DETECTION AND ANALYSIS OF MUSICAL EVENTS USING MODEL-BASED SIGNAL PROCESSING

Thesis by

Randall Lee Owen

In Partial Fulfillment of the Requirements

For the Degree of

Mechanical Engineer

California Institute of Technology

Pasadena, California

1999

(Submitted May 1999)

Acknowledgments

This work was made possible only through the encouragement and support of several important people. First, I would like to thank my advisor and friend, Professor Fred Culick, for never giving up on me even when it seemed unlikely that I would ever find the time to finish this work. I would also like to thank my committee members Professors P. P. Vaidyanathan and Richard M. Murray for their insightful comments and suggestions, without which this work would have reached less than its full potential.

I would also like to thank my father, Don Owen, for his encouragement, support and the "gentle reminders" regarding the completion of this work. Special thanks to Dr. Paul G. Bamberg, both for teaching me the wonders of basic physics at Harvard and for providing access to software and personnel at Dragon Systems, Inc. A special thanks also goes to Eddie Van Halen for his initial enthusiasm for the project, and for encouraging me to pursue this work when almost everyone else deemed it impossible. Others who provided suggestions or encouragement, both technical and otherwise, over the years include Jeff Shumway, Larry Staib, Roy Ashen, Steve Vai, Wolf Marshall, George Joblove and, last but certainly not least, Ken Johnson. Never tell me that something cannot be done, Ken!

Finally, I dedicate this work to my maternal grandfather, Merideth A. Deringer, who instilled in me an intellectual curiosity and thirst for knowledge that has never been quenched. Throughout his life, he never stopped reading, learning and inventing, and set the example by which I have studied and learned. Though he is now gone, he remains my most important influence and mentor.

ABSTRACT

DETECTION AND ANALYSIS OF
MUSICAL EVENTS USING
MODEL-BASED SIGNAL PROCESSING

Randall Lee Owen
California Institute of Technology
1999

The present work is directed to the detection and analysis of notes, chords and other musical events produced by a stringed musical instrument, specifically the guitar. The chords generated by a guitar are polyphonic, meaning that they comprise multiple notes sounded simultaneously. Each note is also spectrally complex, in that it comprises a fundamental tone and several harmonics. Despite this complexity, the statistics of the signal containing the notes and chords are expected to be similar to those of human speech. This similarity will allow the signal to be characterized as a parametric random process so that established mathematical and speech recognition techniques can be used to extract the events from the signal. The analysis of musical signals is an important application since it is a logical extension to the problem of speech recognition. Moreover, a robust computer-based solution to this problem could have both research and commercial applications.

A system for automated detection and analysis of musical events, such as notes and chords, has been designed. The system is comprised of two main elements: the event library and a set of match measures. The event library contains a hierarchy of event models each corresponding to a distinct musical note or chord. Each event model is structured as a hidden Markov model (HMM), $\lambda = (A, B, \pi)$, having the four distinct states labeled attack, sustain, decay or silence, that correspond to the specific physical

states of the musical event. Associated with each model state $Q = \{q_1, \ldots, q_4\}$ are a set of M observation symbols $V = \{v_1, v_2, \ldots, v_M\}$ and a set of three probability distributions: a transition probability distribution $A = \{a_{ij}\}$, an observable probability distribution $B = \{b_j(k)\}$ and an initial probability distribution $\pi = \{\pi_i\}$. Three match measures are developed for solving the recognition problem: one for estimating the HMM parameters, one for determining the optimal state sequence of the HMM and one for evaluating the probability that a given observation sequence was produced by a specific HMM. The observation sequence is derived from the input signal by sampling, converting to a spectral representation, and digitally coding using standard speech recognition techniques. The three match measures correspond, respectively, to training the model, refining the model and matching an event to a model, each of which is performed using conventional speech processing algorithms.

TABLE OF CONTENTS

LIST OF FIGURES AND TABLES

Figures

Tables

Chapter 1

Introduction

The detection and analysis of musical events, such as notes and chords, is part of the more general problem of extracting information from complex dynamic signals. The goal of musical event analysis is to identify a specific musical note or chord from an input signal. The musical events embedded in the input signal must first be segmented as a first step to more detailed analysis and identification. This work is concerned with the segmentation and identification of individual musical events from time-varying input signals.

Extracting a musical event from a time-varying input signal requires both signal data and additional information as to how the event is encoded into the signal. This additional information is based on a model of the musical event. Such models can range from simple templates (*e.g.*, spectral coefficients) to those derived from complex statistical processes (*e.g.*, discrete-time Markov processes).

Prior approaches to musical information processing have used only the signal data without reference to models or extrinsic information. These approaches are based on the extraction of low-level signal features that are then grouped into identifiable patterns. However, such grouping processes have generally only been successful if used in conjunction with special input transducers or other hardware.

Most musical events are combinations of fundamental tones that form the spectral components of the signal. Thus, the amplitude and frequency of the individual spectral components can provide basic musical cues. However, these properties are often

inconsistent and incomplete due to random signal noise or spectral overlap. All signals include some amount of noise that tends to mask those spectral components of low amplitude. In addition, the spectra of even a single note will include a fundamental tone and at least several harmonics or overtones. The fundamental tone may not have the greatest amplitude. Further, the spectral components of signals representing complex musical events such as intervals and chords are often harmonically related and thus overlap in the frequency domain.

Speech signals are similar to music signals in that their spectra are complex and comprise many time-varying spectral components. Automated recognition of speech by computer has been a research goal for more than four decades. A powerful approach from automated speech processing is to characterize the speech signal as a parametric random process. The parameters of the process can then be estimated using a statistical model and mathematical techniques that are precise and well understood. The use of a statistical model can help resolve ambiguous information provided by the signal data. This approach can be directly applied to musical signals in order to augment the information provided by the individual spectral components.

The problem of musical event identification will be formulated as a process of deriving an observation sequence from the input signal and then matching the sequence with a parametric statistical model of the event. This work deals with the mathematical modeling of musical events for use in model-based analysis of musical notes, intervals and chords.

1.1 Introduction to the Problem

Previous attempts to detect and analyze musical events using only data present in the signal have been frustrated by the presence of random noise and the mathematically ill-posed nature of the problem. Imperfect signal data can be augmented with extrinsic information from a parametric statistical model. In order to use model-based information to the fullest extent, it should be incorporated explicitly and as early in the analysis as possible. This will result in a more overall consistent and reliable solution.

The present work is directed to the detection and analysis of notes, chords and other musical events produced by a stringed musical instrument, specifically the guitar. The chords generated by a guitar are polyphonic, meaning that they comprise multiple notes sounded simultaneously. Each note is also spectrally complex, in that it comprises a fundamental tone and several harmonics. Despite this complexity, the statistics of the signal containing the notes and chords are expected to be similar to those of human speech [13]. This similarity will allow the signal to be characterized as a parametric random process so that established mathematical techniques can be used to extract the events from the signal. The analysis of musical signals is an important application since it is a logical extension to the problem of automated speech recognition. Moreover, a robust computer-based solution to this problem could have both research and commercial applications.

1.2 Overview

A system for automated detection and analysis of musical events, such as notes and chords, has been designed. The system is composed of two main elements: the event

library and a set of match measures. The event library, described in Chapter 4, contains a hierarchy of event models each corresponding to a distinct musical note or chord. Each event model is structured as a hidden Markov model (HMM), $\lambda = (A, B, \pi)$, having the four distinct states labeled attack, sustain, decay or silence, that correspond to the specific physical states of the musical event. Associated with each model state $Q = \{q_1, \ldots, q_4\}$ are a set of M observation symbols $V = \{v_1, v_2, \ldots, v_M\}$, a temporal duration density T and a set of three probability distributions: a transition probability distribution $A = \{a_{ij}\}$, an observable probability distribution $B = \{b_j(k)\}$ and an initial probability distribution $\pi = \{\pi_i\}$. Three match measures are developed in Chapter 5: one for estimating the HMM parameters, one for determining the optimal state sequence of the HMM and one for evaluating the probability that a given observation sequence was produced by a specific HMM. The observation sequence is derived from the input signal by sampling, converting to a spectral representation, and digitally coding using standard speech recognition techniques [13]. The three match measures correspond, respectively, to training the model, refining the model and matching an event to a model, each of which is performed using conventional speech processing algorithms. The key processing steps of the system may be summarized as follows:

- Convert the musical event to sequence of observation symbols
- If models have not been trained, then estimate and refine model parameters using the observation symbols
- Else, given the observation symbols compute model likelihood for each stored model
- Select the musical event model having the highest model likelihood

1.3 Summary of Contributions

The present work demonstrates the structural and mathematical similarities between the speech and music production processes, and shows how the large body of research in automated speech recognition can be applied to the recognition and analysis of musical events. A practical and mathematically sound approach to the analysis and detection of musical events has been designed. The approach of the present work models the music signal as a parametric random process that allows the incorporation of explicit information about the statistics of the embedded musical events. Previous music analysis approaches have not used parametric statistical models, but instead have relied on non-parametric signal processing techniques or special hardware. An event library and set of match measures have been developed that provide for the accurate identification of musical events embedded in an input signal. Model training, refinement and event matching are formulated as optimization problems that can be solved by conventional signal processing and speech recognition algorithms.

Chapter 2

Related Work in Musical Chord Detection

Most of the prior work in musical chord detection has focused on low-level feature detection followed by organizing or grouping processes. This work, discussed in Section 2.1, has focused on the detection of individual pitches as the basis for the bottom-up construction of more complex intervals and chords. The process of grouping individual pitches into more complex intervals and chords is discussed in Section 2.2.

2.1 Pitch Detection

Several researchers have done work on detecting individual musical tones (*i.e.*, pitches) from complex signals comprising a fundamental component and a weighted sum of harmonics [1], [2]. Typically, pitch detection involves sampling the input signal and performing a transform from the time domain to the frequency domain. The pitches are then detected using the resulting spectral components. The simplest methods involve sampling the signal, performing a Fourier transform to obtain the spectral components then selecting the component having the largest amplitude. These methods depend on the fundamental component actually having the largest amplitude and being able to set a threshold that selects only the component of interest. However, in practice neither of these conditions can be met on a consistent basis. In many cases, the amplitude of each spectral component varies rapidly in time as the energy decays and is redistributed between the fundamental component and its harmonics. With some stringed instruments

the fundamental component may initially have the largest amplitude, which will quickly decay as the string continues to vibrate. In addition, the presence of random signal noise makes it difficult to set a single threshold that excludes the noise while selecting only the fundamental component.

The pitch detection problem is further complicated when analyzing multiple pitch (*i.e.*, polyphonic) signals. In addition to the problems discussed above, the spectral components of the individual tones overlap in the frequency domain. This problem is particularly severe when analyzing harmonically related musical tones, such as intervals or chords. By definition, chords are composed of musical tones whose fundamental frequencies are combined in an aesthetically pleasing manner. This tends to mask the identity of the individual tones and make separation of the underlying pitches, without additional information or special hardware, extremely difficult.

A number of approaches to musical pitch detection are discussed below. The discussion is not intended to be an exhaustive list, but should give an idea of the range of approaches, while emphasizing those that are relevant to the present work. Although not directly relevant to the present work, an excellent overview of several pitch detection algorithms for use in speech processing can be found in Rabiner et al. [2].

2.1.1 Frequency-To-Voltage Conversion

One commercial product uses solid-state frequency-to-voltage converters to transform the individual string vibrations of a guitar to proportional voltages [3]. The resulting analog voltage levels are then sampled and digitized to allow a computer program to determine the corresponding pitch. The accuracy of the method depends on the frequency

resolution of the converters along with the accuracy of the sampling and digitization. It also depends on the fundamental component having the largest amplitude for a period of time sufficient to perform the frequency-to-voltage conversion. Moreover, the method works only for detecting individual pitches and is therefore used with, and has the same limitations as, the multiple transducer discussed below in Section 2.1.3.

2.1.2 Adaptive Thresholds

Adaptive methods for setting the amplitude thresholds have been proposed [4]. Some approaches are based on estimating the amplitude of the background signal noise and setting the threshold above the noise, but below the fundamental component. These methods can work reasonably well with signals in which the fundamental component initially has the largest amplitude, and in which the fundamental amplitude does not decay too quickly relative to the amplitudes of the harmonics. However, accurately estimating the signal noise, particularly with non-stationary signals, is in itself a difficult signal processing problem.

2.1.3 Multiple Transducers

Multiple transducers are being used commercially to simplify the problem of polyphonic pitch detection for stringed instruments such as the guitar [5], [6]. The method assigns a dedicated electromagnetic transducer or pickup to detect the vibrations of each string, thus reducing the polyphonic pitch detection problem to several individual pitch detection problems. However, in order to minimize the detection of adjacent string vibrations, each transducer must be positioned as close as possible to its assigned string. In practice, this

often requires that the transducers be placed so close to the strings that they make physical contact with the strings during normal playing. This can result in unwanted "buzzing" and other sounds. Moreover, manual sensitivity adjustments are required for each transducer to compensate for the relatively crude mechanical positioning of the transducers.

2.1.4 Neural Networks

The application of neural networks to pitch perception and tonal analysis has been studied by Sano and Jenkins [7]. They proposed a biologically motivated neural network model that, given a complex input tone comprising of a fundamental frequency and a weighted sum of harmonics, identifies both the pitch and octave of complex tone. The model is based on the place theory of pitch discrimination from studies of the human inner ear [8], and uses the synthetic mode of pitch perception [9] in which all of the tonal components are perceived as a unified pitch. The basic model is not designed to handle multiple tone inputs; however, in an addendum Jenkins describes an extension to the pitch perception network that permits multiple complex tones to be input simultaneously for identification of each corresponding pitch and octave.

2.2 Grouping Processes

Once the individual pitches have been detected, they must be analyzed and grouped together to form more complex musical events, such as intervals, triads and chords. These processes should use models based on music theory for the groupings; however, relatively little work has been focused in this direction. A few prior approaches use

simple heuristics or other *ad hoc* approaches to form chords, while others have applied neural networks to tonal analysis and chord classification [10].

2.2.1 Tonal Induction Networks

Scarborough et al. [11] presents two simple connectionist networks for performing tonal induction and musical key identification from a sequence of input notes. The first is a three-layer linear network comprising a layer of pitch class nodes, a layer of major chord nodes and a layer of major key nodes. The occurrence of one or more input pitches activates the corresponding pitch nodes. Activation then flows to each chord node that includes the active pitches, and from the chord nodes to any key nodes for which the active chords are the tonic, dominant or subdominant chords. The amount of pitch node activation is proportional to the duration of the input pitches and, when a pitch ends, the activation of the corresponding pitch node does not stop immediately, but instead decays with time. The activation of the chord and key nodes similarly decay with time. Overall, the weights interconnecting each of the node layers, along with the decay parameters, are critical to the performance of the network. Thus, an important issue is how to best estimate the values of these parameters and the authors admit that, in their work, the parameter values were selected largely based on intuitive guesses.

The second network augments the first with an additional multilevel, nonlinear architecture that maps individual notes onto scale degrees. This provides for an analysis of some additional aspects of human music perception, such as how intervals and chords are recognized despite transposition. Further, by regarding the network nodes as components of a vector in a tonal pitch space, the perceptual proximity of pitches, chords

and key is also modeled. Finally, competitive learning algorithms are suggested for learning the first network, while more complicated learning algorithms would be required for the second.

