently the same pitch, but not in a completely pure manner; the Frenetic Motif represents the singing style of the chorus when the participant is not singing according to the established goals of the Singing Tree.

The six Boston Camerata singers were recorded by professional sound engineers from Erato Disques in Weston, Massachusetts. Each motif had several versions, and each version was recorded at minor-third intervals throughout the range of each singer for the vowels [ah] as in 'father', [ee] as in 'feet', [oo] as in 'pool', and [oh] as in 'so'. [42] The problem with recording in this manner is that the singers' pitch is not perfect, and relative pitch from sample to sample and person to person varied significantly. In addition, sample attacks and volumes varied greatly. The result was a very large number of samples which had to be screened and edited.

Sample editing is a time consuming process. Although the K2500R has on-board sample editing options, the samples were all recorded to computer, and there are several excellent sample editing software packages available. It made a great deal of sense to edit first and then port to the K2500R only the samples to be used in the Singing Tree, rather than port all the raw data first and then edit. The process required two programs: Sound Designer and Alchemy. Using Sound Designer, the samples were first normalized in amplitude. The next step was to clean up the attacks of the samples. After cutting the empty padding at the start of a particular sample, an amplitude envelope with a sharp rise was applied at the beginning of the sample. This eliminated any initial pre-attack noise and clicking that might have resulted from cutting too little or too much empty padding. The next problem was the variance in sample lengths. Since each sample was recorded separately, tempo varied considerably from sample to sample. Because the samples were to be played together, the sample lengths had to be normalized to a consistent value for each of the three motifs. The amplitude envelope was held constant after the attack, and a decay with a relatively slow decay rate was applied at the end of the sample. The placement of the decay determined the length of the sample, and the same amplitude envelope was used for all samples of a particular motif. Alchemy was used to change the pitches of the samples, and this gave a rough tuning. The fine-tuning was done by ear, similar to the way a piano is fine-tuned. Chords of vocal samples were played along with chords of instruments of known pitch, and fine adjustments were made to create the final tuning. With this finished, the best samples were selected and ported to the K2500R via SCSI2. Our budget allowed only 64 Mb RAM, and this was fully utilized. Unfortunately, many good samples could not be used.

The next step in preparing the samples was to create a keymap for each of the classes of motif and vowel. A keymap is an assignment of samples in the K2500R's memory to the keys on a 'keyboard'. The word 'keyboard' is in quotes, because the K2500R is a rack mount and does not have a keyboard. However, it does use MIDI and can be operated like a keyboard via MIDI commands. For a given motif, the keyboard was laid out from D1 to E5, running through the bass notes in the lowest register, through the tenor, alto, and finally soprano at the highest register. Due to memory constraints, it was not feasible to have one sample for each note on the keyboard, and so interpolation was used. The Singing Tree keymaps use approximately one sample for every +/- two halfsteps. Although interpolated samples are literally played back at the 'wrong speed', the effect is not noticeable over two halfsteps. The last step was to apply reverberation and delay to the vocal samples to 'soften' them. The samples were recorded in a dry environment, and the addition of effects made the choir sound much more realistic. These effects were K2500R local effects. They did not change the sample data, but rather were applied to the audio before the output from the K2500R [43].

2.4.3 A Short Digression into MIDI

Thus far, I have been using the term MIDI without introducing what the acronym means. MIDI stands for Musical Instrument Device Interface, and it is a communications protocol used primarily in music and video production to link computers, electronic musical instruments, video equipment, and controllers [44]. MIDI allows the transfer of messages or commands from one device to another. More specifically, a device such as a computer can be used as a master to drive several other slave devices which are linked via MIDI. MIDI utilizes 16 independent channels, and messages can be sent along all 16 channels. One of the advantages of MIDI is that it was designed for real-time performance and has very low latency [45]. MIDI can be used to control most functions of a musical instrument including, but not limited to [44],

- Over 10 octaves of discrete pitch control

- Velocity and pressure dynamics for each note
- Pitch bending
- A group of 95 real-time controllers such as volume, modulation, and sustain
- Instrument and/or program selection
- Sample data transfer

For each of the controllable parameters, a unique set of MIDI messages is defined. The message definitions comprise the protocol which all MIDI instruments use when transmitting or receiving MIDI information. Simply stated, the MIDI message is a stream of numbers which indicate which parameter is to be changed and the value to which it is changed. For example, the MIDI message used to change an instrument is a number (or set of numbers) referring to the location of the sound patch in the slave. The values for parameters such as volume, on the other hand, typically run from 0-127 [44]. The Singing Tree's music is generated at the computer, and it is via MIDI that the computer 'tells' the K2500R what to play. This will be discussed in more detail in Section 3.3.

Chapter 3

Singing Tree Implementation: Technical and Artistic Issues

3.1 Technical Decisions in an Artistic Context

This chapter will present the technology and artistry behind the Singing Tree. While these topics could be discussed in separate chapters, I have decided to include them in the same chapter. In my opinion, the Singing Tree project is an artistic project which is realized through the appropriate use of technology. While the technology is what makes the Singing Tree function, it is the artistry which makes it interesting. After all, it is the artistic content which makes the Singing Tree experience a success or a failure. Many of the technical decisions, particularly the musical mappings, were intimately related to the artistic goals of the project. As a result, the technology and artistry are presented together with their relationship clarified where appropriate.

3.2 Voice Analysis

The voice analysis algorithms used in the Singing Tree measure pitch, noisiness (ambiguity), brightness, energy, and vowel formant. These algorithms were written by Eric Metois, four of which were part of his DSP toolkit used in several other projects and his Ph.D. thesis [30]. In this section, I will describe how these algorithms work and the information they send to the Singing Tree. I will also discuss alternatives where applicable.

3.2.1 Volume

The volume parameter used in the Singing Tree is the energy of the signal on a logarithmic scale [30]. This is not to say that loudness, as perceived by humans, matches this algorithmic estimate. While an engineer's measuring devices are sensitive to energy in a direct manner, the ear and the microphone are not [42]. This is the fundamental difference between the 'energy' of a sound, a physical parameter, and its 'loudness', a perceptual parameter. Ears and microphones operate by detecting differences in air pressure. The fluctuations in air pressure above and below standard atmospheric pressure, at a rate which is within a certain range of frequencies, lead to an audible sound. The human auditory system is extremely sensitive, able to detect sounds corresponding to pressure changes as low as 3.53 billionths standard atmospheric pressure, and as high as 1 million times that amount (threshold of pain) [42]. However, this enormous variation in air pressure does not correspond to a variation in loudness of 1 million times. Thus, loudness is considered a logarithmic function of signal amplitude. Using a dB scale, one can write the volume (energy) of a signal relative to a standard, such as 1 mV in the dBmV scale, or 1 mW in the dBm(W) scale. Of course, loudness is frequency dependent, while signal amplitude is not. The perceived loudness of a signal is highest around 3000 Hz, and approaches indiscernible at 30 Hz and 20000 Hz [42]. In addition, although loudness is logarithmic in nature, it is not strictly a logarithmic function. Thus, loudness is more typically measured in units called *sones* which are defined as follows: A person with healthy hearing sits in an anechoic chamber facing a distant loudspeaker, and she will hear a sound whose loudness is 1 sone when a source having a frequency of 1000 Hz produces a sound pressure level of 40 dB at her ear. A rule of thumb is that a 10 dB increase in sound pressure level corresponds to a doubling of the loudness in sones. However, it should be noted that, especially in musical examples, loudness is typically *additive* while decibels are not. In other words, a sound source of 2 sones played simultaneously with another of 3 sones will be perceived as a single sound source of 5 sones. Figure 3-1 shows a loudness diagram measured in sones over the audible frequency range.

3.2.2 Pitch Estimation

Pitch estimation appears to be a rather simple task on the surface, but there are many subtleties that need to be kept in mind. While there are many successfully implemented pitch estimation algorithms, none work without making certain assumptions about the sound being analyzed. Indeed, the notion of 'pitch' is rather ill-defined. In the context of algorithmic estimation, finding the 'pitch' typically refers to estimating a sounds fundamental frequency or period. However, humans can identify pitch in sounds which have a slightly varying period, or even in ones which lack a fundamental frequency [30]. The human ear and nervous system can make pitch estimations from complex sounds by finding patterns, seeking subsets of 'almost harmonically related' components, and making 'best estimates' [42]. In fact, it has been shown that the manner in which the brain operates on sensory data is oftentimes analogous to a 'maximum likelihood estimation' [47]. Some examples of the ear's extraordinary ability to perceive pitch are noted below [42].

- Tuning Fork and Glockenspiel Bar: Striking either the tuning fork or the glockenspiel softly will excite a single sinusoidal oscillation. However, striking either with a hard hammer will excite two or more characteristic frequencies. There is no particular harmonic relationship between the characteristic

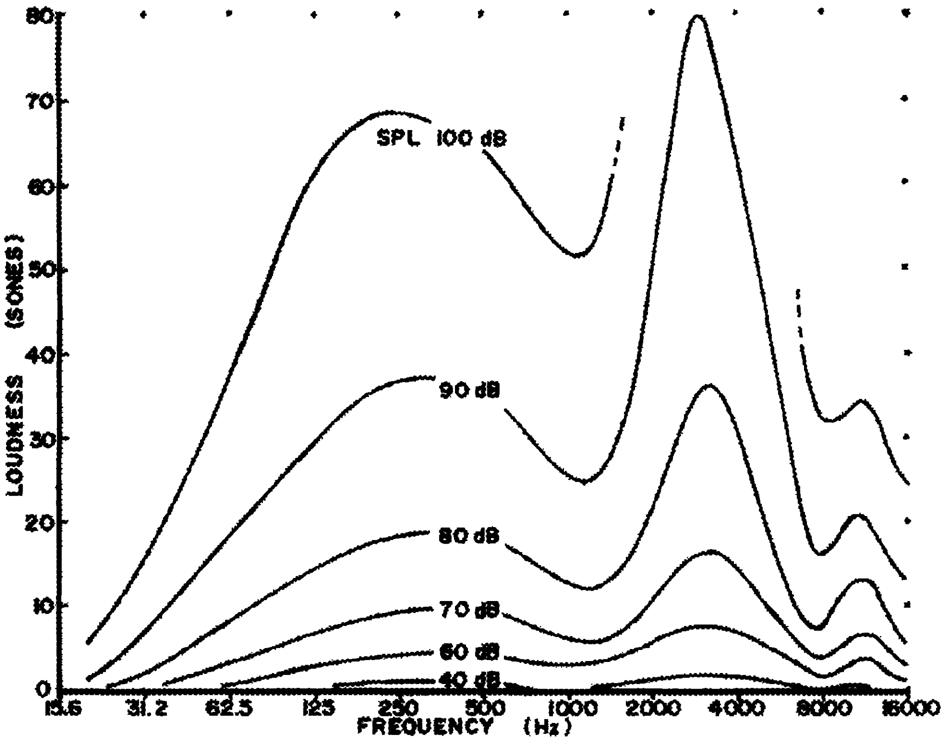

Figure 3-1: Perceived Loudness of Single-Component Sounds as a Function of Frequency

frequencies; they are far apart, and the human ear perceives two or more sounds with different pitches.

- Plucked or Struck Musical Strings: Plucking or striking a musical string provides a large number of partials arranged in a nearly exact integer relationship, approximately matching the exact harmonic prototype. Due to slight departures from the integer relationship, a mathematical calculation would lead to a repetition frequency of 2-3 Hz. However, the human auditory system identifies the frequency which most closely aligns with the harmonics as the fundamental frequency, and that is interpreted as the pitch of the sound.

- Clock Chimes: There are several quasi-harmonic patterns in this complex sound. People hear the pitch of the sound depending on which harmonic pattern they can derive. Once a person attaches a particular pattern to the sound, changes in the relative amplitudes of the harmonic components do not change the perceived pitch. In other words, even though the harmonics on which the pattern is established are changing, the auditory system maintains its original pattern interpretation and pitch.

- Church Bells: Experiments with church bells confirm that humans do not discriminate harmonic structure by the relative amplitudes of the components, but, rather, by the frequencies present and the harmonic patterns they form.

- Suppressed Harmonic Components: Experiments [42] in which all but two or three harmonic components were suppressed reveal that humans can perceive sound as an anticipated or implied pattern projected onto the harmonic components. An analogous situation occurs with the human visual system. A pattern can be reconstructed from an incomplete picture such that a person can visualize the entire pattern.

In addition, humans have an amazing ability to extract patterns from extremely noisy sensory input. One of the first signs of hearing loss is the loss of ability to differentiate the voice of an acquaintance from a large crowd of people (i.e., the loss of ability to extract the desired pattern from the noisy input).

With these complexities in the human auditory system, one might assume that our ability to estimate pitch is indeed faster than that of a computer. However, this is not necessarily true [30]. Both humans and computers must analyze at least a full period of 'clean' sound (i.e., sound without distortion from the attack) in order to detect the fundamental frequency. In this, there is ambiguity in pitch estimation. Both humans and computers will make estimates of the pitch through the attack and during the first period, with each successive guess, ideally, becoming more reliable. Thus, ambiguity in pitch becomes a parameter or characteristic of the pitch estimation process. It is in this ambiguity parameter that a human auditory system is superior to a computer. Humans possess the ability to incorporate new information over time into the estimation process, knowing what to accumulate and what to throw away, whereas the computer may not be able to do the same in real-time [30].

The pitch estimation algorithm used assumes a monophonic signal, such as a human voice. It follows from this assumption that the signal has a periodicity corresponding to the fundamental frequency or pitch. The algorithm, as derived below, is from [30] with minor additions for clarity, and the technique upon which it is based, cross-correlation, is commonly used for pitch estimation. The advantage to the form used in [30] is that it leads naturally to a definition for the ambiguity parameter.

Starting with a signal $s(t)$ that is assumed to be periodic, or more precisely, quasi-periodic, it follows that $s(t) \approx as(t+T)$ where T is the quasi-period of the signal and the scalar a accounts for inevitable amplitude changes. Considering a window of d samples of $s(t)$ taken with a sampling period, τ, one can define the vector, $\mathbf{v}(t) = [s(t), s(t+\tau), ..., s(t+(d-1)\tau)]^T$. Two such windows, $\mathbf{v}(t)$ and $\mathbf{v}(t+T')$, are separated in time by T'.

Using a Bayesian approach detailed in Appendix B, the best estimate for the quasi-period is the T which maximizes the expression

$$\lambda(t, T') = \frac{\mathbf{v}(t)^T \mathbf{v}(t+T')}{\| \mathbf{v}(t) \| \, \| \mathbf{v}(t+T') \|} \tag{3.1}$$

which is a 'normalized' cross correlation between $\mathbf{v}(t)$ and $\mathbf{v}(t+T')$. This is what one expects intuitively. $\mathbf{v}(t)$ can be thought of as a fixed window and $\mathbf{v}(t+T')$ as a sliding window with the cross correlation of the two windows a maximum at $T' = T$. Of course, the use of a sliding window in a computer algorithm implies a discrete time implementation, which can be written

$$\lambda[n,k] = \frac{\mathbf{v}[n]^T \mathbf{v}[n+k]}{\| \mathbf{v}[n] \| \, \| \mathbf{v}[n+k] \|}. \tag{3.2}$$

This equation can be recast in a more compact notation by defining $\mathbf{u} \equiv \mathbf{v}[n=0] = [s[0], s[1], ..., s[d-1]]^T$ to be the fixed window (the first sample in the fixed window is arbitrarily assigned to $n = 0$) and $\mathbf{v}[k] \equiv \mathbf{v}[n+k]|_{n=0} = [s[k], s[k+1], ..., s[k+d-1]]^T$ to be the sliding window. Equation (3.2) becomes

$$\lambda[k] = \frac{\mathbf{u}^T \mathbf{v}[k]}{\| \mathbf{u} \| \, \| \mathbf{v}[k] \|}, \tag{3.3}$$

and the best estimate for the pitch given this discrete cross-correlation is $T_0 = N_0 \tau$, where τ is the sampling period and $k = N_0$ is the smallest integer which maximizes equation (3.3).

It is not necessary to compute this function for every k. It is sufficient to align the fixed window $\mathbf{u}$ with a local maximum of $s(t)$ and then choose several candidates from the sliding windows $\mathbf{v}[k_1], \mathbf{v}[k_2], ...$*etc.*, which are aligned with local maxima within an amplitude range, δ, of the maximum associated with $\mathbf{u}$. This works very well and provides a relatively small set of candidates. The process for resolving local maxima is greatly aided if the signal, $s(t)$, is first low-pass filtered to reduce high-frequency anomalies [30].

As stated above, T_0 is the closest estimate to the actual period, T, as calculated using the discrete cross-correlation. In audio applications, T_0 is typically not a sufficient estimate, because the error is too large. Consider a pitch of frequency, f. It is known that the octave in an equally tempered tuning system is divided into twelve semitones (also known as half-steps) which are logarithmically spaced by the factor $2^{\frac{1}{12}}$ [42]. This implies that two frequencies, f and f_1, which are separated by a semitone are related by the expression,

$$f_1 = 2^{\frac{1}{12}} f \tag{3.4}$$

Where it is assumed that $f_1 > f$. Defining $\Delta f = f_1 - f$ and using equation (3.4) gives the expression,

$$\frac{\Delta f}{f} = 2^{\frac{1}{12}} - 1 = 0.05946 \tag{3.5}$$

Since $T_0 = N_0 \tau$ is the closest estimate to T, it is known that $|T - T_0| \leq \tau$. The error in the frequency domain is given by [30],

$$\frac{\Delta f}{f} = \frac{\left| \frac{1}{T} - \frac{1}{T_0} \right|}{\frac{1}{T}} = \frac{|T - T_0|}{T_0} \leq \frac{\tau}{N_0 \tau} \implies \frac{\Delta f}{f} \leq \frac{1}{N_0} \tag{3.6}$$

This implies that for $N_0 \leq 20$ the error in frequency will approach that of a semitone! For reference, a semitone is 100 cents (abbreviated by 'ct'), and humans can detect differences in pitch on the order of 10-20 ct. The integer estimate N_0 will have to be further resolved by finding a β_0 such that $N_0 + \beta_0$ lies between the $k = N_0$ and $k = N_0 + 1$ samples and better approximates the actual period. In other words, the goal is to interpolate between the window vectors at $\mathbf{v}[N_0]$ and $\mathbf{v}[N_0 + 1]$ to find an interpolated vector, $v_{interp}(N_0, \beta_0)$, such that

$$\lambda_{max} = \lambda(N_0, \beta) \mid_{\beta=\beta_0} = \frac{\mathbf{u}^T \mathbf{v}_{interp}(N_0, \beta_0)}{\| \mathbf{u} \| \, \| \mathbf{v}_{interp}(N_0, \beta_0) \|}, \quad \beta \in [0, 1]. \tag{3.7}$$

Interpolating to find a β_0 which maximizes equation (3.7) will lead to a best interpolated estimate for the period, T_{interp}. It was found that linear interpolation was not only a sufficient method, but also a computationally inexpensive one, because it requires no further iteration on $\lambda(N_0, \beta)$. Considering a plane, $\mathbf{\Pi}$, spanned by the vectors, $\mathbf{v}_1 = \mathbf{v}[N_0]$ and $\mathbf{v}_2 = \mathbf{v}[N_0 + 1]$, the best least-squares estimate for β_0 will correspond to the projection, $\mathbf{p}$, of $\mathbf{u}$ onto $\mathbf{\Pi}$ [30],[48]. In other words,

$$\begin{aligned} \mathbf{p} \parallel \mathbf{v}_{interp}(N_0, \beta_0), \quad \textit{where} \quad \mathbf{v}_{interp}(N_0, \beta_0) &= (1 - \beta_0)\mathbf{v}[N_0] + (\beta_0)\mathbf{v}[N_0 + 1] \\ &= (1 - \beta_0)\mathbf{v}_1 + (\beta_0)\mathbf{v}_2 \end{aligned} \tag{3.8}$$

If one writes $\mathbf{p} = p_1\mathbf{v}_1 + p_2\mathbf{v}_2$, then it follows from equation (3.8) that p_1 and p_2 are simply proportional to $(1 - \beta_0)$ and β_0 through the same proportionality constant. Thus, one can write

$$\frac{p_1}{(1 - \beta_0)} = \frac{p_2}{\beta_0} \quad \Longrightarrow \quad \beta_0 = \frac{p_1}{p_1 + p_2}. \tag{3.9}$$

The vector, $\mathbf{p}$, is the least squares solution to $\mathbf{Ap} = \mathbf{u}$, where the column space of $\mathbf{A}$ spans the plane $\mathbf{\Pi}$.

$$\mathbf{p} = \begin{bmatrix} p_1 \\ p_2 \end{bmatrix} = (\mathbf{A}^T\mathbf{A})^{-1}\mathbf{A}^T\mathbf{u}, \quad \textit{where} \quad \mathbf{A} = \begin{bmatrix} | & | \\ \mathbf{v}_1 & \mathbf{v}_2 \\ | & | \end{bmatrix} \tag{3.10}$$

This result for $\mathbf{p}$ yields a solution for β_0 via equation (3.9) [30], [48],

$$\beta_0 = \frac{(\mathbf{v}_2^T\mathbf{u}) \| \mathbf{v}_1 \|^2 - (\mathbf{v}_2^T\mathbf{v}_1)(\mathbf{v}_1^T\mathbf{u})}{(\mathbf{v}_2^T\mathbf{u})[\, \| \mathbf{v}_1 \|^2 - (\mathbf{v}_1^T\mathbf{v}_2)\,] + (\mathbf{v}_1^T\mathbf{u})[\, \| \mathbf{v}_2 \|^2 - (\mathbf{v}_1^T\mathbf{v}_2)\,]} \tag{3.11}$$

With β_0 in terms of known parameters, the best estimate for the actual period, T, is written

$$T = T_{interp} = (N_0 + \beta_0)\tau. \tag{3.12}$$

The above method works assuming that N_0 is the integer part (floor) of the discrete period. If it is not, then $\beta \notin [0,1]$ and the window must be shifted accordingly such that N_0 is the integer part (floor) of the discrete period. Specifically, for $\beta_0 < 0$, consider a new plane spanned by the vectors $\mathbf{v}_1$ and $\mathbf{v}_2$ shifted down by one sample (i.e., $\mathbf{v}_1 = \mathbf{v}[N_0 - 1]$ and $\mathbf{v}_2 = \mathbf{v}[N_0]$) and project $\mathbf{u}$ onto this new plane to solve for a new β_0. Conversely, for $\beta_0 > 1$, consider a new plane spanned by the vectors $\mathbf{v}_1$ and $\mathbf{v}_2$ shifted up by one sample (i.e., $\mathbf{v}_1 = \mathbf{v}[N_0 + 1]$ and $\mathbf{v}_2 = \mathbf{v}[N_0 + 2]$) and re-project $\mathbf{u}$ onto this new plane to solve for a new β_0. If there is still no $\beta_0 \in [0,1]$ then, depending on the real-time constraints, this shifting should be continued, or a fresh segment of signal should be analyzed [30],[50].

3.2.3 Noisiness

The pitch estimation technique used for the Singing Tree is an interesting implementation, because it naturally leads to a definition of pitch ambiguity and noisiness. It is this point primarily which makes the above technique preferable to alternative solutions, such as the chirp-z transform or the cepstrum which will also give high frequency resolution. The ambiguity in pitch arises in both humans and computers because of the real-time nature of pitch estimation and the inharmonic attack transients associated with most sounds.

It is shown in [30] that this ambiguity in pitch is directly related to the signal to noise ratio (SNR) and the maximization of the cross-correlation used in the pitch estimation implementation. Specifically, considering again a fixed window of the signal $\mathbf{u}$, a sliding window of the signal $\mathbf{v}$, the amplitude scaling factor $a(t)$, and introducing additive noise $\mathbf{e}(t)$, one can write

$$\mathbf{u} = a(t)\mathbf{v}(t) + \mathbf{e}(t). \tag{3.13}$$

The additive noise, $\mathbf{e}(t)$, is the error between the windows $\mathbf{u}$ and $\mathbf{v}(t)$ at time t, and it will decrease as $t \longrightarrow T$, the period of the signal [30]. Defining the signal to noise ratio,

$$SNR(t) = \frac{\| \mathbf{u} \|^2}{\| \mathbf{e}(t) \|^2} = \frac{\| \mathbf{u} \|^2}{\| \mathbf{u} - a(t)\mathbf{v}(t) \|^2} = \frac{\| \mathbf{u} \|^2}{\| \mathbf{u} \|^2 - 2a(t)(\mathbf{u}^T\mathbf{v}(t)) + a^2(t) \| \mathbf{v} \|^2} \tag{3.14}$$

one can maximize the signal to noise ratio with respect to $a(t)$ (i.e., minimize the denominator) as demonstrated in equation (B.2) with the results repeated here.

$$\frac{\partial}{\partial a(t,T)} (\| \mathbf{u} \|^2 - 2a(t)(\mathbf{u}^T\mathbf{v}(t)) + a^2(t) \| \mathbf{u} \|^2) = 0 \qquad \Longrightarrow \qquad a(t) = \frac{\mathbf{u}^T\mathbf{v}(t)}{\| \mathbf{v}(t) \|^2} \tag{3.15}$$

Substituting this relation for $a(t)$ into equation (3.14) yields,

$$SNR(t) = \frac{1}{1 - \frac{\|\mathbf{u}^T\mathbf{v}(t)\|^2}{\|\mathbf{u}\|^2\,\|\mathbf{v}(t)\|^2}} \implies \frac{1}{1 - \frac{\|\mathbf{u}^T\mathbf{v}[n]\|^2}{\|\mathbf{u}\|^2\,\|\mathbf{v}[n]\|^2}} = \frac{1}{1-\lambda^2[n]} \tag{3.16}$$

which demonstrates the relationship between between maximization of the SNR and maximization of the normalized cross-correlation. The maximization of the normalized cross-correlation will estimate the period of the incoming signal, and the SNR will also be maximized at this value. In that sense, the evolution of the SNR acts as a reliability measurement. However, the relationship is more intrinsic. The SNR is also a measurement of timbre, in the sense that sounds with an identical pitch but different timbre will have different values of SNR. For example, a lower SNR may imply additive noise in the pitched sound while a higher SNR may correspond to more pure sound. Furthermore, consider an ideal noiseless channel from the sound source to the computer. The implied additive noise from a low SNR measurement is strictly an implication. The actual origin of the differences in SNR values in this ideal case are characteristics of the various sounds. These characteristics are what make sounds different, and are therefore a measurement of timbre. In fact, timbre is defined as the characteristics of a sound which allow humans to differentiate sounds of identical pitch. It is thought that Metois is the first to use this notion of pitch noisiness or ambiguity as a basic measurement for musical characterization [30].

3.2.4 Brightness

The concept of brightness as used in the Singing Tree is an algorithm based on the works of David Wessel [30]. Simply stated, Wessel defines a sound's brightness in terms of the energy distribution of the signal's harmonics. In other words, given that the fundamental frequency of the incoming monophonic signal has been determined, the brightness can be parameterized in terms of the size of the frequency gap from the signal's fundamental frequency to its center frequency. In this case, the signal's center frequency is the weighted average of the frequencies present in the signal. In general, signals with a larger frequency gap will sound 'brighter' than those with a smaller frequency gap.

A first approach might be to determine the DFT (discrete Fourier transform) and derive the center frequency. After all, a standard method for determining a sampled signal's frequency representation is to take the DFT via the FFT (fast Fourier transform) algorithm. However, as explained through example in [30] and mathematically in [49], one can easily misinterpret the results of a DFT. Having misinterpreted a DFT once or twice before, this is a subject to which the author has given much thought. The main source of error is in assuming that the frequency response one calculates and a simple 'connect-the-dots' version represent the same signal; they do not. The issue is frequency representation versus frequency estimation. To give a more relevant example, if one is trying to find the period of a signal by peak-picking the fundamental frequency from the DFT, one needs to remember that the DFT is merely a sampled version of the DTFT (discrete time Fourier transform). This implies that an estimate based on the peak frequency point of the DFT, f_{peak}, has a tolerance, Δf, attached to it (as do all frequency points in the DFT) of

$$\Delta f = F_s / N_{FFT}, \quad \Longrightarrow \quad f_{actual} = f_{peak} \pm \Delta f \tag{3.17}$$

where F_s is the sampling frequency and N_{FFT} is the number of points in the DFT. Note that the number of points in the DFT is written N_{FFT} in anticipation of calculating the DFT using the FFT algorithm.

As an example, consider the following scenario. A frequency analysis of the singing voice is to be done in real-time using the FFT algorithm. In this case, assume that real-time implies a Δt window size of 15 ms. In other words, a 15 ms chuck of signal is sampled, and while it is being analyzed, the next 15 ms chunk of signal is being sampled to the buffer. This in turn constrains the computation time of the analysis to 15 ms. This constraint is chosen because the human ear also works in real-time on a time scale of $\sim$ 15 ms [30],[42],[50]. The next constraint is the tolerance of error that is aurally allowed. In equation (3.5), a semitone was shown to give a $\Delta f / f$ of 0.05946, and humans can distinguish changes in frequency on the order of 0.1 $\sim$ 0.2 semitones (10 $\sim$ 20 ct). In addition, it is known that, in general, the range in pitch frequency of human singers is approximately 60 $\sim$ 1600 Hz [42]. Assuming the lowest possible pitch is 100 Hz, the highest possible pitch is 1500 Hz, and a 'loose' tolerance of 0.2 semitones, Δf becomes,

$$\frac{\Delta f}{f} = 2^{\frac{0.2}{12}} - 1 = 0.01162 \quad \Longrightarrow \quad \begin{array}{lcccc} \Delta f\,|_{f=100Hz} & = & (100)(0.01162) & \approx & 1.2Hz \\ \Delta f\,|_{f=1500Hz} & = & (1500)(0.01162) & \approx & 17.4Hz \end{array} \tag{3.18}$$

For each case, any error in frequency larger than Δf is unacceptable. It is also known that the upper bound on the frequency of the first three vowel formants is less than 5000 Hz [42]. Since one anticipates attempting some sort of algorithm to determine vowel structure, the lower bound on the sampling frequency is about 10000 Hz. Considering this lower bound of $F_s = 10000Hz$ and the conventional 'upper bound' for audio, F_s = 44100 Hz (CD quality), one can demonstrate the need for caution in the direct application of a DFT. For each of the four cases, the DFT lengths, N_{FFT}, required to meet the established tolerances are,

$$\begin{array}{lcccccr} N_{FFT}\,|_{F_s=10000Hz, f=100Hz} & = & \frac{F_s}{\Delta f} & = & \frac{10000}{1.2} & \approx & 8300\,points \\ N_{FFT}\,|_{F_s=10000Hz, f=1500Hz} & = & \frac{F_s}{\Delta f} & = & \frac{10000}{17.4} & \approx & 575\,points \\ N_{FFT}\,|_{F_s=44100Hz, f=100Hz} & = & \frac{F_s}{\Delta f} & = & \frac{44100}{1.2} & \approx & 36750\,points \\ N_{FFT}\,|_{F_s=44100Hz, f=1500Hz} & = & \frac{F_s}{\Delta f} & = & \frac{44100}{17.4} & \approx & 2600\,points \end{array} \tag{3.19}$$

Since this is a worst case analysis, the larger windows must be used at each sampling frequency.

$$i.e. \quad \begin{array}{lccr} worst\ case\ @\ F_s = 10000Hz: & N_{FFT}\,|_{F_s=10000Hz, f=100Hz} & \approx & 8300\,points \\ worst\ case\ @\ F_s = 44100Hz: & N_{FFT}\,|_{F_s=44100Hz, f=100Hz} & \approx & 36750\,points \end{array} \tag{3.20}$$

Thus, depending on the sampling frequency one chooses to use, the DFT-size required to meet the established tolerance would need to be in the range from 8300 points to 36750 points. In the first case, F_s = 10000 Hz implies a sampling period $T_s = 0.1$ ms, and the number of samples in the Δt window is only $N = 15/0.1 = 150$. In the second case, $F_s = 44100$ Hz implies a sampling period, $T_s = 0.023$ ms, and the number of samples in the Δt window is $N = 15/0.023 = 650$. Thus, in both cases, the real-time condition will only give approximately 1/50 the required number of points. Zero padding the signal and then determining the DFT is a first step, but one must keep in mind the following:

- There is really only one type of transform that one is concerned with in real-time applications, the FEFT (fast enough Fourier transform). Longer windows imply a need for faster hardware as one approaches the established latency limit for real-time operation.
- The inherent trade-off is window size vs. upper frequency limit. A sampling frequency of 10000 Hz yields not only the minimum window size, but also the minimum Nyquist frequency. While adequate for the first three vowel formants, information on fricatives or other high frequency information is lost. The more important the high frequency components of the signal become, the longer the DFT window required.
- Luckily, the definition of 'real time' in this case is 15 ms; it takes a full 10 ms simply to get one period of the 100 Hz signal. As one decreases the lower bound on frequency, one must be careful not to cut off the signal before one period has arrived. Note that 60 Hz has a period of 16.6 ms, larger than the established real-time constraint.

The situation is not hopeless, however, because the analysis was worst case. There are many options to improve results from the choice of windowing function to a periodogram or chirp-Z transform method. Nonetheless, in considering improvements, one must also consider the real-time nature of the problem and make trade-offs accordingly. Alternatively, one can try to reduce the restrictions of the constraints, and of all the constraints, the one that can be dramatically improved is Δf. The largest values of Δf occur when the number of periods represented in the windowed signal on which the DFT is applied happens to be non-integer. In other words, one can make the DFT samples fall on the fundamental frequency and its integer ratios if one windows the signal such that an integer number of periods falls within the window. Furthermore, doing so eliminates the need for a special (i.e., non-rectangular) windowing function. However, it is a catch-22. One needs to know the pitch to apply the correct length window, but this algorithm is used to determine several metrics, including pitch. Nonetheless, if one has *a priori* knowledge of the pitch, one can greatly reduce the window size. This method is known as ***pitch-synchronous frequency analysis,*** an idea first used in analysis of music by Michael Portnoff in a phase vocoder [30], but can be traced back much earlier in the analysis of voiced sounds [51].

In the Singing Tree, the time-domain, cross-correlation algorithm was used to determine the pitch, avoiding many of the issues associated with a frequency-domain pitch estimation. Once the pitch was determined, the signal window size could be adjusted such that an integer number of periods fit within it, and a pitch-synchronous frequency analysis performed. The pitch-synchronous frequency analysis was used primarily to estimate the center frequency used in determining the brightness parameter. As described in [30] by example, an 'asynchronous' estimate for the center frequency of an incoming signal can give widely varying results, while a 'synchronous' estimate is far more accurate. The specific case considered was a signal with pitch frequency $f=$ 90 Hz, window size N_{FFT} = 512, and sampling frequency F_s = 44100 Hz. Note that the window size does not represent an integer number of signal periods (an 'asynchronous' analysis). Several DFT estimates of the center frequency were considered for several starting points of the window, each lagging the previous by only 0.1 ms. The result was a range of center frequency estimates from 425 Hz to 640 Hz. The actual center frequency, as determined by a long-term frequency analysis, was estimated to be 551 Hz. As determined by a single, pitch-synchronous frequency analysis, the center frequency was estimated to be 542 Hz [30]. This simple example demonstrates the need for a long-term frequency analysis, or alternatively, a pitch-synchronous short-term analysis to acquire accurate results.

3.2.5 Formant Estimation

Formant analysis can be traced back to the 1950's, and there are numerous techniques available, some more exotic than others [53], [42]. Most of the early work can be regarded as frequency domain techniques, such as 'peak-picking' spectral peaks in the short-time amplitude spectrum, or 'analysis by synthesis' in which one generates a best match to the incoming signal. Later, techniques were developed to 'peak-pick' a ceptrally-smoothed spectrum or a linear predictive coding (LPC) spectrum, or to find the roots of the LPC polynomial. Other techniques that were reviewed included formant tracking using hidden Markov models and vector quantization of LPC spectra, quasilinearization, Kalman filters, and energy separation [54],[55],[56],[57],[58],[59]. In the end, we decided to use a formant estimation algorithm developed by Rabiner and Schafer based on cepstrum analysis and the chirp-z transform [53]. This method was chosen primarily for two reasons: First, it was feasible to implement in real-time given the other operations, particularly video, that also had to occur within the latency window; second, the technique dealt with two topics, cepstrum analysis and the chirp-z transform, in which the author was interested. To allow the algorithm to work seamlessly with his DSP toolkit, Eric Metois coded the version used in the Singing Tree.

Modeling Human Speech

Speech production can be modeled as a lumped parameter, linear, quasi-time-invariant system. Given this model, a speech waveform can be produced through an excitation of a series/parallel connection of resonators. The complex natural frequencies of the resonators are assumed over short-time to be constant, but vary slowly and continuously over long-time to approximate the time-varying eigenfrequencies of the vocal tract. The excitation which drives the system can be random noise for unvoiced sounds, a quasi-periodic pulse train for voiced sounds, or a combination of the two for voiced fricatives. In the Singing Tree, the analysis assumes a quasi-periodic pulse train as the source of the excitation. The excited resonant frequencies are the formant frequencies, and knowledge of their values can help one derive the formant structure of a signal

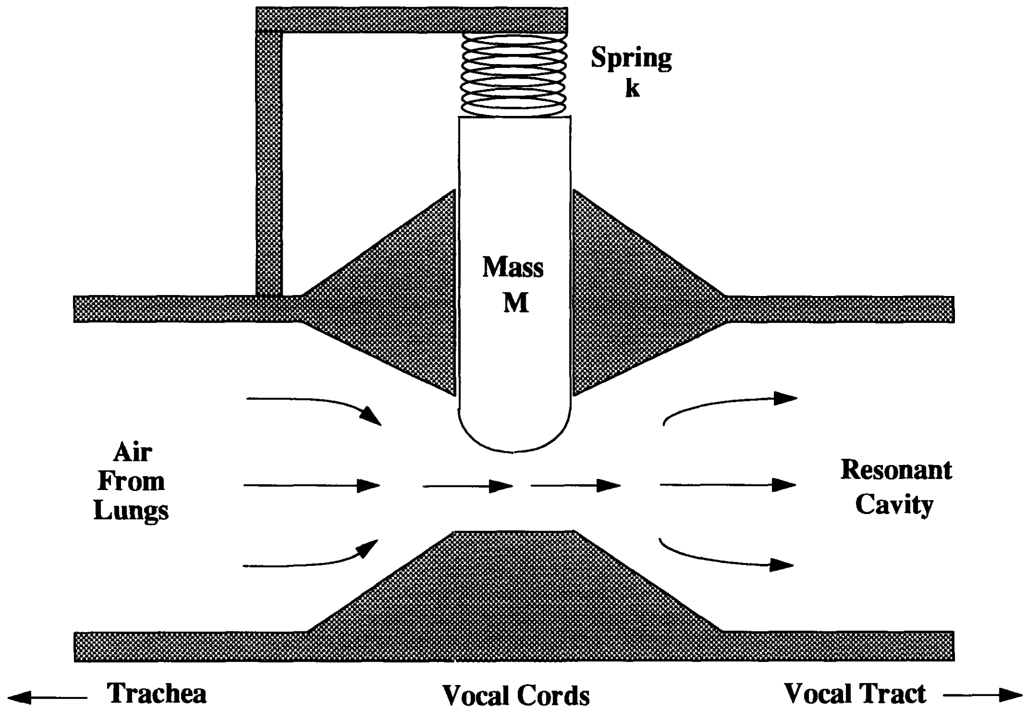

Figure 3-2: Mechanical Model of Speech Production

and, subsequently, the vowel represented in the signal [42],[53].

A mechanical model of the pulse train source and resonant cavity is shown in Figure 3-2. In this model, air from the lungs flows through a large diameter duct (the trachea), through a constriction (vocal cords), and back into a larger resonant cavity (the vocal tract). The vocal cords can be further modeled as a mass and spring system, in which the mass acts as a valve which constricts the size of the ducts and the spring adjusts the position of the mass from outside the duct. As air passes by the vocal cords, two conflicting forces are felt. The first is an inward force which tends to pull the mass into the duct, further constricting the airflow. This is due to the increase in speed of the air and the subsequent drop in air pressure. As the mass falls further into the duct, the speed of the air further increases, the air pressure further decreases, and the mass is pulled further inward. Of course, this is a catastrophic situation, and the second force must act to counter the first force. The second force results from the added frictional resistance produced in the constricted opening. The frictional resistance tends to reduce the total volume of air which passes through the constricted region, and thus, the flow-dependent pressure will not change in the manner described by force one. This second force turns out to be oscillatory, and is called the oscillating Bernoulli Force. In this simple model, changing the spring constant will change the frequency of oscillation. Analogously, changing the supporting air pressure from the lungs, establishing an initial spacing of the vocal cords, and establishing the tension of the vocal cords allows humans to produce sounds with differing pitch.

The vocal tract is a resonant cavity, and it extends from the larynx to the mouth and includes the nasal cavity. Oscillations from the larynx excite vibrational modes in this cavity, transforming the simple airflow spectrum which leaves the larynx into the acoustical patterns required for speech and music. Changes in the position of the tongue, the mouth, and the throat can significantly change the vibrational modes and the

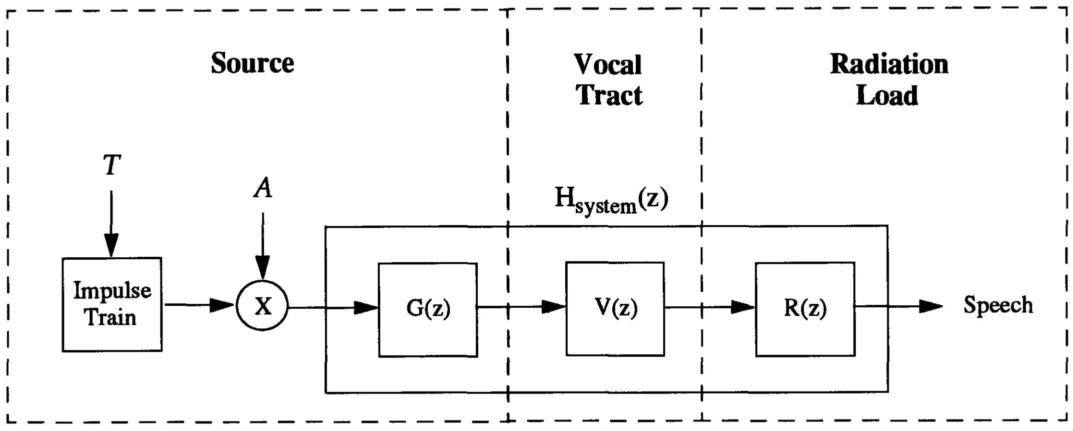

Figure 3-3: Linear, Quasi-Time-Invariant System Model

resulting sound that leaves ones mouth.

There are two additional spectra which are not accounted for in the above model. The first is the glottal source spectrum, which arises from the actual 'shape' of the air pulse which leaves the larynx. Above, this air pulse was assumed to be sinusoidal, but it is actually more triangular, with a DC-offset (i.e., there is always some air passing through the larynx). The second is the radiation load spectra, which occurs in the coupling of the vocal tract to the outside world via the mouth and nostrils. Note that the above model is limited to the physical source of the sound; it represents the physical characteristics of the sound which enters the outside world, Modeling the sound as perceived by another who is in the room would require additional spectral envelopes which model the human auditory system [42],[53],[55].

Linear, Quasi-Time-Invariant System Model

The model used in the Singing Tree for voiced waveforms includes the glottal source and radiational load spectra as shown in Figure 3-3 and Figure 3-4. The impulse train has period, T, which is the pitch period of the resulting signal. The impulse train is multiplied by a variable, $A(t)$, which is the time-dependent gain control of the system, although it will be assumed that $A(t)$ is quasi-time invariant. The glottal-source spectrum is approximated by,

$$G(z) = \frac{1 - e^{-aT_s}}{1 - z^{-1}e^{-aT_s}} \tag{3.21}$$

where a is a constant which characterizes the speaker and T_s is the sampling period to be used. A typical value for a is $a = 400\pi$ [53].

The vocal tract is modeled as a cascade of resonators, each with a resonant frequency, F_i. The system function due to the vocal tract can be written,

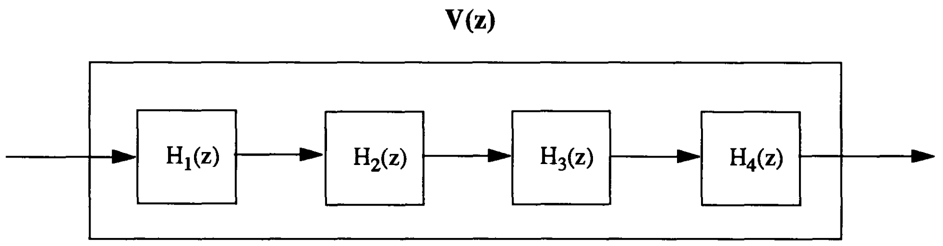

Figure 3-4: Expanded View of the Vocal Tract Model

$$V(z) = \prod_{i=1}^{\infty} H_i(z) \approx \prod_{i=1}^{4} H_i(z) \tag{3.22}$$

where H_i is the system function for the $i-th$ resonator. In this work, only the first three formants, modeled by three resonators, will be considered. The fourth resonator will account for high-frequency components to ensure proper spectral balance. It should be noted that an additional pole and zero are required in the system model to account for nasal consonants, but are ignored in this analysis [53]. Each individual resonator can be approximated as,

$$H_i(z) = \frac{1 - 2e^{-\alpha_i T} cos(\omega_i T) + e^{-2\alpha_i T}}{1 - \left(2e^{-\alpha_i T} cos(\omega_i T)\right) z^{-1} + \left(e^{-2\alpha_i T}\right) z^{-2}}, \quad \alpha_i \in [0, \infty) \tag{3.23}$$

where $\omega_i = 2\pi F_i$ is the resonant frequency of the $i-th$ resonator, and α_i is the neper frequency. Note that DC excitations will yield a gain of one, as one would expect physically. The poles associated with the $i-th$ formant frequency are located at,

$$z = (e^{-\alpha_i T}) e^{\pm j\omega_i T} \tag{3.24}$$

in the *z-plane* . Thus, one can approximate the bandwidth of the $i-th$ formant frequency estimate on the unit circle by $2\alpha_i$. In addition, the peaks in the spectrum along the unit circle correspond very closely to the formant frequencies. Although F_i and α_i are time varying, they do so on a time scale which justifies the quasi-time-invariant assumption of the system. Thus, for short-time analyses, the F_i are constant and will give the correct formant frequencies of the signal.

The radiation load can be modeled by,

$$R(z) = \frac{1 + e^{-bT_s}}{1 + z^{-1} e^{-bT_s}} \tag{3.25}$$

where b is a constant which characterizes the radiation load for a particular speaker, and T_s is the sampling period to be used. A typical value for b is $b = 5000\pi$ [53].

The entire system can thus be written as,

$$H_{system}(z) = G(z)V(z)R(z) = (G(z)R(z))\prod_{i=1}^{4} H_i(z) \tag{3.26}$$

where $G(z)$ and $R(z)$ are grouped together because they are speaker dependent, and thus can be approximated as constant for a particular user. Furthermore, it is assumed that they can be well approximated for all users with the typical values of a and b. In other words, it is assumed that,

$$G(z)R(z) \equiv \frac{1-e^{-aT_s}}{1-z^{-1}e^{-aT_s}}\frac{1+e^{-bT_s}}{1+z^{-1}e^{-bT_s}} \quad \left\{ \begin{array}{ccr} a & = & 400\pi \\ b & = & 5000\pi \\ T_s & = & \textit{sampling period} \\ f & < & 5000Hz \end{array} \right. \tag{3.27}$$

Thus, $G(z)R(z)$ acts as an envelope which scales V as a function of z. The poles corresponding to the formant frequencies are located inside the unit circle, and their associated frequencies are approximated by the peaks in the spectral envelope, $|H_{system}(e^{j\omega})|$. The approximation holds if the poles are relatively spread in frequency and relatively close to the unit circle.

The system for determining the first three formants as described in [53] is asynchronous, and *a priori* knowledge of the exact period is not required. In addition to the first three formants, the system also allows a method for calculating the pitch period and the gain of the signal. As applied in the Singing Tree, the pitch period was already determined through the normalized cross-correlation method described previously, and thus a synchronous analysis was feasible and pitch detection was unnecessary. Nonetheless, both formant and period estimation will be introduced, and the trade-offs between the two methods of pitch estimation will be discussed.

Formant Estimation

A block diagram for the entire estimation algorithm is given in Figure 3-5. A segment of signal, $s[n]$, is first windowed using a Hanning window, $w[n]$, yielding the signal $x[n]$. The DFT of $x[n]$ is evaluated using the FFT algorithm and yields $X[k]$. Next, the log of the magnitude of the samples in $X[k]$ is calculated to give, $\hat{X}[k]$, and then the IDFT is found using the FFT algorithm to finally give the *real cepstrum* $\hat{x}[k]$. The terminology, *real cepstrum*, simply refers to the fact that $\hat{X}[k]$ is a real sequence (i.e., magnitude of $X[k]$ is real, so use the real logarithm); it does NOT imply that the *real cepstrum* $\hat{x}[n]$ is a real sequence. However, due to the DFT symmetry properties, it does imply that the *real cepstrum* $\hat{x}[n]$ is an even function. In

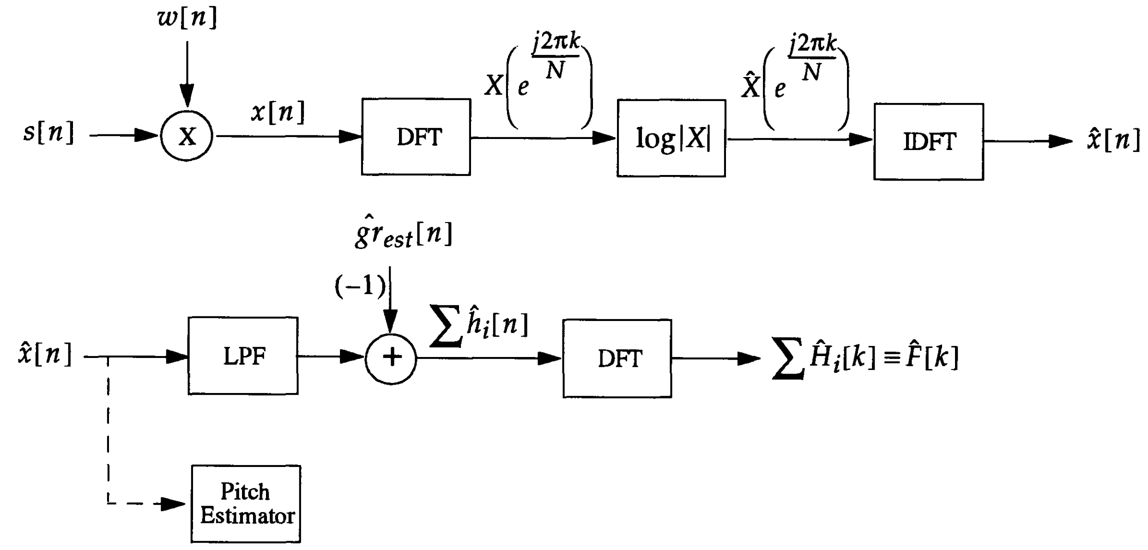
$w[n]$
$s[n]$
$x[n]$
DFT
$X\left(e^{\frac{j2\pi k}{N}}\right)$
$\log|X|$
$\hat{X}\left(e^{\frac{j2\pi k}{N}}\right)$
IDFT
$\hat{x}[n]$
$\hat{g}r_{est}[n]$
(−1)
$\sum \hat{h}_i[n]$
LPF
DFT
$\sum \hat{H}_i[k] \equiv \hat{F}[k]$
Pitch Estimator

Figure 3-5: Block Diagram of the Formant Estimation Algorithm

addition, since $x[n]$ is a real sequence (i.e., the samples of a singer's voice), it is known that $\hat{X}[k]$ is an even function, which in turn does imply that $\hat{x}[n]$ is a real sequence via the symmetry properties. The point, nonetheless, is that the term *real cepstrum* does not imply a real-valued cepstrum. It is simply the subset of cases for which the input signal is real that the cepstrum is real. For comparison, a *complex cepstrum* $\hat{y}[k]$ refers to the sequence defined as the IDFT of the log of a complex sequence, $Y[k]$ (i.e., one which uses the complex logarithm).