2.2.2 Chord Classification Networks

Laden and Keefe [12] explored alternative representations of musical pitch for input into a neural network for musical applications. They then tested the feasibility of each representation with neural networks designed to classify chords as root position major, minor or diminished triads. The representations studied included the 12 tones of the dodecaphonic well-tempered chromatic scale (*i.e.*, the pitch-class approach), and the harmonic and sub-harmonic complex representations (*i.e.*, the psychoacoustical approach). Both adjacent layer and fully connected networks with 12 input nodes, 3 to 25 hidden nodes, 3 output nodes and 2 forms of output encoding were used to test the pitch-class representation. The key findings were that an adjacent layer network with 25 hidden units and interval encoding correctly identified 94 percent of the chords, and a fully connected network with 3 hidden units and simple output encoding correctly identified 72 percent of the chords. The networks were also found to be very sensitive to starting state and learning parameters.

The psychoacoustical approach comprises two representations: the harmonic representation and the sub-harmonic representation. An adjacent layer network with 47 input nodes, 25 hidden nodes and 3 interval-based output nodes was used to test the harmonic representation. The network correctly identified 36 out of 36 chords. Similarly, an adjacent layer network with 50 input nodes, 25 hidden nodes and 3 interval-

based output nodes was used to test the sub-harmonic representation. This network correct identified 35 out of 36 chords and was thus 97 percent accurate. Thus, the psychoacoustic approaches were better able to learn the mapping of input pitch to output chord type because the harmonic and sub-harmonic input signals have more structure than the simple pitch representation. The latter were also able to classify incomplete harmonic patterns, identify chord inversions from harmonic and sub-harmonic representations, and identify chords from harmonic patterns as the input values were modified to simulate variations in power spectra.

2.3 Discussion

Most of the prior work in musical chord detection has focused either on pitch detection or forming specialized groupings of musical tones. Relatively little work has been done to combine pitch detection and grouping into an integrated system for detecting musical notes, intervals and chords. Moreover, while the neural network approaches to pitch detection and chord classification contain *implicit* music theory models in the network architecture and interconnecting weights, none of the other pitch detection approaches are explicitly model-based. In addition, none of the prior work on tonal induction or chord classification discusses how the input signal is processed to obtain the idealized note or pitch representations that form the input to their networks. Indeed, the inherently random nature of the input signal is generally ignored in all of the above work except that of adaptive thresholds, and problems of noise and training time are not discussed. Thus, there is a need for an approach to musical event analysis that *explicitly* models both the musical theory and statistics of the musical signal.

Chapter 3

Deterministic Event Models

3.1 Design Considerations

The models used in detecting and analyzing musical events in the present work embody both the physical processes of musical signal production, along with a theory regarding the musical information contained in the signal. The goal of this chapter is to analyze and compare the physical processes of speech and music.

3.2 Physical Models

The models to be reviewed and compared in this chapter are based on physical speech and musical production mechanisms. Each mechanism is modeled as a linear time-invariant system and the corresponding transfer functions are derived and compared. The complex roots of each transfer function are its poles and zeros. The poles of each transfer function are of particular interest in the present work, since they correspond to the resonant frequencies of the vocal tract or the vibrating strings of the guitar. Complex changes in the speech or music production mechanisms result in changes in the parameters of the transfer function which are, in turn, reflected in the time-varying behavior of the poles. This provides a direct link between the physical mechanisms and the spectral content of the speech or music signal.

In the present chapter, the input to the speech and music production systems is assumed to be a deterministic function of time and/or space without the addition of

random noise. Each output is therefore a deterministic function of time. The discussion of Chapter 4 will include the addition of a random component to the input signal.

3.2.1 Speech Production

The speech production mechanism in humans is a complex acoustic-mechanical system that converts air pressure supplied by the lungs and diaphragm into the distinctive sounds, or phonemes, which are used to construct words [13]. The air is forced through the vocal cords that are caused to vibrate by the glottal air flow. The air flow is then divided into quasi-periodic pulses and is frequency modulated by a number of dynamic physical processes when passing through the vocal tract comprising the throat, mouth and nasal cavities. These physical processes continually change the acoustic impedance, and thus the transfer function, of the vocal tract.

The events in the speech signal may be broadly classified into three states corresponding to the state of the vocal cords during speech production. Voiced speech occurs when the vocal cords are tense and vibrating, thereby generating a speech waveform that is quasi-periodic. With unvoiced speech, the vocal cords are not vibrating and the resulting speech waveform is random in nature. Finally, during silence no speech is produced.

A simplified model of the speech production mechanism is shown in Figure 3.1. A switch selects between a quasi-periodic pulse train representing the glottal pulses of voiced speech, random noise representing unvoiced speech, and silence. The periodicity of the glottal pulses is the fundamental frequency, or pitch, of the speech signal, while the random noise can be modeled, for all practical purposes, as white Gaussian noise (WGN).

Note that both the quasi-periodic pulse train and random noise can be simultaneously selected. This allows the modeling of speech components that include elements of both voiced and unvoiced speech.

The selected inputs are multiplied by a gain factor and form the input to a time-varying digital filter that models the dynamics of the vocal tract. The time-varying parameters of the digital filter correspond to the dynamic physical processes of the vocal tract, and a key assumption is that the parameters are slowly varying functions of time. This allows the speech signal to be analyzed in short time intervals of 5 to 25 milliseconds, within which the speech signal is assumed to be time invariant or quasi-stationary. The assumption of time invariance greatly simplifies the analysis of the speech signal.

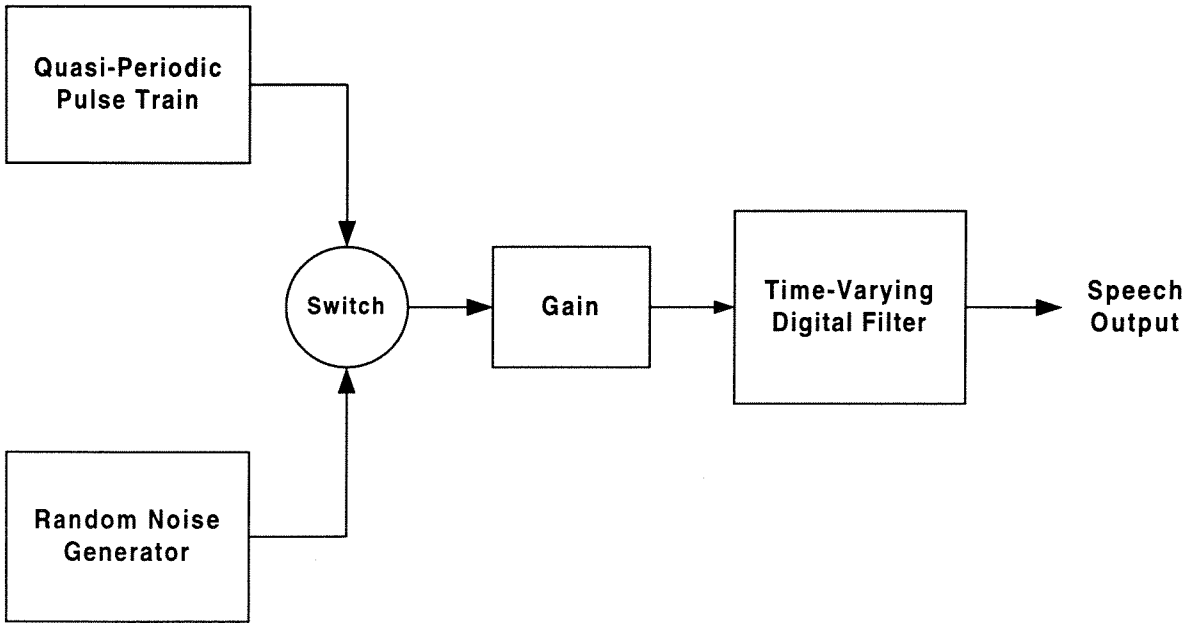

Figure 3.1:
Speech Production Model

The transfer function of the vocal tract is modeled by the system function of the time-varying digital filter in Figure 3.1. An explicit mathematical expression for the system function may be derived as follows [13]. A given speech sample at time n is modeled as a linear combination of the previous p samples plus the current input:

$$s(n) = \sum_{i=1}^{p} a_i s(n-i) + Gu(n)$$

This equation describes a time-invariant, P^{th}-order auto-regressive (AR) or (with a white noise input) Markov process that can be implemented as a recursive digital filter as follows. As mentioned above, the coefficients $\{a_i\}$ are assumed to be real and constant over the speech analysis interval. Taking the z-transform of the above equation gives

$$S(z) = \sum_{i=1}^{p} a_i z^{-i} S(z) + GU(z)$$

which can, in turn, be rearranged to obtain the system function of the vocal tract

$$\frac{S(z)}{GU(z)} = \frac{1}{1 - \sum_{i=1}^{p} a_i z^{-i}} = \frac{1}{A(z)} = H(z)$$

$H(z)$ is an all-pole representation of the vocal tract system function that effectively models the resonant frequencies, or formants, of the vocal tract. The roots of the denominator $A(z)$ are the poles of the system function. The locations of the poles in the complex z-plane can be shown explicitly by first factoring the polynomial denominator $A(z)$ into the following product-of-terms form:

$$H^S(z) = \frac{1}{\prod_{i=1}^{p} \left(1 - d_i z^{-1}\right)}$$

where each d_i is a root of the denominator and is thus the location of a corresponding pole in the complex z-plane. Next, perform a partial fraction expansion to obtain a parallel combination of first-order filter sections:

$$H_P^S(z) = \sum_{i=1}^{p} \frac{c_i}{1 - d_i z^{-1}}$$

where either or both of the c_i and d_i may be complex. Note that for the coefficients of the above AR process to be real, all of the complex d_i must only occur in complex-conjugate pairs. $H_P^S(z)$ may now be rewritten explicitly as a parallel combination of first-order real and second-order complex-conjugate sections:

$$H_P^S(z) = \sum_{i=1}^{k} \frac{c_i}{1 - d_i^R z^{-1}} + \sum_{i=k+1}^{p'} c_i \left(\frac{1}{1 - d_i^C z^{-1}} + \frac{1}{1 - \tilde{d}_i^C z^{-1}} \right)$$

where the d_i^R are all real, $\tilde{d}_i^C$ is the complex-conjugate of d_i^C and $p' = (p - k + 1)/2$ is even. This form will be useful when comparing the vocal tract and guitar system functions later in this chapter.

In order to implement $H(z)$, the prediction coefficients $\{a_i\}$ must be estimated. The estimation is often done using linear predictive coding (LPC), which is a P^{th}-order linear predictor that attempts to predict the value of any point of a time varying linear system based on the values of the previous P samples. The LPC method will be fully discussed in Chapter 5.

Since $H(z)$ models the resonant frequencies of the vocal tract, it is a good model for the vocal tract spectral envelope corresponding to steady state voiced speech. However, it is a poor model for processes that have no poles, such as unvoiced speech,

certain nasalized sounds and breath noise. Indeed, many nonlinear signal components such as noise have an adverse effect on the estimates of the prediction coefficients $\{a_i\}$.

The fundamental sounds of human speech are called phonemes [13]. Phonemes are the linguistically distinct speech sounds that are used to build words, and are formed in direct response to the physical processes of the vocal tract discussed above. There are approximately 48 phonemes in American English, including 18 vowels or vowel combinations (the latter called diphthongs), 4 semi-vowels, 21 consonants, 4 syllabic sounds and 1 glottal stop. The speech formants correspond to vowels, semivowels and diphthongs and most practical speech recognition systems rely heavily on vowel recognition to achieve high performance. As a model for the speech formants, $H(z)$ is also an effective model for these phonemes. Moreover, as will be discussed later in this chapter, vowels correspond to musical notes, intervals and chords, and diphthongs to smooth transitions between the notes, intervals and chords. Therefore, $H(z)$ should work with musical signals at least as well as it does with speech.

3.2.2 Music Production

Music is acoustical sounds consisting of pleasing or expressive combinations of tones. Musical composition is the creative process of designing a pleasing or expressive combination of tones. A musical performance is the creative process of interpreting a musical composition and producing the musical sounds contained in the composition. Original musical sounds are produced by the human voice or by playing musical instruments.

For purposes of the present work, musical events may be broadly classified as notes, intervals, chords and rests. Notes are complex musical tones composed of a fundamental pitch along with one or more harmonics or overtones. The presence of the overtones depends on the particular musical instrument and is important to the timbre of the note. The interval between two sounds is the spacing between them in pitch or frequency and there are two types of intervals: harmonic and melodic. Harmonic intervals consist of two simultaneous musical tones whose fundamental pitches are separated by a specific frequency. Melodic intervals consist of two musical tones sounded one after the other. A chord is a combination of three to seven musical tones that are sounded simultaneously (strummed) or successively (arpeggiated). A rest is a period of silence during which no musical sound is produced.

Musical instruments are mechanical or electronic devices used to produce musical sounds. Most employ resonant or multi-resonant systems for producing the definite and discrete tones of Western music [14], along with a radiating system for producing sound waves in air corresponding to the musical tones. The resonant systems include at least one element in which kinetic energy is stored, and another element in which potential energy is stored. At resonance, energy flows from one element to the other and vice versa. The key classes of musical instruments are string, wind, percussion and electronic, according to the type of vibrating element used to produce the musical tones.

The present work is concerned with string instruments, in particular, the guitar. The vibrations of string instruments give rise to a full range of overtones which are harmonics of a fundamental frequency in the ratio 1, 2, 3, 4, 5, ... etc. The fundamental frequency is determined by the length of the vibrating string. As will be discussed below,

the number and amplitude of the harmonics depend on how and where the string is excited. Certain combinations of tones are used to construct a musical scale, defined as a series of tones arranged from low to high frequency by definite intervals suitable for musical purposes. Although a vibrating string is theoretically capable of generating all of the intervals of the harmonic series, the large number of frequencies in the resulting scale (called the scale of just intonation) makes building a musical instrument with fixed tones impractical. Thus, the musical scale produced by the modern guitar consists of 12 equally spaced intervals and is called the scale of equal temperament or the chromatic scale.

The interval of two frequencies having the ratio 2:1 is called an octave. The chromatic scale is a division of the octave into 12 equal intervals, called tempered half tones or semitones. A semitone is the frequency ratio between any two tones whose frequency ratio is the twelfth root of 2. A further division of the octave exists and is termed the cent. A cent is the interval between any two tones whose frequency ratio is the twelve-hundredth root of 2. Thus, there are 1200 cents in an octave and each semitone contains 100 cents. The mathematical relationship between cents and semitones is given by

$$cent \cong 3986 Log_{10}\left(\sqrt[12]{2}\right)$$

where $\left(\sqrt[12]{2}\right) \cong 1.059463 \equiv \beta$. The intervals and frequency ratios of the chromatic scale are shown in Table 3.1.

Given the tonic or root frequency f of any chromatic scale, the remaining tones of the chromatic scale can be determined using

$$f^n = \beta^n \times f$$

Using this formula, the ratios of the tones and tone frequencies in an octave in terms of the E_2 tonic in the chromatic scale are shown in Table 3.2.

Table 3.1: Chromatic Scale

Interval Name	**Frequency Ratio From Starting Point**	**Cents From Starting Point**
Unison	1:1	0
Semitone or minor second	1.059463:1	100
Whole tone or major second	1.122462:1	200
Minor third	1.189207:1	300
Major third	1.259921:1	400
Perfect fourth	1.334840:1	500
Augmented fourth	1.414214:1	600
Perfect fifth	1.498307:1	700
Minor sixth	1.587401:1	800
Major sixth	1.681793:1	900
Minor seventh	1.781797:1	1,000
Major seventh	1.887749:1	1,100
Octave	2:1	1,200

Table 3.2: Tonal Frequency Ratios in Chromatic Scale
(f = frequency of tonic E_2)

Note	**Frequency Ratio** $\beta^n \times f$	**Frequency (Hz)** f^n
E	1.000000f	82.41
F	1.059463f	87.31
$F^{\#}$	1.122462f	92.50
G	1.189207f	98.00
$G^{\#}$	1.259921f	103.83
A	1.334840f	110.00
$A^{\#}$	1.414214f	116.54
B	1.498307f	123.47
C	1.587401f	130.81
$C^{\#}$	1.681793f	138.59
D	1.781797f	146.83
$D^{\#}$	1.887749f	155.56
E	2.000000f	164.81

Using the notes of the chromatic scale, major and minor scales, intervals and chords for any musical key may be defined. A major or minor scale is defined as a sequence of tones having a specific pattern of semitone and whole tone intervals

separating the tones. For example, the interval pattern (*i.e.*, specific sequence of notes) for a major scale is whole-whole-semitone-whole-whole-whole-semitone. The pattern is the same for all keys; the only difference is a shift in the position of the pattern. There are analogous patterns for the harmonic and melodic minor scales.

Intervals are indicated by the combination of a name and number, the latter derived from the order of the notes in the scale. The interval of a minor second is one semitone, the interval of a major second is two semitones or one whole tone, the interval of a minor third is three semitones, and so forth. Chords are defined in a similar manner. The lowest note, or fundamental, of a chord is called the root. The simplest chord, which contains three notes, is called a triad. A major triad consists of the root, the third and the fifth. A minor triad consists of the root, the minor third and the fifth. More complex chords are constructed in a similar manner using up to seven[1] notes designated as the third, fifth, seventh, ninth, eleventh and thirteenth.

The above discussion demonstrates the highly structured nature of music and musical signals. Notes, intervals and chords are all composed of combinations of fundamental tones and harmonics. The latter correspond to the resonant frequencies of the musical instrument that is being played.

In the case of string instruments such as the guitar, the resonant frequencies are generated by the transverse vibrations of one or more stretched strings. With transverse vibrations, each part of the string vibrates in a plane perpendicular to the line of the string. The guitar comprises six strings stretched between an integrated bridge and tailpiece mounted on the top of the body and the end of a fretted fingerboard. For an acoustic guitar, the body is typically constructed of two flat parallel panels fastened

together along the outside edges. The bottom panel of the body is mechanically coupled to the top panel by a wood post. Vibrations of the strings are transmitted from the tailpiece to the top panel of the body and through the post to the bottom panel. The top panel forms a sounding board and the hollow cavity of the body forms a Helmholtz resonator that is coupled to the outside air by an opening in the top panel. The guitar body has its own resonance characteristics which contribute to the tones produced by the acoustic guitar. The guitar body converts the transverse string vibrations into longitudinal sound waves.

For an electric guitar, the body is typically solid and is constructed of wood, fiberglass or plastic. The vibrations of the strings are transmitted to an electromagnetic pickup mounted on the top of the body, underneath and adjacent to the strings. The vibrating string produces a change in the magnetic flux, supplied by a permanent magnet, through a coil. This induces an alternating voltage corresponding to the vibrations of the string. The voltage is sent to an electric amplifier for amplification, and to a loudspeaker for converting the electric signal to longitudinal sound waves.

The open strings of the standard guitar are tuned to E_2, A_2, D_3, G_3, B_3, and E_4. Most acoustic guitars have 12 to 15 usable frets, so the corresponding frequency range is 82.41 Hertz (E_2) to 783.99 Hertz (G_5). Electric guitars have 21 to 24 frets and a corresponding frequency range of is 82.41 Hertz (E_2) to 1,318.51 Hertz (E_6). Allowing for "drop D" tuning can extend the lower frequency range to 73.42 Hertz (D_2), while harmonics can extend the upper frequency range to 5,274.04 Hertz (E_8). By contrast, the frequency range of human speech nominally extends from 10^2 Hertz to 10^4 Hertz. The

---

[1] Only six notes can be played simultaneously on a six string guitar.

frets on a guitar are spaced so that pressing on a string on any two adjacent frets produces notes that are one semitone apart in pitch.

A simplified model of the music production mechanism for a guitar is shown in Figure 3.2. The guitar model of the present work comprises two digital filters that are cascaded to model the vibrating strings and the resonator/transducer. The resonator model applies only to hollow body acoustic and electric guitars. The transducer model applies to both hollow and solid body guitars, since in either case the transverse string vibrations must be converted to corresponding electric signals for analysis and detection of musical events. The input to the string model, $f(x,t)$, represents the strings being strummed, plucked, tapped or otherwise caused to vibrate. The output of the string model, $y_p(x,t)$, forms the input to the resonator/transducer model. The output of the resonator/transducer, $\hat{g}_p(t)$, is the musical output of the guitar model.

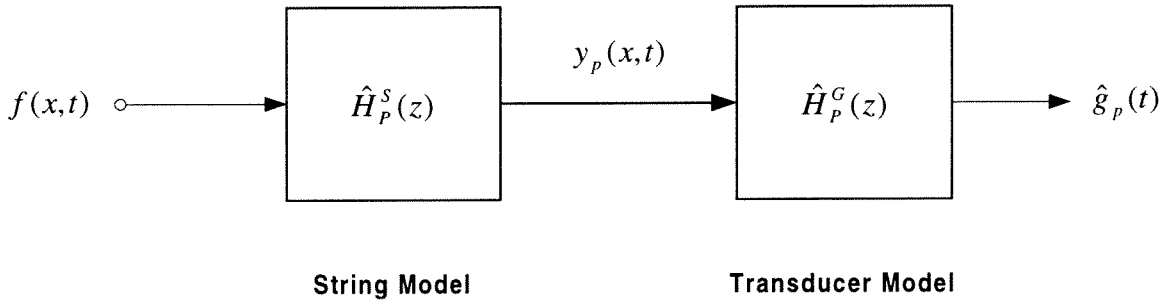

Figure 3.2:
Music Production Model – Guitar

Each guitar string is modeled using the forced, damped wave equation given by

$$\frac{\partial^2 y}{\partial t^2} = c^2 \frac{\partial^2 y}{\partial x^2} - 2b\frac{\partial y}{\partial t} + f$$

where $c = \sqrt{T/\rho}$ is the speed of wave propagation on the string determined, in turn, by the string tension T and mass per unit length of the string ρ, and b is the damping coefficient of the string. For a string of length L, the above hyperbolic PDE will be solved subject to the following boundary and initial conditions

$$y(0,t) = y(L,t) = 0$$

$$y(x,0) = f(x)$$

$$\frac{\partial}{\partial t} y(x,0) = g(x)$$

Assume the solution $y(x,t)$ and forcing function $f(x,t)$ can both be expanded in Fourier series as follows:

$$y(x,t) = \sum_{n=1}^{\infty} h_n(t) \sin k_n x$$

$$f(x,t) = \sum_{n=1}^{\infty} f_n(t) \sin k_n x$$

$$f_n(t) = \frac{2}{L} \int_0^L f(x,t) \sin k_n x dx$$

with $k_n = n\pi/L$ the wave number of wave n. By differentiating the first series and substituting the first two series into the wave equation and simplifying, an ordinary differential equation ODE for the time dependent expansion coefficients $h_n(t)$ is obtained

$$\ddot{h}_n(t) + 2b\dot{h}_n(t) + \omega_n{}^2 h_n(t) = f_n(t)$$

where $\omega_n{}^2 = k_n^2 c^2$ is the frequency of harmonic n. The general solution of this ODE as the form $h_n(t) = h_n^c(t) + h_n^p(t)$ with the first term on the right being the homogeneous solution (sometimes called the complementary function) and the second term on the right

being a particular solution. Since the vibrating string is assumed to be under-damped, $h_n^c(t)$ can be shown to have the general form

$$h_n^c(t) = e^{-bt}\left(A_n \cos\sqrt{\omega_n^2 - b^2}\,t + B_n \sin\sqrt{\omega_n^2 - b^2}\,t\right)$$

Let $\hat{\omega}_n = \sqrt{\omega_n^2 - b^2}$ and by substituting $h_n^c(t)$ into the Fourier series for $y(x,t)$ the complementary function for the homogeneous wave equation becomes

$$\begin{aligned} y_c(x,t) &= e^{-bt}\sum_{n=1}^{\infty}\left(A_n \cos\hat{\omega}_n t + B_n \sin\hat{\omega}_n t\right)\sin k_n x \\ &= \sum_{n=1}^{\infty} h_n^c(t)\sin k_n x \end{aligned}$$

with

$$A_n = \frac{2}{L}\int_0^L f(x)\sin k_n x\,dx$$

$$B_n = \frac{2}{\hat{\omega}_n L}\int_0^L g(x)\sin k_n x\,dx + \frac{b}{\hat{\omega}_n}A_n$$

It can be further shown that $y_c(x,t)$ represents a superposition of standing waves of spatial frequency k_n and temporal frequency $\hat{\omega}_n$.

The particular solution $h_n^p(t)$ is derived using the method of variation of parameters. The Wronskian is given by the following determinant:

$$\begin{aligned} W &= \det\begin{vmatrix} y_1 & y_2 \\ \dot{y}_1 & \dot{y}_2 \end{vmatrix} \\ &= \det\begin{vmatrix} e^{-bt}\cos\hat{\omega}_n t & e^{-bt}\sin\hat{\omega}_n t \\ -e^{-bt}\left(b\cos\hat{\omega}_n t + \hat{\omega}_n \sin\hat{\omega}_n t\right) & -e^{-bt}\left(b\sin\hat{\omega}_n t - \hat{\omega}_n \cos\hat{\omega}_n t\right) \end{vmatrix} = \hat{\omega}_n e^{-2bt} \end{aligned}$$

Define

$$\dot{u}_1 = -\frac{y_2 f(x)}{W} = \frac{-e^{bt} f(t)\sin\hat{\omega}_n t}{\hat{\omega}_n}$$

$$\dot{u}_2 = \frac{y_1 f(t)}{W} = \frac{e^{bt} f(t) \cos \hat{\omega}_n t}{\hat{\omega}_n}$$

The particular solution then has the general form

$$h_n^p(t) = u_1 y_1 + u_2 y_2$$

By direct substitution we obtain the above components of the particular solution

$$u_1 y_1 = \frac{-1}{\hat{\omega}_n} \int_0^t e^{-b(t'-\tau)} f_n(\tau) \cos \hat{\omega}_n t' \sin \hat{\omega}_n \tau d\tau$$

$$u_2 y_2 = \frac{1}{\hat{\omega}_n} \int_0^t e^{-b(t'-\tau)} f_n(\tau) \sin \hat{\omega}_n t' \cos \hat{\omega}_n \tau d\tau$$

By combining the above two components and simplifying using trigonometric identities, the particular solution for the wave equation is obtained:

$$\begin{aligned} y_p(x,t) &= \sum_{n=1}^{\infty} \left[\frac{1}{\hat{\omega}_n} \int_0^t e^{-b(t'-\tau)} f_n(\tau) \sin \hat{\omega}_n (t'-\tau) d\tau \right] \sin k_n x \\ &= \sum_{n=1}^{\infty} h_n^p(t) \sin k_n x \end{aligned}$$

Therefore, the general solution to the wave equation is given by the sum of the complementary function $y_c(x,t)$ and the particular solution $y_p(x,t)$:

$$\begin{aligned} y(x,t) &= y_c(x,t) + y_p(x,t) \\ &= \sum_{n=1}^{\infty} \left[h_n^c(t) + h_n^p(t) \right] \sin k_n x \end{aligned}$$

where $h_n^c(t)$ is the complementary function and $h_n^p(t)$ is the particular solution, respectively, to the above ODE. The complementary function is a transient term that depends on the initial conditions $f(x)$ and $g(x)$, while the particular solution is a steady state term that depends on the forcing function $f(x,t)$. Playing the guitar amounts to

dynamically altering the initial conditions, boundary conditions and forcing function, at times independent of, and at other times simultaneous to, one another.

For example, the strings are plucked by pulling them away from their equilibrium positions and then releasing them. Strumming is a similar process applied to multiple strings. Both plucking and strumming are typically modeled using a non-zero value of $f(x)$. Rapid picking, continuous strumming and tapping of the strings can be modeled as a non-zero forcing function $f(x,t)$. Fretting the strings along the neck changes the effective string length and therefore the boundary conditions.

Assuming the strings of the guitar are initially at rest, the complementary function vanishes and the musical sound is generated only by $y_p(x,t)$. The term in brackets has the form of a convolution integral from linear systems theory. In order to determine the impulse response of the string model, recall that $f(t)$ is given by

$$f(t) = \frac{2}{L}\int_0^L f(x,t)\sin k_n x dx$$

Assume an impulse input of the form $f(x,t) = \delta(x-\beta)\delta(t-\tau)$ and substitute to obtain

$$\begin{aligned} f(t) &= \frac{2}{L}\int_0^L \delta(x-\beta)\delta(t-\tau)\sin k_n x dx \\ &\le \frac{2}{L}\int_{-\infty}^{\infty} \delta(x-\beta)\delta(t-\tau)\sin k_n x dx \\ &= \frac{2}{L}\delta(t-\tau)\sin k_n \beta \end{aligned}$$

provided that $0 < \beta < L$, otherwise, $f(t) = 0$. Substituting into $y_p(x,t)$ gives the impulse response, or Green's function, for the string model:

$$
\begin{aligned}
g_p(x,t;\beta,\tau) &= \sum_{n=1}^{\infty} \frac{2}{\hat{\omega}_n L} e^{-b(t-\tau)} \sin \hat{\omega}_n (t-\tau) \sin k_n x \sin k_n \beta \\
&= \sum_{n=1}^{\infty} \frac{1}{\hat{\omega}_n L} e^{-b(t-\tau)} \sin \hat{\omega}_n (t-\tau) \left[\cos k_n (x-\beta) - \cos k_n (x+\beta)\right] \\
&= \sum_{n=1}^{\infty} \frac{1}{2\hat{\omega}_n L} e^{-b(t-\tau)} \left\{\sin\left[\hat{\omega}_n (t-\tau) + k_n (x-\beta)\right] + \sin\left[\hat{\omega}_n (t-\tau) - k_n (x-\beta)\right]\right\} \\
&\quad - \sum_{n=1}^{\infty} \frac{1}{2\hat{\omega}_n L} e^{-b(t-\tau)} \left\{\sin\left[\hat{\omega}_n (t-\tau) + k_n (x+\beta)\right] + \sin\left[\hat{\omega}_n (t-\tau) - k_n (x+\beta)\right]\right\}
\end{aligned}
$$

which describes the response at point x and time t to an impulse at point β and time τ. More particularly, $g_p(x,t;\beta,\tau)$ represents a superposition of distinct modes of vibration, each a standing wave of spatial frequency k_n and temporal frequency $\hat{\omega}_n$. Finally, note that a weak solution to the original wave equation can be obtained using $g_p(x,t;\beta,\tau)$ by solving the integral equation

$$y_p(x,t) = \iint g_p(x,t;\beta,\tau) f(\beta,\tau) d\beta d\tau$$

The output of the string model $y_p(x,t)$ now forms the input to the resonator/transducer model. The primary resonator for an acoustic guitar is the hollow body whose top panel is coupled to the vibrating strings by the combination bridge and tailpiece. The primary resonator for the solid body electric guitar is the strings themselves, with little or no resonance provided by the body. In both cases a transducer converts the string vibrations into a corresponding electric signal. And in either case, in the present work the transducer is assumed to have the following two properties. First, it acts as an ideal temporal low-pass filter that passes all frequencies $\hat{\omega}_n$ from zero up to its passband frequency. Second, it also acts as a spatial low-pass filter in that it couples to only a narrow spatial segment of the vibrating string.