Considering the model presented for speech production, and defining $p[n]$ to be the impulse train which drives $h_{system}[n]$ (the inverse z-transform of $H_{system}(z)$), one can write an expression for $x[n]$ as,

$$x[n] = s[n]w[n] = (p[n] * h_{system}[n])\, w[n]. \tag{3.28}$$

It is assumed that the system is short-time time-invariant, which implies that for a suitably short windowing function, $h_{system}[n]$ is time-invariant. Furthermore, $w[n]$ varies on a much longer time scale than $s[n]$ does. Thus, the windowing function's role is two-fold: first, it greatly improves the approximation that a segment of speech is well modeled by a convolution of a periodic impulse train with a time-invariant (constant formant frequencies) $h_{system}[n]$, because it creates a signal segment on the time-invariant time scale; second, the windowing function performs its traditional role of reducing the effects of a non-integer number of signal periods in the signal segment [53]. Given these approximations, it is reasonable to write,

$$x[n] \;\approx\; (p[n]w[n]) * h_{system}[n] \;=\; p_w[n] * h_{system}[n], \quad p_w[n] \equiv p[n]w[n]. \tag{3.29}$$

Continuing with the block diagram, the following relations hold using equations (3.26) and (3.29). To avoid ambiguous and occasionally incorrect notation, all transforms are written in terms of the z-transform rather than the DFT. The discrete forms for the equations are then obtained by sampling the z-transform expressions at the points $z = e^{j\frac{2\pi}{N}k}$ where N is the number of points in the DFT.

$$X(z) = P_w(z)H_{system}(z) = P_w(z)(G(z)R(z))\prod_{i=1}^{4}H_i(z) \tag{3.30}$$

$$|X(z)| = |P_w(z)|\,|H_{system}(z)| = |P_w(z)|\,|(G(z)R(z))|\prod_{i=1}^{4}|H_i(z)| \tag{3.31}$$

$$\hat{X}(z) = log|X(z)| = log|P_w(z)| + log|(G(z)R(z))| + \sum_{i=1}^{4} log|H_i(z)| \tag{3.32}$$

$$\begin{aligned} \hat{x}[n] = IDFT\{log|X(z)|\} &= IDFT\{log|P_w(z)| + log|(G(z)R(z))| + \textstyle\sum_{i=1}^{4} log|H_i(z)|\} \\ &= \hat{p}_w[n] + \hat{gr}[n] + \textstyle\sum_{k=1}^{4}\hat{h}_i[n] \end{aligned} \tag{3.33}$$

When considering the DFT rather than the z-transform, the usual care must be taken to ensure that the input, $x[n]$, is sufficiently zero-padded such that the logarithm of the spectrum is sufficiently sampled and, thus, the cepstrum has no aliasing. In the final expression for the cepstrum, $\hat{x}[n]$, the additive terms $\hat{p}_w[n]$, $\hat{gr}[n]$, $\sum_{i=1}^{4}\hat{h}_i[n]$ represent the cepstrum of each corresponding term (IDFT of the respective log-magnitude spectra). Given this additive nature of the cepstrum, one can approximately separate $\hat{x}[n]$ into

$$\hat{x}[n] \approx \begin{cases} \hat{gr}[n] + \sum_{i=1}^{4}\hat{h}_i[n] & nT_s < T \\ \hat{p}_w[n] & nT_s \geq T \end{cases} \tag{3.34}$$

where T is the period of the signal. Assuming that the period of the signal is unknown (asynchronous analysis), T in the above equation would have to be replaced with the lowest anticipated period, T_{min}. However, in the case of the Singing Tree, the period was known *a priori* and the actual pitch, T, minus a small tolerance to account for error was used. Note that if the period of the signal was not already known, it could be found from equation (3.34) by analyzing the cepstrum for $nT_s \geq T_{min}$. This alternative method for finding the period is discussed in more detail in Section 3.2.6.

The task at hand is to determine the formant frequencies. The cepstrum has separated into two components, one containing information about the period of the signal, and one containing information about the formant frequencies, the glottal pulse, and the radiation load. Since this segment is located at times, $nT_s < T$, low-time filtering (LTF) this segment from the cepstrum will retain only the formant, glottal, and radiational information, and also act to smooth $\hat{X}[k]$, the log magnitude of the DFT of the cepstrum. The only problematic point left is the glottal pulse and radiation load. Assuming that these are both approximated as per equation (3.27) evaluated at the unit circle, one can define

$$\hat{gr}_{est}[n] \equiv IDFT\{log|G(z)R(z)|_{sampled\ unit\ circle}\}. \tag{3.35}$$

Note that $\hat{gr}_{est}[n]$ in equation (3.35) is an estimate for the glottal and radiational cepstrum, because the form of equation (3.27) and its values of a and b were approximations. In contrast, $\hat{gr}[n]$ from equation (3.34) is the cepstrum of the actual glottal and radiation information contained in the input signal. Defining $\hat{x}_{LTF}[n]$ as the filtered cepstrum and subtracting equation (3.35), one is approximately left with only the formant cepstrum, $\hat{f}[n]$.

$$\hat{f}[n] \equiv \hat{x}_{LPF}[n] - \hat{gr}_{est}[n] = \hat{gr}[n] + \sum_{k=1}^{4} \hat{h}_i[n] - \hat{gr}_{est}[n] \approx \sum_{i=1}^{4} \hat{h}_i[n] \tag{3.36}$$

Lastly, to find the formant spectrum, all that remains is to take the DFT of the formant cepstrum.

$$\hat{F}[k] = DFT\{\hat{f}[n]\} = \sum_{k=1}^{4} \hat{H}_i[k] \tag{3.37}$$

Of course, greater computational efficiency can be achieved by subtracting off the glottal and radiational spectra before the initial calculation of the cepstrum, or after one retakes the DFT of the filtered cepstrum. Doing so eliminates the need to take the IDFT of $log|G(z)R(z)|_{sampled\ unit\ circle}$. The derivation shown above keeps the real glottal and radiation terms throughout simply for clarity of approach, but it is best to remove them before initially calculating the cepstrum in equation (3.33).

At this point, the formant frequency information is contained in $\hat{F}[k]$, the cepstrally smoothed log spectrum (in the frequency domain). Assuming the poles corresponding to the formant frequencies are sufficiently separated and near the unit circle, peaks will appear in the log spectrum corresponding to the formant frequencies [53]. However, if the poles are located too close together, the peaks may smear together, making it impossible to identify the individual formant frequencies from the unit circle. The solution is to leave the unit circle by using the chirp-z transform. However, determining when to resort to the chirp-z transform is what makes algorithmic peak-picking difficult. Assumptions regarding the frequency range of the formant frequencies will help. As derived from [53], [64], and [42], it is assumed that the formant frequencies will fall into the ranges shown in Table 3.1. In general, the formant frequencies for females and children are higher than those of men. This is convenient, because women and children also tend to sing pitches with higher frequency. Thus, a metric exists to approximate the frequency range based on the pitch of the signal. Given

Table 3.1: Formant Frequency Range

Formant Frequency	Singer	Frequency Range (Hz), [Fmin - Fmax]
F1	men	200 - 900
	women and children	300 - 1050
F2	men	550 - 2700
	women and children	700 - 2900
F3	men	1100 - 3100
	women and children	1400 - 3500

an assumed frequency range, the issue at hand is to develop a peak-picking algorithm which can account for indistinguishable formant frequencies. Paraphrasing the algorithm described in [53],

1. The first step is to find the maximum peak in the range 0 Hz - F1max. Typically, this peak will occur at the first formant frequency. However, occasionally the peak will occur at a frequency less than F1min, as a result of residual glottal pulse spectra. In this case, the first formant peak may be indistinguishable from the glottal peak with no other peaks present in the first formant frequency range, or another peak may be present, but it is unknown whether it corresponds to formant one or formant two. If an additional peak occurs within the first formant frequency range, then it must be less than 8 dB below the glottal peak to be considered the first formant. If not, it is the second formant and the first formant is indistinguishable from the glottal peak. In the cases where there is an indistinguishable peak or no peak is found, the chirp z-transform is used to expand and enhance the 0 Hz to F1max region and the analysis is repeated. If still no peak can be determined, the first formant, F1, is arbitrarily set to F1min. (due to real-time constraints, repeated iteration is computationally too expensive.)

2. The second step is to search the second formant frequency range, provided F1 is less than F2min. However, if F1 is determined to be larger than F2min, it could be that F2 has been mistaken for F1. In this case, the search occurs over the range F1min to F2max. After a second peak has been found, it is compared with the value of F1, and, the lower of the two is assigned to F1 and the higher is assigned to F2. The second formant should be 8.7 to 24.3 dB below the first formant frequency. If only one peak exists over both the first and second frequency ranges, then the chirp z-transform is again used to enhance and expand the region. Once F1 and F2 have been determined, the difference in the height of their peaks is compared to a threshold value (which is itself a function of frequency) to assure the choices are legitimate.

3. Following a similar argument as for F2, the third formant frequency F3 is found.

4. Given the three estimates (F1, F2, and F3), the calculation is repeated for new segments of signal. The values of F1, F2, and F3 are stored as running averages of several lag values, which slightly delays

convergence when a formant change occurs. However, this is necessary to account for erroneous data. Another non-linear smoothing method would be to assume that changes in formant frequency are less than Δ, and estimates which fall further than Δ from the previous point will be reassigned based on previous points. For a critical number of consecutive points further than Δ away, a new formant frequency trace is established.

Having estimated the first three formant frequencies, the next task is to derive the vowel. It is shown in [64] that vowels can typically be determined by the first two formants as shown in Figure 3-6. In this graph, the first formant frequency is displayed along the x-axis, and the second formant frequency is along the y-axis. This result can also be displayed as a 'vowel triangle' as shown in Figure 3-7 also from [64]. In this graph, the axes are opposite to those in Figure 3-6 such that the y-axis corresponds to the 'up and down' movement of the vowels. That is, as the first formant frequency increases, the 'position' of the vowel in the mouth goes from low to high. As the second formant frequency increases, the 'position' of the vowel moves from the back of the mouth to the front.

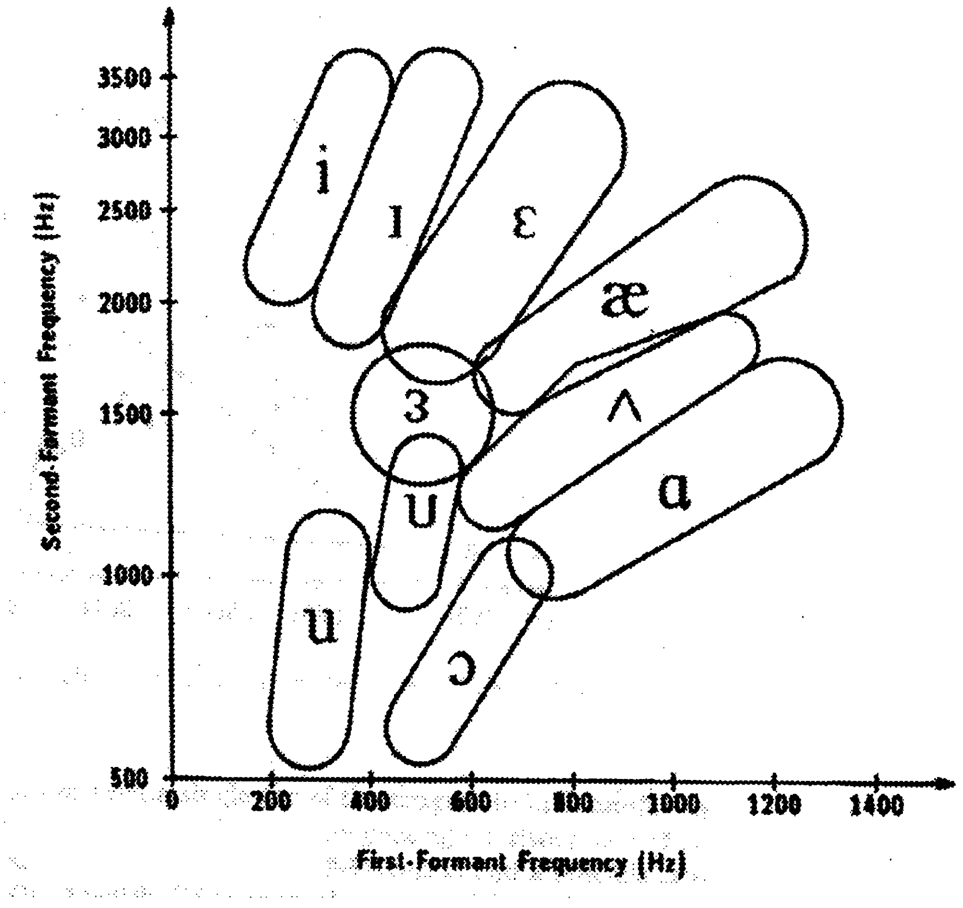

Figure 3-6: Vowel as a Function of F1 and F2

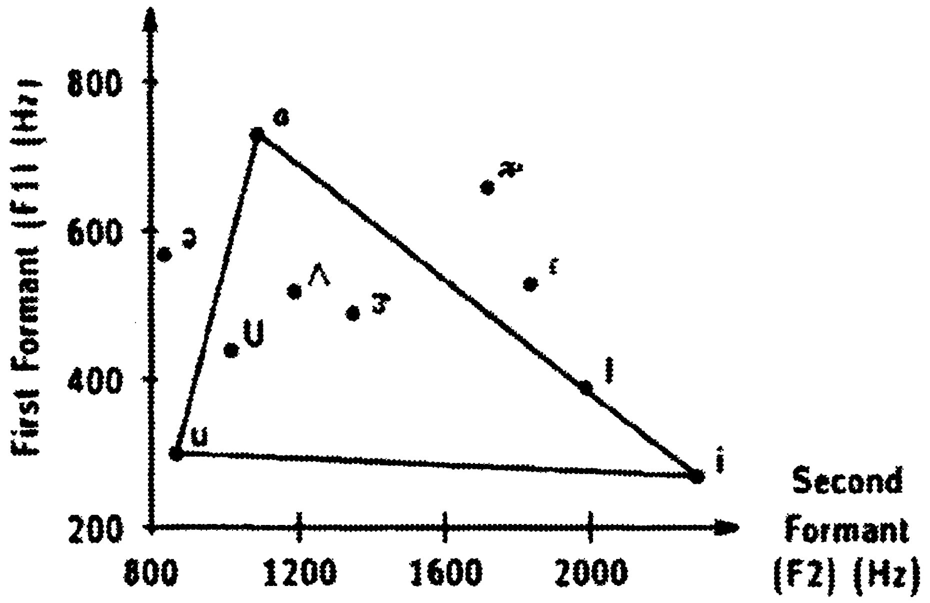

Figure 3-7: The Vowel Triangle

Table 3.2: Average Formant Frequencies for Several Vowel Sounds

Formant Frequency	Singer	/i/	/I/	/ɛ/	/æ/	/a/	/ɔ	/U/	/u/	/ʌ/	/ɝ
F1	men	270	390	530	660	730	570	440	300	640	490
(Hz)	women	310	430	610	860	850	590	470	370	760	500
F2	men	2290	1990	1840	1720	1090	840	1020	870	1190	1350
(Hz)	women	2790	2480	2330	2050	1220	920	1160	950	1400	1640
F3	men	3010	2550	2480	2410	2440	2410	2240	2240	2390	1690
(Hz)	women	3310	3070	2900	2850	2810	2710	2680	2670	2780	1960

If the vowel is still unresolved from the first two formants, a comparison of the average third formant frequencies may help determine the vowel. Table 3.2 (after O'Shaughnessy [64]) gives the average formant frequencies for spoken English vowels. The vowel frequencies of both male and female children most closely resemble those of women.

Chirp z-Transform

Up to this point, the chirp z-transform has been mentioned several times without formal introduction [53]. The chirp z-transform algorithm is a means by which one can determine samples of the z-transform equally spaced along a contour in the $z-plane$. Strictly speaking, the 'chirp transform' finds these samples along the unit circle, while the 'chirp z-transform' algorithm refers to samples along a general contour (which, of course, may be the unit circle) [49]. The advantage of the chirp z-transform is that it enables one to calculate the samples of the z-transform equally spaced over an arc or a spiral contour with an arbitrary starting point and arbitrary frequency range. In other words, if the formant frequency of interest is in the range 200 Hz to 900 Hz, then samples of the z-transform may be calculated specifically in this region. In contrast, the frequency range of the DFT is strictly related to the sampling frequency. Furthermore, the chirp z-transform allows one to calculate an arbitrary number of samples along this contour, which arbitrarily reduces the error in frequency representation. As discussed previously for the DFT case, if one requires a frequency representation of higher resolution, the only recourse is to zero-pad the signal (i.e., the number of sample points in the DFT is the same as the number of points in the signal). Lastly, the chirp z-transform allows one to leave the unit circle and calculate samples in the $z - plane$, while the DFT is limited to the unit circle. This is particularly useful for cases in which the frequencies corresponding to poles inside the unit circle cannot be distinguished along the unit circle contour. One disadvantage of the chirp-z transform is that it may be slower than the FFT algorithm, depending on the number of sample points. The following is a derivation of the chirp z-transform from [49],[53],[60].

The z-transform of a sequence, $x[n]$, is defined to be,

$$X(z) \equiv \sum_{n=0}^{N-1} x[n]z^{-n} \tag{3.38}$$

where N is the number of samples in sequence $x[n]$. The goal is to sample the z-transform at equally spaced points along a general contour in the $z - plane$. In general, one can define the points, z_k, of an M-point contour as

$$z_k = AW^{-k}, \quad for\ k = 0, 1, \ldots, M-1 \tag{3.39}$$

where A is the starting point of the contour,

$$A = A_0 e^{j\omega_0}, \quad \omega_0 = 2\pi\theta_0 \tag{3.40}$$

and W determines the incremental evolution of the contour.

$$W = W_0 e^{j\Delta\omega_0}, \quad \Delta\omega_0 = 2\pi\phi_0. \tag{3.41}$$

Substituting equation 3.39 into equation 3.38 yields

$$X(z_k) = \sum_{n=0}^{N-1} x[n](AW^{-k})^{-n} = \sum_{n=0}^{N-1} x[n]A^{-n}W^{nk} \tag{3.42}$$

which apparently requires NM multiplies and adds to calculate. However, the expression can be re-written as a convolution using the identity,

$$nk = \frac{1}{2}[n^2 + k^2 - (k-n)^2] \tag{3.43}$$

to yield

$$X(z_k) = W^{\frac{k^2}{2}} \sum_{n=0}^{N-1} (x[n]A^{-n}W^{\frac{n^2}{2}})W^{\frac{(k-n)^2}{2}} \quad k = 0, 1, \ldots, (M-1). \tag{3.44}$$

Switching the indices n and k such that X is indexed by n rather than k, and defining $g[n]$ to be

$$g[n] \equiv x[n]A^{-n}W^{\frac{n^2}{2}}, \tag{3.45}$$

the convolution is simply the sequence $g[n]$ convolved with the sequence $W^{\frac{n^2}{2}}$.

$$X(z_n) = W^{\frac{k^2}{2}} \sum_{k=0}^{N-1} g[k]W^{\frac{(n-k)^2}{2}} \quad n = 0, 1, \ldots, (M-1) \tag{3.46}$$

Evaluating the convolutional form of the equation will require approximately $(N+M)\log(N+M)$ multiplies and additions if one uses the FFT algorithm. Of practical interest, the sequence $W^{\frac{n^2}{2}}$ need only be evaluated over the interval $-(N-1) \leq n \leq (M-1)$ (and zero elsewhere), because $g[n]$ is finite and of length N. This implies that the size of the FFT required to evaluate the convolution must be at least $N + M - 1$, but may be arbitrarily larger than this, such as the next largest power of two. Also note that $W^{\frac{n^2}{2}}$ is a complex exponential with linearly increasing frequency, which is a 'chirp' and thus the name chirp z-transform. This type of analysis is used in radar and sonar signal processing [49].

The chirp z-transform is used to resolve formant frequencies which are indistinguishable from the unit circle. The method is to choose a contour inside the unit circle which is closer to the poles of the system, and then evaluate the z-transform along this contour. Since the contour lies closer to the poles, the resolution between poles will be greater. In [53], a circular contour of radius $e^{-0.0314} = 0.97$ was used, and a similar value was also adopted for the Singing Tree.

To give an example of how the chirp z-transform can help resolve formant frequencies, consider the following (unrealistic) formant system with only two resonant frequencies (four complex conjugate poles). Assume that the poles are located at

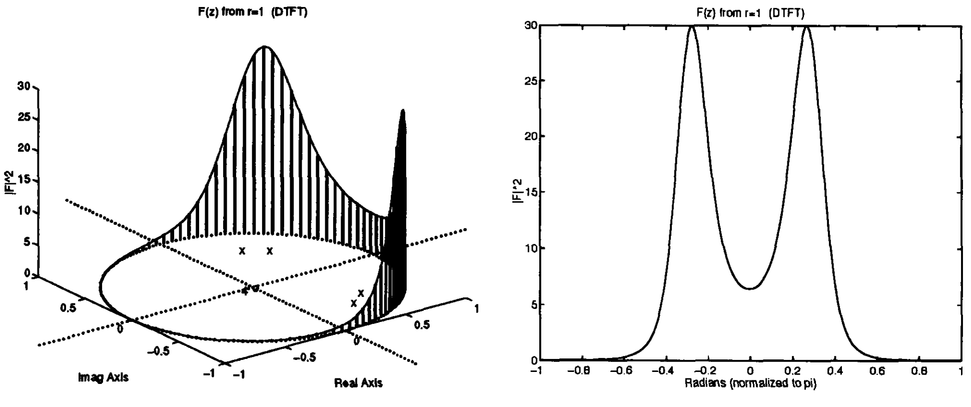

Figure 3-8: Pole-Zero Plot and the DFT in the z-plane

$$\begin{array}{rclcrcl} a & = & 0.7e^{\frac{j\pi}{3}} & & a^* & = & 0.7e^{\frac{-j\pi}{3}} \\ b & = & 0.7e^{\frac{j\pi}{4}} & & b^* & = & 0.7e^{\frac{-j\pi}{4}} \end{array} \tag{3.47}$$

with formant frequencies approximately equal to

$$\begin{array}{rclcl} f_a & = & \frac{\pi/4}{2\pi}F_s & = & \frac{F_s}{8} \\ f_b & = & \frac{\pi/3}{2\pi}F_s & = & \frac{F_s}{6} \end{array} \tag{3.48}$$

for some appropriate sampling frequency F_s. Thus, the system function can be written,

$$\begin{array}{rcl} F(z) & = & \left(\frac{z^2}{(z-a)(z-a^*)}\right)\left(\frac{z^2}{(z-b)(z-b^*)}\right) \\ & = & \left(\frac{1}{1-2\Re\{a\}z^{-1}+|a|^2z^{-2}}\right)\left(\frac{1}{1-2\Re\{b\}z^{-1}+|b|^2z^{-2}}\right). \end{array} \tag{3.49}$$

The system's pole-zero plot, a three dimensional representation of the discrete Fourier transform (DFT) along the unit circle in the *z-plane,* and the DFT are shown in Figure 3-8. The DFT is interpolated in this example for clarity.

It is clear to see from the plot of $F(e^{j\omega})$ that the two poles are indistinguishable from the unit circle. However, using the chirp z-transform (CZT) method, one can choose a contour closer to the poles, giving a higher resolution. Choosing a constant radius of $r = 0.75$ and sweeping through $0 \leq \omega \leq 2\pi$ yield the two graphs of the CZT in Figure 3-9.

The chirp z-transform method reveals that there are indeed two distinct poles, and that they occur at radial frequencies $\omega = \pi/3$ and $\omega = \pi/4$ as expected. Furthermore, the $\Delta\omega$ in equation (3.41) can be made arbitrarily small with the CZT algorithm by either increasing the number of points in the CZT or decreasing the frequency range of observation. In this example, the CZT was calculated at 200 points along the circle of $r = 0.75$. This corresponds to an angular frequency resolution of $\Delta\omega_{CZT} = 2\pi/200 \approx 0.0314$. The

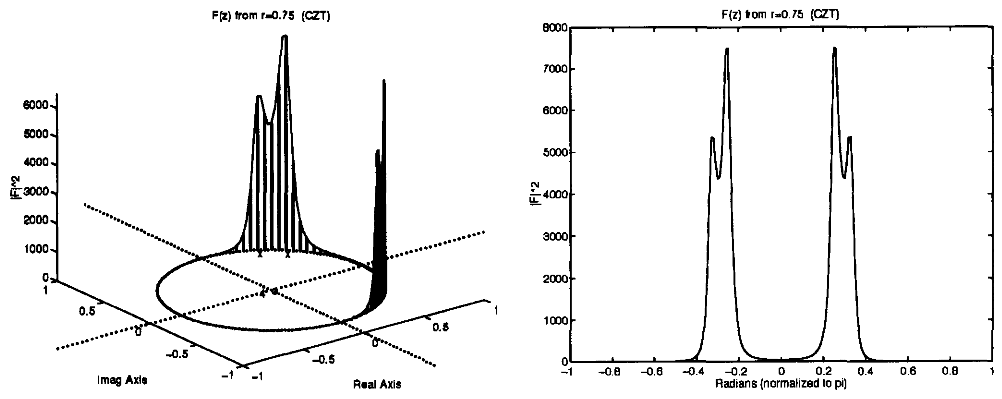

Figure 3-9: Pole-Zero Plot and the CZT along r=0.75 in the z-plane

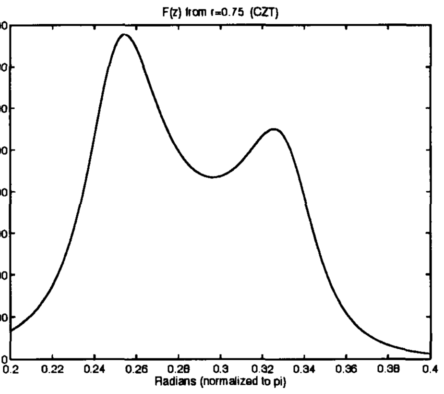

Figure 3-10: Chirp z-Transform over a Smaller Frequency Range for Higher Resolution

DFT of 200 points also has an angular frequency resolution of $\Delta\omega_{DFT} = 2\pi/200 \approx 0.0314$. However, the resolution for the CZT can be greatly improved by reducing the frequency range of analysis such that it covers only those frequencies in which we expect to find formants. Without changing the number of points, a CZT calculated over the frequency range $0.2\pi \leq \omega \leq 0.4\pi$ will have an angular frequency resolution of $\Delta\omega_{CZT} = 0.2\pi/200 \approx 0.00314$, which is a factor of 10 better. Figure 3-10 shows the CZT along the smaller frequency range. The DFT in this example would require a 10-fold increase in the number of points to achieve the same frequency resolution.

Formant Estimation Example

The following is an example of formant frequency estimation using data directly from the Singing Tree. The author sang into the Singing Tree and saved his voice samples for the vowels ‘aah’, ‘ee’, and ‘ooh’. The complete results for the vowel ‘aah’ at a pitch f=290 Hz are shown below, with graphs and results for the other vowels in Appendix C.

Referring to Figure 3-5, the first step is to window a segment of the voice for analysis. Given the real-time

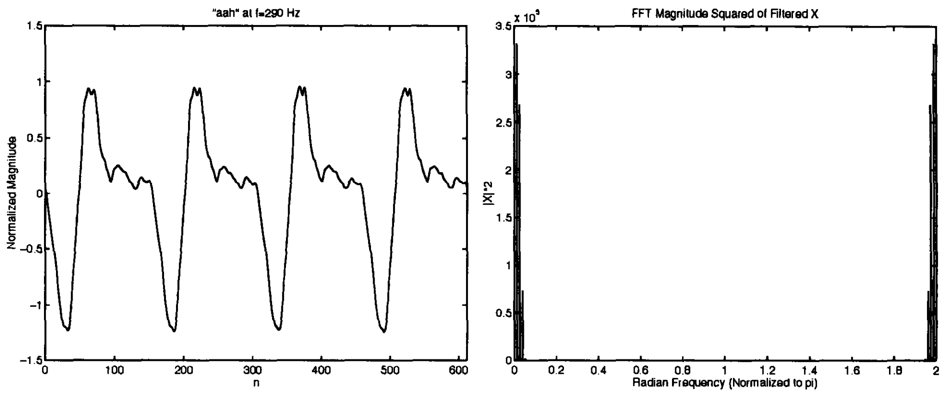

Figure 3-11: Segment of Voice Singing 'aah' and its Power Spectrum,

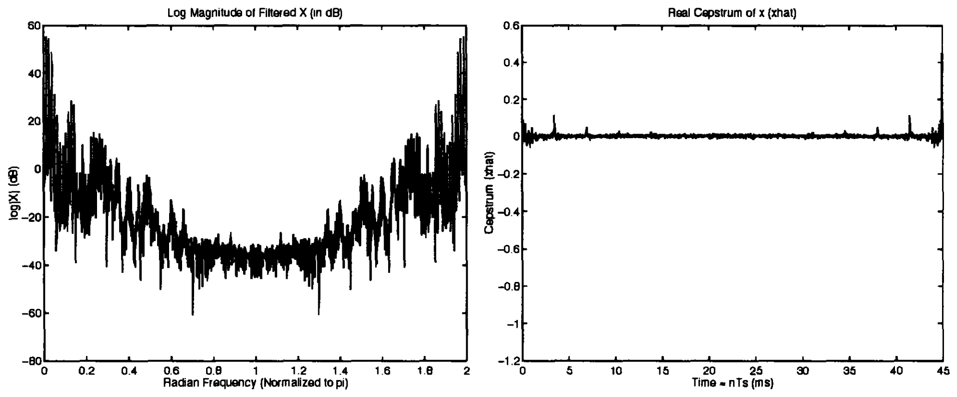

Figure 3-12: Log-Magnitude and Real Cepstrum

constraints of the system, this is approximately a 15 ms window. Since it is assumed in this example that the pitch has already been determined via the normalized cross-correlation algorithm, a pitch synchronous approach is taken. Thus, an integer number of periods is selected for analysis as shown in Figure 3-11 along with its power spectrum (DFT magnitude squared). Note that if the pitch is not already known, a windowing function such as the Hanning window should be used to select a 10-15 ms section of voice. The next two graphs in Figure 3-12 show the log magnitude of the DFT and the real cepstrum of the signal. Although it is difficult to discern from this graph, the cepstrum has a large, negative impulse near $n = 0$ corresponding to the glottal pulse and radiational load.

Since an approximation is made for the form of the glottal pulse and radiational load, these can be effectively removed from the signal as explained previously. The approximation given in equation (3.27) is assumed to be valid over the region f=[0,5000] Hz, and equal to its value at 5000 Hz for frequencies greater than 5000 Hz. Figure 3-13 shows the spectrum of $G(z)R(z)$ evaluated along the unit circle and its real cepstrum. It is further assumed that the spectrum of $G(z)R(z)$ evaluated along contours near the unit circle are approximately the same as the spectrum of $G(z)R(z)$ evaluated along the unit circle. This allows the

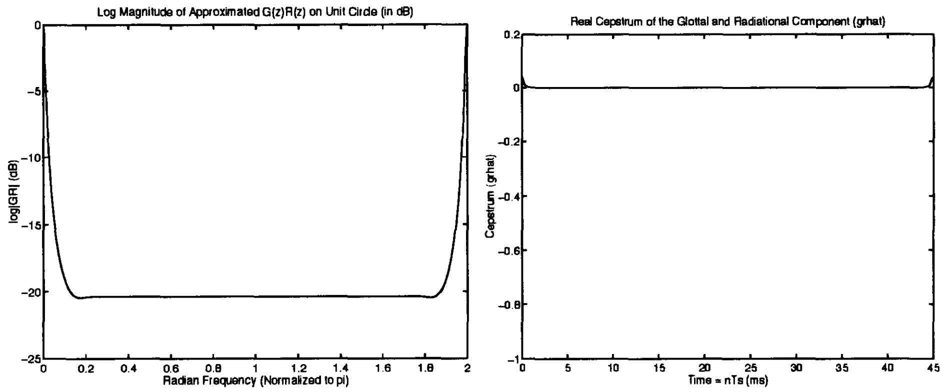

Figure 3-13: Log-Magnitude and Real Cepstrum of the Glottal and Radiational Component

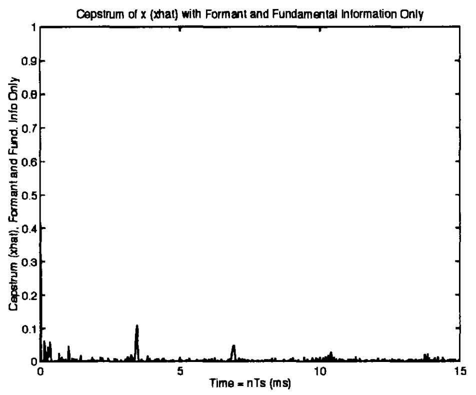

Figure 3-14: Approximation of the Real Cepstrum with no Glottal and Radiational Component

following analysis to progress without the need to recalculate the $G(z)R(z)$ spectrum for differing values of the radius used in the chirp z-transform. Note that since $G(z)R(z)$ is assumed to be known, several potential chirp z-transform radii could be determined in advance, allowing the computer to calculate each spectrum off-line. Thus, calculation of the $G(z)R(z)$ spectrum (or its cepstrum, or its cepstrally smoothed version) is not a concern for real-time analysis. However, each chirp z-transform iteration will require the $G(z)R(z)$ spectrum (or its cepstrum or its cepstrally smoothed version) to be subtracted from the actual signal, and in this, there is a marginal delay due to the retrieval of the $G(z)R(z)$ spectrum from memory and its subtraction from the real cepstrum. Putting these issues aside, the $G(z)R(z)$ spectrum is assumed equal to its values along the unit circle for contours near the unit circle. Although difficult to discern from the graph, the cepstrum of the glottal and radiational component (approximated g-r cepstrum) also has a large, negative impulse near $n = 0$ similar to that in the real cepstrum of the signal. Subtracting the glottal and radiational component from the signal yields the graph in Figure 3-14.

As per equation (3.34), the formant frequency information is contained in the cepstrum at times less than the period of the signal. A simple cosine window is used to low-time filter out the samples of the cepstrum

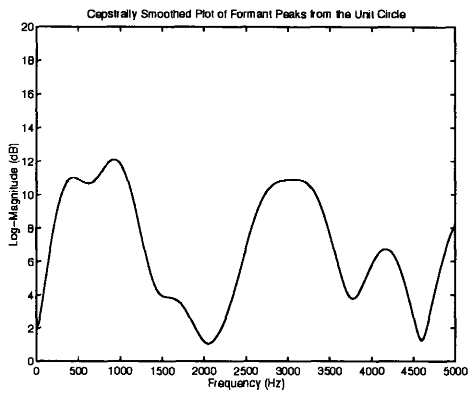

Figure 3-15: Cepstrally Smoothed Log Spectrum

(with no g-r component) for small times,

$$lpf[n] = \begin{cases} 1 & nT_s < \tau_1 \\ \frac{1}{2}cos[\pi(nT_s - \tau_1)/\Delta\tau] & \tau_1 \leq nT_s \leq \tau_2 \\ 0 & nT_s > \tau_2 \end{cases} \tag{3.50}$$

where τ_2 is less than the period of the signal and τ_1 determines how quickly the window will cut-off. In this example, $\tau_1 = 1.15\ ms\ (i.e., n \approx 50)$ and $\tau_2 = 1.58\ ms\ (i.e., n \approx 70)$. Note that for the general case, the higher the pitch frequency, the lower the pitch period and the lower the values of τ_1 and τ_2. Multiplying equation (3.50) with the cepstrum (with no g-r component) yields the cepstrally smoothed log spectrum shown in Figure 3-15.

What remains is to determine the formant frequencies using the previously described algorithm of [53]. Figure 3-16 shows a close up of frequency regions one and two. Considering the frequency range of the first formant, there are two peaks: one at approximately 450 Hz and one at approximately 940 Hz. The peak at 935.5 Hz is larger, and so it is designated the first formant frequency. Recall that it has been shown through experiment that the second formant should be lower than the first formant by between 8 and 27 dB [53]. Considering the second formant frequency region, the largest peak is also at 935.5 Hz, with no other obvious peak. However, there appears to be a plateau near 1600 Hz. Thus, the frequency will have to be resolved using the chirp z-transform, which is shown for all regions and regions 1 and 2 in figure 3-17. The CZT has shown that the plateau is indeed a peak corresponding to the second formant frequency, f2=1686 Hz. Plots for the third region from the unit circle and the chirp z-transform contour are shown in Figure 3-18. The third frequency region has a very broad bump at around f=3074 Hz. However, a chirp z-transform analysis shows this to be a double-peak. The third formant frequency is the lower of the two peaks, at f=2709 Hz. The estimated formant frequencies are shown in Table 3.3.

Referring to Figure 3-6, the formant frequencies correspond to the vowel sound /Λ/ which is an 'aah'

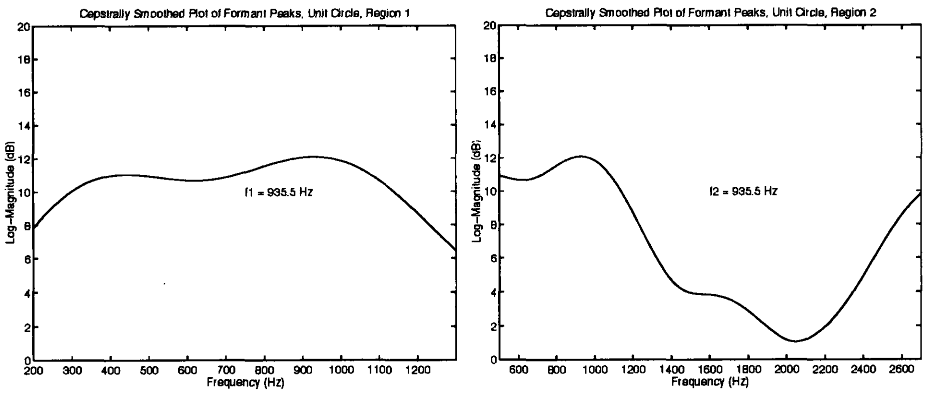

Figure 3-16: Cepstrally Smoothed Log Spectra in Regions 1 and 2

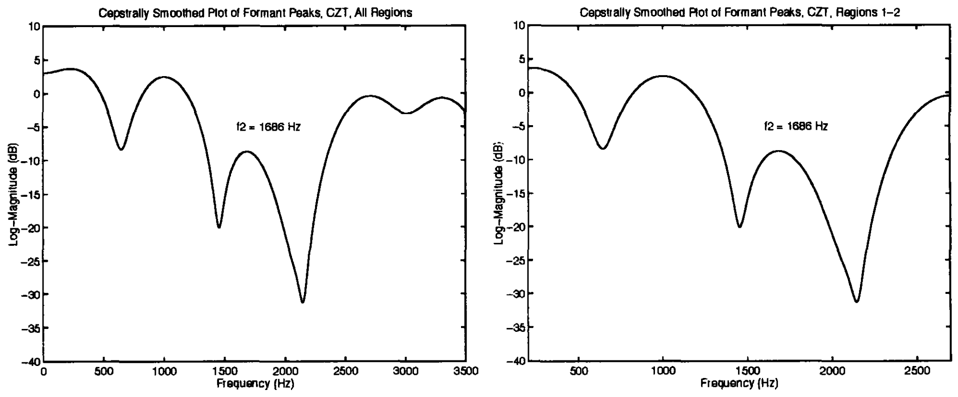

Figure 3-17: Cepstrally Smoothed Log Spectra via the Chirp z-Transform

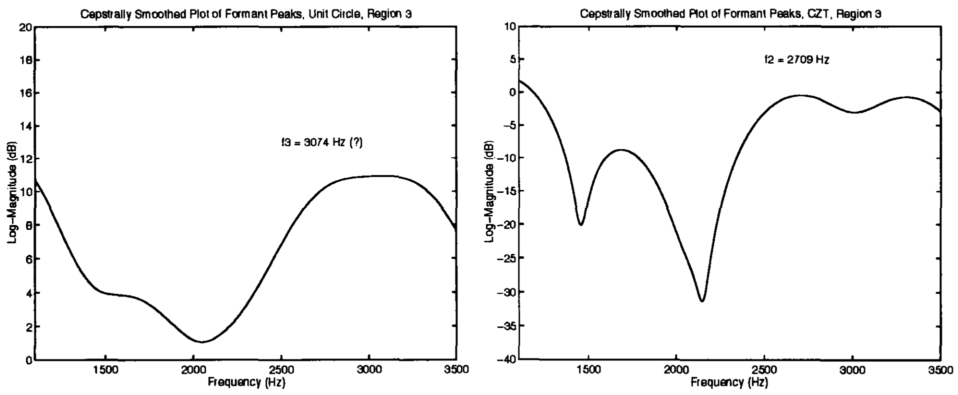

Figure 3-18: Cepstrally Smoothed Log Spectra of Region 3 (Unit Circle and CZT)

Table 3.3: Estimated Formant Frequencies

Formant Frequency	Frequency (Hz)
1	936
2	1686
3	2709

sound. The first and third formants are rather high, though, and this may indicate that the vowel has some /a/ quality. There are several difficulties in determining a vowel based on formant structure which should be noted. First, the correlation between formant frequencies and vowel sounds is approximate. Second, the formant frequencies vary between age, gender, and pitch frequency. Third, singers can manipulate their formant frequencies significantly while singing. Professional singers, in particular, can arrange their formant frequencies such that the words they are singing become recognizable over a wide range of pitch. It is known that, regardless of the particular piece of music being performed, the long-term average sound pressure (LTAS) of an orchestra rises quickly from low frequencies to a peak at f=450 Hz, and then falls at about 9 dB/octave thereafter. Speech and non-operatic singing styles have a similar LTAS to an orchestra. When singing with an orchestra, many of the formants in the singer's voice may be covered or masked by the frequency components of the orchestra. Thus, trained singers will often add or emphasize the higher frequency partials in the 2000-3000 Hz range to 'cut through' the sound of the orchestra. In addition, singers will often 'tune' their vowel formants by moving them up and down to match the harmonics of the music being played. There are many such formant alterations commonly employed by singers which make vowel estimation rather difficult in the Singing Tree application. In the end, changes in formant structure were used in the musical mappings rather than the specific vowel. In other words, whatever the formant structure of the participant at any time, if it stays relatively constant, then the participant is maintaining her vowel. If not, then the vowel is assumed to be changing. These vowel dynamics proved useful in creating the musical mapping algorithms.

3.2.6 Alternative Methods of Pitch Estimation

The normalized cross-correlation algorithm used for pitch estimation in the Singing Tree is an efficient and reliable means to extract the pitch from a monophonic signal. It also naturally leads to the concept of noisiness/pitch ambiguity and timbre. However, there are other methods to determine the pitch frequency of a signal, two of which are described below. While these methods are not explicitly used in the Singing Tree, they are based on the CZT and the cepstrum, and therefore could be implemented without significantly changing the system architecture.

Pitch Estimation via the Chirp z-Transform

The two interesting characteristics of the chirp z-transform discussed thus far are its utility to calculate the z-transform along an arbitrary circular or spiral contour anywhere in the *z-plane,* and to do so at an arbitrarily large number of points. While the former allowed the resolution of formant frequencies which were indistinguishable from the unit circle, the latter allows one to effectively calculate the DFT of a signal along a segment of the unit circle using an arbitrarily large number of points. In the discussion of the benefits of pitch-synchronous analysis, the issue of frequency tolerance between samples was the major factor that led to unreliable results when peak-picking the DFT of a signal to determine a frequency in real-time applications. Since the DFT implies a full-spectrum of equally spaced samples of the DTFT, the only option one has to reduce the inter-sample frequency tolerance is to increase the number of sample points. The CZT, on the other hand, has an additional degree of freedom; the CZT allows one to shorten the frequency spectrum over which one samples the DTFT in addition to increasing the number of sample points, offering two means be which to reduce the frequency error associated with peak-picking.

The method for pitch estimation using the CZT is to first define the desired frequency accuracy required in the estimation and the anticipated frequency range of the incoming signal. In the case of the Singing Tree, the the frequency range is simply that of the human voice, or approximately 60 Hz to 1600 Hz. Equation (3.18) showed that a frequency error of approximately 1 Hz is marginally perceptible at a pitch of 100 Hz. Thus, a tolerance less than 1 Hz is a reasonable goal. It was shown that DFT's with a very large number of point would be required meet the 1 Hz goal, let alone a tighter constraint. The solution there was to use pitch-synchronous frequency analysis, but that assumes the pitch was known *a priori* . The solution here is to use the CZT.

Defining f_2 and f_1 to be the fmax and fmin of the frequency range respectively and m to be the number of sample points, one can write the incremental evolution of the contour given in equation (3.41) as

$$W = W_0 e^{j\Delta\omega_0} = W_0 e^{j\frac{2\pi}{F_s}\frac{f2-f1}{m}}, \quad \Delta\omega_0 = \left(\frac{2\pi}{F_s}\right)\left(\frac{f2-f1}{m}\right) \tag{3.51}$$

which emphasizes the advantage of the CZT over the DFT. Comparing the CZT and the DFT (i.e., setting $W_0 = 1$), the incremental frequency of the CZT compared to the DFT can be written

$$\Delta\omega_{CZT} = \left(\frac{2\pi}{m}\right)\left(\frac{f2-f1}{F_s}\right) = \Delta\omega_{DFT}\frac{(f2-f1)}{F_s}, \quad \frac{(f2-f1)}{F_s} \in [0,1] \tag{3.52}$$

which shows that, for a given number of sample points, the CZT will always have a smaller frequency tolerance than the DFT. The trade-off is that the CZT covers a smaller frequency domain. For the f_2 and f_1 over the assumed vocal frequency range, the number of points necessary to achieve a frequency tolerance Δf of 1 Hz or less is

$$m \geq \frac{(f_2 - f_1)}{\Delta f} \Longrightarrow m \geq 1540pts. \tag{3.53}$$

Thus, choosing 1600 points, or the next power-of-2 at 2048 is reasonable. Remembering that the real-time constraint implies an input signal segment no longer than 15 ms, a signal of sampling frequency 44100 Hz will have fewer than 661 points. To compare the DFT to the CZT under the condition of an equal number of sample points, the remainder must be zero-padded in the DFT case. Using a 660 point Hanning window to extract three back-to-back sections of 660 samples of the signal shown in Figure 3-11 (and not assuming pitch-synchronous analysis), the plots in Figure 3-19 compare the DFT and CZT for three sections of signal.

The frequency estimates of the three sections are 288.6 Hz, 289.3 Hz, and 291.1 Hz. It is clear that the author does not have perfectly consistent pitch, but the running average of the CZT pitch frequency estimate is f=289.7 Hz, which is within tolerance of the long-time estimate of 289.9 Hz. As a comparison, the running average of the DFT pitch frequency estimate is f=287.1 Hz, which is an error of about 20 cents. This is a marginally detectable frequency error for most people. The fact that 290 Hz falls approximately halfway between the DFT sample points of 279.9 Hz and 301.5 Hz helps the DFT approximation. Since most people will naturally shift frequency slightly around their desired pitch when they sing, this 'frequency dithering' between the two nearest DFT sample points yields a result which is nearly correct for this case. However, for frequencies away from the mid-point between DFT samples, for example, a pitch of 285 Hz in this case, the DFT estimate will be very close to 279.9 Hz, which is a large error (about 35 cents). Thus, the CZT consistently gives precise frequency estimates, which makes it a valuable tool for determining the pitch of a signal over short time (i.e., real-time) scales.

Pitch Estimation via the Cepstrum

Equation (3.34) revealed that the cepstrum can be separated into two components: a short-time component which contains the formant frequency, glottal pulse, and radiational load information; and a long-time component which contains period information. Referring to Figure 3-20, there are peaks located at the fundamental period of the signal and multiples of the period thereafter with decreasing amplitude. So long as the the log magnitude of the DFT of the signal is sufficiently sampled, the cepstrum will not have any aliasing over these first few periods. Picking the peak or peaks of the first few periods and estimating a fundamental period is another method of estimating the fundamental frequency of the signal.

Figure 3-20 displays the cepstra of the three sections of the signal shown in Figure 3-19 and considered in the CZT example. The estimated frequencies for the three sections are 288.2 Hz, 290.1 Hz, and 292.1 Hz respectively. The running average of the three sections is 290.1 Hz. The precision of the cepstral estimation is similar to that of the CZT.