The physical coupling between the vibrating strings and the top panel of the acoustic guitar body can best be described as "lossy." The strings run parallel to the top panel and the power transfer to the body is greatest when the strings are excited in a direction perpendicular with respect to the top panel of the body. However, in most cases the strings are excited in a direction parallel to the panel. Once set in motion, the actual movement of the strings is elliptical, so that a portion of the motion is perpendicular to the top panel. Note that the greater the power transfer from the strings to the body, the greater the damping of the string vibrations.

For an electric guitar, there is no physical coupling and therefore no mechanical power transfer between the vibrating strings and the electromagnetic pickup. Instead, the coupling is through the magnetic field produced by the permanent magnet of the pickup. As mentioned above, the vibrating string induces a voltage in the coil of the pickup that has the same frequency characteristics as the vibrating string. Since there is no mechanical power transfer between the strings and the pickup, there is little mechanical damping of the strings. The electric guitar is therefore able to generate sustained notes and chords that are not possible with the acoustic guitar.

The present work is concerned with the detection of musical notes and chords from electric signals. Thus, the remainder of this work will focus on the electric guitar, including both solid and hollow body structures. Further, the present work will assume that the hollow body structure does not alter the basic form of $y_p(x,t)$, but merely provides a uniform gain to all frequency components $\hat{\omega}_n$. The remainder of this section will therefore focus on the conversion of the transverse string vibrations to corresponding electric signals by the electromagnetic pickup.

If the input to the electromagnetic pickup is $g_p(x,t;\beta,\tau)$, the output will be the impulse response for the cascaded combination of the vibrating string and the pickup. More generally, the pickup converts the two-dimensional signal $y(x,t)$ into a one-dimensional signal $\hat{f}(t)$, by summing the input along a small spatial segment of the string. Physically, this is equivalent to having the spatial vibrations concentrated on a limited segment of the string. For example, the electromagnetic pickup on an electric guitar senses only the vibrations of a segment of the strings located very near the pickup. The location of the pickup along the length of the string determines the tone generated by the pickup. In fact, many electric guitars have two or even three pickups that may be selected individually or in combination, thus allowing the same guitar to generate a variety of musical tones.

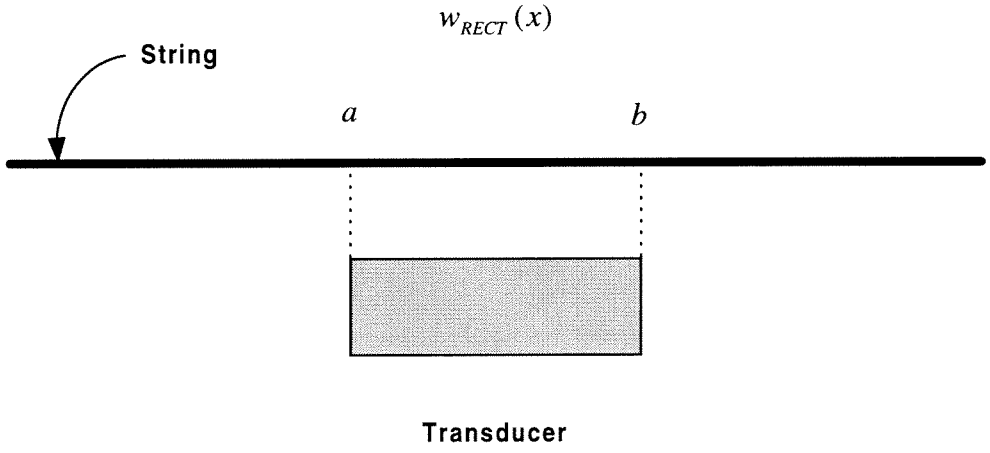

Figure 3.3:
Rectangular Spatial Windowing

Mathematically, this spatial summing is modeled by integrating $\sin k_n x$ along a short segment of the string. This is equivalent to multiplying $g_p(x,t;\beta,\tau)$ by a window function $w(x)$ and integrating the result over x. Assume the detection aperture for each pickup appears as shown in Figure 3.3, which is equivalent to multiplying by a rectangular window function $w_{RECT}(x)$ of length $a-b$. Integrating the windowed impulse response from a to b results in

$$g_p(t;\beta,\tau) = \sum_{n=1}^{\infty} \frac{2}{n\pi\hat{\omega}_n} e^{-b(t-\tau)} \sin\hat{\omega}_n(t-\tau)\left[\cos k_n a - \cos k_n b\right]\sin k_n \beta$$

A more physically accurate window function could be based on a Gaussian detection aperture $w_{GAUS}(x)$ integrated along the entire length of the string, as shown in Figure 3.4.

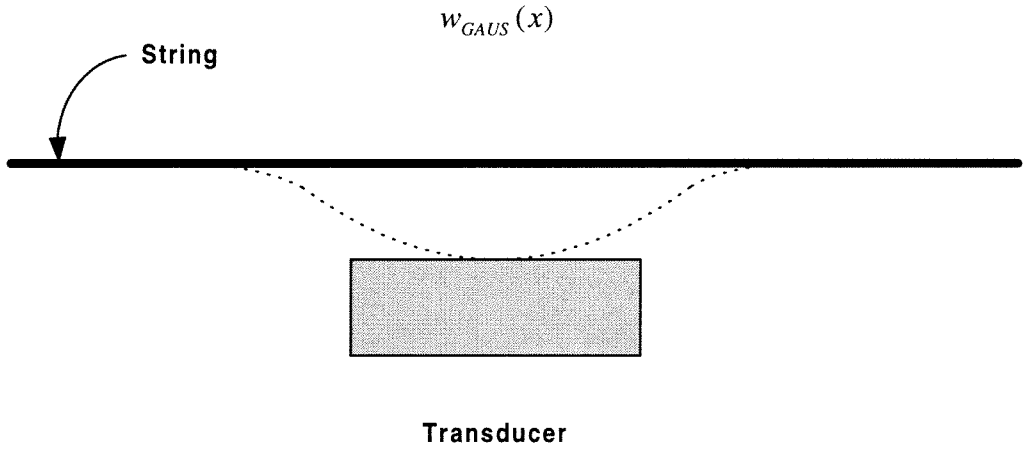

Figure 3.4:
Gaussian Spatial Windowing

However, the key point regarding $g_p(t;\beta,\tau)$ is that a and b are constants for a given electromagnetic pickup independent of the specific window function used. Further, β is also a constant for each musical note or chord so that the above impulse response $g_p(t;\beta,\tau)$ is a function of time t only. Therefore, the above impulse response may be rewritten as

$$g_p(t;\beta,\tau)=\sum_{n=1}^{\infty}\frac{2C(k_n)}{n\pi\hat{\omega}_n}e^{-b(t-\tau)}\sin\hat{\omega}_n(t-\tau)\equiv\hat{g}_p(t)$$

where $C(k_n)=(\cos k_n a-\cos k_n b)\sin k_n\beta$ is constant for each value of k_n.

The system function for the overall guitar model may now be determined by taking the temporal z-transform of $\hat{g}_p(t)$. For simplicity, let time delay τ be equal to zero and sample $\hat{g}_p(t)$ to obtain a corresponding discrete right-hand sequence $\hat{g}_p(m)$, where the number of samples m is selected to satisfy the Nyquist sampling criteria. Then take the z-transform as follows:

$$\begin{aligned}
\hat{H}_p(z)&=\sum_{m=0}^{\infty}\hat{g}_p(m)z^{-m}\\
&=\frac{2}{\pi}\sum_{n=1}^{\infty}\frac{C(k_n)}{n\hat{\omega}_n}\sum_{m=0}^{\infty}(e^{-bm}\sin\hat{\omega}_n m)z^{-m}\\
&=\frac{2}{\pi}\sum_{n=1}^{\infty}\frac{C(k_n)}{n\hat{\omega}_n}\sum_{m=0}^{\infty}\frac{1}{2i}\left[(e^{-b+i\hat{\omega}_n}z^{-1})^m-(e^{-b-i\hat{\omega}_n}z^{-1})^m\right]\\
&=\frac{2}{\pi}\sum_{n=1}^{\infty}\frac{C(k_n)}{n\hat{\omega}_n}\frac{1}{2i}\left(\frac{1}{1-e^{-b+i\hat{\omega}_n}z^{-1}}-\frac{1}{1-e^{-b-i\hat{\omega}_n}z^{-1}}\right)\\
&=\frac{2e^{-b}}{\pi}\sum_{n=1}^{\infty}\frac{C(k_n)}{n\hat{\omega}_n}\frac{\sin(\hat{\omega}_n)z^{-1}}{(1-e^{-b+i\hat{\omega}_n}z^{-1})(1-e^{-b-i\hat{\omega}_n}z^{-1})}
\end{aligned}$$

The system function $\hat{H}_p(z)$ for each vibrating string is thus a parallel combination of second-order complex-conjugate terms that converge for $\left|e^{-b}z^{-1}\right|<1$. The numerator of

each term generates a pole at $z=0$ and a zero at $z=\infty$, so the vibrating string system is auto-regressive/moving-average (ARMA). The denominator of each term generates two zeros at $z=0$ and a pair of complex-conjugate poles at $z=e^{-b\pm i\hat{\omega}_n}$. By definition, $e^{-b}\le 1, \forall b\ge 0$ so all of the complex-conjugate poles are inside the unit circle and the region-of-convergence is outside the circle defined by $|z|>e^{-b}$. As expected, for each term the total number of poles is equal to the total number of zeros. Therefore, the vibrating string system is causal and its output stable and decays over time.

In order to compare $\hat{H}_p(z)$ for the vibrating guitar string with $H_P^S(z)$ derived above for speech, the above is rewritten as the following finite sum:

$$\begin{aligned}\hat{H}_p(z) &= \frac{2}{\pi}\sum_{n=1}^{M}\frac{C(k_n)}{n\hat{\omega}_n}\frac{1}{2i}\left(\frac{1}{1-e^{-b+i\hat{\omega}_n}z^{-1}}-\frac{1}{1-e^{-b-i\hat{\omega}_n}z^{-1}}\right)\\ &= \sum_{n=1}^{M}C_n\left(\frac{1}{1-D_nz^{-1}}-\frac{1}{1-\tilde{D}_nz^{-1}}\right)\end{aligned}$$

where $C_n = C(k_n)/in\pi\hat{\omega}_n$, $D_n = e^{-b+i\hat{\omega}_n}$ and $\tilde{D}_n = e^{-b-i\hat{\omega}_n}$. Note that all the poles D_n occur only in complex-conjugate pairs (*i.e.*, there are no real poles) and that the total number of harmonics included in the model is *M*. The system function for all six strings of the guitar model then becomes a parallel combination of the individual system functions for six vibrating strings

$$\hat{H}_p^G(z) = \sum_{l=1}^{6}\sum_{n=1}^{M}C_{nl}\left(\frac{1}{1-D_{nl}z^{-1}}-\frac{1}{1-\tilde{D}_{nl}z^{-1}}\right)$$

where C_{nl} and D_{nl} are the coefficients and complex pole locations for string *l*, respectively, as defined above, with $\hat{\omega}_{nl}=[\hat{\omega}_n]_{String(l)}$ and $k_{nl}=[k_n]_{String(l)}$. Finally, due to tuning of the guitar strings and tonal structure of the chords and intervals, there will be

substantial overlap between the harmonics $\hat{\omega}_n$ produced by different strings, *i.e.*, the coefficients and pole locations will often satisfy $C_{ij} = C_{kl}, D_{ij} = D_{kl}$ for $i \neq k$ and $j \neq l$. Additionally, here the damping coefficients for each string are assumed to be nearly identical; in reality, the tones comprising each chord will actually decay at different temporal rates. Thus, assume the damping coefficients for all the strings are equal in magnitude, *i.e.*, $b_l = b, \forall l$ and define an integer $q_n \in \{1,2,\ldots 6\}$ as the number of times that harmonic $\hat{\omega}_n$ appears in the summation, *i.e.*, the number of repeating harmonics over different strings. The system function for the guitar model then reduces to the single summation

$$\hat{H}_P^G(z) = \sum_{n=1}^{M'} q_n C_n \left(\frac{1}{1 - D_n z^{-1}} - \frac{1}{1 - \tilde{D}_n z^{-1}} \right)$$

where $M' = 6M - \sum_{n=1}^{M} (q_n - 1)$ must be determined *a priori*. This form of system function is similar in structure to the complex-conjugate term of the system function derived above for the human vocal tract.

Now, the difference equation corresponding to the j^{th} second-order term of the system function $\hat{H}_P(z)$ for the k^{th} vibrating string is given by

$$\begin{aligned} \hat{s}_{jk}(n) &= \sum_{i=1}^{2} a_{ijk} \hat{s}_j(n-i) + G_{jk} u(n-1) \\ &= a_{1jk} \hat{s}_j(n-1) + a_{2jk} \hat{s}_j(n-2) + G_{jk} u(n-1) \end{aligned}$$

where the coefficients are given by

$$a_{1jk} = 2e^{-b} \cos(\hat{\omega}_{jk})$$

$$a_{2jk} = -e^{-2b}$$

$$G_{jk} = \frac{2e^{-b}C(k_{jk})\sin(\hat{\omega}_{jk})}{n\pi\hat{\omega}_{jk}}$$

with $\hat{\omega}_{jk}$ and k_{jk} the j^{th} resonant frequency and wave number for the k^{th} string as defined above. Each harmonic is thus generated by a second-order, auto-regressive/moving average (ARMA) section. The overall difference equation for the k^{th} vibrating string is given by the parallel combination of M of these second-order sections:

$$\begin{aligned}\hat{s}_k(n) &= \sum_{j=1}^{M} \hat{s}_{jk}(n) \\ &= \sum_{j=1}^{M}\left[\sum_{i=1}^{2} a_{ijk}\hat{s}_j(n-i) + G_{jk}u(n-1)\right] \\ &= \sum_{j=1}^{M}\left[a_{1jk}\hat{s}_j(n-1) + a_{2jk}\hat{s}_j(n-2) + G_{jk}u(n-1)\right]\end{aligned}$$

where the input $u(n)$ is applied simultaneously to all of the second-order sections. Note that additional zeros are generated in the total system function in making the parallel connections. However, in general determining the explicit locations of the zeros of the parallel combination of second-order filter sections is a non-trivial task. The key point is that the coefficients of the difference equations are generally independent of time. Finally, the difference equation for the overall guitar model is given by the parallel combination of six of the vibrating string difference equations:

$$\begin{aligned}\hat{s}(n) &= \sum_{k=1}^{6} \hat{s}_k(n) \\ &= \sum_{k=1}^{6}\sum_{j=1}^{M}\left[a_{1jk}\hat{s}_j(n-1) + a_{2jk}\hat{s}_j(n-2) + G_{jk}u(n-1)\right]\end{aligned}$$

Thus, the output of the guitar model is the sum of the individual outputs of the six vibrating strings.

3.3 Comparison of Deterministic Models

The above analysis suggests the structural similarity between speech and music production systems. In particular, the assumption of constant coefficients within the speech analysis interval and for the wave equation allows comparable system functions to be derived. The P^{th}-order AR process for speech production leads to the derivation of an all-pole system function that can be rewritten as a parallel combination of first-order real and second-order complex-conjugate filter sections. Similarly, the particular solution to the wave equation leads to the derivation of a system function for a vibrating string that comprises a parallel combination of second-order complex-conjugate filter sections. Further, the vibrating string system function leads directly to the derivation of multiple (*e.g.*, six) parallel M^{th}-order ARMA processes for music production by the guitar.

There are several key differences between the system functions for speech and music produced by the guitar. First, the system function for speech, $H_P^S(z)$, may include real-valued poles which suggests that the speech model is essentially a low pass filter. This indicates that the vocal tract passes all frequencies from zero up to a maximum; however, the vocal cords have a lower frequency limit below which they cannot physically vibrate. Thus, the vocal tract is actually a band pass filter, albeit with a minimal lower frequency. By contrast, the system function for the guitar, $\hat{H}_P^G(z)$, includes only complex-valued poles which indicates that the guitar model is essentially a band pass filter. The latter is supported by the physics of the vibrating string that has a fixed minimum frequency of vibration determined by the string length.

Second, each second-order complex-conjugate segment comprising the system function for the guitar includes a zero in the numerator. The effect of each zero is to

delay the input to the guitar by one unit. By comparison, the system function for speech has an all-pole structure and there is thus no delay in the input to the vocal tract. Finally, the system function for speech is modeled as a single resonator while the system function for the guitar comprises six parallel resonators.

The key similarities between the speech and guitar system functions are the dominance of the poles and the constant filter coefficients. Both systems are dominated by the their respective resonant frequencies and, within their respective analysis windows, both systems are linear time-invariant. More particularly, for speech the coefficients are assumed to be constant with the 5 to 20 millisecond analysis interval. For the guitar, the coefficients are constant by virtue of the constant coefficients in the wave equation. In fact, the guitar coefficients depend explicitly on the resonant frequencies $\hat{\omega}_n$ and wave numbers k_n, so they are constant for the duration of each given musical event.

Given the structural similarities between the speech and music production processes, it seems clear that the large body of speech recognition research can be applied to music. The remainder of the present work develops models and methods for detecting and analyzing musical events using the techniques of automated speech recognition.

Chapter 4

Stochastic Event Models

4.1 Design Considerations

The analysis of Chapter 3 implicitly assumed that the sounds generated by the human vocal tract and the guitar are deterministic, *i.e.*, known with certainty. Thus, given a known input to each system, the output can be predicted with certainty. The models developed in Chapter 3 for the vocal tract and guitar have included no provision for a random signal component.

However, the actual sounds generated by both the vocal tract and the guitar include random components. For example, breath noise and variability among speakers are examples of random speech signal components. Similarly, differences in strumming and picking, along with unwanted string noise, improper fretting and audio feedback are examples of random musical signal components. Therefore, the signal sources for both speech and music production should be modeled as parametric random processes.

4.2 Parametric Statistical Models

In order to model the stochastic processes of speech and music production, a probability distribution for each type of signal should be assumed [13]. The simplest assumption is that both speech and music production may be modeled as wide-sense, second-order Gaussian processes, in which the mean vector and covariance matrix provides all necessary and sufficient information to model the statistics of the process. Typically, the

signal is assumed to have zero mean. Recall the deterministic models developed in Chapter 3 for speech and music production, respectively:

$$s(n) = \sum_{i=1}^{p} a_i s(n-i) + Gu(n)$$

$$\hat{s}(n) = \sum_{k=1}^{6} \sum_{j=1}^{M} \left[\sum_{i=1}^{2} a_{ijk} \hat{s}_j(n-i) + G_{jk} u(n-1) \right]$$

If the input signal $u(n)$ to either model is a Gaussian random process, then the output signals $s(n)$ and $\hat{s}(n)$ will also be Gaussian random processes. Thus, for speech the AR signal is generated by passing white noise through the all-pole discrete time system. For music, the ARMA signal is generated by passing white noise through the pole-zero discrete time system.

Another important assumption concerns the temporal behavior of the signal statistics, *i.e.*, whether the components of the mean vector and covariance matrix change over time. If the signal statistics do not change over time, the signal is called stationary; otherwise, the signal is called non-stationary. For each of the above models there are two independent ways of creating a non-stationary signal. First, if the input signal is non-stationary, then the output signal will also be non-stationary. Second, if any of the process coefficients are time-dependent, the output signal will be non-stationary even if the input signal is stationary. In either case, the analysis of non-stationary signals is much more complex than that of stationary signals.

A final technical assumption is related to the averaging of the signal samples to calculate the mean and other statistical properties. Specifically, the present work assumes that ensemble averages can be replaced with temporal averages. Signals satisfying this condition are often referred to as ergodic.

In the present work, the input signals to the speech and music production models are assumed to be locally stationary. Thus, the modeling and analysis operations must be repeated for each analysis interval over which the stationarity assumption holds. In addition, the coefficients of both models are assumed to be independent of time within the analysis interval. Thus, for speech the coefficients are assumed to be constant over the speech analysis window. Similarly, the coefficients for the music production model are assumed to be constant for the duration of each musical event (*e.g.*, chord, interval or note).

Given that the speech and music signals can be characterized as parametric random processes, the next task is to design a parametric statistical model that captures the spectral properties of the signals. For the particular case of music, a set of these statistical models will form the components of a stored library of musical events. The remainder of this chapter will focus on the development of a parametric statistical model for the musical signals produced by the guitar.

4.2.1 Discrete-Time Markov Process

The parametric statistical model for musical signals is based on a state machine that may be in any one of N distinct states. The states are indexed by the integers $\{1, 2,...N\}$. Transitions from state i to state j are assumed to occur at regularly spaced time intervals. Note that a transition to the same state is allowable, and that the state transitions are controlled by a set of state-transition probabilities associated with each state. This type of state machine is often called a discrete-time, N^{th} order Markov process.

More particularly, let the time intervals associated with each state transition be denoted as t = 1,2,..., and denote the state of the system at time t as q_t. In the most general case q_t would depend on all of the previous states $q_{t-1}, q_{t-2}, \ldots, q_{t-N}$. However, for many problems of practical interest, including speech and music recognition, the current state may be assumed to depend on only the preceding state:

$$P\left[q_t = j | q_{t-1} = k, q_{t-2} = l, \ldots\right] = P\left[q_t = j | q_{t-1} = k\right]$$

In other words, all of the information contained in the previous states is combined into just the preceding state. The state-transition probabilities a_{ij} are then given by

$$a_{ij} = P\left[q_t = j | q_{t-1} = i\right]$$

where $1 \le i, j \le N$. The state-transition probabilities must also satisfy the usual stochastic constraints

$$a_{ij} \ge 0, \forall i, j$$

$$\sum_{j=1}^{N} a_{ij} = 1, \forall i$$

A state machine satisfying these criteria is called a discrete-time, first-order Markov process.

Figure 4.1 is a simplified graph of the amplitude envelope for a *generic* musical event (chord or note) generated by the vibrating strings of a guitar. The amplitude envelope for a musical event generally comprises three weakly distinct acoustic segments: attack, sustain and decay. The attack segment occurs when the strings are plucked, strummed or otherwise excited, and is characterized by a rapid increase in amplitude as kinetic energy is transferred to the strings. The sustain segment follows the

attack segment and is characterized by a nearly level amplitude, *i.e.*, there is minimal energy dissipation. Note, however, that there is always some energy dissipated by the vibrating strings. Finally, the decay segment is characterized by an exponential decrease in amplitude as the potential and kinetic energy of the vibrating string is dissipated.

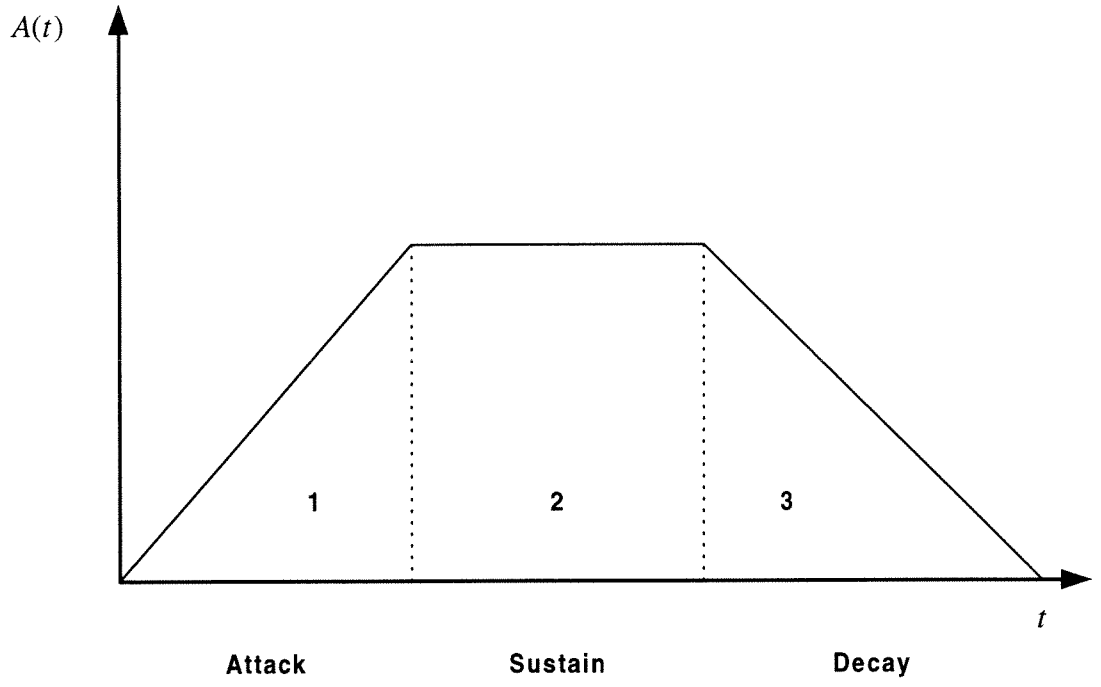

Figure 4.1:
Amplitude Envelope for Generic Music Signal

Continuing with Figure 4.2, a simple three-state Markov process corresponding to the amplitude envelope of Figure 4.1 is shown. Each state directly corresponds to one of the three segments discussed above, as follows:

- State 1: Attack
- State 2: Sustain
- State 3: Decay

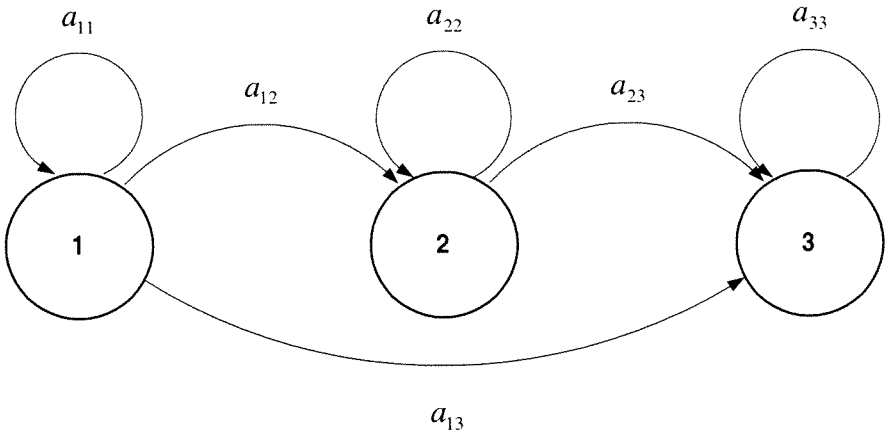

Figure 4.2:
3-State Markov Process for Generic Music Signal

The process of Figure 4.2 also has the property that, as time increases, the state index either increases or stays the same, *i.e.*, the states proceed from left to right. This is called a left-right or Bakis model and has been used for modeling signals whose properties change over time in a successive manner. The state-transition probabilities for this model have the property $a_{ij} = 0, j < i$, so the state transition matrix is upper-triangular

$$A = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ 0 & a_{22} & a_{23} \\ 0 & 0 & a_{33} \end{bmatrix}$$

In addition, the initial state probabilities have the form

$$\pi_i = \begin{cases} 0, i \neq 1 \\ 1, i = 1 \end{cases}$$

which assures that the state sequence begins in state 1 and ends in state 3.

Given a set of numerical values for the initial and state-transition probabilities, the above three-state Markov process can be used as a conceptual model of generic musical events produced by the vibrating strings of the guitar. Moreover, since virtually every

musical event will transition through each of these three states, the basic model can be used for almost all musical events. By contrast, the wide variation in the structure of speech (*e.g.*, number of phonemes, etc.) leads to similar models for speech having up to 15 states or more, with different numbers of states for each word [13]. Thus, the basic modeling of musical events is somewhat simpler than that of speech signals.

However, as with speech, since each state corresponds to a deterministically observable event, the output of the model in any given state is also deterministic. Stated differently, the simple of model of Figure 4.2 assumes only one type of attack, sustain and decay for the signal being modeled and therefore can model only one realization of the corresponding musical event. However, a given musical event may have many types of attack, sustain and decay characteristics, depending on how the underlying chord or note was played. Therefore, what is needed is a statistical model in which the observables for each state are drawn from a statistical distribution, *i.e.*, in which the observations are probabilistic functions of the state. This will provide for the modeling of musical signals that have a range of attack, sustain and delay characteristics, and thus will provide a more realistic and robust model for the generation of musical events.

4.2.2 Hidden Markov Model (HMM)

The HMM was introduced by Baum in 1972 as a statistical method of estimating the probabilistic functions of a Markov chain or process [15]. As discussed in the previous section, Markov processes are systems with discrete, time-dependent behavior characterized by common, short-time processes and probabilistic transitions between them. The processes are modeled as the discrete states $Q = \{q_1, q_2, \ldots, q_N\}$ of a finite state

machine, and the transitions between the states are controlled by the probabilistic elements of a state transition matrix A. For a left-right model the state transition matrix A is upper triangular or echelon.