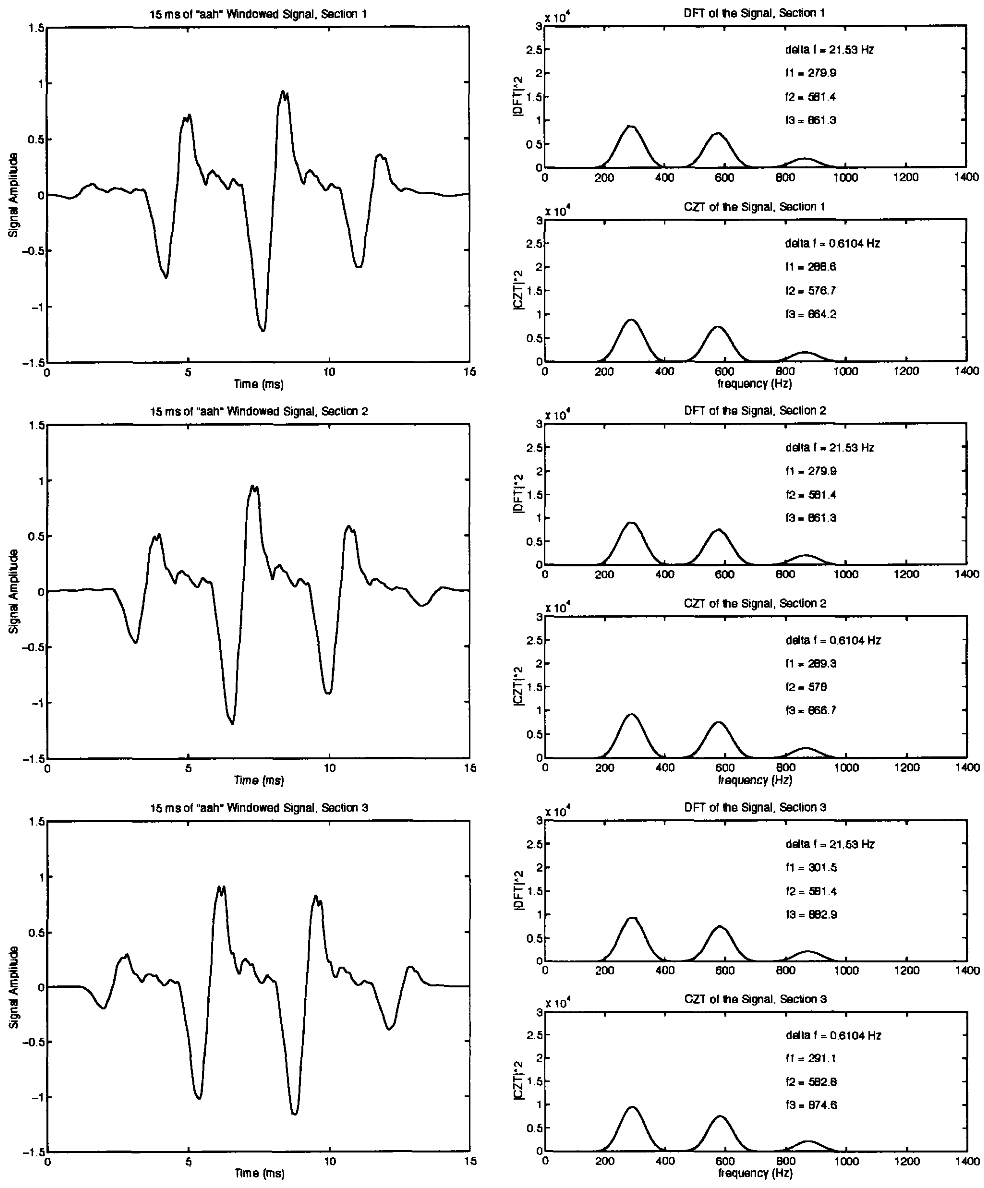

Figure 3-19: Pitch Estimation via the DFT and CZT for Three Signal Segments

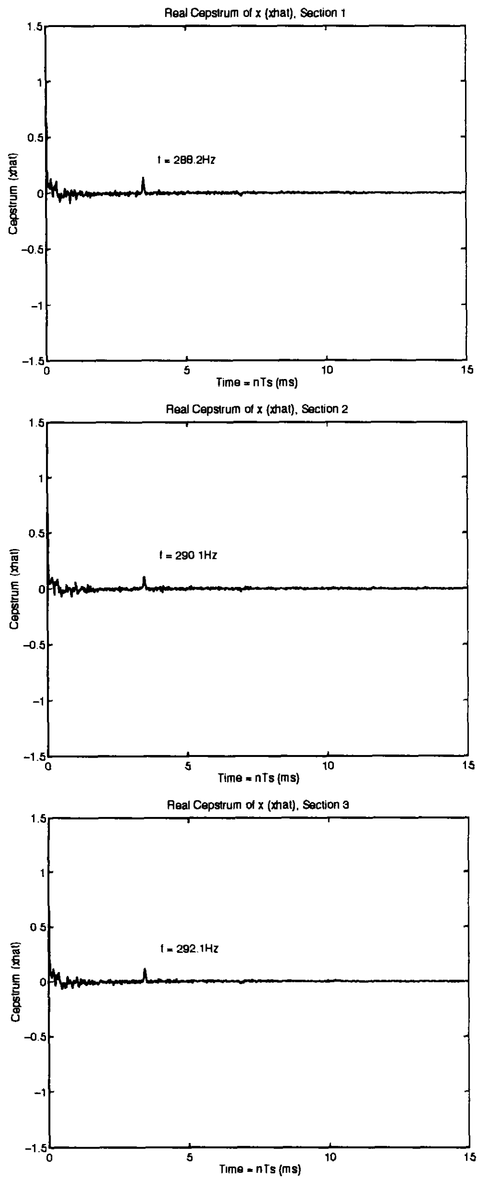

Figure 3-20: Pitch Estimation via the Real Cepstrum for Three Signal Segments

3.3 Music Generation, Interpretation, and Mappings

The next topic presented is the music generation algorithm, Sharle, developed by John Yu [36]. Although Sharle is the final component in the Singing Tree architecture (voice is analyzed, then the parameters are interpreted and mapped, then Sharle is driven using these mappings), it is presented before interpretation and mappings, because its fundamentals must be understood before the mapping algorithms can be developed.

3.3.1 Sharle

Sharle is a sophisticated MIDI generation algorithm written in C++ which composes music in real-time. It is an expert-based system, which simply means that it follows a set rules which are established by an 'expert' and assumed to summarize the fundamental operation of the system [36]. The rules are applied to musical seeds through several levels of data hierarchy. Musical decisions are made randomly, but the set of musical options from which Sharle can choose are constrained through probabilistic weighting. Readers interested in the specifics are referred to [36]. What follows is a summary of the pertinent concepts that were required to design the Singing Tree.

Sharle was first introduced as a user-controllable music generation application. The user would adjust various musical parameters, and the software would generate music with these user-defined characteristics. In [36], the algorithm is described as a cross between an instrument and a radio. An instrument is described as any interactive system which generates or transforms music via a constant input from the user. It is an example of a complicated control system in which the user is typically highly skilled. While an instrument offers very precise control of its output, it cannot fundamentally change its functionality. A piano will, for the most part, sound like a piano whether the user is playing jazz or classical. On the other hand, the radio is essentially independent of the user. The user merely turns it on and selects the music from the various stations available. In this, there is a style parameter that the user controls. The user does not have low-level control of what is broadcast on a particular station, but she can change stations. Sharle is an attempt to bridge these two extremes. It offers high-level control, similar to a radio, but also allows control of certain lower level functions. In addition, Sharle allows one to shift, to an extent, the emphasis of control between the two extremes. The Singing Tree takes vocal parameter information and interprets it such that it can be used to meaningfully control the music generation parameters.

Data Hierarchy and Music Generation Parameters

Sharle's main objective is straightforward: for a particular instrument, decide which notes to play and which notes to stop playing at any given moment [36]. To make this decision, there is a hierarchy of data objects which reflect various characteristics of the music from lowest level objects (such as notes) to the highest level objects (such as sections).

The lowest level object is the *MIDI Note,* and it is controlled by the duration parameter. The next level is the *Line.* This is simply a list of notes in the order they are to be played. At the same level as *Line* is the *Rhythm* object. This is a sequence of note-on events to be played by a given line. While not required of a *Rhythm* object, periodicity within and between rhythm objects creates a sense of continuity. *Rhythm* is controlled using the tempo, density, length, and consistency parameters. Also at the same level, is the *Layer* object. A *Layer* is analogous to a specific instrument in an ensemble. It contains the characteristics of the instrument's player, and determines how successive lines will be played. The parameters which control *Layer* are consonance, direction, pitch center, and instrument. The next higher level is the *Compound Line,* which represents the vertical arrangement of individual lines (cf., *Layer* represents the horizontal or temporal arrangement of lines). Next highest in the hierarchy is the *Stanza,* which is a collection of related compound lines. Within a *Stanza,* compound lines are repeated for unity, and are only re-generated for a new *Stanza.* This degree of unity is controlled by the cohesion parameter. The highest level object is the *Section,* which contains the *Stanzas* and is controlled by the scale, key, cohesion, and change parameters. In addition, there are other designations, such as cadence, which act to incorporate conventional musical trends and associations into the algorithm by working within one of the levels of the hierarchy. For example, cadence will slow the tempo, increase the velocity of each note attack, and end on a tonic note to indicate the end of a musical idea.

The parameters which control Sharle are listed below with a brief description of their function. This represents the 'palette' of musical controls that were available for use in designing the mapping algorithms [36]. The parameters are divided into two groups: general parameters and *Layer*-specific parameters. The general parameters are:

Cohesion Cohesion regulates unity vs. variety at the *Line* level. Increasing cohesion favors unity, while decreasing cohesion favors variety. It is used in controlling the *Line* repetition, *Line* choice, pitch offset between *Lines,* amount actually copied when a *Line* is copied, rhythm adjustment, and *Stanza* length.

Scale Scale is a discrete control. There are six scales which can be selected: 2 Major Scales, 2 Minor Scales. and 2 Pentatonic Scales which define the 12 tones in the particular scaling. The tones of each scaling are weighted 1-5 according to the likelihood of interval consonance as measured relative to the tonic. The relationships are: 1 = Tonic, 2 = Frequent, 3 = Common, 4 = Occasional, and 5 = Chromatically Related.

Tempo Tempo is the speed at which notes are produced.

Rhythm Consistency The higher the consistency, the more likely it is that the note attacks will fall at regular intervals. The lower the consistency, the less likely this will happen.

Rhythm Density Density determines the likelihood that a given attack is inserted at an available interval. As density increases, the probability that an attack is inserted increases. As density decreases, this probability goes down.

Rhythm Length Rhythm Length determines the number of available slots in a *Line.* Longer *Lines* slow down the rate of musical progression given a constant Tempo. Rhythm Density and Rhythm Length determine how many notes are played for a given *Line.*

Change Change is the rate of self-mutation. As change increases, the probability that the current *Stanza* will end in a cadence and a new set of parameters will be used for the next cadence increases.

The *Layer*-specific parameters are:

Consonance Consonance determines the probability of hearing inharmonic notes. As Consonance increases, the probability of hearing inharmonic notes decreases. As Consonance decreases, this probability increases.

Pitch Center Pitch Center is the 'center note' around which most notes will be generated. It affects the pitch range of the generated music, helps determine the note at the end of a *Line,* and also helps determine the starting note of the next *Line.*

Instrument Sets the particular instrument that is played by a particular *Layer.*

Rhythm Modification Rhythm Modification modifies the primary rhythms generated using the general parameters: consistency, density, and length. There are four modifications of primary rhythm. The first is the primary rhythm as generated. The second modification inserts new attacks in the interval between the original attacks of the primary rhythm. The third modification is a mathematical beat representation, which results in a regular beat pattern, and the fourth modification is a deterministic pattern which affects periodicity of the primary rhythm.

Sharle is a superior music generation algorithm, comparable in quality to Microsoft's recently released Microsoft Music Producer [61]. Sharle's strengths lie in its approach to composition; Sharle starts with the most fundamental musical elements, the melodic and rhythmic 'seed', and incrementally builds the music from motif to musical piece. There are numerous parameters which allow control of the composition at the various hierarchical levels. Furthermore, the algorithm uses 'expert' rules, which help to maintain musicality in the face of sudden changes in control inputs. A weakness of Sharle, in its original form, was the interface. One used a computer mouse to adjust parameters, one at a time. With such a large number of control parameters, musically interesting changes in composition can only occur when several parameters are adjusted simultaneously in an organized and interdependent manner. The Singing Tree allows a means by which participants can adjust multiple control parameters in Sharle simultaneously.

3.3.2 General Interpretation and Mapping Objectives

Given the full 'palette' of control parameters, there are two philosophies of approach to address the mapping and interpretation control problem. One could try to control each individual parameter, directly or indirectly, analogous to putting a machine together part-by-part. This was determined to be a far too difficult method for a number of reasons. First, and foremost, the musical 'gesture' or 'intention' contained in the human voice is not obvious in any context other than singing. Subtle analogies between vocal volume, pitch direction, etc., and meaningful musicality could be (and ultimately were) made, but the amount of information contained in the voice was not deemed sufficient to, for example, compose a piece of music from the ground up. Furthermore, developing a mapping for each control parameter would be very complicated. The author believes that the Singing Tree is not a case in which 'the more the merrier' applies. As it turned out, a few vocal interpretations gave a great deal of control information and, thereby, an identifiable and interesting experience. Had too many parameters been used, the control of the system would likely have been bogged down in conflicting arguments, resulting in a rather bland and mediocre musical response. Analogously, in mathematics, an over-determined system of equations will often have no explicit solution, and the only recourse is to use approximation methods. The second philosophy is to initiate Sharle into a certain playback mode, modify its output in a coarse manner to match the state of the singing, and then perturb the output in a fine manner to reveal the subtle changes in the vocal quality. This was a successful approach, because it constrained the objectives of the mappings. The potential pitfalls were 'discontinuities' in the musical experience due to flipping between coarse adjustments. Keeping this problem in mind, the interpretation and mapping algorithms were successfully designed to control Sharle in a way that would eliminate these 'discontinuities'.

The goal of the Singing Tree, as used in the Brain Opera, was to sing a single note as purely as possible. Doing so was to result in a beautiful 'aura' of arpeggiation, and failure to do so was to result in a gradually more chaotic, inharmonic response. Technically, the response was intimately related to the state of the singing voice, and the first objective was to determine what 'states' were meaningful. Artistically, the response was, again, intimately related to the state of the voice, but the first objective was to determine what instruments would play what type of music given the particular 'state' of the voice. Thus, interpretation can be considered a technical issue, and mapping an artistic issue, even though the two are truly inseparable.

Starting with the interpretation issues, the DSP toolkit [30] provided pitch, noisiness, brightness, amplitude, and, later, formant information. The interpretation of pitch was simple; the first pitch a participant sings is the intended pitch, hereafter referred to as the 'basic pitch' or 'tonic pitch'. Every subsequent measurement of pitch would be compared to the basic pitch to determine the pitch stability. It was found that throwing away a small Δt time interval at the beginning of a participant's initial vocal sample greatly improved the experience by accounting for odd transients or slides into the intended pitch. The fact that pitch changes over time, even when a person is intending to sing purely, was shown in Figure 3-19 for the author's vocal sample. While one could strictly map every variation in pitch to a control parameter, this is a rather naive approach. Algorithmic pitch estimation always involves a tolerance, or error. Thus, a small variation of Δf around the basic pitch was allowed. However, the manner in which pitch was changing around the basic pitch was deemed very interesting. Thus, the change in pitch, or 'pitch instability', the change in pitch over time, or 'pitch velocity', and the change in pitch velocity over time, or 'pitch acceleration', were used as an interpretation of musical meaning. For example, if the pitch instability changes in a periodic manner, then the participant is likely singing with vibrato, which, although contrary to the goal of constant pitch, should not be treated in the same manner as a person who's pitch instability is erratic. Or, if pitch velocity averages to zero and is of relatively small magnitude at any given time, the participant is likely doing a good job at maintaining pitch; she is simply wavering naturally. The noisiness parameter was considered to be a measurement of how well the Singing Tree knew the participant's pitch at any given time. Thresholding the noisiness parameter allowed the Singing Tree to know when to start mapping. Otherwise, it would be acting on erroneous information. Brightness was a parameter that was not used individually. Rather, it was considered a supplement to the formant information. In the end, Brightness was measured, but not mapped. Amplitude was used to dictate the volume of the response. Finally, the change in formant frequencies $f1$ and $f2$ was assumed to be an indication of change in vowel structure. It should be noted that most parameters were defined over the range of MIDI values, i.e., [0,127].

The mappings that were developed were based largely on intuition and supposition. A model would be established *a priori* for each individual parameter-interpretation matching, but it was experimentation which dictated the manner in which the mappings would be utilized. Starting with energy, the Singing Tree was either being used, or it was not. When nobody was using the Singing Tree, it was set in an incoherent, 'sleeping' state meant to resemble a sleeping brain with random thoughts passing about; quiet sounds, but with no clear musical goal. Artistically, the author believed it was very important to have the Singing Tree always singing. Many people consider interactivity as an on-off switch: I sing to you and you sing to me. However, the Singing Tree successfully demonstrated that an interactive experience can and should interact with its environment, even if no particular participant is the focus of that interaction. The artistic goals of the sleeping state are

- Instrumentation should be bizarre, synthesized, 'spacey'.
- Voices from the chorus should sing the frenetic motifs.
- Tempo should be slow, with little consistency in rhythm; it should sound as if instruments are playing at random, but in a sporadic manner (i.e., not in a dense manner).
- Pitch should be inharmonic, or better stated, there is no tonic for reference. However, the choice of instrumentation is such that many instruments are not pitched in any real sense. Those that are, such as the voices, are fundamentally inharmonic.
- All video should be in the initial state (front view of a sleeping woman's head, side view of a sleeping woman's head, side view of a hand holding a flower bulb).

When there was an energy value, the Singing Tree was put into the 'awake' state, at which time it would await further mappings. Assuming the participant is singing her pitch purely, it was decided that the following response should happen.

- A string should quietly hold the base pitch behind all other musical events.
- Another string should always match the present pitch the participant was singing (which is essentially the same as the base pitch in this case).
- Bassy strings should round out the low-end of the sound in tonal consonance.
- Woodwinds and flutes should arpeggiate up and down the register in tonal consonance with a pulsating and consistent rhythm.
- Voices from the chorus should sing the basic motifs (defined in Section 2.4.2) in consonance with the participant.
- Video should progress forward through the video frames at the standard 30 frames/second.

The higher timbre of the woodwinds and the bassy strings give a broad, expansive feel to this musical experience. It is angelic and calm. Finally, if the participant's pitch varied outside the allowable range, the following response was in order.

- A string should quietly hold the base pitch behind all musical events.
- A string should always match the present pitch the participant was singing (which is now different than the base pitch).
- Instrumentation should gradually change to brassier, more percussive sounds.
- Rhythm and tonality should gradually become more loose and less coherent.
- Voices should sing the intermediate motifs with gradually less coherence.
- Video should regress through the video frames.

Given the above phenomenological outline of how the interpretation and mappings worked, the following contains the specific implementations for the most important mappings.

3.3.3 Defining Pitch Instability

The pitch instability is a measurement of how poorly the participant is maintaining the desired goal of the system, a steadily sung pitch. The first note a participant sings into the Singing Tree is the basic pitch. Thereafter, all deviations in pitch are measured with reference to the basic pitch. Pitch instability is the difference between the current pitch the participant is singing and the basic pitch (i.e., delta pitch). The farther one is from the basic pitch, the larger his pitch instability. The basic pitch is maintained until the energy of the signal drops below a small threshold level for approximately two seconds. This means that the participant has stopped singing for a reason *other than to take a breath.* In other words, the two second pause was included to allow a participant to take a breath without the Singing Tree 'resetting'. Even if the

Table 3.4: Instrument Groups Used in the Singing Tree

Group Name	Program Number and Instrument Name			
Airy Sounds Ins-group 1	26	Synth Caliope	32	Fluty Lead
	123	Baroque Flute	139	Horn and Flute with String
Airy Vocals Vox-group 1	200	I-OO	203	V-3
	206	I-OO Reverb	209	V-2 Reverb
	212	I-OO Reverb-lots	213	I-Ahh Reverb-lots
Percussive Sounds Ins-group 2	10	Dyna E-Piano	171	Multi Marimba
	176	Mallatoo		
Miscellaneous Sounds Ins-group 3	29	Hyper Guitar	132	Trumpet Section
	153	Fuzz Lite	156	Timer Shift
Miscellaneous Vocals Vox-group 3	210	VI-1 Reverb	211	VI-2 Reverb
	216	VI-1 Reverb-lots	217	VI-2 Reverb-lots
Harsh Sounds Ins-group 4	15	Big PWM	21	New Shaper
	45	Prophet Sync	67	Mark Tree
	93	Cee Tuar	129	Almost Muted
	153	Fuzz Lite	161	PPG 4
	197	Doomsday		
Effects Ins-group 5	62	Rhythmatic	157	Aurora
	164	Spaced	172	Wave Power
	192	A No Way CS	193	Environments
	194	Gremlin Group		
Brass Instruments Ins-group 6	127	Miles Unmuted	131	Sfz Bone
	135	Orchestral Brass	143	W Tell Orchestra

participant takes a deep breath which lasts longer than two seconds, she will very likely commence singing once again on the same note, again, if the intention was simply *to take a breath.* Regardless of intentions, if the two second limit is up and the participant sings a new pitch, then this becomes the new basic pitch and all pitch instability is measure with reference to this basic pitch.

3.3.4 Instrumentation

The instrumentation of the Singing Tree is divided into seven groups, representing the various styles and timbre of instrument to be utilized. A summary of the instruments along with their K2500R program numbers (as used in the Singing Tree) is given in Table 3.4. Each group name characterizes the type of instrument within the particular group.

These instruments were chosen by the author considering the various modes of the Singing Tree. For example, to the author, the response for singing purely (i.e., no pitch instability and no noisiness) should consist of strings and woodwinds mostly. The reason for this is, simply stated, that these instruments best fit the description of 'angelic feedback'. Listening to all the sounds on the K2500, the Synth Caliope, Fluty Lead, Baroque Flute, and Horn and Flute with String provided an 'angelic instrumentation'. Originally, the horn sound was not included, but the response without the horn proved to lack a 'majestic' character that the author felt should be there. However, horn alone was too up-front. The Horn with Flute and String was an appropriate balance between 'majesty' and 'reserve'. Similarly, the remaining seven categories of instrument were chosen by the author, considering the desired instrumentation of the playback. The Airy Sounds and Airy Vocal groups were chosen primarily for the 'angelic' playback. The Percussive Sounds, Miscellaneous Sounds, and Miscellaneous Vocals were chosen as 'transition' instruments. Instruments from these groups would be included in the playback 'gradually' as the pitch instability became greater. The Harsh Sounds, Brass Instruments, and Effects were used for significant deviation from the basic pitch (i.e., the most chaotic and agitated response). The sleeping state uses the Effects and the Frenetic Vocals. Note that the Frenetic Vocals are not listed above, because they are only used in the sleeping state, and therefore are not mapped by the voice (but, rather, by the absence of voice).

The selection of an instrument group for playback was dependent on the selected tolerance of the system. There were four levels at which one could use the Singing Tree. Each level defined the meaning of pitch stability differently, resulting in a difficulty level that ranged from easy to most difficult. Given a pitch-instability (PI) for a given vocal signal which is defined over the range of MIDI values [0,127], one can define an initial pitch stability measurement, W.

$$W = 127 - PI \tag{3.54}$$

Based on this initial measurement of the pitch stability, and defining f to be the reliability based on a running average of the noisiness parameter normalized to values [0,1], the following mapping algorithm scaled the pitch stability depending on the difficulty level.

Difficulty 0	$W' = 127$		always perfect

Difficulty 1	$W' = \begin{cases} int\left[\frac{0.5+127W}{127-bonus}\right] \\ 127 \end{cases}$	bonus = a number
		if $W' > 127$
			easy

Difficulty 2	$f' = \begin{cases} 1.0 \\ 4f \end{cases}$ $W' = f'W$	$f > 0.25$ otherwise	difficult
Difficulty 3	$W' = fW$		most difficult

(3.55)

In Difficulty 1, the *bonus* variable is scalable to make the mapping easier or harder. In Difficulty 2, the reliability f is itself scaled before it is used to scale the pitch stability W. Given this new value for the pitch stability, W', the instrumentation can be determined via the following algorithm. First, the instruments and vocal channels are considered separately. Considering the dynamic instrument channels, increment through each channel number, c, which is using a dynamically selected instrument. In the Singing Tree, channels 1-2 were dedicated to the two string sounds Big Strings (#163) and Big Strings with reverb (#218). These are used respectively to hold the basic pitch and follow the participant's pitch via pitch bend. Channels 3-9 were used for instruments which could be dynamically updated. Using a random generator, establish a 75 percent probability that an instrument on this channel will be re-assigned via the first algorithm, and a 25 percent probability that the algorithm will be re-assigned via the second algorithm. Then, increment to the next channel.

75%	if	$(W'+2c)$	$<$	64	then	Ins-group	$=$	4 Harsh Sounds
	else if	$(W'+2c)$	$<$	108	then	Ins-group	$=$	3 Miscellaneous Sounds
	else					Ins-group	$=$	1 Airy Sounds
25%	if	$(W'+2c)$	$<$	50	then	Ins-group	$=$	5 Effect
	else if	$(W'+2c)$	$<$	70	then	Ins-group	$=$	2 Percussive Sounds
	else					Ins-group	$=$	1 Airy Sounds

(3.56)

The vocal instruments are all set using the same algorithm. Here, d is a number defined such that $9+d$ is the channel on which the voice is played. The algorithm allows voices on channels 10-16, but, due to voice stealing phenomena, typically channels 12-16 are turned off at the K2500R.

if	$(W'+2d)$	$<$	108	then	Vox-group	$=$	3 (Miscellaneous Vocals)
else					Vox-group	$=$	1 (Airy Vocals)

(3.57)

A synopsis of the instrumentation algorithm is as follows. Channels 1-2 are dedicated to the strings which hold the basic pitch and follow the participant's voice; these are never changed. Channels 3-9 are assigned instruments dynamically, based on the stability of the participant's pitch and the difficulty level (which is a function of reliability). The instrument on each channel has a 75% chance of being assigned via one algorithm, and a 25% chance of assignment via the other algorithm. The assignment is a function of channel increment, c, and so each channel has a slightly different assignment range. This helps to prevent discontinuities between instruments of different groups, which may have markedly different timbre and styles. The vocal channels are all changed via the same algorithm, and are a function of d, where $d+9$ is the channel on which the voice is found.

3.3.5 Dynamic Parameter Mapping and Control

Having assigned the instruments, the algorithms playback instruments can be assigned using information from the voice. However, the style of music does not change. The next mapping is designed to change the consonance (as defined in Section 3.3.1) of the playback as a function of the pitch stability. Continuing with W', the pitch instability as defined by the difficulty level, the following mapping is used to assign the consonance parameter C over the range [0,127].

$$\boxed{\begin{array}{lcclcccl} \text{if} & W' & < & 64 & \text{then} & C & = & 0 \\ \text{else if} & W' & > & 110 & \text{then} & C & = & 100 \\ \text{else} & & & & & C & = & int\left(127\frac{W'-64}{110-64}\right) \end{array}} \tag{3.58}$$

An interesting implementation here is that C is a maximum at $W' = 110$. The background to this decision lies in the following logic. Originally, the consonance was scaled linearly to 127. However, for very high consonance, the playback becomes rather dull; it is far too consonant. Scaling linearly to some maximum value would be one solution, but consider the following. The goal of the participant is to sing purely at the basic pitch. No instructions are included with the Singing Tree, and, except for a brief introduction, most participants are discovering the experience as they sing into the Singing Tree. Thus, it is a 'responsibility' of the instrument to lead the participants to the basic pitch, or at least clarify the goal. The reason that the consonance is not linearly scaled to 100 as W approaches 127 is to help the participant identify the goal. As a participant starts to deviate from the basic pitch, the consonance will actually *rise* to help her hear the basic pitch more clearly. With her bearings straight again, she begins to slide back towards the basic pitch, much like a damped pendulum returns to its equilibrium position. This, in turn, causes the consonance to *drop* slightly and provide a richer musical experience. The participant hears the richer response and maintains the basic pitch. This is one of the more subtle examples of the reward-oriented response provided by the Singing Tree. One might question why the participant does not stay at the pitch which causes a consonance of $C = 127$. The answer lies in the use of the Big Strings to both follow the participant's voice and, simultaneously, play the basic pitch. If the participant does sit on the pitch which is far enough from the basic pitch to result in a consonance of $C = 127$, the result is a sound in which most instruments, the voices, and one of the Big Strings are played back on the basic pitch, while the participant and the other Big Strings are at a pitch which is slightly out of tune. Thus, it is easy to hear the dissonance, and yet the overwhelming majority of instruments are playing the basic note, 'calling' the participant to return. In other words, $C = 127$ stops the harmonic embellishment and accentuates the pitch discrepancy between the basic pitch and the participant's pitch.