The *hidden* Markov model (HMM) describes a stochastic process that produces a sequence of observed events or symbols. In addition to the N discrete states $Q = \{q_1, q_2, \ldots, q_N\}$ and the state transition matrix A, the HMM also includes a set of M distinct observation symbols $V = \{v_1, v_2, \ldots, v_M\}$ for each of the N states of the basic Markov model. The observation symbols V correspond to the physical output of the system being modeled, and are generated according to an observation symbol probability distribution $B = \{b_j(k)\}$ associated with each state transition. The observation symbol probability distribution B describes the probability with which an observation symbol o_t will occur during a given state transition. The distribution of the observation symbols for states $j = 1,2,\ldots,N$ is defined by

$$b_j(k) = P[o_t = v_k | q_t = j]$$

for $1 \le k \le M$. In the present work, the inclusion of the observation symbols and their probability distribution for each state allow for the modeling of musical events having different types of attack, sustain or decay, as shown in Figure 4.3, within the basic framework of the Markov process. For the amplitude envelope of Figure 4.3, there are four possible combinations of attack, sustain and decay that form a valid musical event. Note also that with the HMM, the underlying states are not directly observable. Instead, for each state one of the observation symbols will be observed with probability $b_j(k)$.

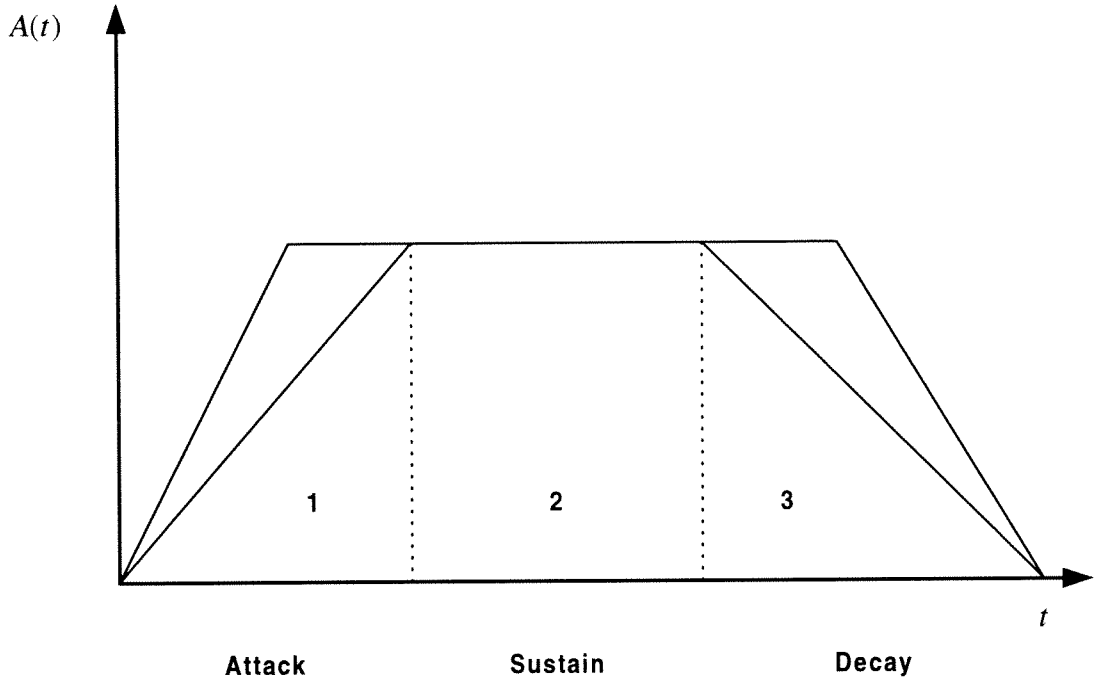

Figure 4.3:
Amplitude Envelope for 4 Combinations
Of Attack, Sustain and Decay

In the detection and analysis of musical events, the observation symbols may be low-level signal parameters computed at regular time intervals, such as windowed Fourier or LPC coefficients. Alternatively, the symbols may be high-level acoustic descriptors, such as attack, sustain or decay segments.

An HMM is therefore a model of the process that generates a sequence of signal parameters or acoustic descriptors belonging to a specific musical event (chord, interval or note) in a musical vocabulary. Specific variations between observed sequences within the same musical event, such as chord length and type of attack, sustain or decay, are modeled by the underlying stochastic properties of the HMM. The vocabulary of an HMM-based musical event detection system will comprise one HMM for each class of musical event being detected. Moreover, the complete specification of an HMM will

include the number of states N, the number of observables M, and the three probability distribution matrices $A = \{a_{ij}\}$, $B = \{b_j(k)\}$, and $\pi = \{\pi_i\}$. The complete parameter set of an HMM for a given musical event E may be specified using the compact notation

$$\lambda_E = \{A_E, B_E, \pi_E\}$$

This parameter set defines a probability measure for the observed sequence $O = \{o_1, o_2, \ldots, o_T\}$, so that pattern recognition with an HMM is equivalent to selecting the single model from the vocabulary that maximizes the probability of the observation sequence $P(O|\lambda_E)$.

4.3 HMM-Based Musical Event Model

As discussed above, the type of HMM best suited for signals whose properties change over time of the left-right model. The fundamental property of the left-right model is that the coefficients of the state-transition matrix A satisfy $a_{ij} = 0, j < i$, so that that A is upper triangular or echelon. In addition, in the present work the number of states is selected to correspond to the number of distinct acoustic segments in the musical event to be modeled, with two additional states for modeling the beginning and ending silences.

Figures 4.4a and 4.4b show two versions of an HMM used for modeling musical events in the present work. The HMM in Figure 4.4a is a conventional left-right type comprising five states corresponding to the pre-silence, attack, sustain, decay and post-silence segments of the musical event.

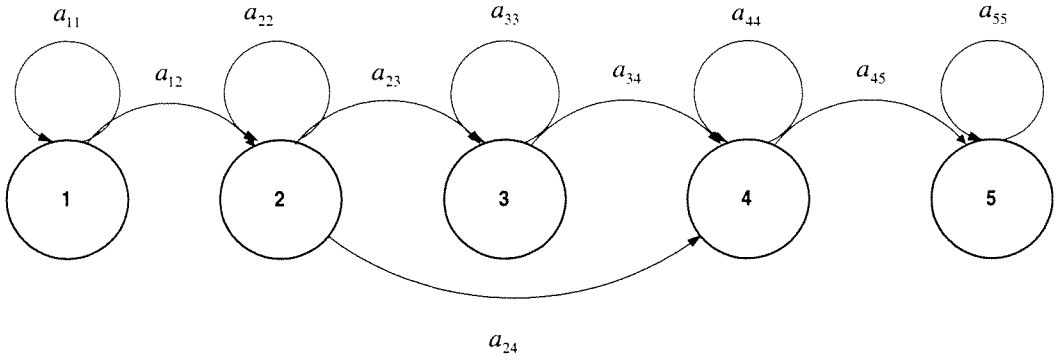

Figure 4.4a:
5-State HMM for a Musical Event

The state transition matrix A_1 corresponding to this HMM is given by

$$A_1 = \begin{bmatrix} a_{11} & a_{12} & 0 & 0 & 0 \\ 0 & a_{22} & a_{23} & a_{24} & 0 \\ 0 & 0 & a_{33} & a_{34} & 0 \\ 0 & 0 & 0 & a_{44} & a_{45} \\ 0 & 0 & 0 & 0 & a_{55} \end{bmatrix}$$

Note that all state transitions are strictly sequential, except for the direct transition from state 2 to state 4. This latter transition provides for the modeling of musical events having no measurable sustain, such as those generated by an acoustic guitar whose strings undergo strong damping due to the mechanical coupling of the strings to the guitar body.

The four-state HMM in Figure 4.4b is similar to that of Figure 4.4a, except that the pre- and post-silences have been combined into a single state. The state transition matrix A_2 corresponding to this HMM is thus

$$A_2 = \begin{bmatrix} a_{11} & a_{12} & 0 & 0 \\ 0 & a_{22} & a_{23} & a_{24} \\ 0 & 0 & a_{33} & a_{34} \\ a_{41} & 0 & 0 & a_{44} \end{bmatrix}$$

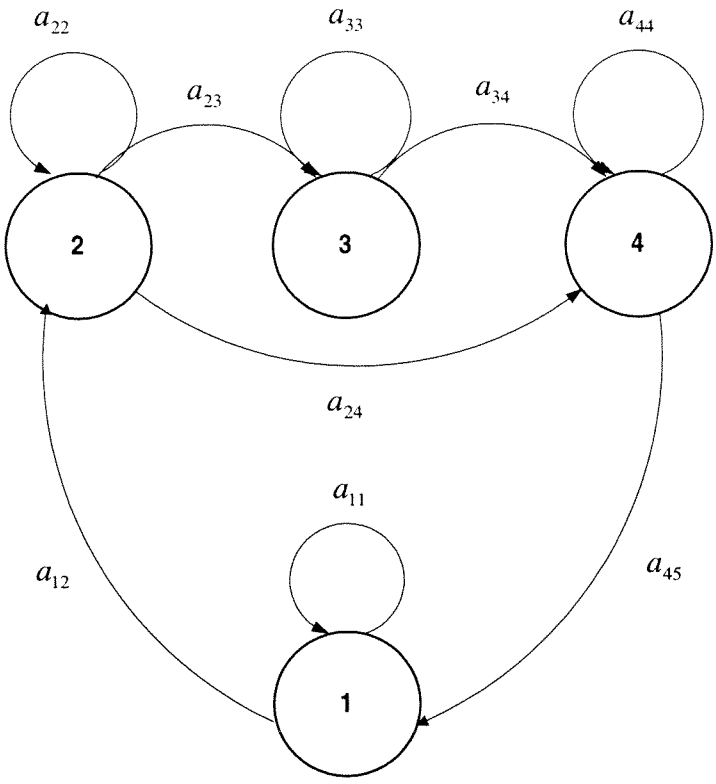

Figure 4.4b:
4-State HMM for a Musical Event

This HMM produces a lower rank state transition matrix, which may result in reduced computation time during optimization and recognition. However, in this case the HMM is not strictly left-right so the state transition matrix is not upper-triangular, which may result in increased computation time when solving the optimization and recognition problem.

For both versions of the HMM, the number of observation symbols per state, M, is identical. The observation symbols and their corresponding observation probabilities are derived from the various ways in which the strings can be picked, strummed or otherwise excited. In the present work, the number of observation symbols per state M is set at 40 although values may be used if needed. This number of symbols allows the

modeling of different intensities (soft through hard) and direction (up or down) of strumming or picking. The observation probabilities $b_j(k)$ are determined empirically by the relative frequency of the different types of string excitation. For example, if all of the ways that the strings may be strummed or picked occur with equal probability, then the observation symbol probability distribution B is uniform and each observation probability is just equal to $b_j(k) = 1/M = 1/40$.

Another modeling issue concerns the number of HMM models needed for each class of musical event. For example, each musical interval or chord corresponding to a given tonic may be played harmonically (strummed) or melodically (arpeggiated). Moreover, if played harmonically the speed of the strumming may vary over a wide range, and if played melodically the individual strings may be picked or harmonically tapped. Each of these examples is essentially a different musical event, which implies the need for a separate HMM for each played version of the chord. Therefore, each tonic-based class of musical event may contain several HMMs depending on the number of ways the event can be played.

4.4 Basic Problems for the HMM

In order for the HMM to be useful in the detection and analysis of musical events, three closely related analytical problems must be solved. The following three sections provide a brief description of these problems, while the detailed methods of solution are described in Chapter 5.

4.4.1 Pattern Matching

The pattern matching problem is: given a model $\lambda = \{A, B, \pi\}$ and a sequence of observations $O = \{o_1, o_2, \ldots, o_T\}$, determine the probability that the sequence was produced by the model, *i.e.* compute $P(O|\lambda)$. Stated differently, the goal is to score each event model based on the given observation sequence, and select the event model whose model score is the highest. This is the musical event recognition problem.

4.4.2 HMM Refinement

The refinement problem is to uncover the hidden part of the model in order to improve its capability in modeling sequences of musical events. Typical goals might be to learn about the structure of the model or to find optimal state sequences for continuous recognition of musical events. Stated differently, given a sequence of observations $O = \{o_1, o_2, \ldots, o_T\}$ and the model $\lambda = \{A, B, \pi\}$, determine a corresponding state sequence $Q = \{q_1, q_2, \ldots, q_T\}$ that is optimal in that it best explains the observations.

4.4.3 HMM Training

The training problem is to optimally estimate the model parameters $\lambda = \{A, B, \pi\}$ for each musical event model in order to maximize $P(O|\lambda)$. The approach is to use a test observation sequence to optimize the model parameters to best describe how a given observation sequence is generated.

4.5 Comparison of Stochastic Models

The above analysis suggests the statistical similarity between speech and music signals. Both signals can be modeled as parametric random processes whose properties change over time in a successive manner. In addition, both may be modeled as wide-sense, second-order Gaussian processes so that the mean vector and covariance matrix provide all necessary and sufficient information to model the statistics of each process. Both speech and music are assumed to be locally stationary within a relatively short analysis window.

Given the above statistical similarities and assumptions, a parametric statistical model is designed which captures the spectral properties of the music signal. Parametric statistical models have long been used in automated speech recognition, both in research and in commercially available products. The model used in the present work is based on a state machine that produces a sequence of observation symbols in each state, and is called a hidden Markov model (HMM). As in speech recognition, the HMM used to model music events has a left-right structure although the number of states is less than the number typically used in speech recognition. Thus, the HMM proposed in the present work for musical events is structurally simpler, in terms of numbers of states, than those used for modeling words in human speech.

Chapter 5

Matching Framework

5.1 Design Considerations

The previous two chapters have provided a theoretical framework for the automated detection and analysis of musical events. Chapter 3 provided a structural comparison of the physical processes of speech and music production, and showed that the well-established body of speech recognition research can be directly applied to the recognition of musical events. Chapter 4 then showed how hidden Markov models, long applied to speech recognition research, could also be used to statistically model musical events such as chords, intervals and notes. The present chapter continues by describing a theoretical approach, based again on speech recognition research, to solving the key analytical problems for automated detection and analysis of musical events.

Consider an HMM based system for recognizing distinct musical events. Specifically, assume a system comprising a stored library of V musical chords, intervals and notes to be recognized, and that each event is modeled by a distinct HMM as described in Chapter 4. A simplified block diagram of the proposed system shown in Figure 5.1. The system of Figure 5.1 includes a signal processing and feature extraction front-end, a library of stored musical event models, and a probabilistic decision-making algorithm.

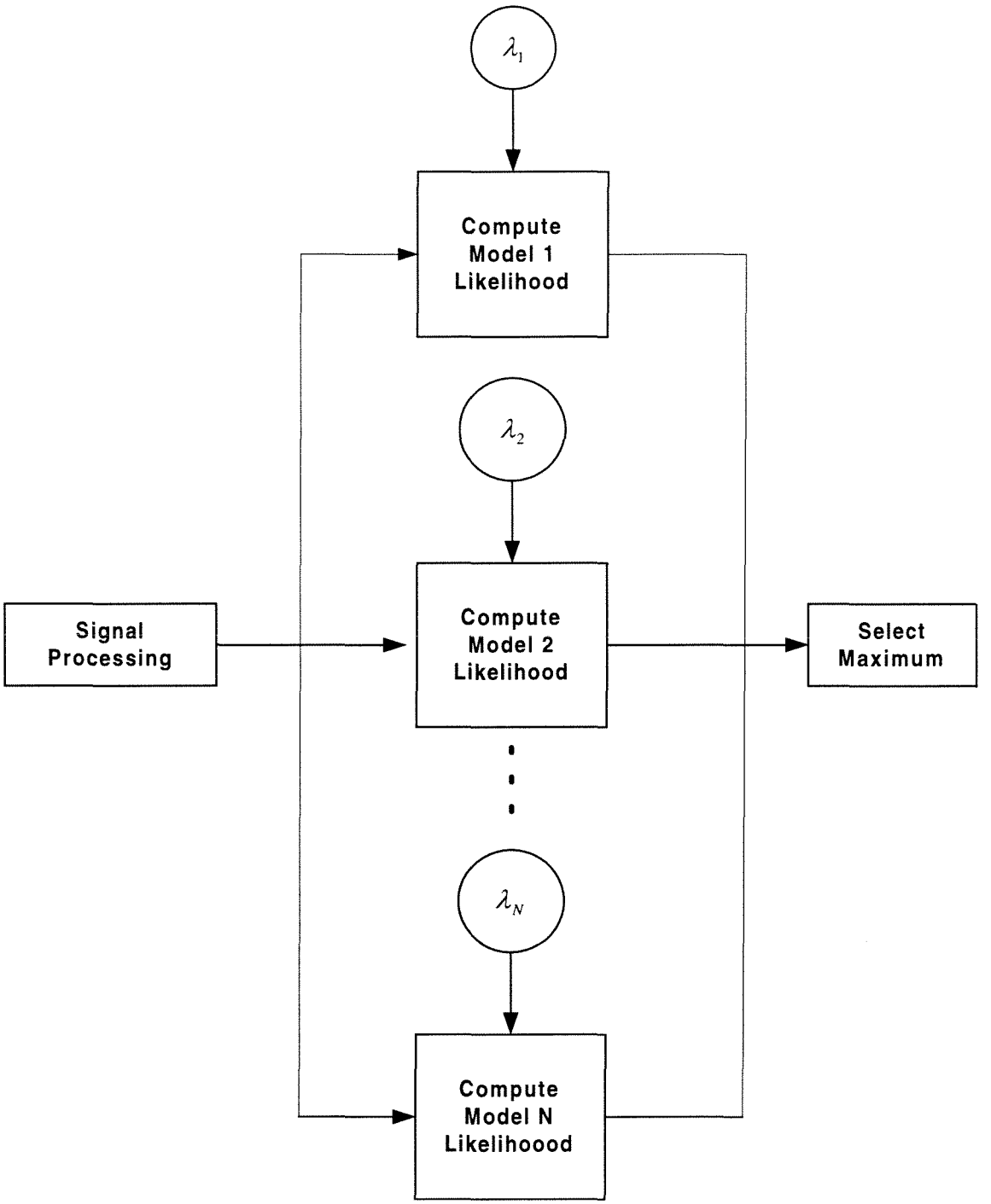

Figure 5.1:
Musical Event Recognition System

While each of these system components will be discussed in detail in the following sections, the overall function of the proposed music recognition system comprises just three basic steps:

- Convert the musical event to sequence of observation symbols
- Given the observation symbols, compute model likelihood for each stored model
- Select the musical event model having the highest model likelihood

As will be discussed later in this chapter, the amount of computation needed to perform these three steps is well within the capabilities of modern signal processing and computation devices.

5.2 Signal Processing

Signal processing and feature extraction are used to convert the analog musical signal into a sequence of observation symbols having a much lower information rate (typically measured in bits per second) than the raw analog signal. More particularly, the signal processing function converts the analog musical signal to a parametric representation that effectively compresses the information contained in the raw analog signal. For example, an 11 kilohertz sampled signal with 16-bit digitized amplitudes has an information rate of 176,000 bits per second in un-compressed form. However, suppose the same sampled signal is converted to the frequency domain via spectral analysis and only the lowest 10 spectral components are kept. Further, assume 100 spectral vectors per second are used. Then again using 16-bit precision reduces the information rate to 16,000 bits per second, an 11:1 reduction in information rate.

The two most common methods of spectral analysis in practical systems are the filter bank model and the linear predictive coding (LPC) model. Each of these methods will be discussed in Sections 5.2.2 and 5.2.3, respectively.

Feature extraction techniques further compress the musical signal by encoding the continuous parametric (*i.e.*, spectral) representation into a finite number of parametric observation symbols. Using, for example, vector quantization (VQ) encoding methods may provide substantial further reductions in the information rate. Vector quantization methods may be applied to any spectral representation without regard to how the spectral analysis is performed. VQ encoding methods as applied to musical events will be discussed in Section 5.3.3.

Since digital signal processing techniques are used in the detection and analysis of musical events, the first steps are to sampling the analog input signal and convert it to digital form.

5.2.1 Sampling and Digitization

The analog musical signal $s(t)$ is first sampled to obtain a corresponding sampled signal $s(n)$, the latter a sequence of numbers representing the amplitudes of $s(t)$ at distinct points of time. The key consideration when sampling the analog signal relates to the number of samples per unit time (*i.e.*, the sample rate) needed to perfectly reconstruct the original signal. Nyquist [16] showed that the minimum sample rate t_s needed for perfect reconstruction of a continuous signal from its discrete samples must be twice the highest frequency component of the signal, or

$$t_s \geq 2 f_{\max}$$

Sampling at a rate lower that the Nyquist rate will result in aliasing of the original signal. Since the maximum frequency for an electric guitar is approximately 5.5 kilohertz, the sample rate must be at least 11 kilohertz. Note that this is lower than that required for high quality human speech.

Once the analog music signal has been sampled, each continuous sample must be encoded into a digital number. The number of bits used to encode the samples is important, since it directly affects the accuracy of the subsequent processing. For example, if 8 bits are used to encode each sample then there are only 255 possible values for each sample and the signal resolution is limited accordingly. However, if 16 bits are used then there are 65,535 possible values for each sample, thus providing a large improvement in resolution. Therefore, the present work assumes 16-bit digitization for all digital samples.

The sampled music signal $s(n)$ comprises a sequence of 16-bit digital numbers derived from the original analog input signal $s(t)$. The sampled signal is now converted into a parametric representation that maintains the information embedded in the original musical signal. As mentioned above, the most common parametric representations are derived using spectral analysis methods, namely, filter banks and linear predictive coding.

5.2.2 Filter Banks

A simplified block diagram for the structure of the filter bank model is shown in Figure 5.2. The sampled signal $s(n)$ is passed through a parallel bank of Q band-pass filters that cover the frequency range of interest. For the musical events produced by the guitar, this

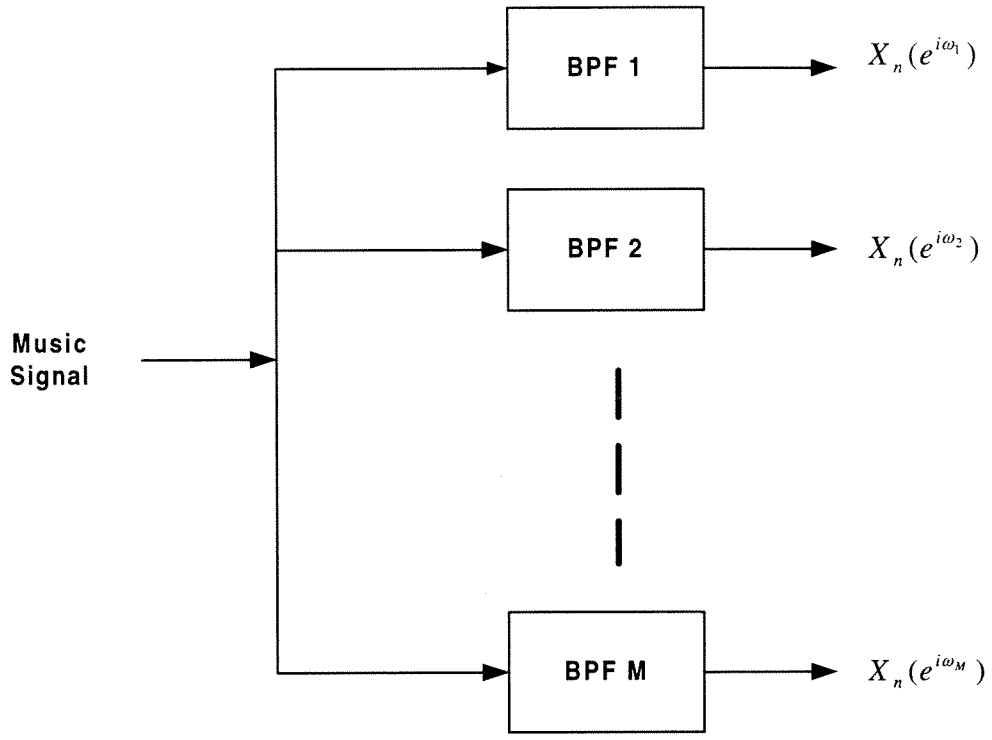

Figure 5.2:
Bank-of-Filters Analysis Model

range should be 50-6,000 Hertz. For each input sample n, the output of the k^{th} filter has the general form $X_n(e^{i\omega_k})$ with ω_k the normalized frequency $2\pi f_k / F_s$ with F_s the sampling frequency derived from the sample rate t_s defined above. This general form of output is called the short-time spectral representation of $s(n)$ at time n.

The key purpose of filter bank analysis is to measure the energy of the musical signal in each frequency band. In order to accomplish this, the output of each band-pass filter $X_n(e^{i\omega_k})$ is processed through a full-wave rectifier followed by a low-pass filter. The rectifier shifts the band-pass spectrum to the low frequency band while simultaneously creating an infinite series of high-frequency images. The low-pass filter selects only the low frequency component, thereby giving a set of signals that represent

the energy in each of the Q filter bands. This approach works well as long as the band-pass filters are narrow enough so that each contains only one strong signal harmonic.

The most common type of filter bank used for spectral analysis is the uniform filter bank, in which the center frequency of the n^{th} filter is given by the relation

$$f_n = \frac{F_s}{N} n$$

where $1 \leq n \leq Q$ and N is the number of parallel filters used to span the frequency range of the guitar. The bandwidth of each filter in the uniform filter band satisfies the relation

$$B_n \geq \frac{F_s}{N}$$

where equality indicates no overlap between adjacent filter bands. In practical systems, the inequality is always satisfied since adjacent filters always overlap to some extent.

Another type of filter bank used for spectral analysis is the non-uniform filter bank, in which the individual filter passbands are spaced in frequency according to a specified criteria. The most commonly used criteria are based on models of the human auditory system and space the filter passbands using a logarithmic frequency scale.

In the case of uniform filtering, the filter banks used for spectral analysis may be implemented using the short-time Fourier transform. The actual transform may be efficiently computed using FFT methods. For non-uniform filter banks, each filter is usually implemented using a direct convolution since an efficient FFT structure is generally not available. However, for certain types of non-uniform filter banks a tree structure in which the signal is filtered in successive stages, and the sampling rate reduced at each stage may be used for implementation. For example, each stage of an

octave-spaced filter bank may be efficiently implemented using quadrature mirror filters (QMFs) and decimation by a factor of 2 at each stage.

5.2.3 Linear Predictive Coding

The advantages of the LPC method for use in modeling voiced speech, especially speech formants such as vowels and diphthongs, were discussed in Chapter 3. A block diagram of the LPC analysis model is shown in Figure 5.3.

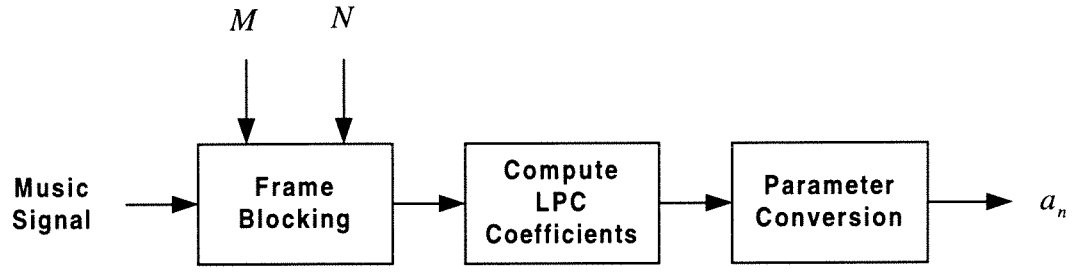

Figure 5.3:
LPC Analysis Model

Since vowels correspond to musical notes and diphthongs to transitions between musical notes, the LPC methods should theoretically work well with musical signals. As discussed in Chapter 3, the LPC method performs spectral analysis on successive frames of the musical signal using an all-pole filtering model. In this case the process output for each frame has the general form $Y_n(e^{i\omega}) = C/A_n(e^{i\omega})$, where the denominator is a p^{th} order polynomial having a z-transform

$$A(z) = 1 + a_1 z^{-1} + a_2 z^{-2} + \cdots + a_p z^{-p}$$

The output of the LPC analysis for each frame is a vector of the coefficients a_i that provide the spectrum of an all-pole filter that optimally matches the spectrum of the musical signal within the frame being analyzed.

The LPC model was explicitly derived in Chapter 3. Recall the transfer function $H(z)$ for an all-pole system given by

$$H(z) = \frac{1}{1 - \sum_{i=1}^{p} a_i z^{-i}}$$

which corresponds to a given signal sample at time n being modeled as a linear combination of the previous p samples plus the current input

$$s(n) = \sum_{i=1}^{p} a_i s(n-i) + Gu(n)$$

The basic problem of LPC analysis is to determine the set of predictor coefficients $\{a_i\}$ directly from the musical signal. This will assure that the spectral properties of $H(z)$ closely match those of the musical signal within the analysis frame. Since the spectral characteristics of the musical signal vary over time, the predictor coefficients must be estimated from a short segment (frame) of the overall musical signal. The analysis is performed on successive (usually overlapping) frames of the musical signal with frame spacing on the order of 10-20 milliseconds. Thus, under LPC analysis each frame generates a single p-dimensional spectral vector.

Once a spectral representation of the musical signal has been computed, either by the filter bank or LPC method, the result is a sequence of p-dimensional vectors that contain the time-varying spectral characteristics of the musical signal. This sequence of vectors may be used to directly form the sequence of observation symbols needed to train and use the HMM. However, it is sometimes possible to further reduce the information

rate by building a table of discrete analysis vectors which then form the sequence of observation symbols used by HMM based recognition. If this is desired, the next step is to encode the sequence of spectral vectors using vector quantization (VQ) methods.

5.2.4 Vector Quantization

The fundamental idea behind VQ methods is to further reduce the spectral representation of musical signals to a small number of distinct vectors, while maintaining sufficient variability for reliable event recognition. The theoretical limit for this idea would be a single, unique vector for each type of attack, sustain and decay since these comprise the basic components of each musical event. However, the wide range of playing styles and techniques make achievement of the theoretical limit impossible. Still, VQ methods may be used to build a table or codebook of "typical" spectral vectors that reduces the information rate and storage requirements of the musical event recognition process. For example, assuming 385 types each of attack and decay and 250 types of sustain, then a VQ table with approximately 1024 unique vectors would be required. Thus, a 10-bit index for each vector would be sufficient to represent an arbitrary input vector.