The next mappings are those relating pitch velocity, scale, and rhythm consistency. Given the pitch velocity (PV) and rhythm consistency (RC) scaled over [0,127], a tight constraint of $PV < 5$ matched well the concept of a purely sung pitch. Thus, the following algorithm was used to in cases in which $PS > 5$.

$$\boxed{\begin{array}{lcccccl} \text{if} & PV & > & 5 & \text{then} & \text{scale} = & \text{minor/pentatonic} \\ & & & & & RC = & 127/PV \\ \text{else} & & & & & \text{scale} = & \text{major} \\ & & & & & RC = & 127 \end{array}} \tag{3.59}$$

In other words, for pitch velocities greater than 5, the scale would change to either minor or pentatonic, and the rhythm consistency would decrease. Thus, even if a person is hovering close to the basic pitch, if she is moving around with a high pitch velocity, the response will change key and have a less consistent rhythmic structure. An analogous mapping was made for formant velocity, defined as the change in formant frequency over time. Surpassing a formant velocity threshold would cause a new scale to be played and a decrease in rhythm consistency.

The formant structure was interpreted in a simple manner: if the ratio $f2/f1$ were high, then voices singing an 'ooh'; if the ratio were low, then the voices would sing an 'aah'. While this is a dramatic simplification of formant analysis, the goal is not particularly to match the vowel of the singer. In reality, one often hears a chorus singing 'aah' behind a soloist who may be singing any number of vowels. Rather, the objective here is to simply recognize variation in the participant's vowel structure and respond accordingly. Thus, when the participant changes vowel structure significantly, the chorus will change as well. Note that the formant frequencies are maintained as running averages (as are many of the vocal parameters) to prevent instantaneous changes from creating discontinuous flip-flops between the chorus singing 'aah' and 'ooh'. The algorithm is as follows.

$$\boxed{\begin{array}{lcccccll} \text{if} & f2/f1 & > & 70 & \text{then} & \text{chorus vowel} & = & \text{ooh} \\ \text{else if} & f2/f1 & < & 58 & \text{then} & \text{chorus vowel} & = & \text{aah} \\ \text{else} & & & & & & & \text{no change} \end{array}} \tag{3.60}$$

Another mapping was based on the thresholding of the running average of the pitch. If the pitch stayed constant within a small threshold Δ, then the *Layer* object was told not to change. However, once this Δ was surpassed, a call is made to the *Layer* object instructing a new layer to be generated.

The Singing Tree was originally designed to supply vocal samples to the performance space. Unfortunately, this function was never realized. This was actually a blessing in disguise. By requiring vocal submissions from the Singing Tree, the Singing Tree could only send samples of voices that were singing at a MIDI note number pitch (i.e., a key on the piano). Since most people do not have perfect pitch, this meant that a participant would not be able to sing *any* note into the microphone and expect a perfect match. In other words, a participant could, of course, sing any note, but the Singing Tree would have to select the closest MIDI note to the participant's pitch as the basic pitch. The consonance algorithm would lead the participant to the MIDI note, but a participant with a good sense of pitch would be able to realize that the basic pitch was slightly different than the initial pitch she had sung. However, this constraint was removed before the Lincoln Center debut, and the Singing Tree was adjusted accordingly. To allow *any* pitch to be sung, Sharle kept account of how far the initial pitch (i.e., the basic pitch, but not a MIDI note number) was from the nearest MIDI note number. Sharle then sent out MIDI commands to the K2500 to pitch bend the playback music up or down the difference. Another consideration was the modulo nature of pitch; considering semitones, octaves are modulo 12. Thus, people were rewarded for singing not only the basic pitch, but any integer octave away from the basic pitch.

The Role of Probability

Sharle is a music generation algorithm which uses randomly generated numbers to achieve probabilistic playback. This means that the mapping algorithms put Sharle into a specific 'probabilistic' state, but its behavior once it is in that state is random. This serves to present a consistent, yet non-deterministic response. Considering an example, singing purely (with no pitch instability and no noisiness) will put Sharle into a state in which the consonance is $C = 100$. This means that the highest probability for playback note is assigned to the Tonic Relationship, with a smaller, but still significant, probability assigned to the Frequent Relationship and the Common Relationship, marginal probabilities assigned to the Occasional Relationship, and nearly zero probability assigned to the Chromatic Relationship (see 'Scale' in Section 3.3.1). First, Sharle will randomly (probabilistically) choose the type of Relationship that will be played. Sharle then randomly chooses the particular note from the set of notes within that particular Relationship. For example, if the Tonic Relationship is selected and the playback scale is the major scale, then only the tonic of the particular scale and its octaves can be selected. For the Frequent Relationship, the 3rd, the 5th, and their octaves are candidates for selection. Note that the 'expert knowledge' of the system and other control parameters may further constrain the selection of the particular notes within a given Relationship set, which is yet another layer of probabilities. What the example demonstrates, is that the role of the mapping algorithms is truly perturbative, making explicit determination of an output as a function of its input impossible. One could construct an instantaneous stochastic representation for the output of the system at some time, t, if and only if one knew all the inputs and the current state of the system.

3.3.6 Examples of Operation

The following are examples of the Singing Tree's operation in three modes: the sleeping mode (high pitch instability), the pure pitch (no pitch instability and no noisiness), and the harsh mode (marginal to moderate pitch instability). Be advised: the description of any mode can only indicate the parameter values and the *likely* behavior of the system, since, ultimately, the music Sharle plays is randomly generated.

Sleeping Mode Example

The sleeping mode is completely 'composed'. Since nobody is singing into the microphone, Sharle is put into a particular mode, characterized by the control parameters being set to certain values, and allowed to randomly generate music. The cohesion parameter is set to zero, which makes the output sound inharmonic. Density is also set to a very low value of five, and the rhythm length is set to a moderately high 32. The volume is set to 45. The instrumentation is selected randomly from the Effects and Frenetic Voices. The result is the desired 'sleeping brain' mode in which strange voices are played sporadically at random times with low volume.

Pure Mode Example

In the Pure case, the pitch instability (PI) and noisiness of the signal are zero. As a result, the scaled pitch stability, W', is equal to 127 and is independent of the difficulty level. This implies that all instrumentation will use the Airy Sounds and Airy Vocals. The Consonance parameter, C, is equal to 100. As described in Section 3.3.4, $C = 100$ implies a high probability of hearing notes from the Tonic, Frequent, and Common Relationship sets, with lower probabilities of hearing notes from the Occasional and Chromatic Relationship sets. The pitch velocity is zero, implying the scale is major and the rhythm consistency, RC, is 127. Thus, the rhythm has a very high probability of falling at regular intervals, and generated rhythm patterns will tend to be repeated. For example, an arpeggiating phrase may contain 8 sixteenth notes, and this same phrase is repeated several times before being regenerated. Depending on ratio of the two formant frequencies, the vocal samples would be either 'aah' or 'ooh'.

The parameters mentioned up to this point perturb Sharle in its Pure Mode setting. In other words, the remaining parameters are set to appropriate values and held constant. These include a rhythm length of 20, which constrains the length of the *Line*. The rhythm modification parameter is set to one, which merely takes the primary rhythm as generated (i.e., no modification). Rhythm density is set at an established level, which is slightly different for the various MIDI channels as a source of variety. Note that while the density remains the same for a given channel, the instrumentation on that channel can change.

Harsh Mode Example

In this mode, the changes in perturbation parameters are few, but significant. The PI and noisiness have values now such that, depending on the difficult level, scale W'. This implies that the instrumentation will be selected from the Miscellaneous Sounds, Miscellaneous Vocals, Harsh Sounds, Effects, Percussive Sounds, and possible the Airy Sounds. The Consonance parameter, C, is likely to be less than 100, and the probability begins to even out among notes of all Relationships. If the participant is changing pitch as well, then the pitch velocity will likely exceed 5, and the scale will change, along with the rhythm consistency. The result is a sound characterized by inconsistent rhythms, increasingly likely inharmonicity, and agitated instrumentation.

3.3.7 Discussion of Design Methodology

At the risk of repetition, the author summarizes the design methodology of the Singing Tree mappings and the results. The design of the Singing Tree mappings was largely experimental in nature. This does not imply, however, that the author and John Yu made wild trial-and-error guesses that eventually paid off. Rather, like an experimental scientist, the 'black-box' hypothesize, test, re-hypothesize methodology was employed. The fundamental assumption was that the various control parameters were independent of each other, and their effects on the output were also independent. While there is really no formal justification for this approximation, intuitively it is reasonable to assume. For example, the notion that changing the volume parameter will affect the volume and not the pitch of the output is obvious. The issue becomes sketchy, however, when choosing instruments of different instrument sets. The Harsh set instruments, on the whole, tend to be louder than those of the Airy set. Thus, this illustrates a case in which instrument selection changes both the instrument (which it should do) and the output volume (which it should not do). It is merely a case given for the purpose of counterexample to the assumption of independence, and it is not intended to make a significant impact on the reader. Independence was assumed.

Given independence of control parameters, one can analyze and adjust them individually. Considering a particular mode of the Singing Tree, the Pure Mode, and a parameter to investigate, Consonance, one can first form a hypothesis and then experiment. For example, hypothesize that Consonance values greater than 110 are acceptable for the Pure Mode, while they are increasingly unacceptable as Consonance drops from 110 to 0. The hypothesis regarding the behavior of Consonance follows directly from its definition in Section 3.3.1. However, its specific effect on Sharle's output, what it sounds like, how quickly the output changes with changes in C, etc., are only clarified through experiment. One listens to Sharle's output for many different trial values of C, and based on the results, discovers that the unacceptable range is actually for values below 90. The next hypothesis that must be made requires both an artistic and technical fluency with the goals and nature of the system. The question is, 'what vocal parameter should control the consonance?', and it is highly dependent on one's knowledge of the voice, the vocal parameters available, the creative interpretation of what they might mean, the artistic goals of the system, and how Sharle might respond to a control parameter manipulated in such a way. Since the Singing Tree should provide a less tonally coherent response as one leaves the basic pitch, one hypothesizes that C should be linearly scaled to the pitch stability (i.e., if the person is right on the basic pitch, pitch stability and C are 127, and they both decrease together to 0). Testing the Singing Tree with this mapping results in an output which is less tonally coherent as one leaves the basic pitch. Note that the notion of independence is still important. No other parameters are being changed. Thus, the instrumentation in this case is always the Airy Sounds and Airy Vocals, the volume is constant, the scale is constant, etc. The only perceptible change in the output is the change in tonal coherence. Continuing with this example, the designer listens to the output, and determines that a coherence of 127 when a participant is singing the basic pitch is not desirable, because the output is not interesting. At $C = 127$, the output is almost entirely octave and 5th intervals. $C = 100$ offers more variety while maintaining the 'angelic' sense, and so the hypothesis is to map a pitch stability of 127 (right on the basic pitch) to $C = 100$ and scale linearly to zero. Testing again reveals that this is acceptable. However, at certain difficulty levels, this mapping may make the experience too difficult for some to enjoy. New hypotheses are formed and tested for the various difficulty levels. This process is repeated for each control parameter that one decides to use. Since independence is assumed, each one can be adjusted individually and results in the same output trends when implemented simultaneously.

An important issue is deciding which parameters to adjust and which to hold constant. In the Singing Tree, this was particularly relevant because of the large number of control parameters on the 'palette'. It is the author's opinion that one should hypothesize which parameters will be the most significant, and adjust those first. One should only use the parameters necessary to adequately model the desired response. This does not mean that one should ignore the more subtle effects, but one should certainly test the parameter mappings for relevance. If a particular mapping makes an artistically important difference and benefits the experience, then it should be used; otherwise it should not. Note that 'artistically important' does not imply 'obvious'. The results may be very subtle, yet profound. The Singing Tree worked using only a fraction of the possible control parameters, and, yet, the ones that it did use provided a unique and rewarding experience.

3.3.8 The Issues of Algorithm, Constraint, and Composition

In its native form, Sharle is an algorithm. When one uses Sharle, one constrains the algorithm with the various control parameters. In Sharle's original interface, this was accomplished with a mouse by adjusting one parameter at a time. In the Singing Tree, multiple constraints were adjusted simultaneously. However, a fundamental and largely philosophical question left for the reader to ponder is, 'Is this composition?' The author believes that the answer is really 'yes' and 'no', depending on one's notion of composition.

Consider a composer who is also a software engineer. The composer is not really all that good, and writes a mediocre piece of music. He decides to program his computer to play it exactly as written. It is certainly a piece of algorithmic music, albeit a very simple algorithm. Did the computer compose this music? Did it generate this music? Our composer has a friend, Mr. Expert, who is both an expert composer and an expert software engineer. He programs the computer with rules based on his expert knowledge of composition to 'fix' the mediocre composer's piece of music, which it does. The result is a very 'nice' piece of music that is certainly better than before. Yet, the best that can be said for it is that there is nothing technically wrong with it. The computer has modified it, but did it compose the music? Did it generate the music? Mr. Expert decides that he wants his computer to 'compose' music in his own aural image, and programs the computer to generate a piece using his expert rule-base. The result is a rather plain example of Mr. Expert's music. Now is the computer a composer?

It reminds the author of composition classes in music school. One is assigned to write a piece of music in the Baroque Style (the style of Bach), which has very strict rules regarding the interval relationships of adjacent notes. He first writes a piece that he finds *very interesting and profound,* and then puts it through the 'Bach Filter'. This changes all the notes which did not follow the rules, as established by Bach, and spits out a piece which sounds a lot like bad Bach and is *neither very interesting nor profound.* Who composed this piece of music?

Disgruntled with this approach to music education, the author quits music school and decides to become a rock star. He furiously writes 101 pieces and presents it to Sony Entertainment for evaluation. Sony tells him that one of the songs is OK, but the remainder sound exactly like the 100 CD's (of famous, expert musicians) he has at home. Since the author did not grow up in a musical vacuum, in a sense, his 'expert rule-base' has been largely determined by his musical environment. Did he compose all 101 songs?

To give a last example, anyone who has listened to the background music of a TV show has been introduced to 'algorithmic music' as produced by a human. Is this composition? These philosophical questions can continue indefinitely. The point one should remember is that Sharle can compose music better than most people, simply because most people do not write music. In this sense, Sharle is a 'composer'. However, Sharle is simply an algorithm, however complicated, which probabilistically follows an expert rule-base. It is not cognizant of its creation, and in this respect, is not even a musician, let alone a 'composer'. In the author's opinion, as long as computers lack the ability to be affected by their music, their music will lack the ability to affect others. They are not 'composers' in the sense that Bach, Beethoven, and Mozart were. However, as generational algorithms incorporate larger and more sophisticated expert rule-bases, the degree to which they can mimic composition will increase. In this respect, computers will become excellent 'composers'. Finally, the issue of constraint will also become increasingly important in generational music. It is the author's opinion that knowing which music *not* to write can be as important to the composer as knowing which notes to place on the staff. Cage has taught us the value of silence. Constraining generational algorithms may make their slight of hand all the more believable.

3.3.9 Visual Feedback Control

While most of the attention of this thesis has been directed toward the aural issues of implementing the Singing Tree, another fundamental part the Singing Tree experience is the visual feedback. Three video streams were designed and produced by the Brain Opera Visual Designer, Sharon Daniel [34], with support from Chris Dodge [62]. A brief description of each video stream is as follows:

The Dancer This video stream starts with frontal view of a sleeping woman's face. As the video progresses, the woman's eye opens and the camera zooms into the pupil to find a spinning dancer dressed in white.

The Singer The Singer is a side shot of a sleeping woman's face. As the video progresses, the woman wakes, opens her mouth, and begins to sing. A visual 'aura' meant to represent the singing voice swirls out of her mouth and begins to get brighter. When the 'aura' is at its brightest point, the face is no longer visible.

The Flower The third video stream is a shot of cupped hands cradling a flower bulb. As the video progresses, the hands close, and then reopen to reveal a rose. The rose then explodes in a bright light and flower petals shower down the screen.

The three videos are common in that they start in a sleeping state, and 'wake' to progress through a series of events which leads to an obvious ending or goal. The mapping used to drive the video was based on pitch instability. If the pitch instability was low, then the video would progress forward. Otherwise, the video would regress back to the 'sleeping' state. The video clips were successful because of their simplicity of purpose; it was easy to identify when the video was rewarding one's singing and when it was not. In two of the three clips, it was possible to hold the final scene indefinitely. This was an interesting design, because it meant that the experience was indefinite in these two cases. For example, when a participant sang purely for a long period, the camera zoomed into the eye, and the dancer continued to spin until the pitch changed. The Flower, on the other hand, repeated after the petal shower. Having seen all three video clips used at many venues, it is the author's opinion that the indefinite experience was better received. Of course, the best implementation would have been an interactive video, in which the color, brightness, contrast, focus, or other algorithmic distortion was controlled by the user's ability to sing the desired note. This approach was not used in the Singing Tree, because the real-time latency limit had already been reached simply by putting the bitmap on the LCD screen. Any further algorithmic operations on the video would have compromised the real-time nature of the interface.

Chapter 4

Results, Observations, and Future Directions

4.1 Observations of The Singing Tree in Various Venues

The Singing Tree was successful as a stand-alone interface and as part of the larger interactive experience called the Brain Opera. The author made a point to ask many participants their thoughts and comments on the Singing Tree experience. Most were positive, some were negative, but many were insightful. The notable feedback along with the author's own observations and evaluations are discussed below.

4.1.1 The Singing Tree as a Stand-Alone Interface

During development and testing of the Singing Tree, the author and his colleagues had more than ample opportunity to demo their interactive instruments as stand-alone interfaces to the sponsors and visitors of the Media Laboratory. Alan Alda was the first person from outside the Media Laboratory to try the Singing Tree, as part of a Scientific American Explorer Documentary on the lab. Although famous for his acting, Alan Alda is also an accomplished singer. Singing with a full operatic voice, he easily held the pitch steadily and was able to hold the angelic aura consistently. After he became accustomed to its operation, he began to deliberately leave the pitch to discover the instrument's behavior. Eventually, he was talking and laughing into the Singing Tree. In talking with him a few months later at the New York City debut, he mentioned that he liked the instrument, that it felt comfortable, intuitive, and fun to experiment with. At the other end of the spectrum, some people are very intimidated with the idea of singing into the microphone. Many are unsure what the response will be, and thus are afraid to sing into the microphone. Others are intimidated by the goal (to sing one note purely), thinking that only a virtuoso could make it work. Some simply think, 'Am I supposed to sing into this thing?' However, most agree afterwards that it is a very meditative and rewarding experience. It is easy to forget one's surroundings when using the interface, because the feedback is both aural and visual. Because of the Singing Tree's high degree of sensitivity, the participants are well aware that they are an integral part of the feedback loop.

4.1.2 The Singing Tree as Part of the Mind Forest

One of the author's favorite experiences was to walk into the Mind Forest, shown in Figure A-3, when nobody was present and listen to the Singing Trees in their 'sleeping' state. The atmosphere was, in the author's opinion, that of a sleeping brain; 40 interactive 'agents' in a well-designed, organic/industrial-looking structure with three Singing Trees quietly humming with an occasional chatter from a rhythm tree. When participants arrived, often in groups of 100 or more, the Mind Forest would 'wake'. With 40 interactive musical instruments in close proximity to each other, the sound level in the room would increase dramatically. Oftentimes, it was too high to hear the 'sleeping state' of the Singing Trees without wearing the headphones. This was part of a broader problem with the Mind Forest audio levels in general. Increasing the volume of the 'sleeping state' would solve the problem in the packed Mind Forest environment, but the 'sleeping state' levels would then be far too loud once the audio levels in the Mind Forest had dropped back to quieter levels. Another criticism of the Singing Tree related to high audio levels in the Mind Forest, was the occasional feedback into the microphone from surrounding experiences. To prevent this, the microphone was gated. However, this in turn required people to sing louder and be closer to the microphone. Participants who were shy or intimidated might have been disappointed if their attempts to sing into the Singing Tree were apparently rejected, simply because they weren't singing loud enough to overcome the gate. The solution lies somewhere in the sticky issue of sound isolation in small spaces, which the author defers to another thesis.

An interesting observation, which was not solely a Singing Tree phenomenon, was that the very young and the very old enjoyed tremendously the Mind Forest experiences. In particular, lines would often form at the Singing Tree because a child, completely unaware that 10-20 others were waiting for him to finish, would be completely involved in the experience. Children, in particular, would hold the note purely for as long as possible. Many would continue for minutes, intently listening to the response. Somewhat surprisingly, the elderly readily accepted and enjoyed the Mind Forest experiences. Although they knew the experiences were computerized, most were not intimidated. The Brain Opera visual designers did a great job keeping the computer out of the experience, and making the instruments approachable, interesting, and user-friendly. In addition, many of the elderly did not have any preconceptions as to what an interactive experience should be. In this, they were more open-minded to the Brain Opera than many others who, for whatever reason, did not take the time to discover what the instruments were and how they worked.

As a last comment, I found great pleasure in watching and listening to participants who were initially singing quietly because they were self-conscious, only to later be singing quite loudly and wildly in an attempt to get unique and interesting responses. One could say that the responses that the Singing Tree evoked from the participants were as interesting as the ones the Singing Tree generated.

4.2 Critical Evaluation of the Singing Tree

The Singing Tree was a very successful and engaging musical interface. It worked as per the established design criteria, and many participant's commented on how much they enjoyed the experience. In a WYSIWYG consumer report, the Singing Tree would likely do quite well. Nonetheless, it is often the designer who can offer the most reliable criticisms and analysis of the research. The designer is thoroughly familiar with the system, and he knows all its strengths and weaknesses. What follows is the author's critical evaluation of the Singing Tree.

4.2.1 Strengths of the Singing Tree System

In the author's opinion, the most successful aspects of the Singing Tree are as follows.

- The Singing Tree always sings.
- There were no discontinuities or 'glitches' in the musical experience.
- The Singing Tree was accessible to people of all ages and musical abilities.
- The Singing Tree-human feedback mechanism worked remarkably well at leading participants to the established goal.
- The physical design removed the computer completely.
- The video streams of indefinite length worked well.

Perhaps the most successful and significant contribution that the author made to the group was the concept of continuous interactivity. The Singing Tree always interacts with its environment, whether that environment is a specific participant or simply the empty Mind Forest. The previous model for musical instruments, interactive or otherwise, was contingent upon the presence of a user. The participant provides some sort of input, and then (and only then) the instrument provides a response. While background music in restaurants, elevators, and stores are examples of continuous musical output, they obviously lack the interactive component. Previous applications of interactive instruments have typically been from the point of view of a participant interacting with the instrument and receiving feedback. The Singing Tree is an example of an instrument which *interacts with the participant* . Although one may dismiss this point as merely an arbitrary, cerebral fixation of the author, consider the following. The Singing Tree is always singing; it *inputs* stimuli into its environment. This *evokes a response* in passers-by to use the Singing Tree. Their *response* is to sing into the tree. The Singing Tree, in turn, *leads* the participant to the established goal. If a participant behaves in an ideal manner and always tries to reach the goal, then the roles of *participant* and *interactive instrument* are interchangeable. The difference, of course, is free will. The participant will not always act in such an ideal manner. Nonetheless, it is the author's opinion that the concept of continuous interactivity is an important distinction between previous works and the Singing Tree. Furthermore, it is one of the most significant reasons that the Singing Tree was considered such an engaging experience. The Singing Tree began to break down the boundary between participant and instrument.

Another significant success was the continuity of the musical experience over wide variations in instrumentation and musical style. A participant who quickly changed between the basic pitch and other pitches heard the music style and instrumentation change significantly without ever losing a sense of musical continuity. There were no glitches, discontinuities, or obvious stops. In addition, the changes were real-time. The primary reasons for this success were the musical mappings and Sharle's multi-layered approach to sound generation [36]. While control parameters would change the internal state of Sharle, Sharle's allegiance was to the expert rule-base. One of the ways this expert rule-base manifested itself was the multi-layered approach to implementation. Since the melodic and rhythmic 'seeds' were generated and embellished through several layers under the guidance of rules, instantaneously changing control parameters would not result in an instantaneous change in output *if it broke an expert rule.* In other words, the rules often dictated how a transition would occur. For example, if the participant were to suddenly leave the basic pitch and sing with a high pitch instability, the control parameter would say, 'change to the harsh and chaotic state immediately', while the rule-base would respond, 'yes, but only after a cadence to finish the current musical thought'.

Another major success was the musical response itself; the instrumentation and musical style for each type of vocal input were appropriate and, simply stated, sounded good. This also was a direct result of the musical mappings and Sharle [36]. The experimental approach to developing the musical mappings and determining the best coordination with the vocal parameters was time well spent.

4.2.2 Weaknesses of the Singing Tree System

While a marketing report would never publicize the following, the author certainly acknowledges the many shortcomings and weaknesses of the Singing Tree which include, but are not limited to, the following.

- The musical goal was not identifiable for all participants without prior instruction.
- The Singing Tree was not programmed to reward multiple pitches.
- The musical mappings did not specifically target random inputs such as yelling, talking, or making noises.
- The vowel formant estimation was not precise.
- The sleeping state of the Singing Tree was too quiet to be heard when large numbers of people were in the Mind Forest.