A block diagram of the VQ process is shown in Figure 5.4. Building a VQ table or codebook requires a large set of spectral analysis vectors as a training set, along with a measure of the distance between pairs of vectors. For each musical event, the training vectors should span the anticipated range of attack, sustain and decay characteristics based on various playing techniques such as strumming, picking, slides, hammer-ons, pull-offs, string bending, vibrato and tapping. Additionally, training vectors from different types of guitars and electromagnetic pickups should also be included.

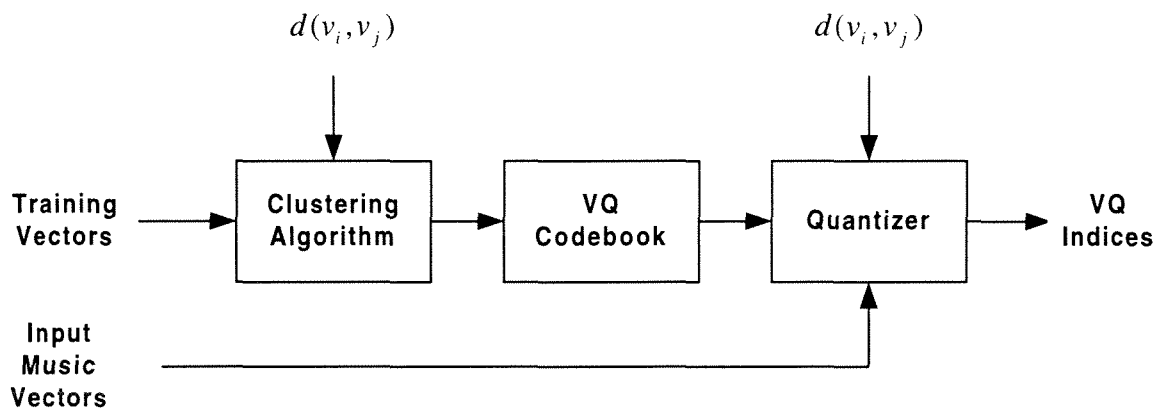

Figure 5.4:
VQ Processing Structure

The specific distance measure used for determining the similarity of spectral vectors is very important to the overall performance of the VQ codebook, both for building the codebook and during the classification procedure. The distance measure for pairs of vectors will have the general form

$$d(v_i, v_j) = \begin{cases} 0, & v_i = v_j \\ > 0, & v_i \neq v_j \end{cases}.$$

There are a number of standard measures including L_1, L_2, weighted cepstral distances or likelihood measures. The latter are particularly useful with LPC derived spectral vectors.

The VQ method also requires a centroid computation or clustering algorithm for partitioning the training vectors into the set of codebook vectors. A commonly used procedure is the Lloyd or k-means clustering algorithm. The inputs to the clustering algorithm are the set of training vectors and the distance measure, and the output is the codebook of analysis vectors. Also required is a classification procedure or quantizer for optimally choosing the closest codebook vector to the input vector. The inputs to the quantizer are the distance measure, the indices of the codebook vectors, and the spectral

vectors derived from the musical event to be recognized. The output of the quantizer is the corresponding index of the closest codebook vector. A sequence of these codebook indices forms the observation symbols used by the HMM based musical event recognition system.

Once the set of observation symbols for a musical event have been formed, they can be used either to:

- Estimate and/or refine the parameters of the corresponding HMM model; or
- Identify the musical event using an existing HMM model.

The following sections will describe both the training and recognition processes in greater detail.

5.4 HMM Training

Before the HMM system can be used to recognize musical events, the model parameters for each event in the library must be estimated. More particularly, for each musical event E, the elements of the three probability distribution matrices $A_E = \{a_{ij}\}$, $B_E = \{b_j(k)\}$, and $\pi_E = \{\pi_i\}$ must be estimated. These elements are optimally estimated using training data derived from signals representing the anticipated range of musical events. The training data is encoded using the VQ method described above to form a set of training observation symbols corresponding to the anticipated range of musical events. Using the set of training observation symbols, the model parameters that optimize the likelihood of the training observation symbols are determined.

Although the problem of determining the model parameters cannot be solved analytically in a closed form, it can be solved using standard iterative optimization

techniques such as gradient ascent or other maximum likelihood methods. For example, given a training observation sequence O and an initial set of model parameters $\hat{\lambda} = (\hat{A}, \hat{B}, \hat{\pi})$, a reestimated set of model parameters $\lambda = (A, B, \pi)$ can be determined using the following reestimation formulae derived by Baum and his colleagues [14]:

$$\pi_i = \frac{P(O, q_0 = i|\lambda)}{P(O|\lambda)}$$

$$a_{ij} = \frac{\sum_{t=1}^{T} P(O, q_{t-1} = i, q_t = j|\lambda)}{\sum_{t=1}^{T} P(O, q_t = i|\lambda)}$$

$$b_i(k) = \frac{\sum_{t=1}^{T} P(O, q_t = i|\lambda)\delta(o_t, v_k)}{\sum_{t=1}^{T} P(O, q_t = i|\lambda)}$$

Using the Baum formulae it has been proven that either the initial model parameters $\hat{\lambda}$ define a critical point of the likelihood function $P(O|\lambda)$, or the reestimated model parameters λ are more likely to have produced the training observation sequence O, *i.e.*, $P(O|\lambda) > P(O|\hat{\lambda})$. Thus, the Baum formulae can be used iteratively to improve the likelihood that the model was produced by the training observation sequence, until a critical point is reached. Note, however, that since the likelihood function is not convex and will therefore have many local maxima, the above approach will only lead to a local, as opposed to global, maxima point.

As mentioned above, the parameter estimation problem can also be solved using standard gradient ascent methods. In this case the problem is formulated as the constrained optimization of $P(O|\lambda)$ subject to the usual stochastic constraints

$$\sum_{i=1}^{N} \pi_i = 1$$

$$\sum_{j=1}^{N} a_{ij} = 1$$

$$\sum_{k=1}^{M} b_j(k) = 1$$

for $1 \le i, j \le N$. Using Lagrange multipliers the reestimation formulae can then be written in the standard Lagrange optimization notation

$$\pi_i = \frac{\pi_i \dfrac{\partial P}{\partial \pi_i}}{\sum_{k=1}^{N} \pi_k \dfrac{\partial P}{\partial \pi_k}}$$

$$a_{ij} = \frac{a_{ij} \dfrac{\partial P}{\partial a_{ij}}}{\sum_{k=1}^{N} a_{ik} \dfrac{\partial P}{\partial a_{ik}}}$$

$$b_j(k) = \frac{b_j(k) \dfrac{\partial P}{\partial b_j(k)}}{\sum_{l=1}^{M} b_j(l) \dfrac{\partial P}{\partial b_j(l)}}$$

These can be shown to be identical to the Baum reestimation formulae.

5.5 HMM Refinement

Once the model parameters for each musical event have been estimated, it may be desirable to determine the optimal state sequence corresponding to a given training observation sequence. This may allow the model to be refined in terms of the number and sequence of states, size of the VQ codebook, and so on, thereby improving its capability to recognize sequences of musical events. In speech recognition, the most widely used criterion for solving this problem is to optimally find the best state sequence

by maximizing $P(q|O,\lambda)$ which is equivalent to maximizing $P(q,O|\lambda)$. The most common method for finding the best state sequence uses the Viterbi algorithm [17], [18], which is discussed in detail in the reference. The Viterbi algorithm is efficiently implemented using a lattice or trellis structure, with its output being the optimal state sequence corresponding to a given training sequence.

5.6 Pattern Matching

Now that a library of musical event models has been trained and refined, the recognition of unknown musical events can be performed. For event recognition, the specific problem is to determine the probability of an input observation sequence O of length T given an event model $P(O|\lambda)$. One obvious approach would be to enumerate through every possible state sequence of length T; however, this would require on the order of $2TN^T$ calculations which becomes infeasible even for small values of N and T.

However, from the research in speech recognition a more efficient procedure exists for computing $P(O|\lambda)$. Define the partial observation sequence as $o_1 o_2 \cdots o_t$ which is the sequence of observation symbols occurring until time t. Next, define the forward variable $\alpha_t(i)$ as

$$\alpha_t(i) = P(o_1 o_2 \cdots o_t, q_{t=i}|\lambda)$$

which is the probability of the partial observation sequence and state i, given the musical event model λ. Now solve for $\alpha_t(i)$ inductively as follows:

1. Initialization

$$\alpha_1 = \pi_i b_i(o_1), \qquad 1 \le i \le N$$

2. Induction

$$\alpha_{t+1}(j) = \left[\sum_{i=1}^{N} \alpha_t(i) a_{ij}\right] b_j(o_{t+1}), \qquad \begin{matrix} 1 \le t \le T-1 \\ 1 \le j \le N \end{matrix}$$

3. Termination

$$P(O|\lambda) = \sum_{i=1}^{N} \alpha_T(i)$$

The amount of computation needed to calculate $\alpha_t(j)$ for $1 \le t \le T$ is on the order of N^2T calculations, a substantial savings over the direct computation method. In fact, the forward procedure is the key computation method that makes automated music speech recognition feasible for practical recognition systems.

5.7 Discussion

The automated recognition of musical events requires that the raw musical signal be converted to a form suitable for digital processing and analysis. A key objective of this processing is to reduce the information rate while retaining the key spectral properties embedded in the original signal. Each of the processing steps discussed in this chapter contributes to the accomplishment of that objective.

Signal processing and feature extraction are used to convert the analog musical signal into a sequence of discrete observation symbols having a lower information rate, as measured in bits per second. The sequence of observation symbols are typically a sequence of *p*-dimensional vectors that contain the time-varying spectral characteristics of the original musical signal. The observation symbols are first used to train and refine the set of parametric statistical models (HMMs) used to represent each musical event. After training, HMMs are used to identify the observation symbols corresponding to

unknown musical events. The key point is that conventional signal processing and speech recognition approaches can be used to perform each step of the analysis.

Chapter 6

Conclusions

This work has presented a theory and methodology for the detection and analysis of musical events, including notes, intervals and chords. The goal of this work was to demonstrate that the theory and methods developed for automated speech processing could be applied to the recognition of musical events. The concept of applying model-based signal processing to the detection and analysis of musical events shows great promise, both for further research and for use in practical applications such as composing [1] and musical education [19].

However, work remains to be done in order to fully demonstrate the feasibility and robustness of the concept when applied to actual musical signals. In particular, a complete software implementation of the proposed method needs to be developed and fully tested in order to validate the stochastic models using actual musical data. Preliminary implementation work has been performed by modifying the signal processing portion of conventional speech recognition software obtained from Dragon Systems, Inc., Newton, Massachusetts. Specifically, the sample rate and frequency range of the signal processing software was modified in order to determine the promise of the proposed approach. Significantly, the stochastic word models were not modified, although a custom display driver was developed so that individual chords and notes could be visually displayed.

The modified software was trained using a single guitar (an Ernie Ball Music Man®) played by a single player. The guitar was connected directly to the high-impedance line input of a Sound Blaster™ audio card and the software was trained using sequences of five chords or notes for each musical event. Included in the training sequences were the same chord played at different fretboard locations and using different tonal inversions. This was done to test the ability of the software to distinguish between events having the same tonic but played at different locations on the fretboard of the guitar.

The results of this preliminary experiment were very promising. Even without modifying the stochastic word models, the modified speech recognition software was able to correctly identify about 80 percent of the notes and chords played by the same individual that trained the software. Especially promising was that different tonal inversions of the same chord were almost always correctly identified. The software was able to track the chord and note changes as long as there was a short silence between each event. This was not surprising, since the speech recognition software obtained from Dragon Systems was specifically designed to recognize discrete words as opposed to continuous speech.

However, as expected, the identification accuracy was substantially reduced when the same chords and notes were played by an individual other than the trainer, or when using a different guitar. This is a direct result of the stochastic models for speech recognition not being optimal for recognizing musical events. It is expected that modifications of the stochastic word models would substantially improve the system performance. These modifications would address issues such as the optimal number of

HMM states, initial values for model parameters, and training by multiple players and instruments. In addition, the effects of using an explicit temporal duration for each HMM state should be fully developed and tested. The results of this work should lead to a practical and robust system for the automated recognition of chords, intervals and notes.

Bibliography

[1] R. Rowe, *Interactive Music Systems*, MIT Press, Cambridge, MA, 1993.

[2] L. Rabiner, M.J. Cheng, A.E. Rosenberg and C.A. McGonegal, A comparative performance study of several pitch detection algorithms, *IEEE Trans. Acoustics, Speech, and Signal Processing*, ASSP-24, 5: 399-418, October 1976.

[3] U.S. Patent No. 5,567,903 entitled *Transducer Assembly for a Stringed Musical Instrument*, issued October 22, 1996, to J. Coopersmith, N. Weiss and H. Madden.

[4] H. Van Trees, *Detection, Estimation and Modulation Theory*, John Wiley & Sons, New York, NY, 1968.

[5] U.S. Patent No. 5,567,903, *ibid.*

[6] U.S. Patent No. 4,357,852 entitled *Guitar Synthesizer*, issued November 9, 1982, to N. Suenaga.

[7] H. Sano and B.K. Jenkins, A neural network model for pitch perception, *Computer Music Journal*, Vol. 13, No. 3: 41-48, Fall 1989.

[8] M.B. Sachs, C.C. Blackburn and E.D. Young, Rate-place and temporal-place representations of vowels in the auditory nerve and anteroventral cochlear nucleas, *Journal of Phonetics*, 16: 37-53, 1988.

[9] E. de Boer, On the residue and auditory pitch perception, *Handbook of Sensory Physiology – Auditory System, Volume 3 – Clinical and Special Topics*, Springer-Verlag, Berlin, 1974.

[10] P.M. Todd and D.G. Loy, *Music and Connectionism*, MIT Press, Cambridge, MA, 1991.

[11] D.L. Scarborough, B.O. Miller and J.A. Jones, Connectionist models for tonal analysis, *Computer Music Journal*, Vol. 13, No. 3, Fall 1989.

[12] B. Laden and D. Keefe, The representation of pitch in a neural model of chord classification, *Computer Music Journal*, Vol. 13, No. 4, Winter 1989.

[13] L. Rabiner and B. Huang, *Fundamentals of Speech Recognition*, Prentice-Hall, Englewood Cliffs, NJ, 1993.

[14] H.F. Olson, *Music, Physics and Engineering*, Dover Publications, New York, NY, 1967.

[15] L.E. Baum, An inequality and associated maximization technique in statistical estimation for probabilistic functions of Markov processes, *Inequalities*, 3: 1-8, 1972.

[16] H. Nyquist, Certain factors affecting telegraph speed, *Bell System Tech. Journal*, 3, No. 2: 324-346, 1924.

[17] A.J. Viterbi, Error bounds for convolutional code and an asymtotically optimal decoding algorithm, *IEEE Trans. Information Theory*, IT-13:260-269, April 1967.

[18] G.D. Forney, The Viterbi algorithm, *Proc. IEEE*, 61: 268-278, March 1973.

[19] U.S. Patent No. 5,585,583 entitled *Interactive Musical Instrument Instruction System*, issued December 17, 1996, to R.L. Owen.