Most of these weaknesses are self-explanatory. A few participants who used the Singing Tree without instruction did not grasp the musical goal. The word 'singing' implies that one should sing multiple pitches, and many people would first try to sing a song into the tree. While most people would proceed to discover that it was a single pitch which would provide the best response, a few people could not. In this, the tree failed to be a completely intuitive interface; some instruction was necessary. Furthermore, many samples retrieved from the 12 Speaking Tree were of people singing or humming one note. People were mistaking the Speaking Trees, in which Marvin Minsky asks a question and participants respond, for Singing Trees. Another problem stemmed from the fact that the basic pitch was determined to be the first pitch sung. A participant would have to stop singing, allow a short time (approximately 2 seconds which accounted for breath) to elapse, and allow the Singing Tree to reset before being able to establish a new pitch. A more 'intelligent' method for determining when the participant intended to sing a new pitch would have been desirable. This method would be intimately related to the concept of rewarding multiple pitches.

The vowel formant recognition works quite well in Matlab simulations, but is rather imprecise in application. There are many reasons for this, but the most significant are the variability in participants and the fact that they are singing, not speaking. The Singing Tree catered to men, women, and children of all ages. The formant frequencies for specific vowels vary considerably for people of different sex and age. Furthermore, the formant structures of singers tend to be differ from those of people who are speaking. The Singing Tree could only reliably detect the corners of the formant triangle, and, as a result, relied heavily on the changes in formant frequencies, rather than a specific determination of vowel, in the mapping algorithms.

4.3 Future Directions for Applications and Research

This sections outlines possibilities for future developments and applications of the Singing Tree to interactive karaoke. Future research directions are also discussed. The research has potential to improve both the Singing Tree and interactive musical mappings and interpretations in general.

4.3.1 Interactive Karaoke

While developing the Singing Tree, many additional mappings were considered. The most compelling extension was to allow the participant to sing multiple pitches. The idea was to maintain the concept of basic pitch, but then allow the participant to sing notes of different intervals relative to the basic pitch. The Singing Tree did have a degenerate case of this more general situation, in that the participants could sing octave intervals from the basic pitch. However, in expanding one's notion of an interactive singing experience, one of the first concepts to be explored was singing multiple pitches. Briefly stated, the concept was to weight each interval, much in the way Sharle weights each interval for playback. Considering which pitch the person was singing, its interval relationship to the basic pitch, and the associated weight as a function of the playback scale/key, a new dimension of playback control was invented on the white-board of the author's office. Although not yet realized, the concept has recently surfaced again in discussions with the Brother Corporation regarding interactive karaoke.

Brother is the largest maker of karaoke systems in Japan, a country in which karaoke plays an important cultural role, in addition to being a source of entertainment. The big karaoke boom of the late 1980's and early 1990's continued even after Japan went into economic recession. In an attempt to explore new karaoke markets, Brother has proposed the idea of using an interactive component analogous to the Singing Tree in conjunction with their existing system. The objectives are three fold: first, improve the experience and enjoyment of present users; second, improve the experience to attract new people who do not currently use karaoke; improve the experience for listeners.

The issues are straightforward. One should first develop a weighting system for the intervals of the various chord structures found in the music used in karaoke systems. The next issue is to make this weighting system dynamic, which implies knowing the music score. Brother's system uses MIDI sequencers and a synthesizer to reproduce the karaoke music, which is a big advantage. This means that all of the music, including the chord structure and melody, are in a MIDI format. It is not difficult to imagine the interval weighting following the score. The challenging issue is the type of interactivity desired.

Simply stated, there are two types of interactivity of interest in karaoke systems: non-generational and generational. Non-generational interactivity includes modifying the singer's voice with dynamic vocal effects such as reverb and delay to maintain pitch consonance, pitch shifting the voice to maintain pitch consonance, and self-harmonization by pitch shifting the voice to create harmony parts. Other non-generational modifications include dynamic mixing, in which the instruments and lines in a piece are mixed up or down to match the style of singing; dynamic instrumentation, in which the instrumentation of a particular line of music is changed; and dynamic global amplification, in which the volume of the accompaniment and the singer's voice is regulated to maintain consistency. More difficult, and potentially more interesting, are the generational modifications to the music. These would work in a manner similar to the Singing Tree operation. A score of music is being played as written. As the voice parameters change, interpretation and mappings work to embellish an existing part through perturbation and ornamentation rather than re-composition.

If an interactive karaoke system were developed which could intelligently map vocal parameters to musically meaningful changes in the karaoke experience, the result could revolutionize the karaoke industry. It would also be an incremental step toward the concept of interactive social events, such as interactive clubs.

4.3.2 Fuzzy Systems, Linguistic Variables, and Interactive Systems

There were two major issues in designing the Singing Tree which made the author consider Fuzzy Systems. First was the interpretation and mapping of complex concepts. The second was the use of instrument sets and nested layers of probabilities to make instrumentation decisions. The following is a brief discussion of the reasoning behind the author's assertion that a fuzzy systems approach to interactive systems is sensible. The author refuses to become muddled in the controversy over whether fuzzy set theory is a new branch of mathematics or simply another perspective. Honestly, he does not have the mathematical background to be making claims one way or the other. However, he does see potential for application, and he believes that more attention should be given to the subject of interactivity and fuzzy systems. For more information regarding fuzzy systems, the reader is referred to [31], [32].

The Singing Tree is far easier to describe in words than it is in mathematical terms. For example, it is rather clear to understand the Singing Tree's behavior simply by reading the design criteria given in Chapter 2. However, it is not entirely clear at all how it will behave by reading the mapping algorithms given in Chapter 3. Nonetheless, interactive systems which utilize computers must somehow translate a linguistic expression into a mathematical concept. This is, in essence, the function of interpretation and mapping algorithms. Given the linguistic description of a system, take available input parameters and map them in such a way that the output matches this description. In reviewing the work done by [36] in implementing the music generation algorithm, Sharle, and the work done on the interpretation and mapping algorithms in the Singing Tree and elsewhere, the author came to the conclusion that many of the solutions attempted in the Singing Tree resemble, in principle, a fuzzy systems approach. The two primary examples are the use of instrument sets and probabilistic selection, and the use of mathematical functions to approximate a linguistic expression. While neither of these in their own right are examples of a fuzzy implementation, they both have compelling similarities to the fundamentals of a fuzzy system.

The instrumentation of the Singing Tree was accomplished by defining sets of instruments with similar timbre qualities. Deciding which instruments would go into which sets was made based on the description of how the Singing Tree was supposed to operate. For example, the airy sounds are made by instruments that the author thought had a high degree of 'angelic' quality. The harsh instruments were those which had a high degree of 'agitation.' The sets from which the instruments were chosen were largely determined by the pitch instability parameter. However, in the case of dynamic instrument assignment, there was a 75% chance of using one algorithm, and a 25% chance of using another. The idea led the author to explore other implementations using probability and set theory in decision making, which led to a book on fuzzy set theory [32], and finally a book on fuzzy sets [31]. The most interesting result of this research is that, given the option to build the Singing Tree again from scratch, the author would be inclined to try using concepts such as fuzzy sets. Briefly described, a system described by fuzzy sets differs from that described using classical sets in that all its elements (the universal set) have a degree of membership in all other fuzzy sets. For example, from a classical set perspective, the author would determine the airy set of instruments exactly in the manner that was used for the Singing Tree. He would listen to all the instrument timbres available to him on the K2500R, decide if they are angelic or not, and if they were, include them in the set. In this approach, the sets are broadly defined concepts, such as 'instruments with angelic sound', and each instrument is given a crisp label: 'angelic' or 'not angelic'. Of course, one might try to derive some statistical justification for his choice (i.e., based on a survey of 560 men, women and children, 53% of those surveyed, on average, say instrument A is angelic, and this estimate has a variance, skewness, and kurtosis that can be calculated). But, in the end, the sound is either classified as angelic, or it is not. The fuzzy approach, on the other hand, is to define the meaning of the sets in a crisp manner, and assign a degree of membership to all the instruments. For example, a set is defined as the 'Set of Angelic Instruments'. Now, the author flips through all the instruments on the K2500R, and, based on his long experience and 'expert' knowledge of all types of instruments, he assigns a degree of membership to each of the instruments. This number is in the range [0,1], and it indicates an instrument's degree of 'angelic characteristic'. When it is time to playback instruments which are angelic, the crisp set has a fixed number of equally weighted possibilities from which to choose. The fuzzy set, on the other hand, has all the instruments on the K2500R available for playback, each weighted by its membership function. What are the options for playback? In the fuzzy case, selecting an instrument using a random number generator or probabilistically will give a consistently angelic playback without a deterministic response. The crisp set, on the other hand, will always have the same instruments. Admittedly, this is a very simple case in which one could easily imagine assigning probabilities to the instruments in the crisp set and then using a random number generator to select an playback instrument. In effect, this is the approach utilized in the Singing Tree and Sharle. Nonetheless, the fuzzy set description is quite elegant at taking the knowledge of an 'expert' about an abstract concept and creating a rule which governs all elements with which one is working. This concept generalizes to the definition of linguistic variables, the fuzzy IF-THEN statement, and the fuzzy rule-base. The theory of fuzzy sets defines how these sets, IF-THEN statements, and rule base interact mathematically. In addition, the concept of defining a mathematical function to represent the condition of abstract ideas is the premise of the Takagi-Sugeno-Kong Fuzzy System. The Singing Tree made such mathematical representations on an independent basis. Fuzzy set theory provides a mathematical framework for their interaction. These are all areas in which the author would like to see more work done, especially in interactive applications.

4.3.3 Higher-Order Spectra Analysis

In following the work of his colleague Beth Gerstein [66], the author was introduced to the use of higher order cumulants in the analysis of heart-rate time series. In short, these higher-order cumulants can remove linearities such as additive white Gaussian noise, reveal non-linearities, extract low amplitude periodicities from the time series, and ideally be used as a diagnostic tool. The issue that the author did not have time to sufficiently address is the concept of musical intention versus musical gesture, and the possibility of using higher order spectra analysis tools to detect or differentiate them.

Much work has been done in the field of gesture recognition [63], [67]. In general, the idea is to recognize a pattern of movement or a pattern in a picture and reference it to a library of primitives. The point the author raises, however, is that there is a distinct difference between gesture and intention. Granted, the primitives are established based on some notion of universality, but the extension to musical gesture may be difficult. Given several people with an identical musical intention, the gesture each uses to represent this intention may be remarkably different. The author's basis for such a statement is his observations of people using the many musical interfaces of the Mind Forest. For example, the gestures used to make music at the Gesture Wall, an interface in which one's hand and arm movements change a musical stream, were as unique as the participants who made them. As a proposal for future work, the author would suggest looking for intentional cues in the time series of some indicative metric. Even having the knowledge that one's intention is to maintain or change a gesture could be useful. While the higher-order spectra analysis techniques are quite computationally intensive, the author thinks such investigation would be very interesting in its own right.

4.3.4 Beyond the Singing Tree

There are numerous applications for a Singing-Tree-type technology. Perhaps the most promising is in the field of 'smart' applications. Making a 'smart' automobile interior or a 'smart' room which responds to voice is currently a hot research topic. Admittedly, many such applications would benefit from full-blown speech recognition. However, there are applications which could use information contained in the voice without resorting to these relatively expensive algorithms. For example, a person could be humming a piece of music in her car or at home, and a computer could identify the piece, or find a station which is currently playing it. Or, a system which monitors events occurring within the room or automobile could be in a stand-by mode, in which a Singing-Tree-type technology is listening for specific cues and turns on the appropriate, more powerful recognition system as a situation warrants it. This would be particularly useful in band-limited, shared systems in which the use of more powerful recognition systems is necessary, but not everyone can use them simultaneously. The educational applications of the Singing Tree are also compelling. The Singing Tree is an excellent educational tool which allows children to experience computer music without having years of experience. The benefits are a greater appreciation for music and, hopefully, a stimulated interest in music.

Chapter 5

Conclusions

5.1 Synopsis of the Singing Tree

The Singing Tree is a novel, interactive musical interface which responds to vocal input with real-time aural and visual feedback. A participant interacts with the Singing Tree by singing into a microphone. Her voice is analyzed for pitch frequency, noisiness, energy, brightness, and the first three formant frequencies. Based on these metrics, musically meaningful mapping algorithms are developed to control both a music generation algorithm named Sharle and a video stream in real-time. The aural and visual feedback are designed to lead the participant to an established goal. In the current version of the Singing Tree, this goal is to sing one note as purely and steadily as possible. Maintaining such a pitch is rewarded with an angelic, meditative response of bassy strings and arpeggiating woodwinds. Deviations from the goal result in a harsher, more agitated response of brass and percussion. The result is a reward-oriented relationship between the sounds one makes and the video and music one experiences.

The Singing Tree has been presented in conjunction with the Brain Opera at the Lincoln Center Festival in New York City, U.S.A.; the Ars Elektronica Festival in Linz, Austria; The Electronic Cafe International (with sponsorship from TeleDanmark) in Copenhagen, Denmark; the Yebisu Garden Center (with sponsorship from NTT Data) in Tokyo, Japan; and at the Kravitz Center (with sponsorship from the Kravitz Center) in West Palm Beach, U.S.A. In addition, it is scheduled to be presented at the Japan Applied Mathematics Society's 1997 International Student Conference in Colorado, and the Design of Interactive Systems (DIS) Conference in Amsterdam, The Netherlands. It will also be presented at future Brain Opera venues.

Based on observations and user feedback, the Singing Tree has been a particularly successful interactive interface. It is an interesting musical interaction experience for both professional and amateur singers, and people of all ages can enjoy the musicality of the Singing Tree. The Singing Tree successfully identified musically meaningful vocal parameters from the singing voice. Furthermore, a methodology for mapping these parameters was developed which can be extended to other interactive music systems. These algorithms provided a musical experience which was intuitive, responsive, and engaging. The music was appropriate and consistent with the participant's quality of singing, without being deterministic. Although the music was based on randomly generated MIDI information, it contained no discontinuous behavior. The interface was seamless, and the computer was successfully 'removed' from the participant's experience. This work introduced the concept of continuous interactivity, in which the Singing Tree interacts continuously with its environment, regardless of the presence or absence of the user. In this, the Singing Tree and the participant often exchanged roles; the Singing Tree elicited a desired response from the participant. This creates a new perspective and direction for the 'inter-active' instrument and 'inter-active' music. Succinctly stated, the Singing Tree achieved its fundamental design specification: to provide users with a new and interesting means by which to make and discover a musical experience.

In the future, the Singing Tree technology will find applications in interactive karaoke systems, educational environment, and 'smart' rooms and automobile interiors. Further research into the use of fuzzy set theory in interactive systems will likely prove fruitful, the benefits of which would include the development of a 'language' which could more adequately translate linguistic concepts into mathematical representations. In addition, it is the hope of the author that more designers of interactive systems take the 'continuous interactivity' approach; designing a system which interacts with its environment, rather than simply an interactive system which is initiated by and responds to a human user, will lead to more meaningful, realistic, and truly 'inter-active' experiences.

Appendix A

The Novel Interactive Instruments of the Brain Opera

The Singing Tree is but one of eight interactive interfaces used in the Mind Forest and the Performance. The Mind Forest Interfaces are: the Singing Tree, the Speaking Tree, the Rhythm Tree, Harmonic Driving, the Melody Easel, and the Gesture Wall. Through these instruments, participants can create music while exploring the themes of the Brain Opera music they will hear in the Brain Opera Performance. The instruments used in the Performance include a combination Gesture Wall/Rhythm Tree, a Digital Baton, and the Sensor Chair.

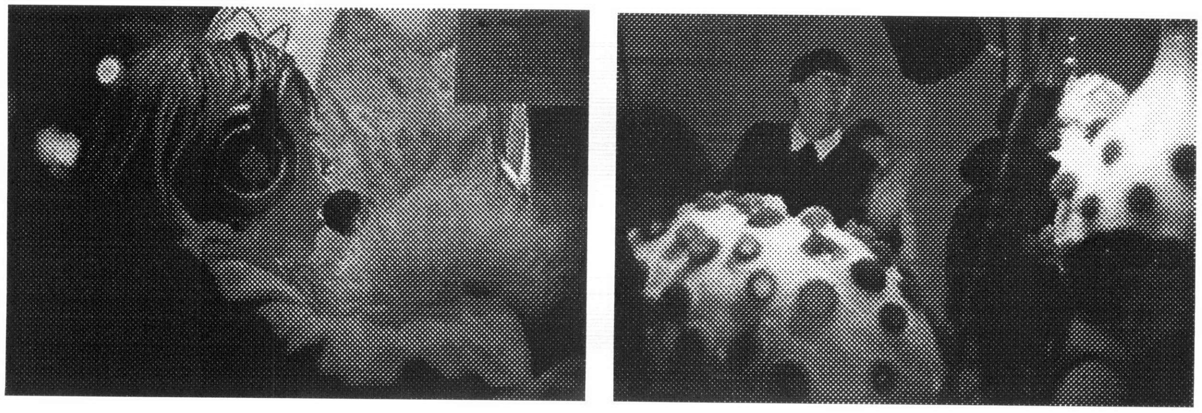

Figure A-1: The Singing Tree (left) and the Rhythm Tree (right) in Tokyo

Figure A-2: Harmonic Driving (left) and the Melody Easel (right)

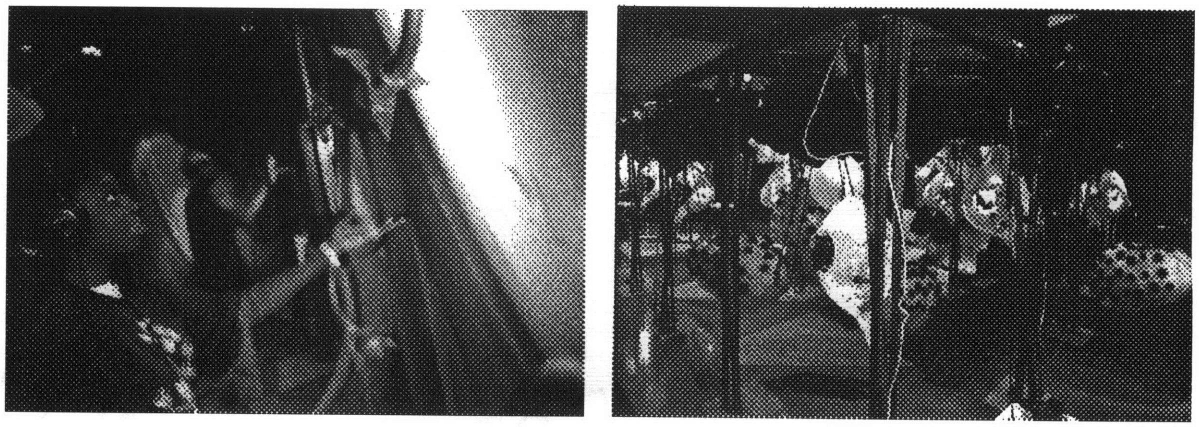

Figure A-3: The Gesture Wall (left) and the Mind Forest (right)

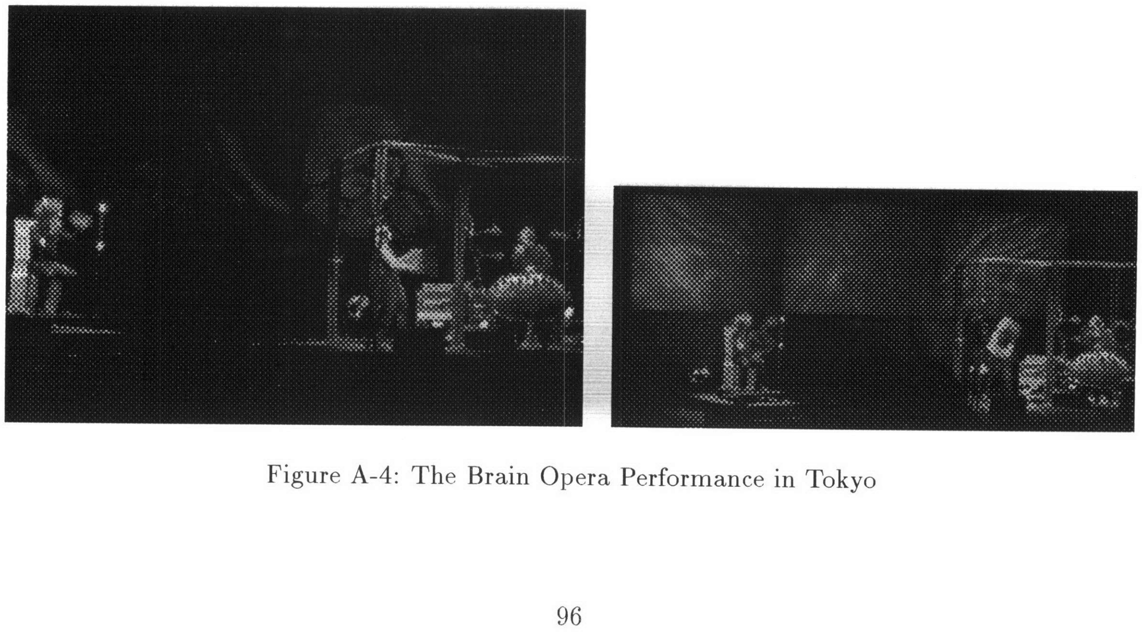

Figure A-4: The Brain Opera Performance in Tokyo

Appendix B

Bayesian Approach to Deriving the Pitch Period Estimator

The cross-correlation of two windows of a periodic signal is maximized when the windows are separated by a period or multiple of the period. While this is an intuitive method for estimating the pitch period of a signal, generalizing the concept to quasi-periodic cases may require justification. Below is one such justification from [30] using a Beysian approach [46].

Starting with a signal $s(t)$ that is assumed to be periodic, or more precisely, quasi-periodic, it follows that $s(t) \approx as(t+T)$ where T is the quasi-period of the signal and the scalar a accounts for inevitable amplitude changes. Considering a window of d samples of $s(t)$ taken with a sampling period, τ, one can define the vector, $\mathbf{v}(t) = [s(t), s(t+\tau), ..., s(t+(d-1)\tau)]^T$. Two such windows, $\mathbf{v}(t)$ and $\mathbf{v}(t+T')$, are separated in time by T'. Using a Bayesian approach, all parameters are considered to be random variables, which can be described by probability density functions [46]. Conditioning on the hypothesis, H_T, that T is the quasi-period, one expects the conditional probability that T' is the best estimate for the quasi-period will be a maximum for the case $T' = T$. Furthermore, the conditional probability should decrease as T' leaves T from the left and the right. Therefore, temporarily side-stepping the issue of multiple periodicity and dropping the prime from T', it is reasonable to postulate the d-dimensional conditional probability density function

$$p_{\mathbf{v}|H_T}(\mathbf{v}(t)|H_T) = \frac{1}{(2\pi\sigma^2)^{d/2}} exp\left(-\frac{\| \mathbf{v}(t) - a(t,T)\mathbf{v}(t+T) \|^2}{2\sigma^2}\right) \tag{B.1}$$

where σ is a 'tolerance' and $a(t,T)$ minimizes the expression in the exponent.

$$\frac{\partial}{\partial a(t,T)}\left(\frac{\| \mathbf{v}(t) - a(t,T)\mathbf{v}(t+T) \|^2}{2\sigma^2}\right) = 0 \qquad \Longrightarrow \qquad a(t,T) = \frac{\mathbf{v}(t)^T\mathbf{v}(t+T)}{\| \mathbf{v}(t+T) \|^2} \tag{B.2}$$

The best estimate for the pseudo-period is simply the choice of hypothesis which will maximize the *a posteriori* probability density function, $p_{H_T|\mathbf{v}}(H_T|\mathbf{v}(t))$ [30]. Using Bayes' rule [46], the goal is to find the T which will maximize

$$p_{H_T|\mathbf{v}}(H_T|\mathbf{v}(t)) = \left(\frac{p_{\mathbf{v}|H_T}(\mathbf{v}(t)|H_T)\, p_{H_T}(H_T)}{p_{\mathbf{v}}(\mathbf{v}(t))}\right) \tag{B.3}$$

where $p_{H_T}(H_T)$ is the *a priori* probability density function that T is the period of the signal, and $p_{\mathbf{v}(t)}(\mathbf{v}(t))$ is the probability density function for $\mathbf{v}(t)$. Assuming an equal *a priori* probability that the signal has a pitch T across all possible periods and recognizing that $p_{\mathbf{v}(t)}(\mathbf{v}(t))$ is independent of T, finding the T which maximizes equation (B.3) is equivalent to finding the T which maximizes $p_{\mathbf{v}(t)|H_T}(\mathbf{v}(t)|H_T)$. This is equivalent to finding the T which minimizes

$$\gamma = \| \mathbf{v}(t) - a(t,T)\mathbf{v}(t+T) \|^2 \tag{B.4}$$

where γ is simply defined as the T-dependent part of the argument of the exponential in equation (B.1). Substituting the expression for $a(t,T)$ from equation (B.2) into equation (B.4) yields,

$$\begin{aligned}\gamma &= \| \mathbf{v}(t) \|^2 - 2\frac{\mathbf{v}(t)^T\mathbf{v}(t+T)}{\|\mathbf{v}(t+T)\|^2}\mathbf{v}(t)^T\mathbf{v}(t+T) + \left(\frac{\mathbf{v}(t)^T\mathbf{v}(t+T)}{\|\mathbf{v}(t+T)\|^2}\right)^2 \| \mathbf{v}(t+T) \|^2 \\ &= \| \mathbf{v}(t) \|^2 \left(1 - \left(\frac{\mathbf{v}(t)^T\mathbf{v}(t+T)}{\|\mathbf{v}(t)\| \, \|\mathbf{v}(t+T)\|}\right)^2\right) \\ &\equiv \| \mathbf{v}(t) \|^2 \left(1 - \lambda(t,T)\right).\end{aligned} \tag{B.5}$$

Since $\| \mathbf{v}(t) \|^2$ is not a function of T, minimizing γ is equivalent to maximizing λ. Thus, the best estimate for the quasi-period is now the T which maximizes the expression

$$\lambda(t,T) = \frac{\mathbf{v}(t)^T\mathbf{v}(t+T)}{\| \mathbf{v}(t) \| \, \| \mathbf{v}(t+T) \|} \tag{B.6}$$

which is a 'normalized' cross correlation between $\mathbf{v}(t)$ and $\mathbf{v}(t+T)$.

Appendix C

Additional Examples of Formant Frequency Estimation

The following are examples taken directly from the Singing Tree. The author sang the vowels 'aah', 'ee', and 'ooh' into the Singing Tree and saved the sampled voice data.

The first example shown is 'aah' with a pitch frequency of 139.5 Hz. The signal and cepstrum (no glottal or radiational (g-r) component) are shown in the upper half of Figure C-1. The cepstrally smoothed log spectrum is the plot on the bottom-left. The peak in region one is f1=883 Hz, the frequencies at 1800 Hz are not distinguishable, and the large peak at 3100 Hz is out of range for the third formant frequency. The chirp z-transform (CZT) log spectrum is the plot on the bottom-right in FigureC-1. It resolves the bump at 1800 Hz into two formant frequencies: f2=1691 Hz and f3=2205 Hz. This corresponds to a borderline case between vowel sounds /ʌ/ and /æ/. The third formant frequency is closer to /ʌ/, and the vowel sound is therefore estimated to be /ʌ/.

Table C.1: Estimated Formant Frequencies for the 'aah' Signal at f=139.5 Hz

Formant Frequency	Frequency (Hz)	Vowel
1	936	/ʌ/
2	1686	
3	2709	

The second example is an 'ee' at a pitch frequency of 296 Hz. The signal and cepstrum (no glottal or radiational (g-r) component) are shown in Figure C-2. The cepstrally smoothed log spectrum appears sufficient in this case to determine all three formant frequencies. As shown in Figure C.2, f1=341 Hz, f2=2045, and f3=2704. If this is the case, the vowel sound corresponding to these formant frequencies is correct, an /i/ or 'ee' sound. However, the algorithm would notice that the estimate for the second formant is not 8 dB below the first formant. Thus, it would run the CZT analysis and find that the second formant is actually f2=1379 Hz and that leaves the third formant at f3=2704 Hz. Thus, the vowel sound estimated is in between an /i/ and an /u/, indicating that my sung /i/ is very dark (in the back of the mouth). Using the third formant to resolve the vowel, f3 is closer to the average /i/ value than the average /u/ value, and thus /i/ is the estimated vowel sound.

The remaining examples of another 'ee' at a pitch frequency of 149 Hz, an 'ooh' at f=290 Hz, and another 'ooh' at f=145 Hz are presented in as similar fashion without discussion.

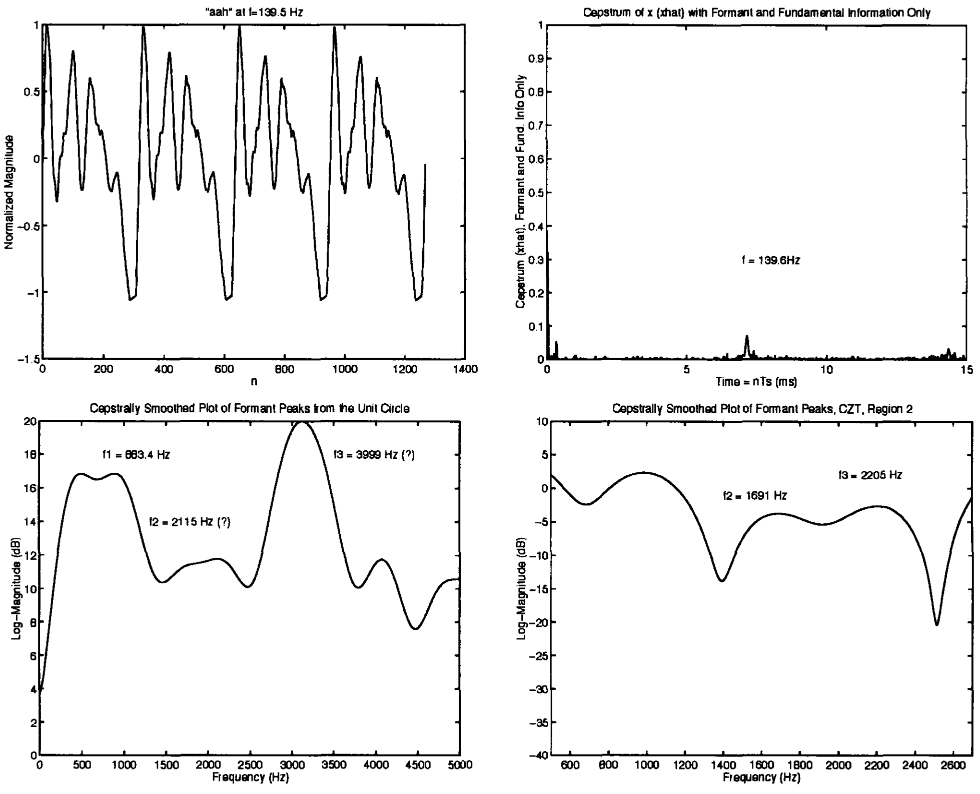

Figure C-1: 'aah' Signal at f=139.5 Hz and Relevant Formant Estimation Plots

Table C.2: Estimated Formant Frequencies for the 'ee' Signal at f=296 Hz

Formant Frequency	Frequency (Hz)	Vowel
1	341	/i/
2	1379	
3	2704	

Table C.3: Estimated Formant Frequencies for the 'ee' Signal at f=149 Hz

Formant Frequency	Frequency (Hz)	Vowel
1	347	/i/
2	969	
3	3157	

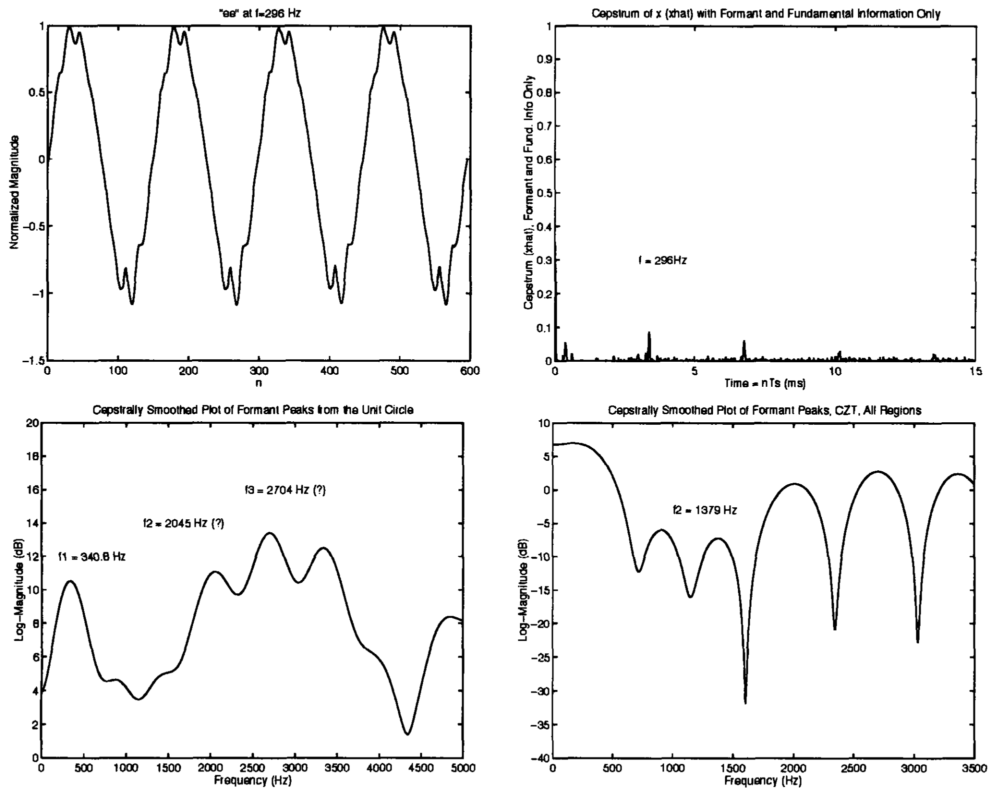

Figure C-2: 'ee' Signal at f=296 Hz and Relevant Formant Estimation Plots

Table C.4: Estimated Formant Frequencies for the 'ooh' Signal at f=290 Hz

Formant Frequency	Frequency (Hz)	Vowel
1	942	/i/
2	2165	
3	2706	

Table C.5: Estimated Formant Frequencies for the 'ooh' Signal at f=145 Hz

Formant Frequency	Frequency (Hz)	Vowel
1	508	/i/
2	1570	
3	2537	

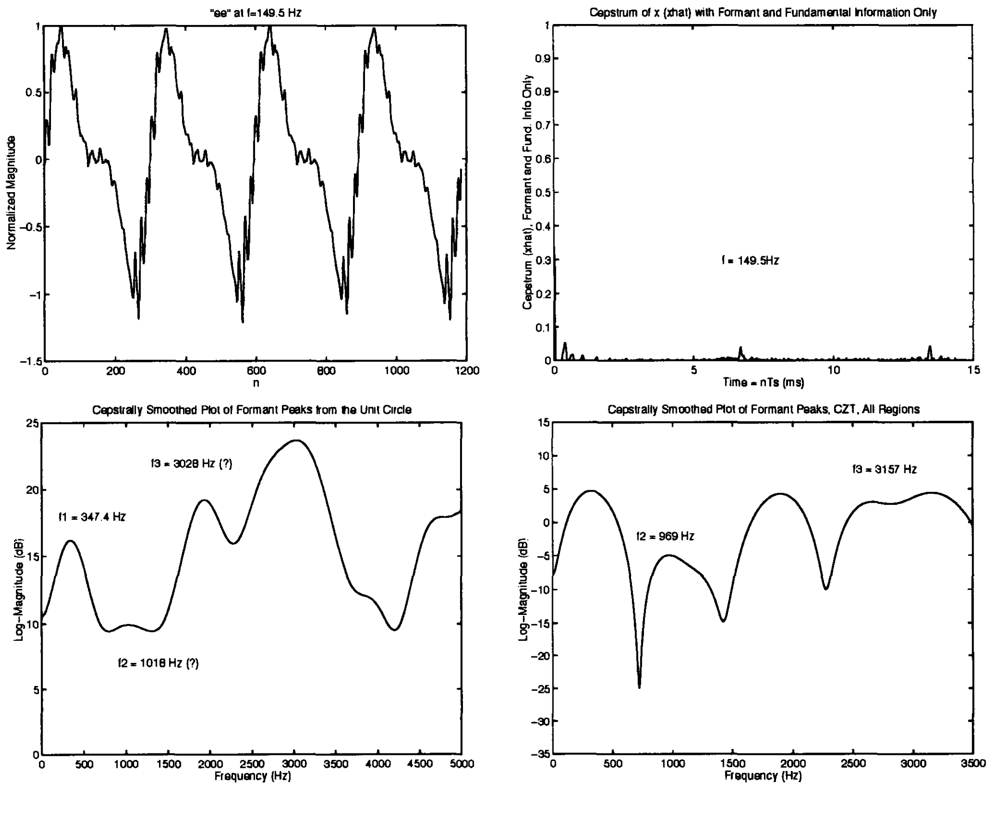

Figure C-3: 'ee' Signal at f=149 Hz and Relevant Formant Estimation Plots

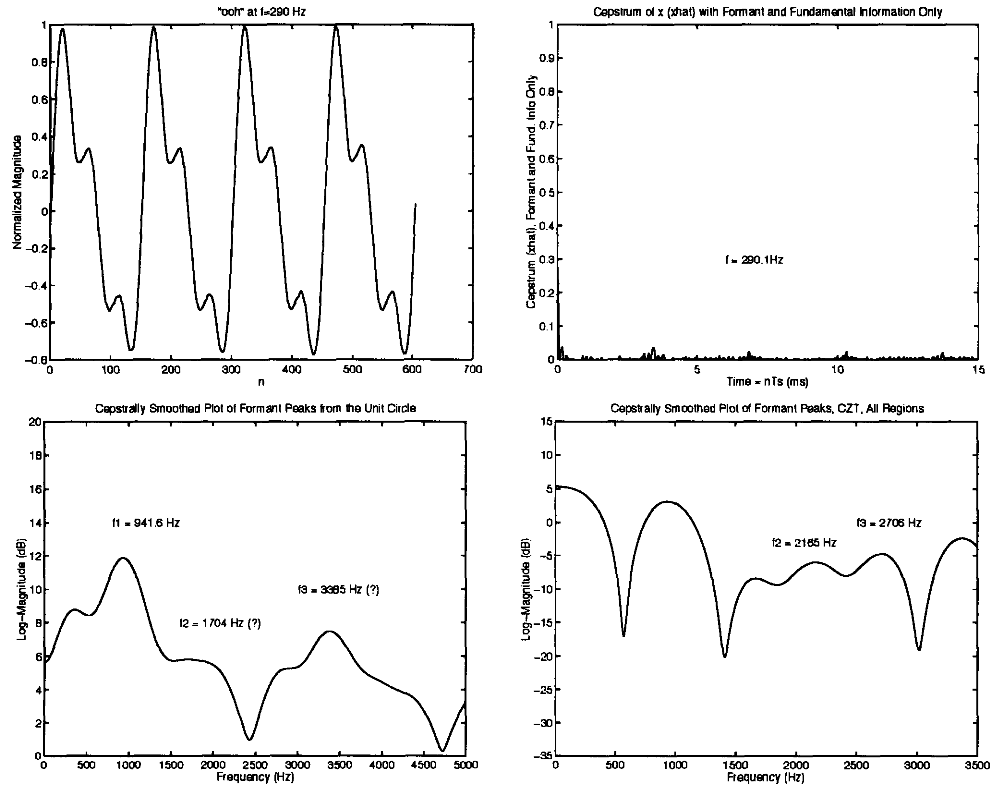

Figure C-4: 'ooh' Signal at f=290 Hz and Relevant Formant Estimation Plots

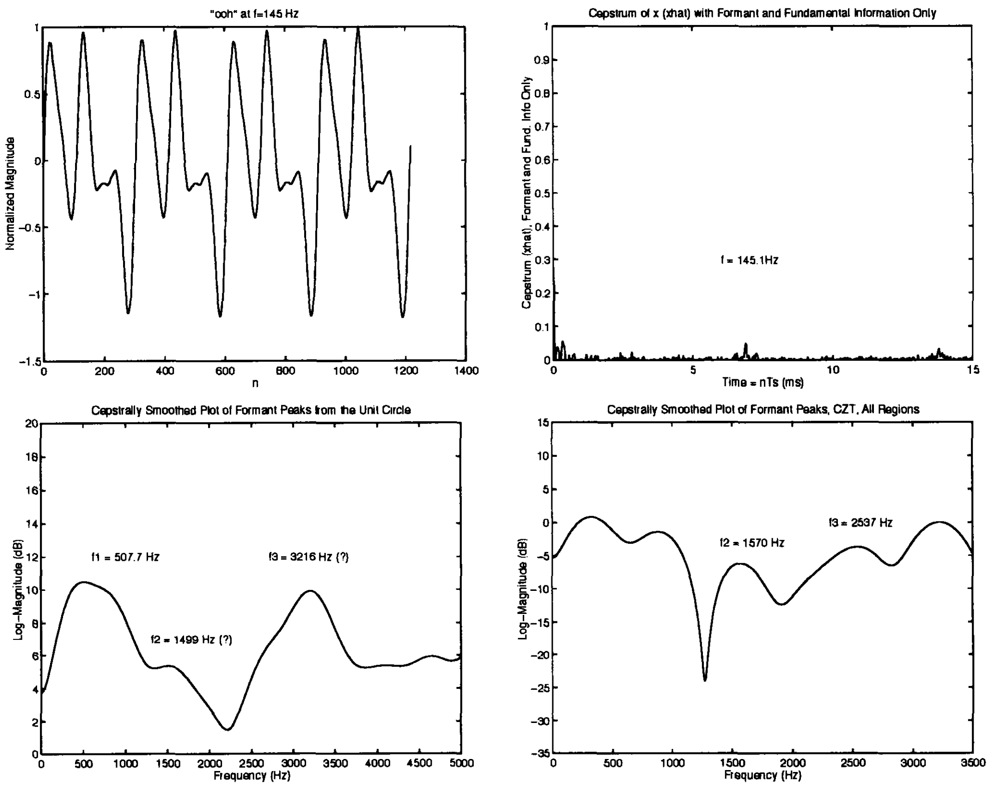

Figure C-5: 'ooh' Signal at f=145 Hz and Relevant Formant Estimation Plots

Bibliography

[1] D. Waxman, *Digital Theremins: Interactive Musical Experiences for Amateurs Using Electric Field Sensing*, M.S. Thesis for MIT Media Laboratory, 1995.

[2] R. Sessions, *The Musical Experience of Composer, Performer, Listener*, Princeton University Press, Princeton, NJ, 1950.

[3] T. Machover et al., *Vox-Cubed*, performance at the MIT Media Laboratory, 1994.

[4] T. Machover, *Hyperinstruments: a Progress Report*, Internal Document, MIT Media Laboratory, 1992.

[5] A. Rigopolous, *Growing Music from Seeds: Parametric Generation and Control of Seed-Based Music for Interactive Composition and Performance*, M.S. Thesis for MIT Media Laboratory, 1994.

[6] F. Matsumoto, *Using Simple Controls to Manipulate Complex Objects: Application to the Drum-Boy Interactive Percussion System*, M.S. Thesis for MIT Media Laboratory, 1993.

[7] J. Paradiso and N. Gershenfeld, "Musical Applications of Electric Field Sensing," *Submitted to Computer Music Journal*, 1995.

[8] M. Minsky, *Society of Mind*, Simon and Schuster, New York, 1988.

[9] T. Machover, *Brain Opera Update, January 1996*, Internal Document, MIT Media Laboratory, 1996.

[10] W. Oliver, J. C. Yu, E. Metois, "The Singing Tree: Design of an Interactive Musical Interface" *Design of Interactive Systems*, DIS Conference, Amsterdam, 1997.

[11] J. Sundberg, "Formant Technique in a Professional Female Singer," *Acoustica,* Vol.32, pp. 89-96, 1975.

[12] J. Sundberg, "A Perceptual Function of the Singing Formant," *Quarterly Progress and Status Report (of the Royal Institute of Technology,* Speech Transmission Laboratory in Stockholm, pp. 61-63, Oct 1972.

[13] J. Sundberg, "Articulatory Interpretation of the Singing Formant" *Journal of the Acoustical Society of America,* Vol. 55, pp. 838-844, 1974.

[14] K. Stevens, H. Smith, R. Tomlinson, "On an Unusual Mode of Chanting by Tibetan Lamas," *Journal of the Acoustical Society of America,* Vol. 41, pp. 1262-1264, 1967.

[15] Rodet, X., Yves Potard, Jean-Baptiste Barriere, "The CHANT Project: From the Synthesis of the Singing Voice to Synthesis in General," *Computer Music Journal,* Vol. 8(3), pp. 15-31, 1984.

[16] Dr. Daniel Huang, "Complete Speech and Voice Assessment, Therapy, and Training," *Tiger Electronics Inc.,* 1997.

[17] P. Rice, Conversation with Pete Rice re. differences in the research of computer music and interactive instruments.

[18] *A Page for John Cage*, http://www.emf.net/ mal/cage.

[19] C. Roads, *The Music Machine, Selected Readings from Computer Music Journal*, The MIT Press, Cambridge, MA, 1989.

[20] M. Mathews and J. Pierce, *Current Directions in Computer Music Research*, The MIT Press, Cambridge, MA, 1989.

[21] K. Stockhausen, *Kontakte*, (On Compact Disk), Universal Edition, Wien, 1959-1960.

[22] K. Stockhausen, *Mantra*, (On Compact Disk), Stockhausen-Verlag, Kurten, 1970.

[23] M. Subotnick, "Interview with Morton Subotnick", *Contemporary Music Review*, Vol. 13, Part 2, pp. 3-11, Amsterdam, 1996.

[24] A. Gerzso, "Reflections on Repons", *Contemporary Music Review*, Vol. 1, Part 1, pp. 23-34, Amsterdam, 1984.

[25] R. Rowe, *Interactive Music Systems, Machine Listening and Composing*, The MIT Press, Cambridge, 1993.

[26] G. Lewis, C. Roads ed., *Composers and the Computer*, William Kaufmann, Inc., Los Altos, pp. 75-88, 1985.

[27] M. Mathews, F. Moore, "GROOVE-A Program to Compose, Store, and Edit Functions of Time", *Communications of the ACM*, Vol. 13, No. 12, 1970.

[28] H. Schenker, F. Salzer, *Five Graphical Music Analyses*, Dover Publications Inc., New York, 1969.

[29] R. Feynman, R. Leighton, M. Sands, *The Feynman Lectures, Vol.3*, Addison-Wesley Publishing Company, Reading, MA, 1965.

[30] E. Metois, "Musical Sound Information: Musical Gesture and Embedding synthesis (Psymbesis," *http://www.brainop.mit.edu*, Ph.D. Thesis for MIT Media Laboratory, October 1996.

[31] L. Wang, *A Course in Fuzzy Systems and Control*, Prentice Hall PTR, Upper Saddle River, NJ, 1997.

[32] L. Zadeh, R. Yager, ed., *Fuzzy Sets and Applications*, John Wiley and Sons, New York, NY, 1987.

[33] T. Machover, from conversation re. the Singing Tree, 1996.

[34] S. Daniel, Visual Director of the Brain Opera.

[35] J. C. Yu, DSP tookit ported from C to C++ by John Yu.

[36] C. Yu, "Computer Generated Music Composition," *http://www.brainop.mit.ed*, M.S. Thesis for MIT Electrical Engineering and Computer Science Department, May 1996.

[37] E. Hammond, T. Machover, Boston Camerata, Music for samples written by Tod Machover, recorded by Ed Hammond, and performed by the Boston Camerata.

[38] M. Orth, Production Manager and Design Coordinator of the Brain Opera

[39] R. Kinoshita, Architect and Space Designer.

[40] The floor mat was developed by Joe Paradiso, Patrick Pelltier, and William Oliver.

[41] Kurzweil, *Kurzweil K2500 Series Performance Guide*, Young Chang Co, 1996.

[42] A. Benade, *Fundamentals of Musical Acoustics, 2nd ed.*, Dover Publications, New York, NY, 1990.

[43] The samples were prepared by William Oliver, Jason Freeman, and Maribeth Back.

[44] S. De Furia, J. Scacciaferro, *MIDI Programmer's Handbook*, M and T Publishing Inc., Redwood City, CA, 1990.

[45] Conversation with Ben Denckla re. advantages of MIDI protocol.

[46] A. Drake, *Fundamentals of Applied Probability Theory*, McGraw-Hill Book Company, New York, NY, 1988.

[47] J. Goldstein, "An Optimum Processor for the Central Formation of the Pitch of Complex Tones" *Journal of the Acoustical Society of America*, Vol.54, pp.1496-1517, 1973.

[48] G. Strang, *Introduction to Applied Mathematics*, Wellesley-Cambridge Press, Wellesley, MA, 1986.

[49] A. Oppenheim, R. Schafer, *Discrete Time Signal Processing*, Prentice Hall, Englewood Cliffs, NJ, 1989.

[50] Conversation with Ben Denckla, MIT Media Laboratory.

[51] M. Mathews, J. Miller, E. David Jr., "Pitch Synchronous Analysis of Voiced Sounds" *Journal of the Acoustical Society of America*, Vol.33, pp.179-186, 1961.

[52] S. McAdams and A. Bregman, "Hearing Musical Streams," *Computer Music Journal*, Vol. 3, No. 4, 1979, pp. 26-43.

[53] R. Schafer and L. Rabiner, "System for Automatic Formant Analysis of Voiced Speech," *The Journal of the Acoustical Society of America*, Vol. 47, No. 2 (part 2), pp. 634-648, 1991.

[54] D. R. Reddy, "Computer Recognition of Connected Speech," *The Journal of the Acoustical Society of America*, Vol. 42, No. 2, pp. 329-347, 1967.

[55] J. P. Oliver, "Automatic Formant Tracking by a Newton-Raphson Technique," *The Journal of the Acoustical Society of America*, Vol. 50, No. 2 (part 2), pp. 661-670, 1971.

[56] H. Hanson, P. Maragos, and A. Potamianos, "A System for Finding Speech Formants and Modulations via Energy Separation," submitted to *IEEE Transactions on Speech Processing*, 1992.

[57] G. Rigoli, "A New Algorithm for Estimation of Formant Trajectories Directly from the Speech Signal Based on an Extended Kalman-Filter," *ICASSP 86*, 1986.

[58] G. E. Kopec, "Formant Tracking Using Hidden Markov Models and Vector Quantization," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, Vol. ASSP-34, No. 4, pp. 709-729, August 1986.

[59] G. Rigoli, "Formant Tracking with Quasilinearization," *ICASSP 88*, 1988

[60] The MathWorks, Inc., *Matlab, Signal Processing Toolbox, User's Guide*, MathWorks, Inc., 1996.

[61] Microsoft Music Producter, http://www.microsoft.edu

[62] C. Dodge, *The Abstraction, Transmission, and Reconstruction of Presence: A Proposed Model for Computer Based Interactive Art*, M.S. Thesis for MIT Media Laboratory, 1997.

[63] T. Starner, *Visual Recognition of American Sign Language Using Hidden Markov Models*, S.M. Thesis, MIT Media Laboratory, 1995.

[64] D. O'Shaughnessy, *Speech Communication, Human and Machine*, pp. 67-71, Addison-Wesley, NY, 1987.

[65] G. Bennett and X. Rodet, *Synthesis of the Singing Voice*, pp. 20-44.

[66] B. Gerstein, *Applying Higher Order Spectra Analysis to the Heart Rate Time Series*, S.M. Thesis, MIT Electrical Engineering and Computer Science Department, 1997.

[67] T. Marrin, *Toward an Understanding of Musical Gesture: Mapping Expressive Intention with the Digital Baton*, S.M. Thesis, MIT Media Laboratory, 1996.

[68] N. Gershenfeld, "An Experimentalist's Introduction to the Observation of Dynamical Systems," *Directions in Chaos, Vol II*, Hao Bai-lin ed., World Scientific, 1988.

[69] N. Gershenfeld, "Information in Dynamics," *Workshop on Physics and Computation PhysComp '92*, reprint, 1992.